Korea-Australia Rheology Journal
Vol. 21, No. 1, March 2009 pp. 1-12

Computation of dilute polymer solution flows using BCF-RBFN based method and domain decomposition technique

Canh-Dung Tran[1,*], David G. Phillips[1] and Thanh Tran-Cong[2]
[1]*CSIRO, Geelong, VIC 3216, Australia,*
[2]*Faculty of Engineering and Surveying, University of Southern Queensland, Toowoomba, QLD 4350, Australia*
(Received March 12, 2008; final revision received July 31, 2008)

Abstract

This paper reports the suitability of a domain decomposition technique for the hybrid simulation of dilute polymer solution flows using Eulerian Brownian dynamics and Radial Basis Function Networks (RBFN) based methods. The Brownian Configuration Fields (BCF) and RBFN method incorporates the features of the BCF scheme (which render both closed form constitutive equations and a particle tracking process unnecessary) and a mesh-less method (which eliminates element-based discretisation of domains). However, when dealing with large scale problems, there appear several difficulties: the high computational time associated with the Stochastic Simulation Technique (SST), and the ill-condition of the system matrix associated with the RBFN. One way to overcome these disadvantages is to use parallel domain decomposition (DD) techniques. This approach makes the BCF-RBFN method more suitable for large scale problems.

Keywords : Brownian dynamics, RBFN, Brownian configuration fields, dilute polymer solution, meshless method, domain decomposition, message passing interface (MPI)

1. Introduction

In this work, the RBFN-BCF method (Tran-Canh and Tran-Cong, 2003; 2004) is considered in conjunction with domain decomposition techniques for approximation of functions and solution of PDEs. The domain under consideration is partitioned into a number of subdomains and the task on each subdomain is carried out on a distributed memory computer with a parallelization procedure. Furthermore, since the tasks of the stochastic and deterministic processes, such as solving the Stochastic Differential Equations (SDEs) for dumbbells and the PDEs governing the flow, can be carried out separately in subdomains, the parallel DD techniques can be applied effectively in both macroscopic and microscopic components. In general either only one subdomain or a group of sub-domains can be handled by a separate processor; the former case is considered here. Hence, the parallel domain decomposition method increases the throughput, and, at the same time, removes the problem of ill-conditioning of the system matrix associated with the RBFN-based method. For the purpose of function approximation, the domain decomposition is relatively simple in the sense that the data on the interfaces between the sub-domains are known. It is more complex for solving PDEs in applying the DD method. Since the interfaces between the subregions are considered as part of the boundary of the sub-domains, the associated unknowns need to be determined as part of the solution of the problem.

The domain decomposition techniques can be classified into two main categories, namely the subregioning and the substructuring DD methods. The former includes iterative non-overlapping schemes (Marini and Quarteroni, 1988; Funaro *et al.*, 1988; Yang, 1996), and the iterative overlapping schemes using the additive or multiplicative Schwarz techniques (Zhou *et al.*, 2003). While the subregioning methods focus on the way to estimate the boundary conditions on the interfaces that ensures the continuity and the smoothness across the interfaces, the substructuring DD method is based on explicit computation and factorization of a sequence of Schur complement matrices. These techniques circumvent the ill-posed problem resulting from using the RBFs as a global interpolant, reduce the memory requirement and achieve a high accuracy of the solution. In the present work, the iterative non-overlapping domain decomposition method is employed in conjunction with the hybrid RBFN-BCF simulation. The paper is organized as follows: In section 2, the RBFN-BCF method for dilute polymer solution is reviewed briefly. Section 3 is to present the domain decomposition for the RBFN-based method in which the RBFN interpolation is reviewed and the iterative non-overlapping DD for solving PDEs is described. The parallelization of the BCF-RBFN method is reported in section 4. Numerical examples are then dis-

*Corresponding author: canh-dung.tran@csiro.au

cussed in section 5 in which the steady state 10:1 planar contraction flow is simulated, followed by a brief conclusion in section 6.

2. Review of RBFN-BCF method for dilute polymer solutions

A dilute polymer solution is considered as an incompressible suspension of non-interacting dumbbells in a Newtonian solvent. Dumbbells are characterized by connector vectors **Q**'s. These connector vectors, which constitute Brownian Configuration Fields (Hulsen *et al.*, 1997), satisfy the dimensionless form of the SDE as follows

$$d\mathbf{Q} = \left[-De(\mathbf{u}\cdot\nabla\mathbf{Q}+\kappa\cdot\mathbf{Q})-\frac{1}{2}\mathbf{F}(\mathbf{Q})\right]dt+d\mathbf{W}(t), \quad (1)$$

where **Q**, **u**, t are the dimensionless forms of connector vectors, velocity fields and time, scaled by $(k_BT/H)^{1/2}$, V, λ_H, respectively; $De=\lambda_H/\lambda_{flow}$ is the Deborah number; $\lambda_H=\zeta/4H$ and $\lambda_{flow}=a/V$ are characteristic relaxation times of the fluid and flow, respectively; ζ is the friction coefficient between the dumbbell and the solvent; H is the spring constant; a, V are characteristic length and velocity; k_B is the Boltzmann constant; T is the absolute temperature; κ is the velocity gradient; **W**(t) is a Wiener process with mean $\langle W_i(t)\rangle = 0$ and covariance $\langle W_i(t)W_j(t')\rangle = \delta_{ij}\min(t,t')$ and accounts for the random displacement of the beads due to thermal motion, and **F** is the dimensionless spring force which is given by, for the FENE and FENE-P models, respectively.

$$\mathbf{F}_{FENE} = \frac{\mathbf{Q}}{1-\mathbf{Q}^2/b}, \quad \mathbf{F}_{FENE-P} = \frac{\mathbf{Q}}{1-\langle\mathbf{Q}^2\rangle/b}, \quad (2)$$

where $b=HQ_o^2/k_BT$ is the square of the dimensionless maximum extensibility. In this work, based on the Brownian configuration fields, the polymer stress tensor is determined by the Kramer formula as

$$\tau^p = -\langle\mathbf{Q}\cdot\mathbf{F}(\mathbf{Q})\rangle+\mathbf{I}, \quad (3)$$

where τ_p is the dimensionless form of the polymer stress tensor scaled by n_dk_BT; n_d is the density of dumbbells; **I** is the unity matrix.

The velocity field **u** in Eq. (1) of steady, isothermal, incompressible flows is governed by the system of momentum and mass conservation equations as follows

$$\nabla\cdot\mathbf{u}=0 \quad (4a)$$

$$-\nabla p+\nabla\cdot\tau=\rho(\mathbf{u}\cdot\nabla)\mathbf{u} \quad (4b)$$

where τ is the extra stress; ρ is the fluid density.

