

Renato N. Elias · Marcos A. D. Martins  
Alvaro L. G. A. Coutinho

## Parallel edge-based solution of viscoplastic flows with the SUPG/PSPG formulation

Received: 1 August 2005 / Accepted: 17 September 2005  
© Springer-Verlag 2005

**Abstract** A parallel edge-based solution of three dimensional viscoplastic flows governed by the steady Navier–Stokes equations is presented. The governing partial differential equations are discretized using the SUPG/PSPG stabilized finite element method on unstructured grids. The highly nonlinear algebraic system arising from the convective and material effects is solved by an inexact Newton–Krylov method. The locally linear Newton equations are solved by GMRES with nodal block diagonal preconditioner. Matrix–vector products within GMRES are computed edge-by-edge (EDE), diminishing flop counts and memory requirements. A comparison between EDE and element-by-element data structures is presented. The parallel computations were based in a message passing interface standard. Performance tests were carried out in representative three dimensional problems, the sudden expansion for power-law fluids and the flow of Bingham fluids in a lid-driven cavity. Results have shown that edge based schemes requires less CPU time and memory than element-based solutions.

**Keywords** Viscoplastic flow · Edge-based data structures · Stabilized finite element formulations · Parallel computing

### 1 Introduction

Several modern material and manufacturing processes involve non-Newtonian fluids, and in particular viscoplastic fluids. Examples of non-Newtonian behavior can be found in processes for manufacturing coated sheets, optical fibers, foods, drilling muds and plastic polymers. Numerical simulations of non-Newtonian behavior represent a particular and difficult case of incompressible fluid flows. In these fluids the dependence between the viscosity and the shear rate amplifies the nonlinear character of the governing equations.

Renato N. Elias · Marcos A. D. Martins · Alvaro L. G. A. Coutinho (✉)  
Center for Parallel Computations  
and Department of Civil Engineering,  
Federal University of Rio de Janeiro, P. O. Box 68506,  
Rio de Janeiro, RJ 21945-970, Brazil  
E-mail: {renato, alvaro, marcos}@nacad.ufrj.br

For a comprehensive presentation of numerical methods for non-Newtonian fluid flow computations we refer to [12] and [40].

The finite element computation of incompressible Newtonian flows involves two sources of potential numerical instabilities associated with the Galerkin formulation of the problem. One is the presence of convective terms in the governing equations. The other is due to the use of inappropriate combinations of interpolation functions to represent the velocity and pressure fields. These instabilities are frequently prevented by addition of stabilization terms to the Galerkin formulation. In the context of non-Newtonian fluids, the rheological equations are inherently nonlinear, thus increasing the difficulties to find an efficient solution method.

In this work we consider the stabilized finite element formulation proposed by Tezduyar [56] applied to solve steady viscoplastic incompressible flows on unstructured grids. This formulation, originally proposed for Newtonian fluids, allows that equal-order-interpolation velocity–pressure elements are employed, circumventing the Babuska–Brezzi stability condition by introducing two stabilization terms. The first term is the streamline upwind/Petrov–Galerkin (SUPG) introduced by Brooks and Hughes [7] and the other one is the pressure-stabilizing/Petrov–Galerkin (PSPG) stabilization proposed initially by Hughes et al. [25] for Stokes flows and generalized by Tezduyar et al. [55] to the Navier–Stokes equations.

Many authors have considered finite element formulations in combination with solution algorithms for nonlinear problems arising from non-Newtonian incompressible flow simulations. For instance, in [6] the Least-Squares method was employed; in [1] a mixed-Galerkin finite element formulation with a Newton–Raphson iteration procedure coupled to an iterative solver was used, while in [44, 50] the authors adopted the Galerkin/least squares formulation (GLS) associated also with Newton–Raphson algorithm; Meuric et al. [37] used the SUPG formulation in combination with Newton–Raphson and Picard iterations as a strategy to circumvent computational difficulties in annuli flow computations. Some of these strategies employ analytical or directional forms of Jacobians in the Newton method.

The inexact-Newton methods associated with iterative Krylov solvers have been used to reduce computational efforts related to non-linearities in many problems of computational fluid dynamics, offering a trade-off between accuracy and the amount of computational effort spent per iteration. According to Kelley [29] its success depends on several factors, such as: quality of initial Newton step, robustness of Jacobian evaluation and proper forcing function choice. Tezduyar et al. [52] presented a large number of 3D problems computed on parallel platforms with Jacobian-Free Newton–Krylov methods. Shadid et al. [48] have shown an inexact-Newton method applied to problems involving Newtonian fluids, mass and energy transport, discretized by SUPG/PSPG formulation and equal-order interpolation elements. In [17] we discussed the use of numerically approximated Jacobians in the inexact Newton solution of viscoplastic flows with the SUPG/PSPG formulation. Recently, Knoll and Keyes [30] discussed the constituents of a broader class of inexact-Newton methods, the Jacobian-Free Newton–Krylov methods. In this work we address only the essentials of the inexact-Newton methods and the interested reader should refer to Knoll and Keyes [30], and references therein, for a more detailed presentation.

Element-based data structures have been extensively used in the implementation of iterative solvers. A wide class of preconditioners specially designed for element-by-element (EBE) techniques can be found in literature [23,24,27,53,54,59]. This type of data structure, besides fully exploiting the sparsity of the system, is suitable for parallelization and vectorization, as matrix-vector (matvec) products and right-hand side evaluations can be written in the form of a single loop structure [3,8,10,22]. The EBE preconditioners in [27,53,54] are also built using element-level (or group-level or cluster-level) computations with loop structures similar to the matvec products. Another way of writing an efficient code for matvec products is by using the compressed storage row (CSR) format [47], in which only global nonzero coefficients are stored. In the CSR format, the algorithm for matrix-vector products requires a two-loop structure: an outer loop over the equations and an inner loop over each nonzero contribution. In comparison with EBE implementations, this format has the advantage of storing only global coefficients, but with the overhead of an inner loop that can not be unrolled, specially if unstructured meshes are to be considered.

Edge-based data structures, or EDE for short, can be viewed as a blend of EBE and CSR formats: only global nonzero coefficients are stored and the single loop structure is maintained. In the context of finite elements, this type of data structure has been used since the early 1990s. Peraire and Morgan [42] and Lyra et al. [33] used edge-based data structures to compute the nodal balance of fluxes for compressible flow. Luo et al. [32] used an edge-based approach in the development of an upwind finite element scheme for the solution of the Euler equations. Löhner [31] showed different ways of grouping edges in the evaluation of residuals for the Laplacian operator, aiming to reduce the number i/a operations. More recently, Ribeiro et al. [45] presented an

edge-based implementation for stabilized semi-discrete and space-time finite element formulations for shallow water equations. Other recent works on the subject include those of Coutinho et al. [11], with applications to nonlinear solid mechanics, Catabriga and Coutinho [9], for the implicit SUPG solution of the Euler equations, and Soto et al. [51] for incompressible flow problems. It has been shown by Ribeiro and Coutinho [46] that for unstructured grids composed by tetrahedra, edge-based data structures offer more advantages than CSR, particularly for problems involving many degrees of freedom.

The present paper is organized as follows: Sects. 2 and 3 present the governing and constitutive equations and the SUPG/PSPG finite element formulation with the inexact nonlinear method. Section 4 introduces the use of the edge-based data structure in the present finite element context. Section 5 describes briefly the parallelism issues related to the solution procedure and the test problems and concluding remarks are presented in Sects. 6 and 7 respectively.

## 2 Governing and constitutive equations

Let  $\Omega \subset \mathfrak{N}^{nsd}$  be the spatial domain, where  $nsd$  is the number of space dimensions. Let denote the boundary of  $\Omega$ . We consider the following velocity-pressure formulation of the Navier–Stokes equations governing steady incompressible flows:

$$\rho(\mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f}) - \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \quad \text{on } \Omega \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{on } \Omega \quad (2)$$

where  $\rho$  and  $\mathbf{u}$  are the density and velocity,  $\boldsymbol{\sigma}$  is the stress tensor given as

$$\boldsymbol{\sigma}(p, \mathbf{u}) = -p \mathbf{I} + \mathbf{T}, \quad (3)$$

where  $p$  is the hydrostatic pressure,  $\mathbf{I}$  is the identity tensor and  $\mathbf{T}$  is the deviatoric stress tensor.

The essential and natural boundary conditions associated with Eqs. 1 and 2 can be imposed at different portions of the boundary  $\Gamma$  and represented by,

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_g \quad (4)$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h} \quad \text{on } \Gamma_h \quad (5)$$

where  $\Gamma_g$  and  $\Gamma_h$  are complementary subsets of  $\Gamma$ .

The relationship between the stress tensor and deformation rate for Newtonian fluids is defined by a proportionality constant, which represents the momentum diffusion experienced by the fluid. Therefore, the deviatoric tensor in Eq. 3 can be expressed by

$$\mathbf{T} = 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) \quad (6)$$

where  $\mu$  is the proportionality constant known as the dynamic viscosity and  $\boldsymbol{\varepsilon}$  is the deformation rate tensor,

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}[\nabla \mathbf{u} + (\nabla \mathbf{u})^T] \quad (7)$$

The fluids that do not obey the relationship expressed in Eq. 6 are known as non-Newtonian fluids. The main characteristic of these fluids is the dependence of viscosity on other flow parameters, such as, deformation rate and even the deformation history of the fluid. In these cases Eq. 6 can be rewritten as

$$\mathbf{T} = 2\mu(\dot{\gamma}) \boldsymbol{\varepsilon}(\mathbf{u}) \quad (8)$$

where  $\dot{\gamma}$  is the second invariant of the strain rate tensor and  $\mu(\dot{\gamma})$  is the apparent viscosity of the fluid [13,40].

In this work the non-Newtonian flows considered are viscoplastic fluids described by power law and Bingham models. The rheology models and non-Newtonian viscosity relations follow the definitions discussed in [2,12,19,40]; thus for the power law fluids we have,

$$\mu(\dot{\gamma}) = \begin{cases} \mu_0 K \dot{\gamma}^{n-1} & \text{if } \dot{\gamma} > \dot{\gamma}_0 \\ \mu_0 K \dot{\gamma}_0^{n-1} & \text{if } \dot{\gamma} \leq \dot{\gamma}_0 \end{cases} \quad (9)$$

where  $K$  denotes the consistency index,  $\mu_0$  is a nominal viscosity,  $n$  is the power law index and  $\dot{\gamma}_0$  is the cutoff value for  $\dot{\gamma}$ . For Bingham fluids we use the bi-viscosity model expressed as,

$$\mu(\dot{\gamma}) = \begin{cases} \mu_0 + \frac{\sigma_Y}{\dot{\gamma}} & \text{if } \dot{\gamma} > \frac{\sigma_Y}{\mu_r - \mu_0} \\ \mu_r & \text{if } \dot{\gamma} \leq \frac{\sigma_Y}{\mu_r - \mu_0} \end{cases} \quad (10)$$

where  $\sigma_Y$  is the yield stress,  $\mu_r$  is the Newtonian viscosity chosen to be at least one order of magnitude larger than  $\mu_0$ . Typically  $\mu_r$  is approximately  $100\mu_0$  to represent a true Bingham fluid behavior [2,5].