For a polymer solution, the extra stress τ can be decomposed as

$$\tau = 2\eta_N\mathbf{D}+\tau^p, \quad (4c)$$

where $2\eta_N\mathbf{D}$ is the Newtonian solvent contribution; η_N is the solvent viscosity; $\mathbf{D}=\frac{1}{2}(\nabla u+(\nabla u)^T)$ is the rate of strain tensor; τ^p is the polymer-contributed stress and p is the pressure arisen from the incompressibility constraint. From Eq. (4c) , Eq. (4b) can be rewritten as follows

$$2\eta_N\nabla\cdot\mathbf{D}-\rho(\mathbf{u}\cdot\nabla)\mathbf{u}-\nabla p=-\nabla\cdot\tau^p \quad (4d)$$

Instead of using the Eqs. (4a) and (4d), the incompressibility condition is enforced via the penalty function method as follows $p=-p_e(\nabla\cdot\mathbf{u})$ where p_e is a sufficiently large parameter. Thus the momentum equation (4d) is rewritten as

$$2\alpha\nabla\cdot\mathbf{D}-Re(\mathbf{u}\cdot\nabla)\mathbf{u}+\beta_1\nabla(\nabla\cdot\mathbf{u})=-\beta_2\nabla\cdot\tau^p, \quad (4)$$

where $\alpha=\eta_N/\eta_o$; $Re=\rho Va/\eta_o$; $\eta_o=\eta_N+\eta_P$; η_N, η_P are the solvent and polymer viscosities; ρ is the fluid density; $\beta_1=p_e/\eta_o$; $\beta_2=n_dk_BT/(\eta_oV/a)$ is a dimensionless parameter. In Equation (4), **D** and τ_p denote the dimensionless form of the rate of strain tensor and polymer stress, respectively. While τ_p is determined by the Brownian configuration field method represented in section 2.1, the solutions (**u**) of the governing PDEs (Eq. (4)) are obtained by using the RBFN-based mesh-free method described in the sections 2.2 and 3.

2.1. Numerical simulation and variance reduction of the configuration fields

The second-order semi-implicit predictor-corrector scheme is applied in solving the governing Eulerian SDE of Brownian configuration fields (1). The time discretisation of the elastic dumbbell configuration fields consists of the predictor and corrector steps, and the polymer stress tensor is then determined by the average of the configuration fields evaluated at each collocation point.

The implementation of the control variate techniques is achieved easily based on the Eulerian nature of the BCF scheme. For illustration, the implementation of the control variate method in conjunction with BCFs is presented for the FENE dumbbell model as follows: at each collocation point, n_d dumbbells are assigned and numbered from i=1 .. n_d where dumbbells having the same index in the whole analysis domain have the same random number (Tran-Canh and Tran-Cong, 2004).

This technique is employed for the numerical calculation of the polymer contribution to stress (3) where the expectations of random variables $\frac{\mathbf{QQ}}{1-\mathbf{Q}^2/b}$ are required. At each time t and position **x**, let $\overline{\mathbf{Q}}(\mathbf{x},t)$ be the control variate corresponding to the configuration field **Q**(**x**,t). The variance reduction method is carried out by splitting the expectation above as follows

$$\left\langle\frac{\mathbf{QQ}}{1-\frac{\mathbf{Q}^2}{b}}\right\rangle = \left\langle\frac{\overline{\mathbf{Q}}\overline{\mathbf{Q}}}{1-\frac{\overline{\mathbf{Q}}^2}{b}}\right\rangle - \left\langle\frac{\mathbf{QQ}}{1-\frac{\mathbf{Q}^2}{b}}-\frac{\overline{\mathbf{Q}}\overline{\mathbf{Q}}}{1-\frac{\overline{\mathbf{Q}}^2}{b}}\right\rangle. \quad (5)$$

When $\bar{\mathbf{Q}}=0$ there is no variance reduction and $\langle(.)\rangle$ means the expectation of (.). From (5), the polymer stress tensor (3) is rewritten as follows

$$\tau^p=-\left\langle\frac{\mathbf{QQ}}{1-\frac{\mathbf{Q}^2}{b}}-\frac{\bar{\mathbf{Q}}\bar{\mathbf{Q}}}{1-\frac{\bar{\mathbf{Q}}^2}{b}}\right\rangle+\bar{\tau}^p, \tag{6}$$

where $\bar{\tau}^p$ is given by

$$\bar{\tau}^p=-\left\langle\frac{\bar{\mathbf{Q}}\bar{\mathbf{Q}}}{1-\frac{\bar{\mathbf{Q}}^2}{b}}\right\rangle+\mathbf{I}. \tag{7}$$

The first term of the RHS of Eq. (6) is calculated by using Brownian dynamic simulation and the second term is determined in a deterministic way. In the present work, since $\bar{\mathbf{Q}}$'s are estimated at equilibrium configuration, $\bar{\tau}^p$ is zero and the configuration vectors $\mathbf{Q}$'s satisfy the following SDE

$$d\bar{\mathbf{Q}}=-\frac{\bar{\mathbf{F}}}{2}dt+d\mathbf{W}(t) \tag{8}$$

where $\bar{\mathbf{F}}$ is given by

$$\bar{\mathbf{F}}=\frac{\bar{\mathbf{Q}}}{1-\frac{\bar{\mathbf{Q}}^2}{b}}. \tag{9}$$

The polymer stress tensor (6) is rewritten as follows

$$\tau^p=-\left\langle\frac{\mathbf{QQ}}{1-\frac{\mathbf{Q}^2}{b}}-\frac{\bar{\mathbf{Q}}\bar{\mathbf{Q}}}{1-\frac{\bar{\mathbf{Q}}^2}{b}}\right\rangle. \tag{10}$$

2.2. RBFN-based element-free method

2.2.1. Review of RBFN interpolation

In the present work, a combination of RBF and Polynomial Basis Function (PBF) is employed in a RBFN as follows (Fig. 1)

$$f(\mathbf{x})=\sum_{j=1}^{m}w^j h^j(\mathbf{x})+\sum_{k=1}^{\bar{m}}\iota^k p^k(\mathbf{x})=\mathbf{H}^T(\mathbf{x})\mathbf{w}+\mathbf{P}^T(\mathbf{x})\iota, \tag{11}$$

where $w^j\in\mathbf{w}(\mathbf{w}^T=[w^1\ w^2\ \dots\ w^m])$ and $\iota^k\in\iota(\iota^T=[\iota^1\ \iota^2\ \dots\ \iota^{\bar{m}}])$ are the synaptic weights; h^j is the chosen radial basis function corresponding to the j^{th} RBF-neuron; p^k is the polynomial basis function corresponding to the k^{th} PBF-neuron; $m+\bar{m}$ is the total number of neurons. **H** and **P** are defined as follows

$$\mathbf{H}^T(\mathbf{x})=[h^1(\mathbf{x})\ h^2(\mathbf{x})\ \dots\ h^m(\mathbf{x})], \tag{12}$$

$$\mathbf{P}^T(\mathbf{x})=[p^1(\mathbf{x})\ p^2(\mathbf{x})\ \dots\ p^{\bar{m}}(\mathbf{x})]. \tag{13}$$