### 3 Finite Element Formulation

Let us assume following Tezduyar [56] that we have some suitably defined finite-dimensional trial solution and test function spaces for velocity and pressure,  $S_u^h$ ,  $V_u^h$ ,  $S_p^h$  and

$V_p^h = S_p^h$ . The finite element formulation of Eqs. 1 and 2 using SUPG and PSPG stabilizations for incompressible fluid flows [56] can be written as follows: find  $\mathbf{u}^h \in S_u^h$  and  $p^h \in S_p^h$  such that  $\forall \mathbf{w}^h \in V_u^h$  and  $\forall q^h \in V_p^h$ :

$$\begin{aligned} & \int_{\Omega} \mathbf{w}^h \cdot \rho(\mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}) \, d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \, d\Omega \\ & - \int_{\Gamma} \mathbf{w}^h \cdot h \, d\Gamma + \int_{\Omega} q^h \nabla \cdot \mathbf{u}^h \, d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{SUPG} \mathbf{u}^h \cdot \nabla \mathbf{w}^h \cdot [\rho(\mathbf{u}^h \cdot \nabla \mathbf{u}^h) \\ & \quad - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h) - \rho \mathbf{f}] \, d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{1}{\rho} \tau_{PSPG} \nabla q^h \cdot [\rho(\mathbf{u}^h \cdot \nabla \mathbf{u}^h) \\ & \quad - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h) - \rho \mathbf{f}] \, d\Omega = 0 \end{aligned} \quad (11)$$

In the above equation the first four integrals on the left hand side represent terms that appear in the Galerkin formulation of the problem (Eqs. 1–5), while the remaining integral expressions represent the additional terms which arise in the stabilized finite element formulation. Note that the stabilization terms are evaluated as the sum of element-wise integral expressions, where  $n_{el}$  is the number of elements in the mesh. The first summation corresponds to the Streamline Upwind/Petrov-Galerkin (SUPG) term and the second to the Pressure-Stabilizing/Petrov-Galerkin (PSPG) term. We have calculated the stabilization parameters according to [55], as follows:

$$\tau_{SUPG} = \tau_{PSPG} = \left[ \left( \frac{2 \|\mathbf{u}^h\|}{h^\#} \right)^2 + 9 \left( \frac{4\nu}{(h^\#)^2} \right)^2 \right]^{-1/2} \quad (12)$$

Here  $\mathbf{u}^h$  is the local velocity vector,  $\nu$  represent the kinematic viscosity and the ‘‘element length’’  $h^\#$  is defined to be equal to the diameter of the sphere which is volume-equivalent to the element.

The spatial discretization of Eq. 11 leads to the following system of nonlinear equations,

$$\begin{aligned} \mathbf{N}(\mathbf{u}) + \mathbf{N}_\delta(\mathbf{u}) + \mathbf{N}_\nu(\mathbf{u}) - (\mathbf{G} + \mathbf{G}_\delta) \mathbf{p} &= \mathbf{f}_u \\ \mathbf{G}^T \mathbf{u} + \mathbf{N}_\psi(\mathbf{u}) + \mathbf{G}_\psi \mathbf{p} &= \mathbf{f}_p \end{aligned} \quad (13)$$

where  $\mathbf{u}$  is the vector of unknown nodal values of  $\mathbf{u}^h$  and  $\mathbf{p}$  is the vector of unknown nodal values of  $p^h$ . The nonlinear vectors  $\mathbf{N}(\mathbf{u})$ ,  $\mathbf{N}_\delta(\mathbf{u})$ ,  $\mathbf{N}_\nu(\mathbf{u})$ , and  $\mathbf{N}_\psi(\mathbf{u})$ , the matrices  $\mathbf{G}$ ,  $\mathbf{G}_\delta$ , and  $\mathbf{G}_\psi$  emanate, respectively, from the convective, viscous and pressure terms. The vectors  $\mathbf{f}_u$  and  $\mathbf{f}_p$  are due to the boundary conditions 4 and 5. The subscripts  $\delta$  and  $\psi$  identify the SUPG and PSPG contributions respectively. In order to simplify the notation we denote by  $\mathbf{x} = (\mathbf{u}, \mathbf{p})$  a vector of nodal variables comprising both nodal velocities and pressures. Thus, Eq. 13 can be written as,

$$\mathbf{F}(\mathbf{x}) = \mathbf{0} \quad (14)$$

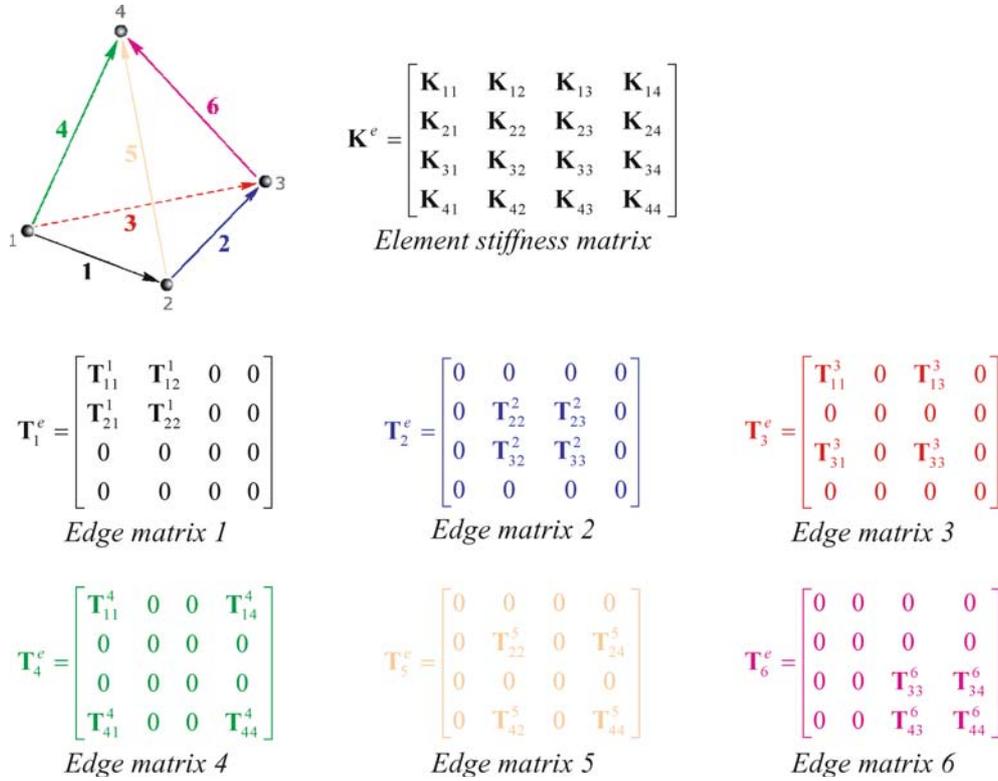
where  $\mathbf{F}(\mathbf{x})$  represents a nonlinear vector function.

A particularly simple scheme for solving the nonlinear system of Eq. 14 is a fixed point iteration procedure known as the *successive substitution*, (also known as Picard iteration, functional iteration or successive iteration) which may be written as

$$\mathbf{K}(\mathbf{x}_k) \mathbf{x}_{k+1} = -\mathbf{r}_k \quad (15)$$

here the nonlinearity is evaluated at the known iterate  $\mathbf{x}_k$  where  $\mathbf{K}$  is a first order approximation of the Jacobian  $\mathbf{F}'(\mathbf{x})$ ,  $\mathbf{r}_k$  is the residual vector, and a non-symmetric linear system must be formed and solved at each iteration. Although the convergence rate of this scheme can be slow (its convergence rate is only asymptotically linear), the method converges for a fair range of Reynolds numbers. This, together with its simplicity of implementation and relativity insensitivity to the initial iterate  $\mathbf{x}_0$ , accounts for its popularity and recommends its use for the solution of a wide variety of problems.

One drawback of nonlinear methods is the need to solve a local linear system at each stage. Computing the exact solution using a direct method can be expensive if the number



**Fig. 1** Edge disassembling of a tetrahedral finite element for viscoplastic flow problems

of unknowns is large and may not be justified when  $\mathbf{x}_k$  is far from a solution [13]. Thus, one might prefer to compute some approximate solution, leading to the following algorithm:

For  $k = 0$  step 1 until convergence do

Find some  $\eta_k \in [0, 1)$  AND  $\mathbf{s}_k$  that satisfy

$$\|\mathbf{r}_k + \mathbf{K}(\mathbf{x}_k)\mathbf{s}_{k+1}\| \leq \eta_k \|\mathbf{r}_k\|$$

$$\text{update } \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k \quad (16)$$

for some  $\eta_k \in [0, 1)$ , where  $\|\bullet\|$  is a norm of choice. This formulation naturally allows the use of an iterative solver like GMRES or BiCGSTAB: one first chooses  $\eta_k$  and then applies the iterative solver to Eq. 15 until a  $\mathbf{s}_k$  is determined for which the residual norm satisfies Eq. 16. In this context  $\eta_k$  is often called the forcing term, since its role is to force the residual of Eq. 15 to be suitably small. The forcing term can be specified in several ways (see, Eisenstat and Walker [16] for details) and in this work the following definition was employed

$$\eta_k = 0.9 \|\mathbf{F}(\mathbf{x}_k)\|^2 / \|\mathbf{F}(\mathbf{x}_{k-1})\|^2. \quad (17)$$

#### 4 Edge-Based Data Structures

Edge-based finite element data structures have been introduced for explicit computations of compressible flow in unstructured grids composed by triangles and tetrahedra

[32,43]. It was observed in these works that residual computations with edge-based data structures were faster and required less memory than standard element-based residual evaluations. Following these ideas, Coutinho et al. [11] and Catabriga and Coutinho [9] derived edge-based formulations respectively for elasto-plasticity and the SUPG formulation for inviscid compressible flows. They used the concept of disassembling the finite element matrices to build the edge matrices. For three dimensional viscoplastic flow problems on unstructured meshes, we may derive an edge-based finite element scheme by noting that the element matrices can be disassembled into their edge contributions as,

$$\mathbf{K}^e = \sum_{s=1}^m \mathbf{T}_s^e \quad (18)$$

where  $\mathbf{T}_s^e$  is the contribution of edge  $s$  to  $\mathbf{K}^e$  and  $m$  is the number of edges per element (six for tetrahedral). For instance, Fig. 1 shows a tetrahedron and its edge disassembling into its six edges.