Let n be the number of training points $(\mathbf{x}_i,\hat{y}_i)$; $\mathbf{x}_i$ is the coordinate of the i^{th} collocation point and $\hat{y}_i$ is the desired value of function f at the collocation point $\mathbf{x}_i$. The RBFs h^j employed here is Thin Plate Splines Radial Basis Function (TPS-RBF) which is given by

$$h^j(r)=h^j(\|x-c^j\|)=r^{2s}\log(r),\quad s=1,2,\dots\quad j=1..m \tag{14}$$

and its first and second order derivatives are written as, respectively, follows

$$\frac{\partial h^j}{\partial x_i}=r^{2(s-1)}(x_i-c_i^j)(2s\log(r)+1),\quad j=1..m,\quad i=1,2,3 \tag{15}$$

$$\frac{\partial^2 h^j}{\partial x_i\partial x_l}=2r^{2(s-2)}(x_i-c_i^j)(x_l-c_l^j)[2s(s-1)\log(r)+(2s-1)]+\chi(r),$$

$$l=1,2,3$$

$$\chi(r)=r^{2(s-1)}(2s\log(r)+1)\quad \forall i=l,$$
$$\chi(r)=0\quad \forall i\neq l, \tag{16}$$

where $\mathbf{r}=(\mathbf{x}-\mathbf{c}^j)$ and $r=\|(\mathbf{x}-\mathbf{c}^j)\|$ is the Euclidean norm of $\mathbf{r}$; $\{\mathbf{c}^j\}$ is a set of centers that can be chosen from among the data points. Since TPS-RBF is C^{2s-1}-continuous, the power index s must be appropriately chosen for a given partial differential operator. In the present work, the existence of the second order derivatives of unknowns, which need to be approximated in the governing PDE (4), requires $m\geq 2$ to satisfy the continuity condition. An extra requirement for the uniqueness of the approximation (11) is imposed as follows (Zerroukat *et al.*, 1998)

$$\sum_{i=1}^{n}p^k(\mathbf{x}_i)w_i=0,\quad k=1,\cdots,\bar{m} \tag{17}$$

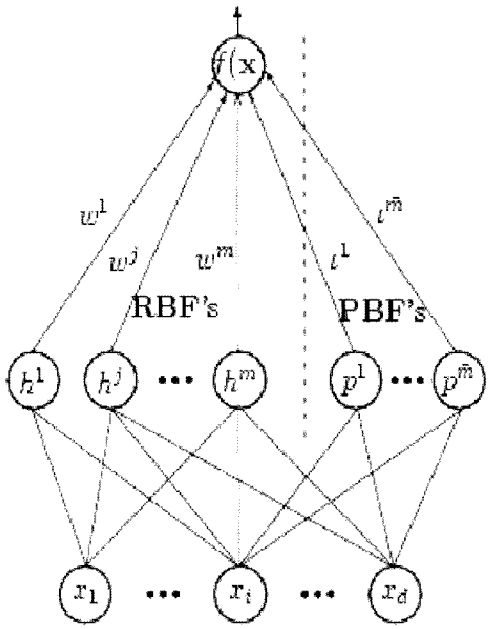


Fig. 1. Schematic of the RBF-PBF neural networks. Each of d (d =2, in the present work) components of the input vector **x** feeds forward to m RBFs and $\bar{m}$ PBFs whose outputs are linearly combined with weights $\{w^j\}_{j=1}^{m}$ and $\{\iota^k\}_{k=1}^{\bar{m}}$, respectively, into the network output f(**x**).

where the PBF of order p for a 2D problem is given by

$$\mathbf{P}^T(\mathbf{x}) = [1\ x_1\ x_2\ x_1^2\ x_1x_1\ x_2^2 \cdots x_2^p]. \quad (18)$$

The choice of the quantity and location of collocation points (x_i, i=1,...,n) depends on the problem geometry and desired solution accuracy. In general, both collocation points and RBF centres can be randomly and separately distributed in the analysis domain. However, in the present work, collocation points are chosen to be the same as RBF centres, *i.e.* m=n. The training of the model Eq. (11), given a training set of n collocation points $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, can be achieved via the minimisation of a cost function based on the sum of squared errors that was represented in Tran-Canh and Tran-Cong (2002). The partial derivatives of f(**x**) then can be calculated analytically as follows

$$lf(\mathbf{x}) = \sum_{j=1}^{n} w^j lh^j(\mathbf{x}) + \sum_{k=1}^{3} \iota^k lp^k(\mathbf{x}), \quad (19)$$

where l is a derivative operator.

2.2.2. RBFN-based method for solving the deterministic PDEs

Considering the following boundary value problem (BVP) in general

$$lu = y \quad \forall \mathbf{x} \in \Omega, \quad (20)$$
$$\zeta u = b \quad \forall \mathbf{x} \in \partial\Omega, \quad (21)$$

where Ω is the volume under consideration; $\partial\Omega$ is the boundary of Ω; ζ is an arbitrary differential operator; b is an operator imposed as boundary conditions; u is an unknown function; y and b are given functions. The problem may be multidimensional; however, without loss of generality, the 1-D problem is presented here for simplicity. Since RBFN is a good universal approximator as mentioned above, **u** is represented by an approximant f defined by Eq. (11) and then lf and ζf can be described by Eq. (19). The numerical problem is therefore reduced to an unconstrained optimization problem of the objective function as follows

$$\Phi = \int_{\Omega} \|lf - y\|^2 d\Omega + \int_{\Gamma_t} \|\zeta f - b\|^2 d\Gamma. \quad (22)$$

For a more detailed mathematical derivation leading to Eq. (22), see Franke (1982); Hornik *et al.* (1989) and Dissanayake and Phan-Thien (1994). Eq. (22) can now be applied particularly to solve macroscopic governing equations.

3. Domain decomposition technique for RBFN-based method

Although the parallel domain decomposition schemes were introduced a long time ago with the advent of powerful supercomputers, they have been well developed only for the element type methods (Quarteroni and Valli, 1999; Smith *et al.*, 1996), and there are only a few attempts to couple RBFN based meshless methods and DD techniques (Dubal, 1994; Beatson *et al.*, 2000 and Zhou *et al.*, 2003).

The non-overlapping domain decompositions are easy for parallel implementation and they also have some advantages over the overlapping method, for example they are efficient for handling elliptic problems with large jumps in coefficients (Xu and Zou, 1998). For further references, these techniques can be found in, for example, Smith *et al.* (1996). A very good survey on the non-overlapping DD method was given by Xu and Zou (1998). In this work, the iterative non-overlapping domain decomposition is employed to couple with RBFN-based meshfree method for the function approximation and numerical solution of PDEs. The method is obviously well suited for parallel computing architectures. Here the implementation is for both macroscopic and microscopic parts of the method. For the purpose of function approximation, since the data are known on the interfaces, the DD technique is very simple and not presented here. For solving PDEs, the interfaces are considered as part of the boundary of subdomains and the associated boundary conditions need to be determined.

Consider a steady-state problem governed by Eq. (20). The boundary condition (21) could be Dirichlet, Newmann or a mixture of both. The solution u can be expressed as follows

$$u \approx \sum_{j=1}^{m} w^j h^j(\mathbf{x}) + \sum_{k=1}^{\bar{m}} \iota^k p^k(\mathbf{x}), \quad (23)$$

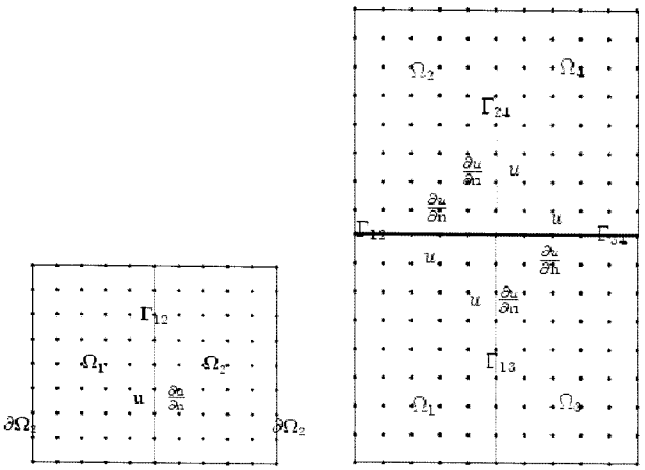
where w^j and ι^k are the weights; h^j is the chosen radial basis function corresponding to the j^{th} RBF-neuron; p^j is the polynomial basis function of the k^{th} PBF. The partial derivatives of f(**x**) can be calculated analytically as follows

$$lu(x) \approx \sum_{j=1}^{m} w^j lh^j(x) + \sum_{k=1}^{\bar{m}} \iota^k lp^k(x), \quad (24)$$

where l is a derivative operator. The chosen PBF is 2-D linear; lh^j for TPS-RBFN is given in Eqs. (15)-(16). The substitution (23) and (24) into (20) and (21) followed by the application of the general linear least squares principle lead to a system of equations in the unknowns w^j and ι^k, which was described in Tran-Canh and Tran-Cong (2004).

3.1. The iterative non-overlapping DD scheme for RBFN

For illustration, the steady state problem governed by Eqs. (20) and (21) using DD technique with two subdomains (see Fig. 2(a)). The domain under consideration is divided into 2 non-overlapping sub-domains Ω_1, Ω_2. Let $\partial\Omega_1$ and $\partial\Omega_2$ be the boundaries of Ω_1 and Ω_2, respectively; Γ_{12} is the artificial boundary (interface) between Ω_1 and Ω_2. The boundary condition imposed on this interface



(a) (b)

Fig. 2. The schematic overlapping domain decomposition techniques with (a) two and (b) four sub-domains.

can be Dirichlet-Neumann, Neumann-Neumann or otherwise. In the present work, the boundary condition imposed on the interface Γ_{12} is of Dirichlet-Neumann type. The algorithm of the iterative non-overlapping DD consists of two main tasks: (i) determining the unknowns on interfaces based on the compatibility conditions on the interfaces, and (ii) solving the smaller boundary value problems on the sub-domains. This algorithm is written for the subdomains Ω_1 and Ω_2, at the step i+1, respectively, as follows

$$
\begin{aligned}
&l\mathrm{u}_1^{i+1}(\mathbf{x}) = \mathrm{y} \quad \mathbf{x} \in \Omega_1 \\
&l\mathrm{u}_1^{i+1}(\mathbf{x}) = \mathrm{b} \quad \mathbf{x} \in \partial\Omega_1 \backslash \Gamma_{12} \\
&\mathrm{u}_1^{i+1}(\mathbf{x}) = \mathrm{c}^{\mathrm{i}} \quad \mathbf{x} \in \Gamma_{12}
\end{aligned} \tag{25}
$$

and

$$
\begin{aligned}
&l\mathrm{u}_2^{i+1}(\mathbf{x}) = \mathrm{y} \quad \mathbf{x} \in \Omega_2 \\
&\zeta\mathrm{u}_2^{i+1}(\mathbf{x}) = \mathrm{b} \quad \mathbf{x} \in \partial\Omega_2 \backslash \Gamma_{12} \\
&\frac{\partial \mathrm{u}_2^{i+1}}{\partial \mathbf{n}}(\mathbf{x}) = \frac{\partial u_1^{i}}{\partial \mathbf{n}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma_{12}
\end{aligned} \tag{26}
$$

where c^{i} is extracted from Ω_2 (the neighbouring subdomain sharing the interface with Ω_1) at the step i and given by

$$\mathrm{c}^{\mathrm{i}} = \theta \mathrm{c}^{\mathrm{i}-1} + (1-\theta)\mathrm{u}_2^{\mathrm{i}}; \quad 0 \le \theta \le 1 .$$

$\mathrm{u}_k^{\mathrm{i}}(\mathrm{k}=1,2)$ is the numerical solutions obtained from the subdomain Ω_k at the collocation points located on the interface at a step i and $\frac{\partial \mathrm{u}}{\partial \mathbf{n}} = \frac{\partial \mathrm{u}}{\partial \mathrm{n}_1}\mathrm{n}_1 + \frac{\partial \mathrm{u}}{\partial \mathrm{n}_2}\mathrm{n}_2$.

This algorithm can be extended for N>2 subdomains by using the black and white coloring method (Quarteroni and Valli, 1999). In order to facilitate the parallel implementation, the block-parallel coloring scheme is used in this work. Let Ω_k be a subdomain ($1 \le k \le N$), $\Gamma_{kl} = \Omega_k \cap \Omega_l$ the common interface between two adjoining subdomain Ω_k and Ω_l, and G_b and G_w be the two groups of sub-domains which are defined by (Fig. 2(b))

$$
\begin{aligned}
&G_b = \{\Omega_k \text{ is black}, 1 \le k \le N\}, \\
&G_w = \{\Omega_k \text{ is white}, 1 \le k \le N\}.
\end{aligned}
$$

The algorithm of this method is given by

$$
\begin{cases}
l\mathrm{u}_k^{i+1}(\mathbf{x}) = \mathrm{y}, \ \mathbf{x} \in \Omega_k, \ \Omega_k \in G_b \\
\mathrm{u}_k^{i+1}(\mathbf{x}) = \theta \mathrm{u}_l^{i} + (1-\theta)\mathrm{u}_k^{i}, \ \mathbf{x} \in \Gamma_{kl}, \ \Omega_l \in G_w
\end{cases} \tag{27}
$$

and

$$
\begin{cases}
l\mathrm{u}_k^{i+1}(\mathbf{x}) = \mathrm{y}, \ \mathbf{x} \in \Omega_l, \ \Omega_l \in G_w \\
\frac{\partial \mathrm{u}_l^{i+1}}{\partial \mathbf{n}}(\mathbf{x}) = \frac{\partial \mathrm{u}_l^{i}}{\partial \mathbf{n}}, \ \mathbf{x} \in \Gamma_{kl}, \ \Omega_l \in G_w, \ \Omega_k \in G_b
\end{cases} . \tag{28}
$$

Hence, the sub-programs corresponding to subdomains Ω_k are independent of one another and they are solved simultaneously with separate CPUs. If any part of $\partial\Omega_k$ coincides with the original boundary $\partial\Omega$, the original boundary conditions would apply there as appropriate.