In Fig. 1, each  $\mathbf{T}_{ij}^s$  is a  $n_{\text{dof}} \times n_{\text{dof}}$  sub-matrix, where  $n_{\text{dof}}$  is the number of degrees of freedom (four for fully coupled incompressible fluid flow). The arrows represent the adopted edge orientation. Denoting by  $E$  the set of all elements sharing a given edge  $s$ , we may add their contributions, arriving to the edge matrix,

$$\mathbf{K}_s = \sum_{s \in E} \mathbf{T}_s^e \quad (19)$$

The resulting matrix for incompressible fluid flow is non-symmetric, preserving the structure of the edge matrices given in Fig. 1. Thus, in principle we need to store two off-diagonal  $4 \times 4$  blocks and two  $4 \times 4$  diagonal blocks per edge. As we shall see below this storage area can be reduced.

When working with iterative solvers like GMRES, it is necessary to compute sparse matrix-vector products at each iteration. A straightforward way to implement the edge-by-edge (EDE) or element-by-element (EBE) matrix-vector product is,

$$\mathbf{K}\mathbf{p} = \sum_{l=1}^{n_e} \mathbf{K}^l \mathbf{p}^l \quad (20)$$

where  $n_e$  is the total number of local structures (edges or elements) in the mesh and  $\mathbf{p}^l$  is the restriction of  $\mathbf{p}$  to the edge or element degrees-of-freedom.

Considering that we may add all the local nodal block diagonals in a global nodal block diagonal matrix and we may store only the off-diagonal blocks (by edges or elements), it is possible to improve the matrix-vector product (Eq. 20) disregarding redundant nodal block diagonal coefficients for each local structure [9]. This scheme is an extension of that proposed by van Gijzen [57] where is considered only the main diagonal instead of the nodal block diagonal. Thus, the matrix-vector product may be rewritten as

$$\mathbf{K}\mathbf{p} = \mathbf{B}(\mathbf{K})\mathbf{p} + \sum_{l=1}^{n_e} \mathbf{A} [\mathbf{K}^l - \mathbf{B}(\mathbf{K}^l)] \mathbf{p}^l \quad (21)$$

where  $\mathbf{B}(\mathbf{K})$  stores the nodal block diagonal of  $\mathbf{K}$  and  $\mathbf{A}$  is the assembling operator. In the above matrix-vector product the first sum involves only global quantities while the second is very similar to the standard matrix-vector product (Eq. 20). Note that we need to store only the off-diagonal blocks associated to each structure (edge or element), thus reducing memory requirements. It is also important to observe that in the case of the edge-based matrix-vector product (Eq. 21) only global coefficients are involved in the computations. The resulting algorithm is,

*For each local structure l do :*

*Recover the global numbering of the local degrees of freedom.*

*Gathering operation :  $\mathbf{p}^l \leftarrow \mathbf{p}$*

*Perform product :  $\mathbf{kp}^l = [\mathbf{K}^l - \mathbf{B}(\mathbf{K}^l)] \cdot \mathbf{p}^l$*

*Scattering and accumulation operations :  $\mathbf{kp} \leftarrow \mathbf{kp} + \mathbf{kp}^l$*

*end for l*

$\mathbf{kp} = \mathbf{B}(\mathbf{K})\mathbf{p} + \mathbf{kp}$

The  $\mathbf{B}(\mathbf{K})\mathbf{p}$  product performed at nodal level follows the same ideas described in the above algorithm (gather, local computations and scatter and accumulation). Considering the standard pointers, localization matrices and destination arrays of finite element implementations [26] applied to edge data structures, it is easy to verify that pointers, localization matrices and destination arrays correspond to a two-node element, that is, an edge. In Table 1 we compare the storage

**Table 1** Memory to hold matrix coefficients and computational costs for element and edge-based matrix-vector products for tetrahedral finite element meshes

Data structure	Memory	flop	i/a
Element	$1,056 n_{\text{nodes}}$	$2,112 n_{\text{nodes}}$	$1,408 n_{\text{nodes}}$
Edge	$224 n_{\text{nodes}}$	$448 n_{\text{nodes}}$	$448 n_{\text{nodes}}$

requirements to hold the coefficients of the element and edge matrices as well as the number of floating point (flop) and indirect addressing (i/a) operations for computing matrix-vector products (Eq. 21) using element and edge-based data structures.

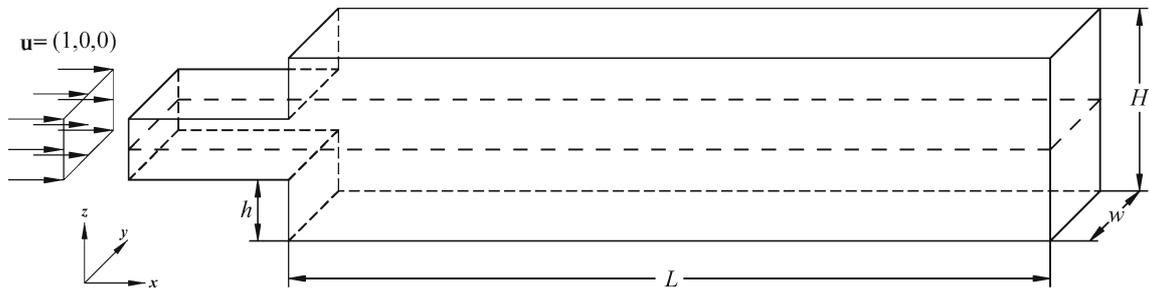
All data in this table is referred to  $n_{\text{nodes}}$ , the number of nodes in the finite element mesh. According to [31], the following estimates are valid for unstructured 3D grids,  $n_{\text{el}} \approx 5.5 \times n_{\text{nodes}}$ ,  $n_{\text{edges}} \approx 7 \times n_{\text{nodes}}$ .

Clearly data in Table 1 favors the edge-based scheme. However, compared to the element data structure, the edge scheme does not present a good balance between flop and i/a operations. Minimizing indirect addressing and improving data locality are major concerns to achieve high performance on current processors. Löhner and Galle [34] and Coutinho et al. [36] discuss several enhancements to the simple edge scheme introduced here, that mitigate the effects of indirect addressing and bad data locality. However they were not employed in this work.

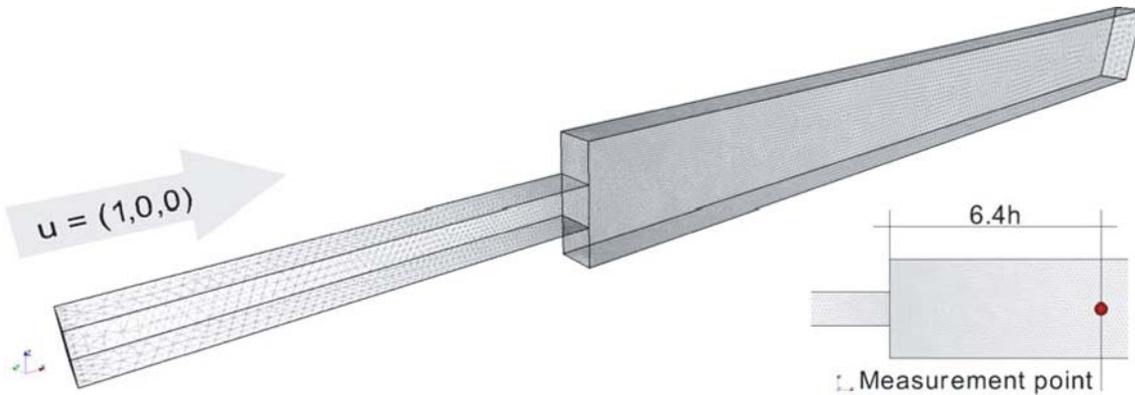
## 5 Parallel Implementation

The edge-based finite element formulation presented in the previous sections was implemented based in the message passing parallelism model (MPI) [14]. The original unstructured grid was partitioned into non-overlapped sub-domains by the use of the METIS\_PartMeshDual routine provided by Metis package [28]. Afterwards, the partitioned data was reordered to avoid indirect memory addressing and IF clauses inside hot loops and MPI communications. Therefore, the equation numbers shared by the partitions were relocated to the last entries of the corresponding arrays. The edge extraction was also parallelized at runtime. Thus, the element partitions were employed at each processor to build a local set of edges. The element incidences at each processor were kept to allow residual computations due to the material nonlinearities. The residual vector (Eq. 15) was computed and assembled elementwise in a single DO-LOOP in each partition.

Most of the computational effort spent during the iterative solution of linear systems is due to evaluations of matrix-vector products or matvec for short. In our tests matvec operations achieved 92% of the total computational costs. In EBE and EDE data structures this task is message passing parallelizable by performing matvec operations at each partition level, then assembling the contribution of the interface equations calling MPIAllReduce routine over the last array entries. Finally, it is important to note that edge (and element)



**Fig. 2** Schematic model of a channel with sudden expansion (1:3),  $h = 1$ ,  $H = 3$ ,  $w = 1$  and  $L = 30$



**Fig. 3** Finite element mesh with 382,985 tetrahedron elements and 72,593 nodes

matrix coefficients are computed in single DO-LOOPS also in each partition.

(45–100) corresponds to the range of Krylov vectors employed.

## 6 Numerical Results

In the following sections two benchmark problems employing power law and viscoplastic fluids are shown to evaluate the performance improvements, due the edge-based data structure, and accuracy of our implementation. The first problem consists in a 1:3 sudden expansion for power law fluids where a phenomenon of symmetry breaking is observed after a critical Reynolds number is reached. The second problem is the well known lid driven cavity problem extended for the three dimensional case where a Bingham fluid flows and the eye vortex position and rigid zones are used as validation parameters.

All computations were made on two SGI Altix 3700 systems (32/64 Intel Itanium-2 CPUs with 1.3/1.5 GHz and 64/128 Gb of NUMA flex memory) and a Quad Dell PowerEdge Itanium-2 1.3 GHz system with 8 Gb of memory. All systems run Linux and Intel Fortran compiler 8.1. No optimizations further those provided by standard compiler flags were used.