The overall procedure can now be described as follows

1. Divide the analysis domain into a number of subdomains and determine G_b and G_w and the interfaces between subdomains.
2. Guess initial boundary conditions at the interfaces for subdomains as appropriate (Dirichlet condition for $\Omega_k \in G_b$ and Neumann condition for $\Omega_l \in G_w$).
3. Solve the boundary value problems described by Eqs. (25)-(26) corresponding to Ω_k, ($1 \le k \le N$) using TPS-RBFN-based element-free method on separate CPUs
4. Check for the compatibility on the whole artificial interfaces.
5. If the procedure is not yet converged, return to step 3, when the boundary conditions on all interfaces are updated as described in the second equation of (27) for $\Omega_k \in G_b$ and (28) for $\Omega_l \in G_w$.
6. Stop the procedure.

The singular value decomposition scheme is employed to get the reciprocal matrices associated with each subregion.

4. Parallelization and DD scheme for RBFN-BCF method

The parallelization performed in the present work is focussed on both micro and macro parts as mentioned in the previous section. The parallelization is carried out in regard to collocation points and based on the domain decomposition technique. In this work the non-overlapping scheme is employed. The basic parallelization scheme is described in Figs. 3 and 4. Parallel implementation of the algorithm is based on the single program, multiple data (SPMD) paradigm with message passing interface (MPI) for parallel communication. Although a CPU could accommodate more than one sub-domain, in the present work,

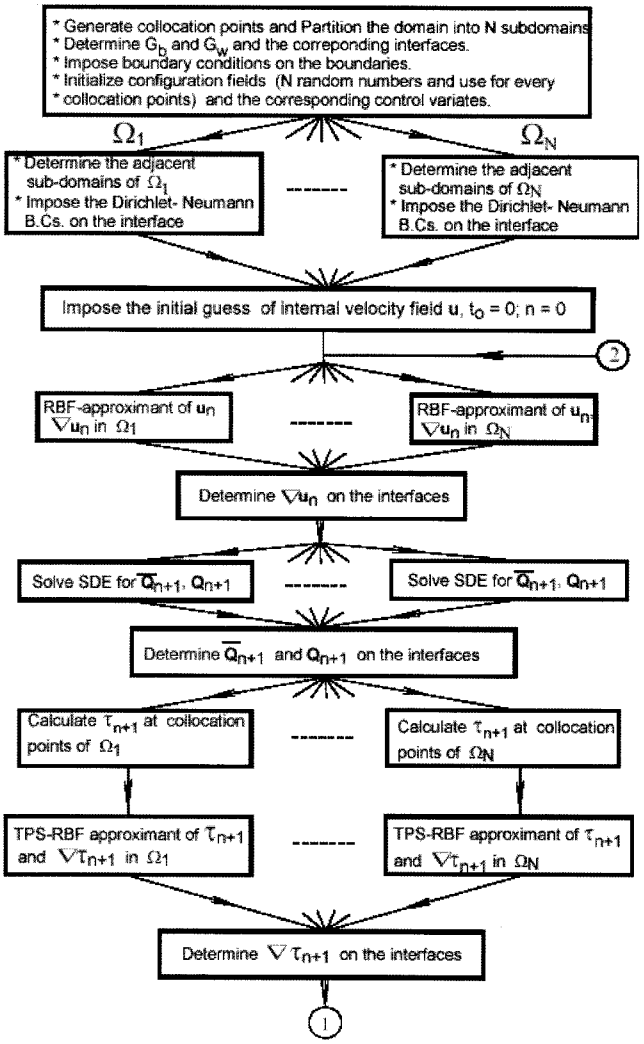


Fig. 3. General flow-chart of the present RBFN-BCF-based element-free method in conjunction with the iterative non-overlapping domain decomposition technique. See Fig. 4.

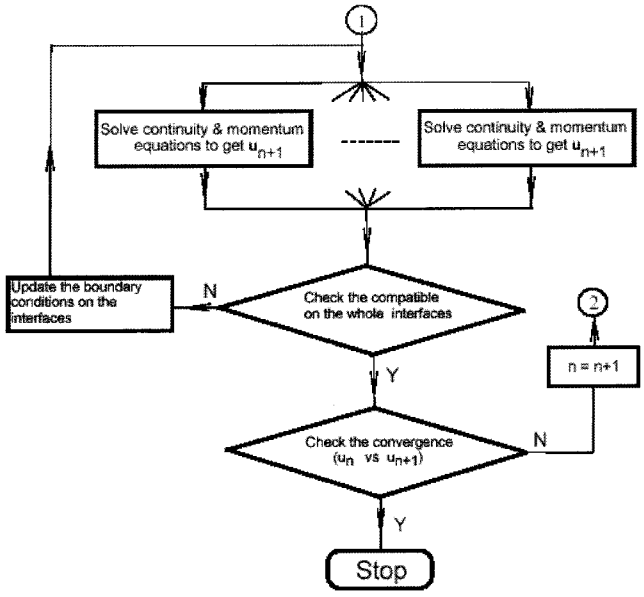


Fig. 4. General flow-chart of the present RBFN-BCF-based element-free method in conjunction with the iterative non-overlapping domain decomposition technique. See Fig. 3.

only one sub-domain is allowed per processor. The parallel implementation is carried out on the AlphaServer ES45 installed at the Australian Partnership for Advanced Computing National Facility. In this paper, we use Fortran-90 supported by the MPI (Gropp *et al.*, 1998; Snir *et al.*, 1998a; 1998b) in parallel programming. MPI is a library specification for message-passing and designed for high performance on both massively parallel machines and on workstation clusters. Processor loads are almost balanced with static balancing where sub-domains contain the same number of collocation points and then the same number of dumbbells. Hence, the parallelization for both microscopic and macroscopic levels is carried out on each sub-domain.

5. Numerical examples

In this section, the ability of the RBFN-BCF method in conjunction with parallel domain decomposition technique to simulate challenging problems is estimated, using the steady state planar 10:1 contraction of the FENE dumbbell model fluids. In order to ensure accurate results, the velocity and stress are obtained by averaging the results of a large number of iterations that were continued after convergence. Furthermore, by using the variance reduction method (see section (2.1)), the statistical errors have been shown to be acceptable for a mesoscopic approach (Laso *et al.*, 1997; 1999).

5.1. 10:1 planar contraction flow

This problem was simulated with different models and schemes by several researchers (Feigl and Ottinger, 1996; Laso, 1998; Laso *et al.*, 1999). For the sake of comparison, the same model and physical parameters as in Laso (1998), Laso *et al.* (1999) are used here. In particular, the results for FENE model are compared with the results of Laso (1998) and Laso *et al.* (1999).

Owing to the symmetry, only half of the analysis domain needs be considered as shown in Fig. 5. Let 2h (h=0.003m) be the height of downstream channel. The height of upstream channel is 2H=20h; the upstream and downstream lengths are chosen to be L_u=6H and L_d=60h, respectively. The fluid parameters are as follows

$$\rho = 1000\ \text{kg/m}^3\ ;\ \eta_N = 0.01\ \text{Pa}\cdot\text{s}\ ;\ b = 50\ ;$$
$$\alpha = \frac{\eta_N}{\eta_o} = 0.2\ ;\ \lambda_H = 2\ \text{s}. \qquad (29)$$

The characteristic length is chosen to be h; the characteristic velocity $\langle u_d \rangle$ (downstream average axial velocity corresponding to the flow rate D_r); the characteristic vis-

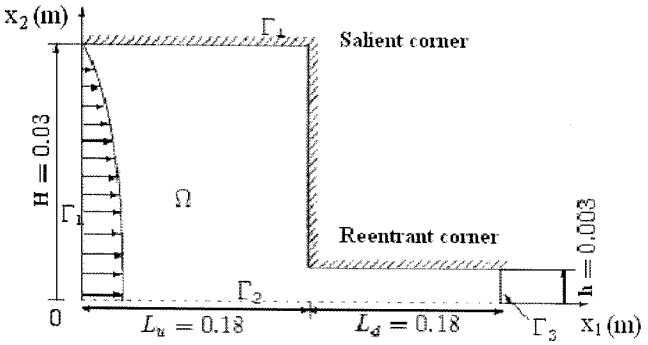


Fig. 5. Planar 10:1 contraction flow problem: non-slip boundary conditions are imposed at the fluid-solid interfaces; fully developed Poiseuille at the inlet and outlet, and symmetry on the centre-line.

cosity $\eta_o = \eta_N + \eta_p$; the characteristic time λ_H and the flow rate is $D_r = 2\times10^{-5}$ m^3/s.m.

The global boundary conditions of the problem are the same as in Laso *et al.* (1999), Ryssel and Brunn (1999), Raghay and Hakim (2001) where no slip condition is imposed along the wall; a fully developed velocity profile is specified at the inlet and the symmetry condition is imposed on the centre-line.

Let Γ_1, Γ_2, Γ_3 and Γ_4 be the boundaries of the domain Ω (Fig. 5). The boundary conditions are given by

1. On the wall Γ_4, there is no slip condition, *i.e.*, $u_{x1}=u_{x2}=0$;

2. At the inlet (Γ_1), the flow is fully developed. In this work, the velocity profile at the inlet is determined by solving the 2D Poiseuille problem with the same flow rate D_r. The approximated velocity profile, and the dumbbells leaving the domain after convergence for a time period Δt are stored and used as the inlet condition in the 10:1 contraction flow problem. The inlet velocity profile obtained from this problem (Fig. 6) is in good agreement of the results of Laso *et al.* (1999);

3. At the outlet (Γ_3), $u_{x2}=0$ and u_{x1} is considered as an unknown and need to be calculated;

4. On the centreline (Γ_2), the symmetry condition is given by

$$\frac{\partial u_{x1}}{\partial x_2}(\mathbf{x})=0,\ u_{x2}=0 \tag{30}$$

Four different densities of collocation points are employed to simulate this problem and several corresponding geometry parameters are given in Table 1. The collocation points are not uniformly distributed. A higher density of collocation points is specified in the reentrant corner area to capture the expected strong flow gradients in that region. The stochastic simulation is time-dependent with simulation time step $\Delta t=0.005\lambda_H=0.01$.

In order to deal with this large scale problem, the analysis domain is divided into a number of sub-domains. Each

Table 1. Planar 10:1 contraction flow problem: four sets of collocation points. N_t the total number of collocation points; N_{iu}: the number of internal points of the upstream flow; N_{id}: the number of internal points of the downstream flow; N_b: the number of boundary points

N_t	N_{iu}	N_{id}	N_b
720	453	116	151
1024	667	186	171
1632	1102	312	218
1943	1286	414	243

sub-domain is governed by a separate CPU. At an iterative step, there are two groups of tasks, one for the stochastic process to determine the stress tensor and the other one is for deterministic operations to calculate the velocity field.

The stochastic task consists of solving SDEs and computing average stresses at collocation points. In general, these tasks are carried out independently, except several data exchange through the interfaces of adjacent sub-domains, especially when generating random numbers. In this work, the average stress at the collocation points fixed on interfaces is the average of the whole configuration fields obtained from adjacent sub-domains.

The macroscopic task is to solve the governing PDEs. Owing to the penalty function method, only unknowns **u** need to be determined in each sub-domain. The algorithm of the parallel mesh-free TPS-RBFN was presented in section 3 and 3.1 in which the sum square error associated with each sub-domain depends on its boundary conditions (*i.e.* the location of sub-domain). For each iterative step, the parallel iterative non-overlapping DD procedure requires the communication between adjacent sub-domains such as the interchanges of Dirichlet and Neumann data at

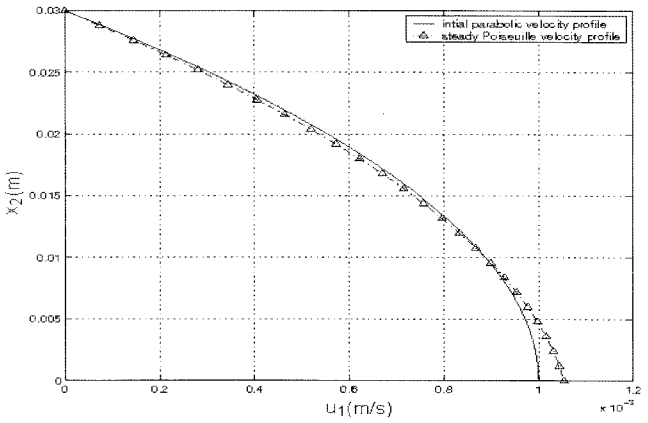


Fig. 6. The steady state planar 10:1 contraction flow problem using the FENE dumbbell model fluid: the inlet velocity profile obtained from the steady Poiseuille flow problem, De=4.4.

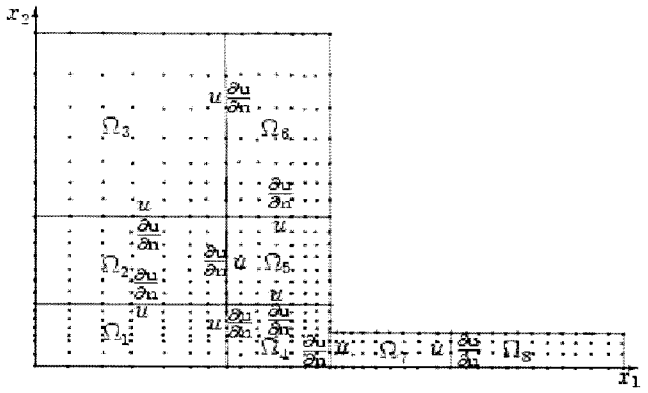


Fig. 7. Planar 10:1 contraction flow problem: Domain decomposition technique using 8 sub-domains; $G_b=[\Omega_1, \Omega_3, \Omega_5, \Omega_7]$ and $G_w=[\Omega_2, \Omega_4, \Omega_6, \Omega_8]$.

their interfaces from the previous iterative step. The results presented in this work correspond to the use of 1632 collocation points with 8 sub-domains and 400 dumbbells assigned at each collocation point. A schematic distribution of collocation points and a sub-regioning with 8 sub-domains are shown in Fig. 7.