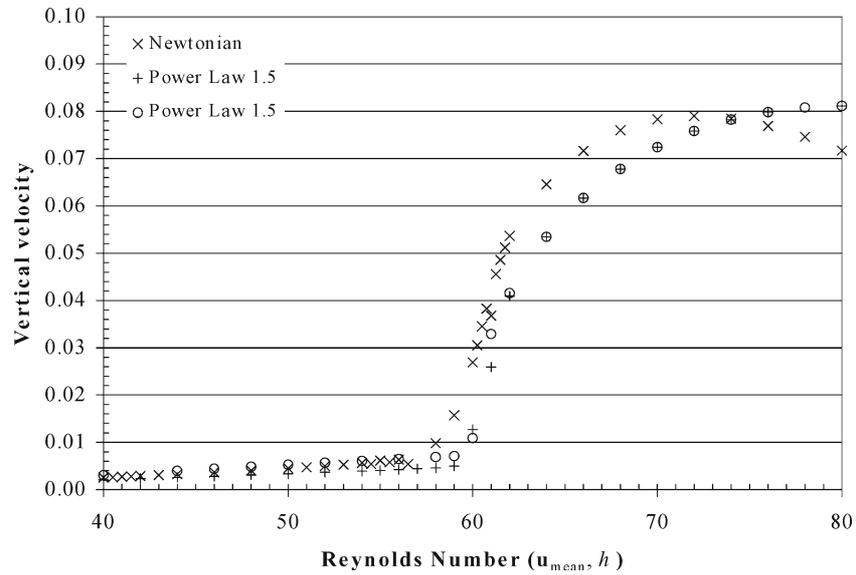
For all tests the numerical procedure considered a fully coupled  $\mathbf{u}$ - $p$  version of the stabilized formulation using linear tetrahedron elements. The parallel solver is composed by an outer inexact-Newton loop and an inner nodal block diagonal preconditioned GMRES(45-100) linear solver, where

### 6.1 Sudden expansion flow for power-law fluids

Considerable attention has been given to the problem of a fluid flow through a channel with a sudden expansion for Newtonian fluids. In this problem, after a critical Reynolds number is reached an instability region develops, a symmetry bifurcation begins to appear and the problem presents a pseudo-steady behavior. This phenomenon is frequently referred as symmetry breaking and has been studied for several authors (see [4, 15, 18, 21, 39, 49] for details).

We have tested the expansion ratio of 1:3 and the schematic model and finite element mesh are shown in Figs. 2 and 3. The unstructured mesh employed was built with 382,985 tetrahedron elements, 72,593 nodes and 473,383 edges, yielding a system with 260,782 equations. The boundary conditions applied were: inlet mean velocity equal to one, no slip boundary condition in the top and bottom walls, no leakage condition at the side walls (the normal velocity was set to zero) and null traction condition at the outlet. For this problem we have considered the Reynolds number as  $Re = \rho u^{2-n} h^n / K$ , where  $u$  is the inlet mean velocity,  $h$  is the inlet height,  $n$  is the power law index (1 for Newtonian fluid) and  $K$  is the power law consistency index (equal the viscosity for Newtonian cases).

Following Fearn et al. [18] we have determined the symmetry breaking when a non null vertical velocity is detected

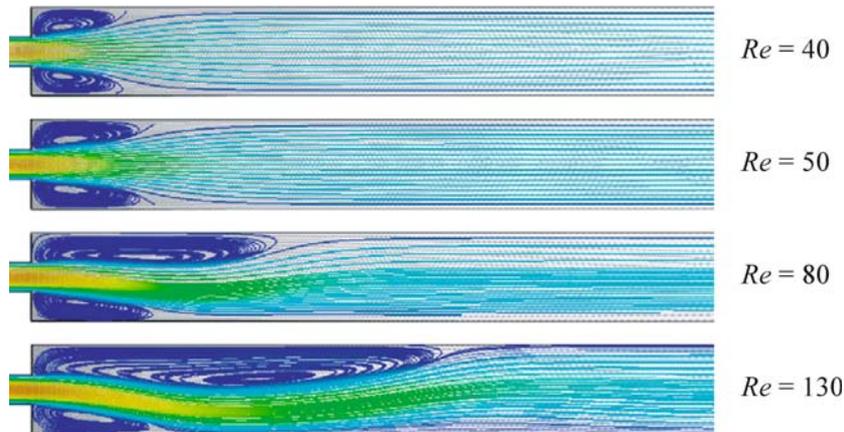


**Fig. 4** Vertical velocity for Newtonian, pseudoplastic ( $n = 0.5$ ) and dilatant ( $n = 1.5$ ) fluids in a measurement point placed at 6.4 h

**Table 2** Critical Reynolds number for symmetry breaking in the 1:3 expansion

	Definition of $Re$	$Re_{cr}$	$Re_{cr} (u_{mean}, h)$
Fearn et al. [18] <sup>a</sup>	$u_{max}, h/2$	40.45/47.3	53.9/63.1
Manica and De Bortoli [35]	$u_{max}, h/2$	44.0–45.0	58.7–60.0
Shapira et al. [49]	$u_{max}, h$	82.6	55.0
Drikakis [15]	$u_{max}, h$	80.0	53.3
Battaglia et al. [4]	$u_{mean}, h$	53.8	53.8
Hawa and Rusak [21]	$u_{mean}, h$	53.8	53.8
This work	$u_{mean}, h$	56.0	56.0

<sup>a</sup>Experimental/simulated

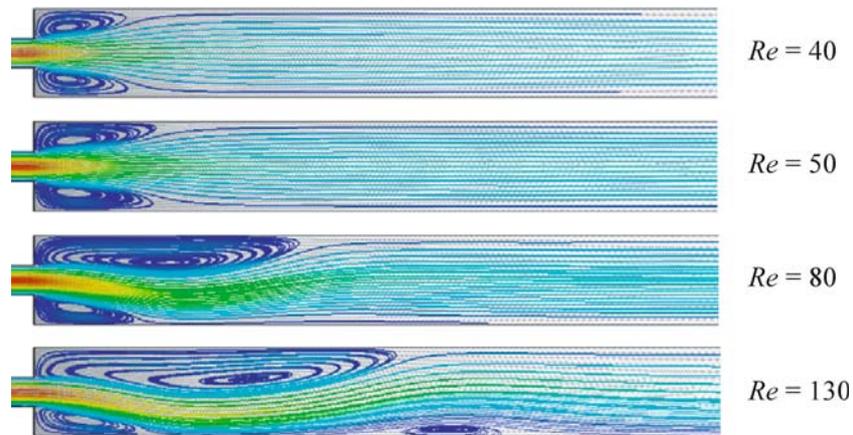


**Fig. 5** Sudden expansion streamlines for the pseudo-plastic case ( $n = 0.5$ )

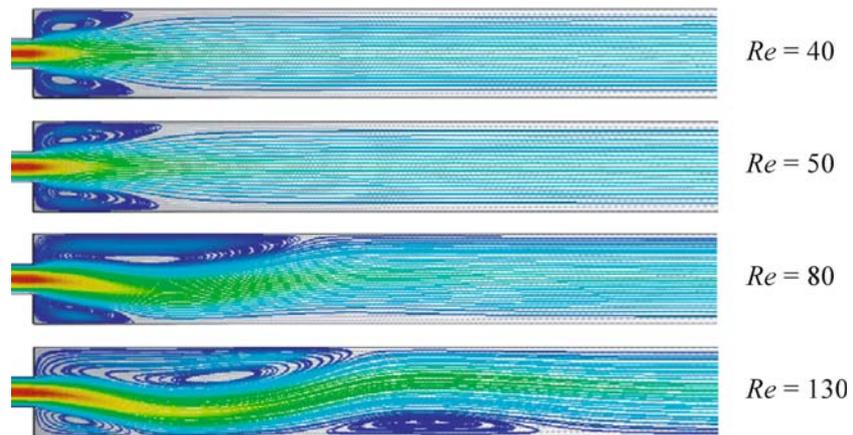
in a measurement point placed at 6.4 h from the expansion as shown in Fig. 3. The magnitude of the vertical velocity at this measurement station is plotted for Newtonian, pseudo-plastic ( $n = 0.5$ ) and dilatant ( $n = 1.5$ ) cases in Fig. 4. In Table 2 we list the characteristic velocity and height used by several authors to compute the critical Reynolds number, where  $u_{max}$  and  $u_{mean}$  are respectively the inlet maximum and mean velocity. In the last two columns we give the critical

Reynolds number computed by the different authors and with the characteristic parameters adopted here.

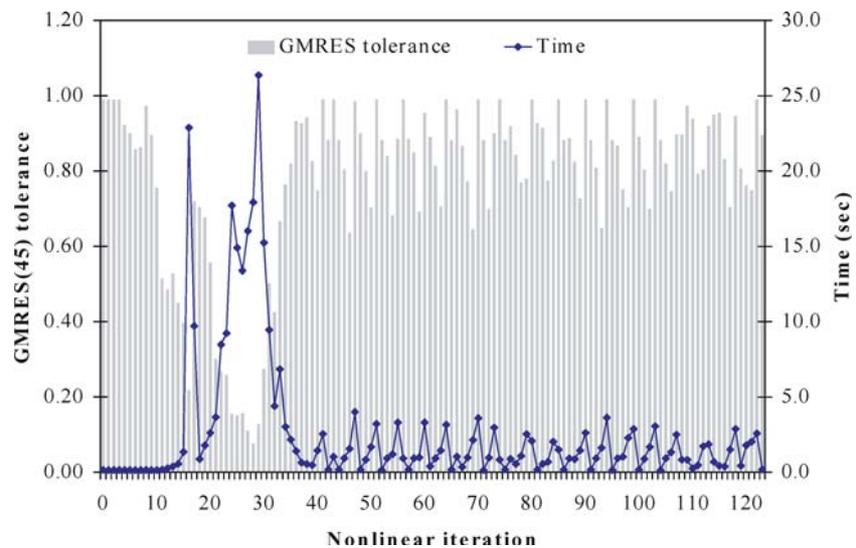
Note in Table 2 that the critical Reynolds number of the present work for the Newtonian case is in good agreement with those listed for other authors. Figure 4 shows that for power law cases the critical Reynolds numbers for the first bifurcation were around the same value than for the Newtonian case, which is consistent with earlier results of



**Fig. 6** Sudden expansion streamlines for the Newtonian case ( $n = 1.0$ )



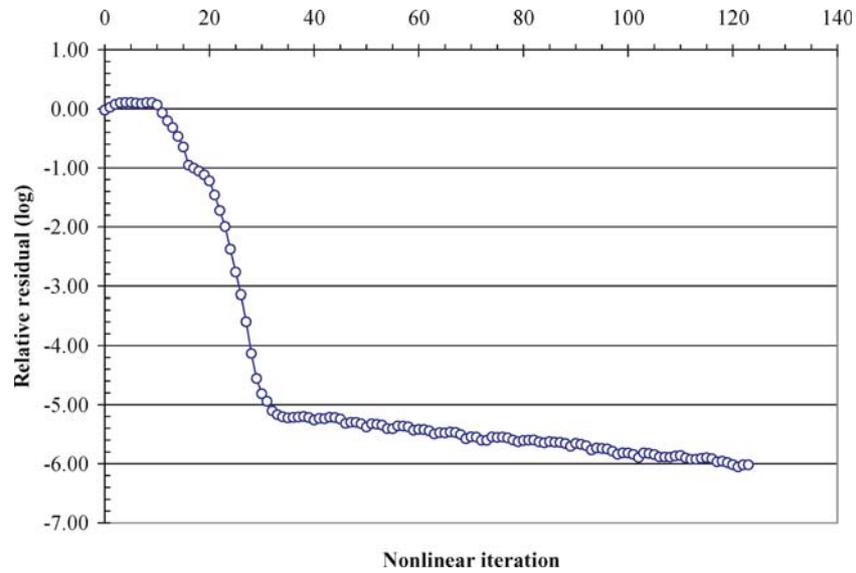
**Fig. 7** Sudden expansion streamlines for the dilatant case ( $n = 1.5$ )



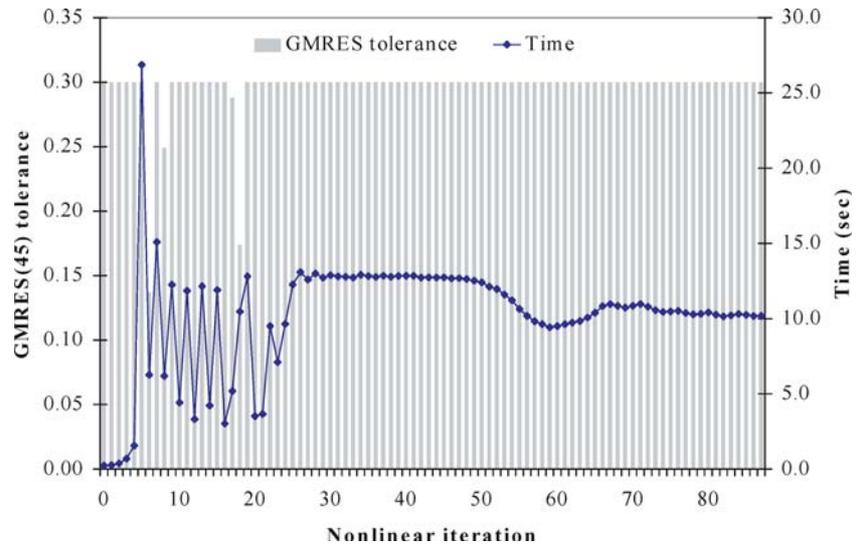
**Fig. 8** GMRES tolerance and nonlinear iteration time for Newtonian sudden expansion flow with Reynolds 50

Manica and De Bortoli [35]. Figures 5, 6, 7 show the streamlines for Reynolds 40, 50, 80 and 130 for dilatant, Newtonian and pseudo-plastic cases respectively. We may verify that

the symmetry breaking is clearly characterized in all cases. Further, a third vortex appears for  $n = 1.0$  and  $n = 1.5$ , while for  $n = 0.5$  it is not present. These observations are



**Fig. 9** Convergence history for Newtonian sudden expansion flow with Reynolds 50



**Fig. 10** GMRES tolerance and nonlinear iteration time for power law ( $n = 1.50$ ) sudden expansion flow with Reynolds 80

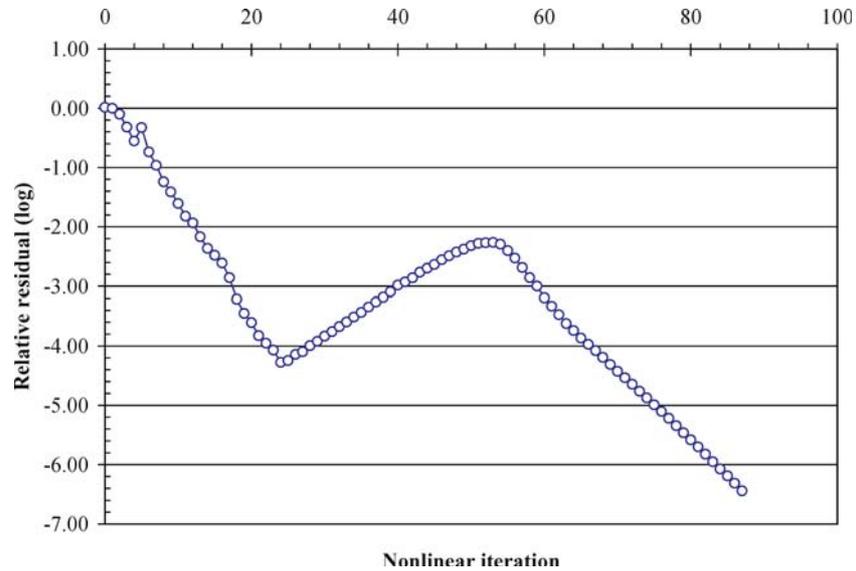
in good agreement with those presented by Manica and De Bortoli [35].

Figures 8 and 9 describe the behavior of the inexact nonlinear solver for the Newtonian case at Reynolds 50. Note that at the beginning of the computation the large linear tolerance (0.99) allowed fast nonlinear solution steps and, according to the ratio of relative residual decrease, the linear tolerance was adapted up to 0.075, at nonlinear iteration 28, increasing again until convergence. The resulting convergence history is shown in Fig. 9, where we can observe a superlinear decrease of the relative residual Euclidean norm up to  $10^{-5}$ , followed by a linear progression towards the desired accuracy ( $10^{-6}$ ).

The GMRES tolerances, nonlinear iteration times and the convergence history for the dilatant case at Reynolds 80 are

plotted in Figs. 10 and 11. For this case a maximum GMRES tolerance of 0.3 was employed to allow a convergent solution. The smallest tolerance adapted by the inexact nonlinear solver was 0.137 and occurred at step 6, where the relative residual suddenly decreased from step 5 to 6. At this nonlinear step GMRES(45) spent 26.87 s to reach the prescribed accuracy. At all other iterations the solution procedure at each nonlinear step spent on the average 10 s to be solved for tolerances around the maximum specified value.

Table 3 lists the characteristic solution parameters for the cases shown in Figs. 5, 6, and 7. All computations were halted when the relative residual ( $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$ ) and the relative solution increment ( $\|\mathbf{s}_k\|/\|\mathbf{x}_k\|$ ) decreased six orders of magnitude. We may see that the easiest cases were those involving Newtonian and dilatant fluids. For non-Newtonian cases, at



**Fig. 11** Convergence history for power law ( $n = 1.50$ ) sudden expansion flow with Reynolds 80

**Table 3** Solution parameters for the 1:3 sudden expansion flows

	Nonlinear iterations	Newtonian iterations	Final relative residual	GMRES tolerance	
				Maximum	Minimum <sup>a</sup>
Pseudo-plastic ( $n = 0.5$ )					
$Re = 40$	45	5	$1.64 \times 10^{-7}$	0.100	0.052 (8)
$Re = 50$	48	5	$2.00 \times 10^{-7}$	0.100	0.049 (7)
$Re = 80$	222	5	$2.11 \times 10^{-7}$	0.100	0.047 (7)
$Re = 130$	109	5	$4.74 \times 10^{-7}$	0.100	0.100 (-)
Newtonian					
$Re = 40$	34	-	$9.37 \times 10^{-7}$	0.990	0.075 (25)
$Re = 50$	124	-	$9.56 \times 10^{-7}$	0.990	0.076 (28)
$Re = 80$	385	-	$6.20 \times 10^{-7}$	0.990	0.066 (84)
$Re = 130$	832	-	$9.89 \times 10^{-7}$	0.990	0.046 (311)
Dilatant ( $n = 1.5$ )					
$Re = 40$	42	3	$8.71 \times 10^{-7}$	0.990	0.226 (13)
$Re = 50$	43	2	$7.19 \times 10^{-7}$	0.990	0.172 (35)
$Re = 80$	87	5	$3.60 \times 10^{-7}$	0.300	0.137 (7)
$Re = 130$	117	5	$6.73 \times 10^{-7}$	0.300	0.150 (10)

<sup>a</sup>The number between parentheses is the nonlinear iteration that the minimum GMRES tolerance occurred

the beginning of the solution procedure the domain does not have a velocity distribution to compute a non null shear stress. We have observed that this may lead to some initialization problems. For instance, for power law fluids, the evaluation of Eq. 10 leads to a constant viscosity value ( $\mu_0$ ) and for Bingham fluids a null shear stress means that the fluid flow cutoff value was not achieved and the domain is filled with a high and “rigid” viscosity. Therefore, to circumvent these initialization problems we have run some initial iterations with Newtonian fluid before activate the non-Newtonian viscosity evaluation as given in Table 3. It is important to note that the cases for Reynolds numbers greater than 50 spent more nonlinear iterations to converge. It coincides with the

range where the symmetry breaking begins to appear. We may observe that the minimum GMRES tolerances have occurred at the beginning of the solution process in all cases. It points to convergence ratios better at the initial nonlinear iterations than at the end when the non-Newtonian rheology has more influence.

For the pseudo-plastic cases listed in Table 3 the maximum linear tolerance in the inexact nonlinear solver was decreased up to 0.1 to allow convergent solutions. Most of the cases employed 45 Krylov vectors for GMRES; the exceptions were the cases with Reynolds numbers 40 and 130 for pseudo-plastic and dilatant fluids that we have used 100 and 55 Krylov vectors respectively.

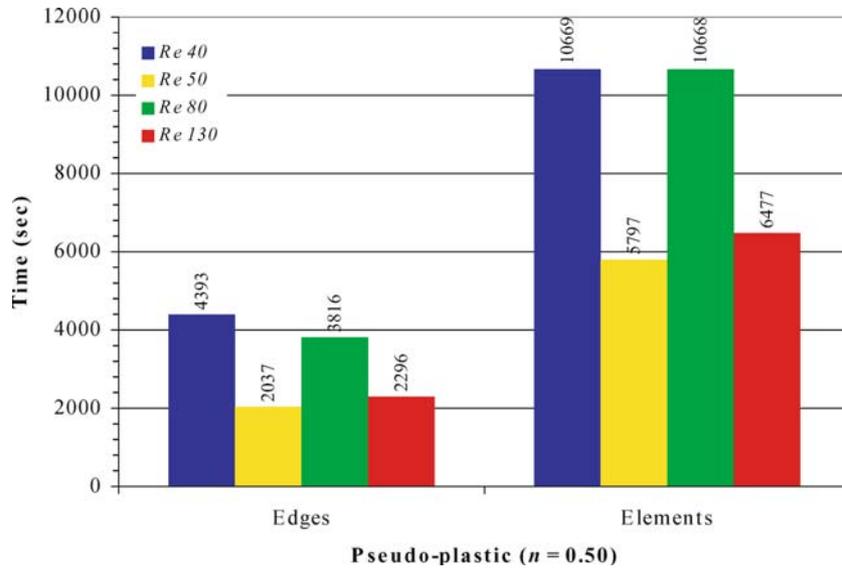


Fig. 12 Data structure performance and influence of the Reynolds number for pseudo-plastic flow in a sudden expansion