5.2. Results and discussion

In order to compare the present results with those obtained by Laso *et al.* (1999), the Deborah and Reynold numbers associated with the flow are chosen to be the same, namely,

$$De = \frac{\lambda_H \langle u_d \rangle}{h} = \lambda_H \frac{D_r}{2h^2} = 4.44,$$

$$Re = \frac{\rho \langle u_d \rangle h}{\eta_o} = 0.4,$$

where $\langle u_d \rangle$ is the downstream average axial velocity corresponding to the flow rate D_r. The velocity and stresses are obtained by averaging the results of 500 iterations that were continued after convergence. For this problem the convergence measures (Please see Tran-Canh and Tran-Cong, 2002; 2003 for the convergence measure formulaes), CM_u<1.0e-4 for the velocity field (Fig. 8) and CM_τ<2.0e-

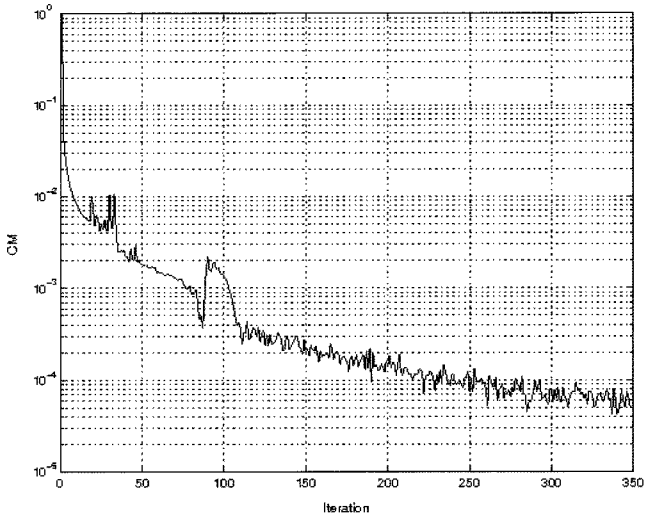


Fig. 8. The steady state planar 10:1 contraction flow problem using the FENE dumbbell model fluid: the convergence behavior (CM_u) of the velocity field **u** with respect to the iteration number, using 8 subdomains.

3 for the stress tensor. Fig. 9 shows the streamlines which are generally in good agreement with the results of Laso *et al.* (1999). However, a small difference in the triangle vortex shape can be seen, for example, the length of edge parallel with the face containing the salient and re-entrant corners is longer for our result, (1.8 vs 1.5). While the vortex hypotenuse of the present work is straight, it is lightly concave into the salient corner for the Laso result. (Please refer to Fig. 6.7, Laso *et al.* 1999).

Fig. 10 shows the velocity profiles at several cross-sections along the downstream channel x_1=0.18123 (near the abrupt contraction) and 0.360 (outlet). These results are in very good agreement with that mentioned in Figure 6.11, Laso *et al.*, 1999. Fig. 11 shows the variation of the axial velocity on several planes x_2=0.0150, 0.0028, 0.0013 and on the centre-line. The present results agree with those obtained by Laso *et al.* (1999), for example the axial velocity on the centreline (see Figure 6.16.a, Laso *et al.* 1999). The current results also depicted the velocity overshoot feature at the abrupt contraction, which was reported in

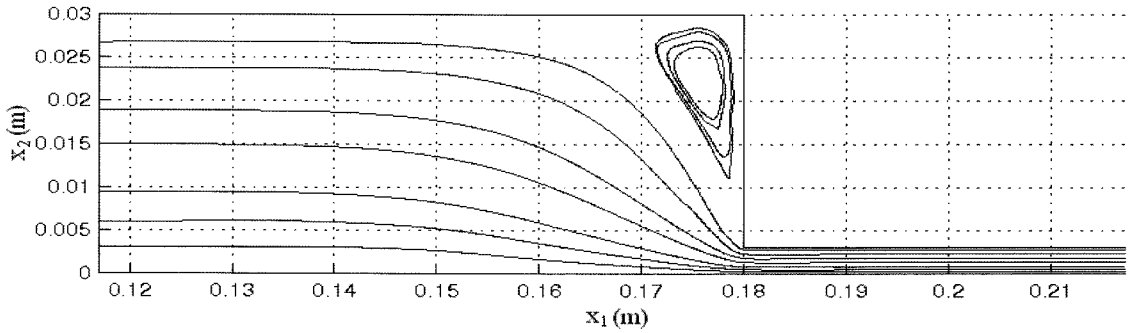


Fig. 9. The steady state planar 10:1 contraction flow problem using the FENE dumbbell model fluid: the streamline of velocity field, De=4.4.

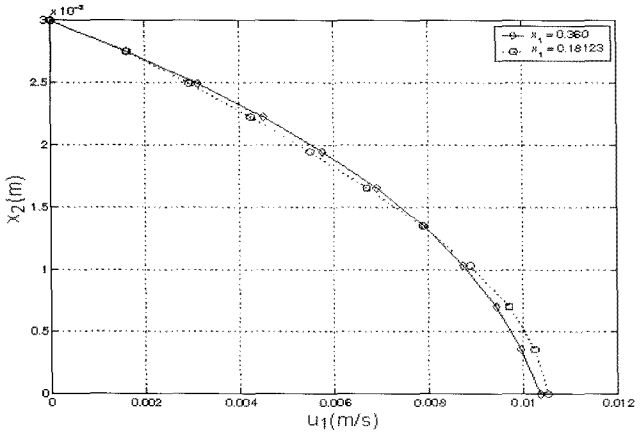


Fig. 10. The steady state planar 10:1 contraction flow problem using the FENE model: The velocity profile on several cross-sections of the downstream channel $x_1=0.360$ (outlet) and 0.1812; De=4.4.

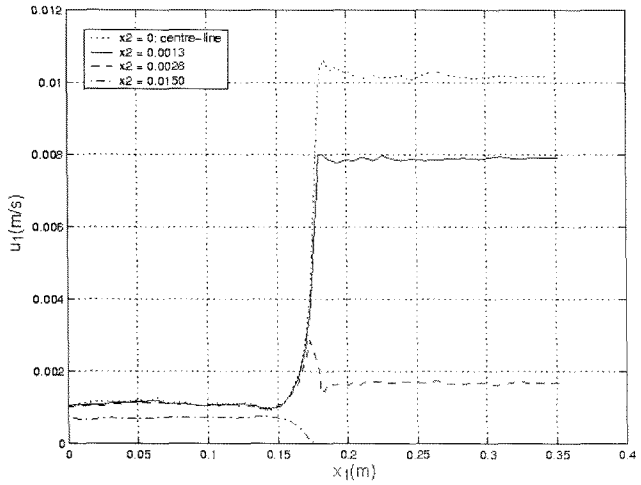


Fig. 11. The steady state planar 10:1 contraction flow problem using the FENE model fluid: The velocity profile on several planes $x_2=0.0013$, 0.0028 and 0.0150 and on the centreline; De=4.4.

Feigl and Ottinger (1996).

Figs. 12 and 13 present the shear stresses along the channel on several planes $x_2=0.00020$, 0.00047, 0.00114 and 0.00284. It can be seen that there is an abrupt change of shear stress at the cross-section of the contraction, which is increasingly pronounced towards the re-entrant corner. Although Figs. 12 and 13 only differ from another by one curve our aim is to show the difference of τ_{12} along x_1 at the position near the re-entrant corner using different scales for the vertical axes.