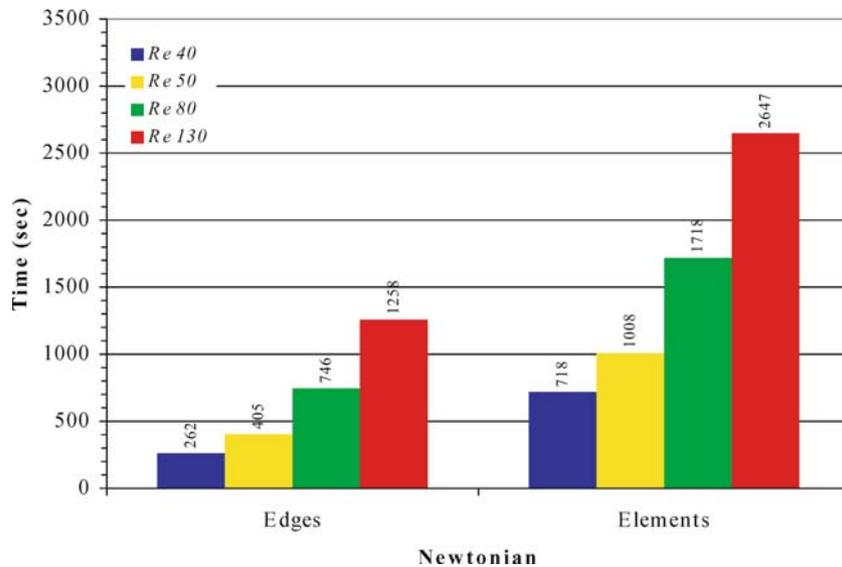


Fig. 13 Data structure performance and influence of the Reynolds number for Newtonian flow in a sudden expansion

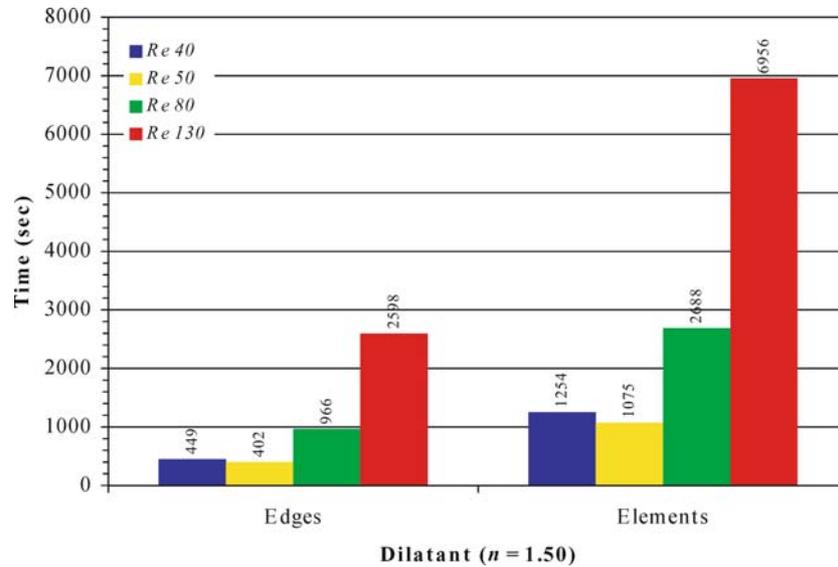
Performance comparisons for edge and element data structures for the rheologies considered are depicted in Figs. 12, 13, and 14. The performance improvement provided by the edge-based data structure may be clearly seen for all cases studied. The solution time for problems employing edges in the matrix-vector products were, on average, 2.5 faster when compared with those using standard EBE matvec computations.

### 6.2 Leaky lid-driven cavity for Bingham fluids

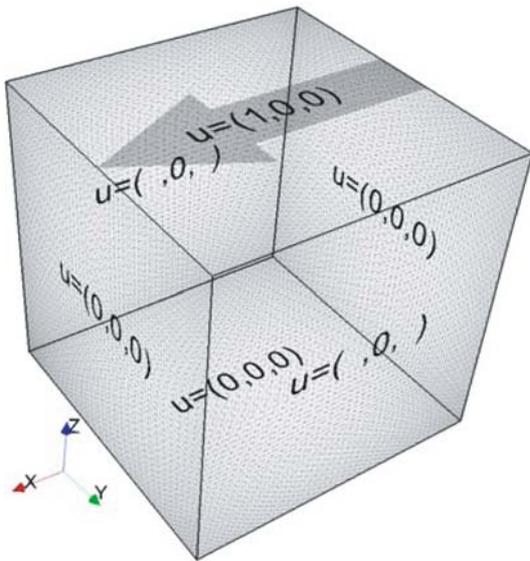
The problem of a fluid flowing in a leaky lid-driven cavity has been extensively used to evaluate CFD codes with Newtonian fluids for many years. In this section we present

our contribution for this problem extended to three dimensional cases employing Bingham fluids. Some authors have recently studied this problem for viscoplastic fluids. Mitsoulis and Zisis [38] studied the position of the main vortex eye and the evolution of rigid zones for Reynolds number equal to one. These authors used the Papanastasiou [41] modification to represent the Bingham constitutive equation with bilinear quadratic finite elements. Vola et al. in [58] have proposed a numerical method based on the characteristic Galerkin formulation and the Fortin–Glowinski decomposition to deal with non differentiable terms of the constitutive law. They study the unsteady Bingham flows for Reynolds numbers up to 1,000 without any regularization of the constitutive law.

The 3D model used in the present work and the corresponding boundary conditions are depicted in Fig. 15.



**Fig. 14** Data structure performance and influence of the Reynolds number for dilatant flow in a sudden expansion



**Fig. 15** Flow in a three dimensional leaky lid-driven cavity—geometry and boundary conditions

**Table 4** Meshes for the three dimensional leaky lid-driven cavity

	Elements	Edges	Nodes	Equations
<i>cav-31</i>	148,955	187,488	32,768	117,367
<i>cav-51</i>	663,255	819,468	140,608	525,556
<i>cav-71</i>	1,789,555	2,193,048	373,248	1,421,776
<i>cav-101</i>	5,151,505	6,273,918	1,061,208	4,101,106

The meshes employed to evaluate parallel performance are listed in Table 4.

Figure 16 shows the Bingham viscosity and streamlines for several Reynolds numbers and plastic threshold values computed with mesh *cav-51*. We may observe that for higher plastic threshold values the rigid zones, where the fluid is

highly viscous, are proportionally larger and the vortex eye tends to be pushed up to the lid of the cavity. Thus, as the viscous zone increases the cavity seems to be smaller and the fluid is forced to flow at the top region of the model. Furthermore, the vortex eye is displaced to the outflow direction according to the Reynolds number. For higher Reynolds number the vortex is displaced to the leakage direction as shown in Fig. 16 for the cases with Reynolds number equal to 1,000.

For the cases shown in Fig. 16 we have carried out the computations up to a relative residual ( $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$ ) and relative solution increment ( $\|\mathbf{s}_k\|/\|\mathbf{x}_k\|$ ) decrease of four orders of magnitude or 1,000 nonlinear iterations were reached. It was not necessary maximum GMRES(45) tolerances smaller than 0.99 to develop convergent solutions for any case considered and the inexact nonlinear solver has adapted the tolerance up to the minimum of 0.251. However, the maximum number of nonlinear iterations was reached in some cases as shown in Table 5. Again we observed that the minimum GMRES tolerances have occurred at the beginning of the solution process in all cases.

As suggested in previous works (see [58,38]) we have validated our results with the position of the main vortex eye inside the cavity. The  $x$  and  $z$  coordinates of the main vortex eye for several plastic thresholds at Reynolds number equal to 1,000 are shown in Fig. 17. We may note a good agreement between the results of this work (dots) with those presented by Vola et al. [58] (lines).

In Fig. 18 (left) is shown the scaled speedup on the SGI Altix computed according to Gustafson law [20] and defined as  $S_s = n + (1 - n)s$ , where  $n$  is the number of processors and  $s$  corresponds to the normalized time spent in the serial portion of the program. For these performance tests we have ran the problem with  $Re = 1,000$  and plasticity threshold 10 up to a relative residual decrease of three orders of magnitude with the models listed in Table 4. The scalability

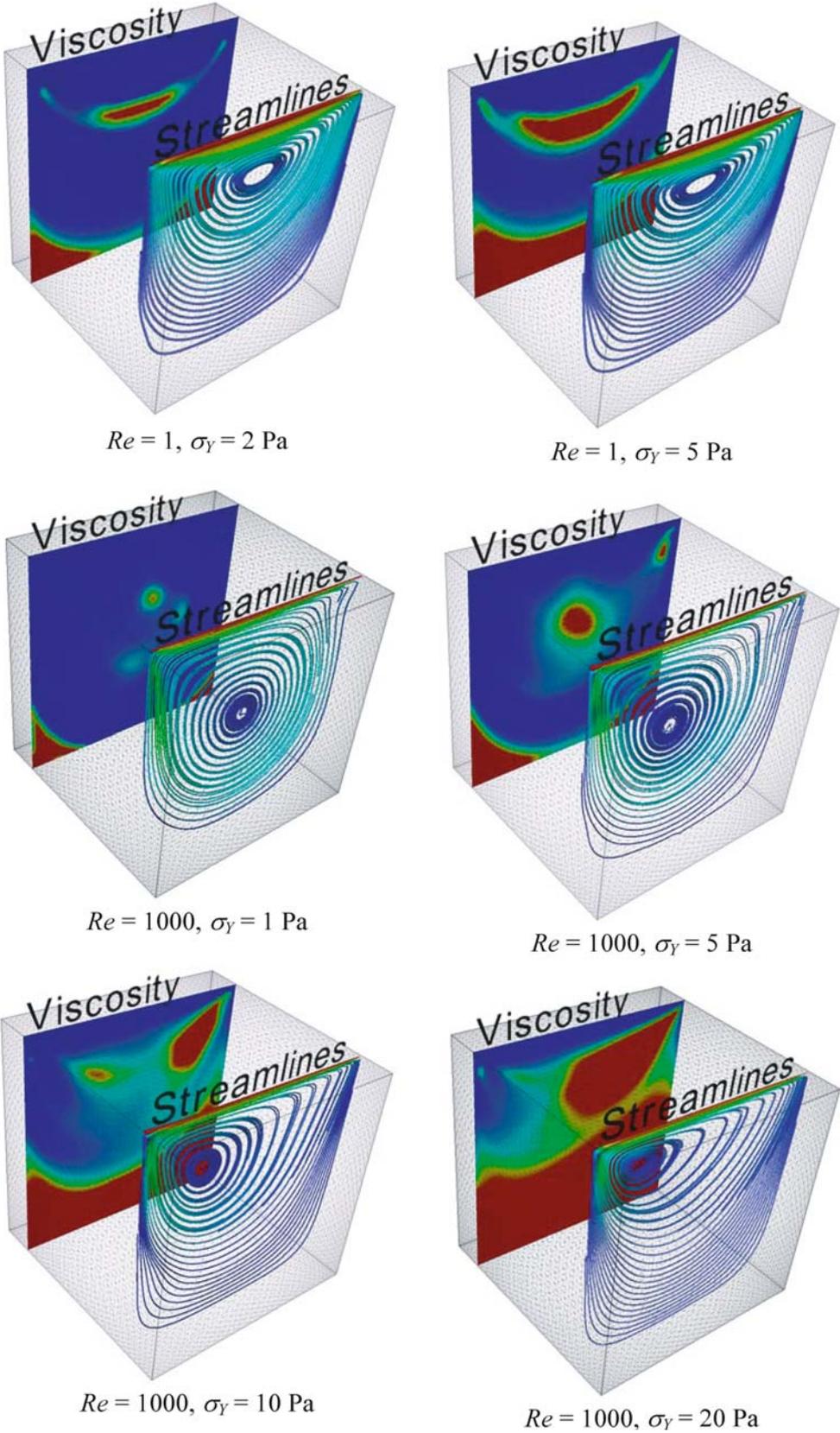
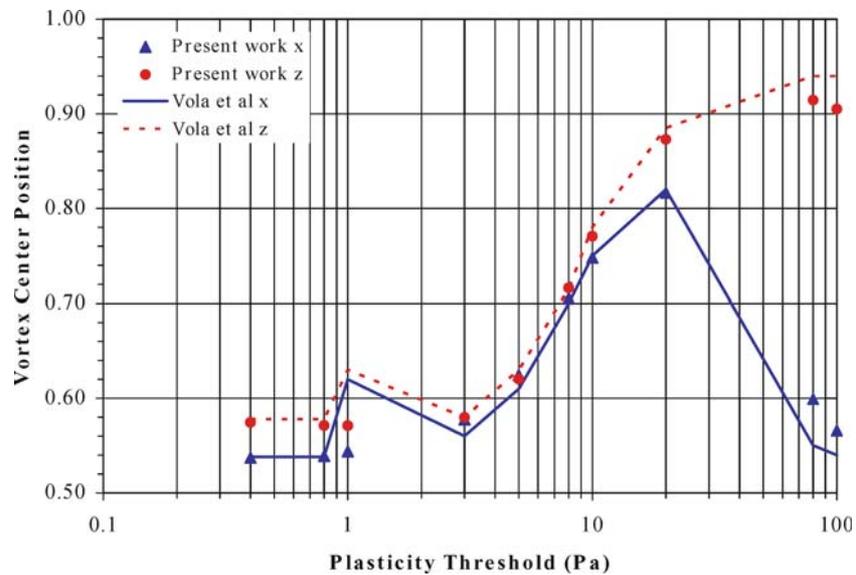
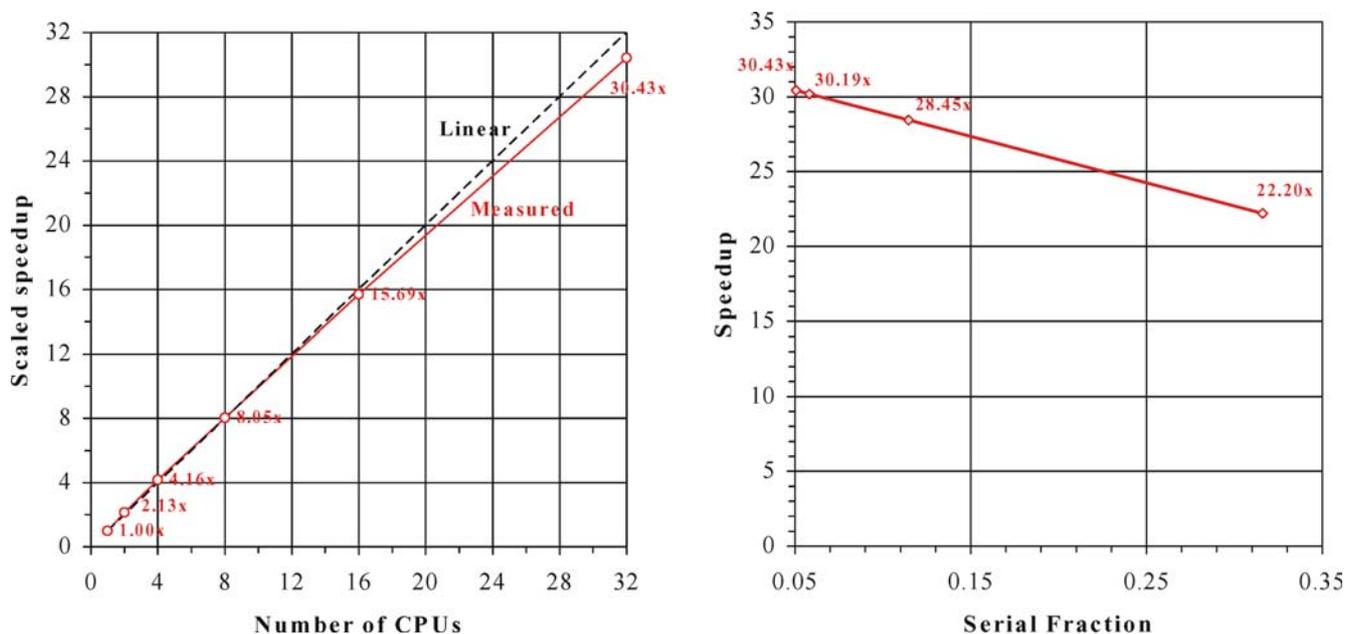
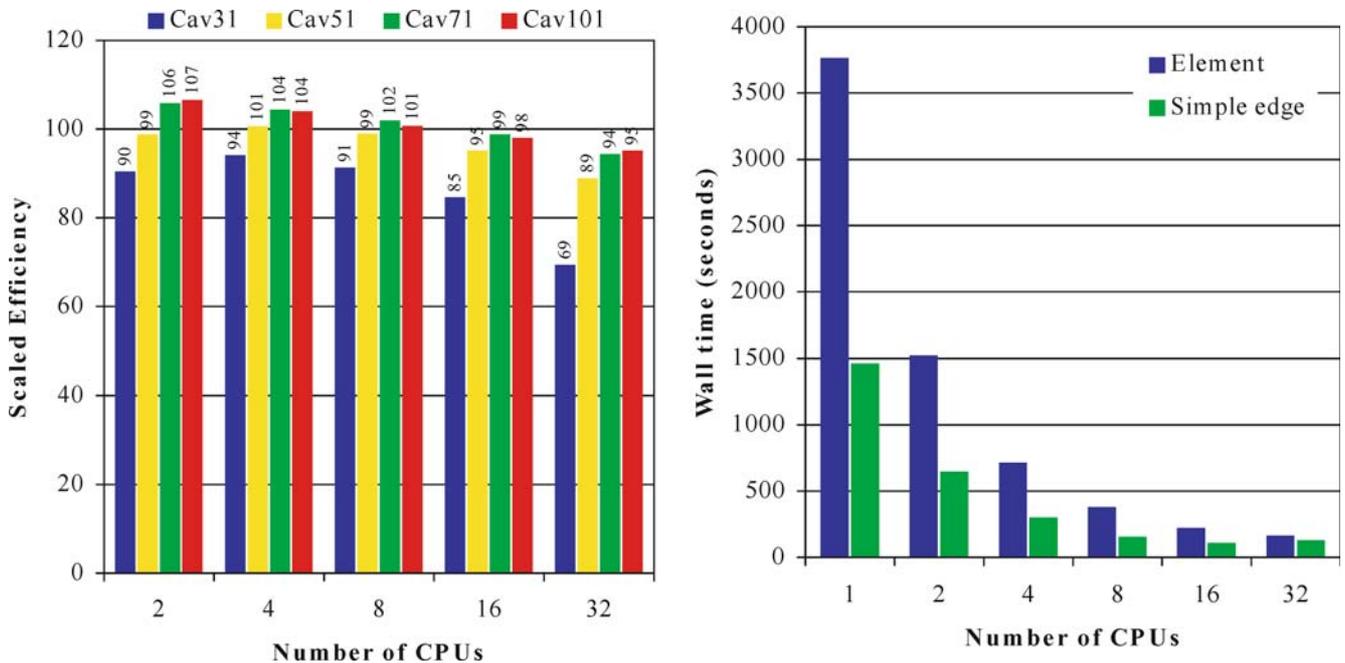


Fig. 16 Bingham viscosity and streamlines for the lid driven cavity flow (cav-51 model)

**Table 5** Solution control parameters for the Bingham leaky lid-driven cavity flow

	Nonlinear iterations	Newtonian iterations	Final relative residual	GMRES tolerance	
				Maximum	Minimum
$Re = 1$					
$\sigma_Y = 2$ Pa	1,000	5	$5.05 \times 10^{-4}$	0.99	0.320 (7)
$\sigma_Y = 5$ Pa	1,000	5	$6.75 \times 10^{-4}$	0.99	0.337 (6)
$Re = 1000$					
$\sigma_Y = 1$ Pa	382	2	$9.63 \times 10^{-5}$	0.99	0.320 (7)
$\sigma_Y = 5$ Pa	1,000	2	$5.56 \times 10^{-4}$	0.99	0.251 (10)
$\sigma_Y = 10$ Pa	572	2	$9.72 \times 10^{-5}$	0.99	0.414 (11)
$\sigma_Y = 20$ Pa	740	1	$9.86 \times 10^{-5}$	0.99	0.551 (5)

**Fig. 17** Position of the vortex eye according to the plasticity threshold for  $Re = 1,000$  (*cav-5I* model)**Fig. 18** Scaled speedup on SGI Altix for EDE data structure and *cav-10I* model (*left*). Scalability for the models listed in Table 4 (*right*)



**Fig. 19** Scaled efficiency for EDE data structure on SGI Altix (left). Wall time comparisons for EBE and EDE data structures for *cav-71* mesh (right)

reached on SGI Altix is shown in Fig. 18 (right). Note that when increasing the problem size the serial fraction  $s$  tends to shrink as more processors are employed. In our tests with SGI Altix 3700 we have employed up to 32 Intel's Itanium-2 processors and according to [20] the scaled speedup should be a linear function with moderate slope  $1 - n$  such as the line we have measured and shown in Fig. 18.

The scaled efficiency on SGI Altix for EDE data structure is presented in Fig. 19 (left). Good results may be observed for the cavity models, especially for those with larger number of degrees of freedom. In some cases efficiencies greater than 100% may be attributed to cache effects. The time spent when solving the *cav-71* problem with EBE and EDE data structures is plotted in Fig. 19 (right). We may observe that the EDE solutions were faster than the EBE in all cases. Nevertheless, the CPU time ratios between EBE and EDE solutions are around 2.5 up to 16 processors. For 32 processors this ratio decreases. This is an indication that as we refine the mesh, CPU time ratios between EBE and EDE has a tendency to remain around this value.

## 7 Conclusions

The SUPG/PSPG finite element formulation for three dimensional incompressible viscoplastic fluid flows was presented and the results validated with those provided by several works. The nonlinear character due to the non-Newtonian viscous and convective terms of the Navier–Stokes equations was treated by an inexact-nonlinear method allowing a good trade-off between convergence and computational effort. At the

beginning of the solution procedure the large linear tolerances produced fast nonlinear steps, and as the solution progresses, the inexact nonlinear method adapts the tolerances to reach the desired accuracy. The linear systems of equations within the nonlinear solution procedure were solved with the nodal-block diagonal preconditioned GMRES. An edge-based data structure was introduced and successfully employed to improve the performance of the matrix-vector products within the iterative solver. The results showed that the computing time when using EDE data structure was on the average 2.5 times faster than for those problems using standard EBE. The computations were performed in a message passing parallelism environment presenting good speedup and scalability.

**Acknowledgements** The authors would like to thank the financial support of the Petroleum National Agency (ANP, Brazil) and the Center for Parallel Computations (NACAD) and the Laboratory of Computational Methods in Engineering (LAMCE) at the Federal University of Rio de Janeiro. The authors are very grateful for the computational resources provided by Silicon Graphics Inc. (SGI/Brazil).

## References

- Alexandrou AN, McGilvray TM, Burgos G (2001) Steady Herschel–Bulkley fluid flow in three-dimensional expansions. *J Non-Newtonian Fluid Mech* 100:77–96
- Ansys Inc. Theory Reference (2003) URL: <http://www.ansys.com>
- Barragy E, Carey GF (1988) A parallel element-by-element solution scheme. *Int J Numer Meth Eng* 26(11):2367–2382
- Battaglia F, Tavener SJ, Kulkarni AK, Merkle CL (1997) Bifurcation of low Reynolds number flows in symmetric channels. *AIAA J* 35:99

5. Beverly CR, Tanner RI (1992) Numerical analysis of three-dimensional Bingham plastic flow. *J Non-Newtonian Fluid Mech* 42:85–115
6. Bose A, Carey GF (1999) Least-squares  $p - r$  finite element methods for incompressible non-Newtonian flows. *Comput Meth Appl Mech Eng* 180:431–458
7. Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equation. *Comput Meth Appl Mech Eng* 32:199–259
8. Carey GF, Barragy É, Mclay R, Sharma M (1988) Element-by-element vector and parallel computations. *Commun Appl Numer Meth* 4(3):299–307
9. Catabriga L, Coutinho ALGA (2002) Implicit SUPG solution of Euler equations using edge-based data structures. *Comput Meth Appl Mech Eng* 32:3477–3490
10. Coutinho ALGA, Alves JLD, Ebecken NFF (1991) A study of implementation schemes for vectorized sparse EBE matrix-vector multiplication. *Adv Eng Softw Workstations* 13(3):130–134
11. Coutinho ALGA, Martins MAD, Alves JLD, Landau L, Moraes A (2001) Edge-based finite element techniques for non-linear solid mechanics problems. *Int J Numer Meth Eng* 50(9):2053–2068
12. Crochet MJ, Davies AR, Walters K (1984) Numerical simulation of non-Newtonian flow. Elsevier, Amsterdam
13. Dembo RS, Eisenstat SC, Steihaug T (1982) Inexact Newton methods. *SIAM J Numer Anal* 19:400–408
14. Dongarra J, Foster I, Fox G, Gropp W, Kennedy K, Torczon L, White A (2002) In: Sourcebook of parallel computing. Morgan Kaufmann, San Francisco
15. Drikakis D (1997) Study of bifurcation flow phenomena in incompressible in sudden expansions. *Phys Fluids* 9:76–87
16. Eisenstat SC, Walker HF (1996) Choosing the forcing terms in inexact Newton method. *SIAM J Sci Comput* 17(1):16–32
17. Elias RN, Coutinho ALGA, Martins MAD (2005) Inexact Newton-type methods for the solution of steady incompressible viscoplastic flows with the SUPG/PSPG finite element formulation. *Comput Meth Appl Mech Eng* (to appear)
18. Fearn RM, Mullin T, Cliffe KA (1990) Nonlinear flow phenomena in a symmetric sudden expansion. *J Fluid Mech* 211:595–608
19. Gartling DK (1992) Finite element methods for non-Newtonian flows. Report SAND92–0886, CFD Department Sandia National Laboratories, Albuquerque
20. Gustafson JL, Montry GR, Benner RE (1988) Development of parallel methods for a 1024-processor hypercube. *SIAM J Sci Stat Comp* 9(4):609–638
21. Hawa T, Rusak Z (2000) Viscous flow in a slightly asymmetric channel with a sudden expansion. *Phys Fluids* 12:2257
22. Hayes LJ, Devloo PRB (1986) A vectorized version of a sparse matrix vector multiply. *Int J Numer Meth Eng* 23(3):1043–1056
23. Hughes TJR, Levit I, Winget J (1983) An element-by-element solution algorithm for problems of structural and solid mechanics. *Comput Meth Appl Mech Eng* 36(2):241–254
24. Hughes TJR, Ferencz RM, Hallquist JO (1987) Large-scale vectorized implicit calculations in solid mechanics on a CRAY X-MP/48 utilizing EBE preconditioned conjugate gradients. *Comput Meth Appl Mech Eng* 61(2):215–248
25. Hughes TJR, Franca LP, Balestra M (1986) A new finite element formulation for computational fluid dynamics: v. circumventing the Babuška–Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Comput Meth Appl Mech Eng* 59:85–99
26. Hughes TJR (1987) The finite element method: linear static and dynamic finite element analysis. Prentice-Hall, Englewood Cliffs
27. Hughes TJR, Winget J, Levit I, Tezduyar TE (1983) New alternating direction procedures in finite element analysis based upon EBE approximate factorizations. In: Atluri SN, Perrone N (eds) Computer methods for nonlinear solids and structural mechanics, vol 54. ASME, New York, pp 75–109
28. Karypis G, Kumar V (1998) Metis 4.0: unstructured graph partitioning and sparse matrix ordering system. Technical report. Department of Computer Science, University of Minnesota, Minneapolis, ; <http://www.users.cs.umn.edu/~karypis/metis>.
29. Kelley CT (1995) Iterative methods for linear and nonlinear equations. In: Frontiers in applied mathematics. SIAM, Philadelphia
30. Knoll DA, Keyes DE (2004) Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *J Comput Phys* 193:357–397
31. Lohner R (1994) Edges, stars, superedges and chains. *Comput Meth Appl Mech Eng* 111(3–4):255–263
32. Luo H, Baum JD, Löhner R (1994) Edge-based finite element scheme for the Euler equations. *AIAA J* 32(6):1183–1190
33. Lyra PRM, Manzari MT, Morgan K, Hassan O, Peraire J (1995) Side-based unstructured grid algorithms for compressible viscous flow computations. *Int J Eng Anal Des* 2:88–102
34. Löhner R, Galle M (2002) Minimization of indirect addressing for edge-based field solvers. *Commun Numer Meth Eng* 18:335–343
35. Manica R, De Bortoli AL (2004) Simulation of sudden expansion flows for power-law fluids. *J Non-Newtonian Fluid Mech* 121:35–40
36. Coutinho ALGA, Martins MAD, Sydenstricker R, Elias RN (2005) Performance comparison of data reordering algorithms for sparse matrix-vector multiplication in edge-based unstructured grid computations. *Int J Numer Meth Eng* (in press)
37. Meuric OFJ, Wakeman RJ, Chiu TW, Fisher KA (1998) Numerical flow simulation of viscoplastic fluids in annuli. *Can J Chem Eng* 76:27–40
38. Mitsoulis E, Zisis Th (2001) Flow of Bingham plastics in a lid-driven square cavity. *J Non-Newtonian Fluid Mech* 101:173–180
39. Neofytou P, Drikakis D (2003) Non-Newtonian flow instability in a channel with a sudden expansion. *J Non-Newtonian Fluid Mech* 111:127–150
40. Owens RG, Phillips TN (2002) Computational rheology. Imperial College Press
41. Papanastasiou TC (1987) Flow of materials with yield. *J Rheol* 31:385–404
42. Peraire J, Morgan K (1992) A 3D finite element multigrid solver for the Euler equations. *AIAA Paper* 92:0449
43. Peraire J, Peiro J, Morgan K (1993) Multigrid solution of the 3d-compressible Euler equations on unstructured grids. *Int J Numer Meth Eng* 36(6):1029–1044
44. Perić D, Slijepčević S (2001) Computational modelling of viscoplastic fluids based on a stabilised finite element method. *Eng Comput* 18:577–591
45. Ribeiro FLB, Galeão AC, Landau L (2001) Edge-based finite element method for shallow-water equations. *Int J Numer Meth Fluids* 36:659–685
46. Ribeiro FLB, Coutinho ALGA (2005) Comparison between element, edge and compressed storage schemes for iterative solutions in finite element analyses. *Int J Numer Meth Eng* 63(4):569–588
47. Saad Y (1996) Iterative methods for sparse linear systems. PWS Publishing Company, Boston
48. Shadid JN, Tuminaro RS, Walker HF (1997) An inexact Newton method for fully-coupled solution of the Navier–Stokes equations with heat and mass transport. *J Comput Phys* 137:155–185
49. Shapira M, Degani D, Weihs D (1990) Stability and existence of multiple solutions for viscous flow in suddenly enlarged channels. *Comput Fluids* 18:239
50. Slijepčević S, Perić D (2004) Some aspects of computational modeling of non-Newtonian fluids based on a stabilised finite element method. European congress on computational methods in applied sciences and engineering — ECCOMAS Finland, 24–28 July 2004
51. Soto O, Löhner R, Cebal JR, Camelli F (2004) A stabilized edge-based implicit incompressible flow formulation. *Comput Meth Appl Mech Eng* 193:2139–2154
52. Tezduyar TE, Aliabadi S, Behr M, Johnson A, Kalro V, Litke M (1996) Flow simulation and high performance computing. *Comput Mech* 18:397–412
53. Tezduyar TE, Liou J (1989) Grouped element-by-element iteration schemes for incompressible flow computations. *Comput Phys Commun* 53:441–453
54. Tezduyar TE, Behr M, Aliabadi S, Mittal S, Ray SE (1992) A new mixed preconditioning method for finite element computations. *Comput Meth Appl Mech Eng* 99:27–42

55. Tezduyar TE, Mittal S, Ray SE, Shih R (1992) Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Comput Meth Appl Mech Eng* 95:221–242
56. Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. *Adv Appl Mech* 28:1–44
57. van Gijzen MB (1995) Large scale finite element computations with GMRES-like methods on a CRAY Y-MP. *Appl Numer Math* 19:51–62
58. Vola D, Boscardin L, Latché JC (2003) Laminar unsteady flows of Bingham fluids: a numerical strategy and some benchmark results. *J Comput Phys* 187:441–456
59. Winget J, Hughes TJR (1985) Solution algorithms for nonlinear transient heat-conduction analysis employing element-by-element iterative strategies. *Comput Meth Appl Mech Eng* 52(1–3):711–815