Figs. 14 and 15 show the first normal stress difference on several planes $x_2=0.00020$, 0.00047, 0.00081, 0.00114, 0.00258 and 0.00284. On the planes nearer to the centreline, stress overshoots can be observed.

Figs. 16 and 17 show the variation in behavior of the

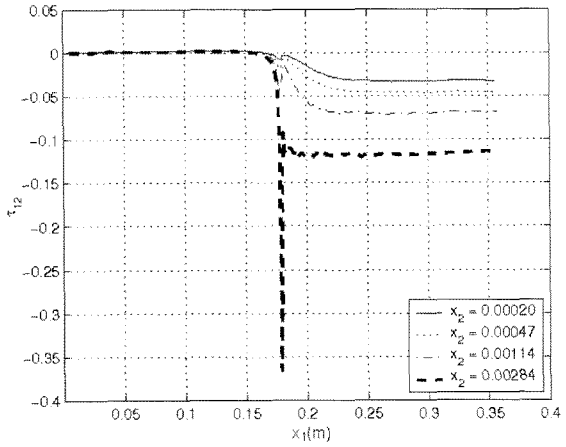


Fig. 12. The steady state planar 10:1 contraction flow problem using the FENE model fluid: The polymer shear stress on several planes $x_2=0.00020$, 0.00047, 0.00114 and 0.00284; De=4.4.

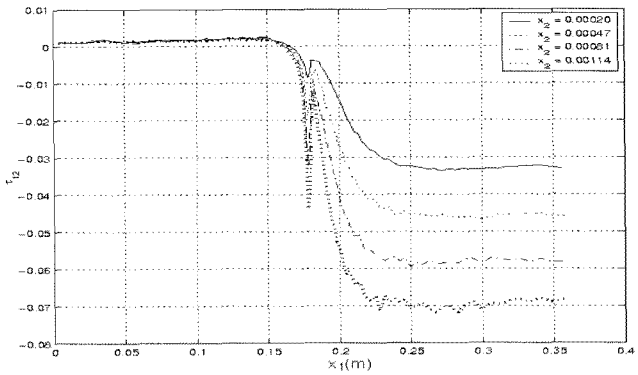


Fig. 13. The steady state planar 10:1 contraction flow problem using the FENE model fluid: The polymer shear stress on several planes $x_2=0.00020$, 0.00047, 0.00081 and 0.00114; De=4.4.

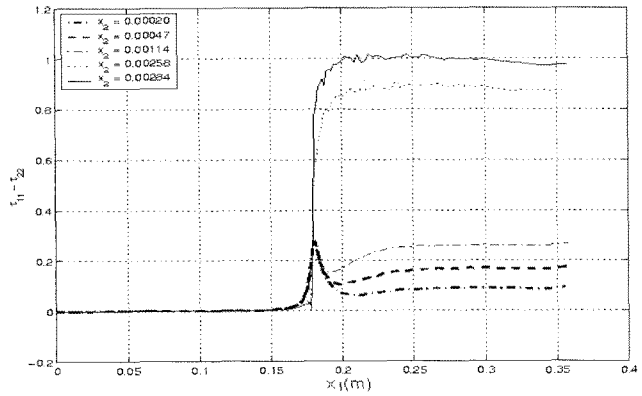


Fig. 14. The steady state planar 10:1 contraction flow problem using the FENE model fluid: The first polymer normal stress difference on several planes $x_2=0.00020$, 0.00047, 0.00114, 0.00258 and 0.00284; De=4.4

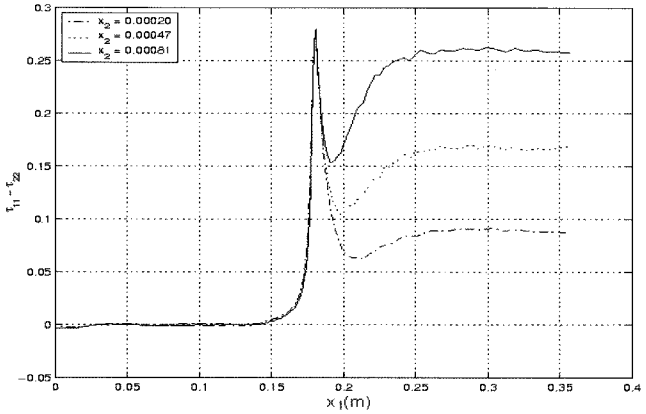


Fig. 15. The steady state planar 10:1 contraction flow problem using the FENE model fluid: The first polymer normal stress difference on several planes x_2=0.00020, 0.00047 and 0.00081; De=4.4.

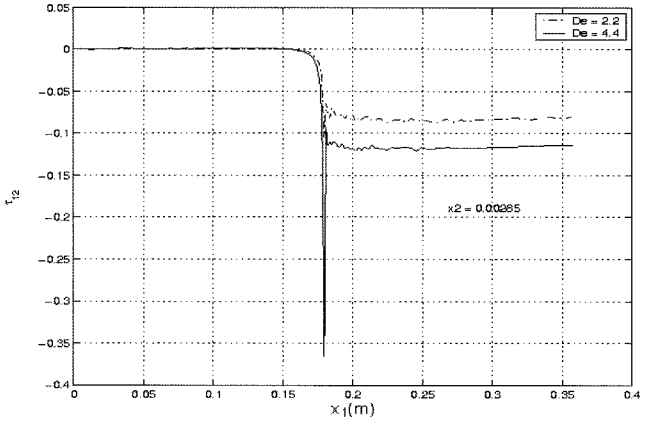


Fig. 16. The steady state planar 10:1 contraction flow problem using the FENE model fluid: the polymer shear stresses on the plane x_2=0.00285 near the reentrant corner, for two Deborah numbers De=4.4 and 2.2.

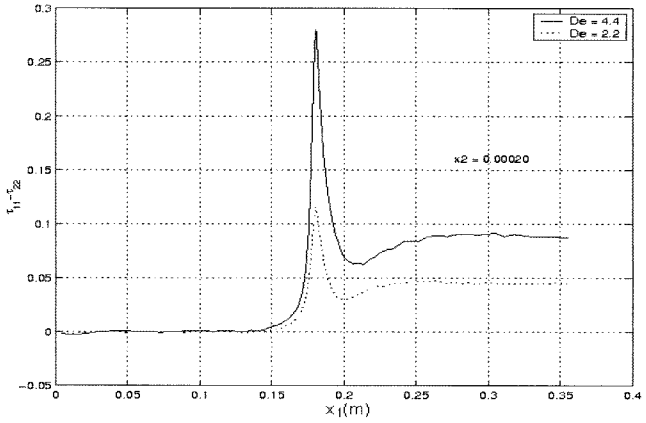


Fig. 17. The steady state planar 10:1 contraction flow problem using the FENE model fluid: the polymer first normal stress differences on the plane x_2=0.00020 near the centre-line, for two Deborah numbers De=4.4 and 2.2.

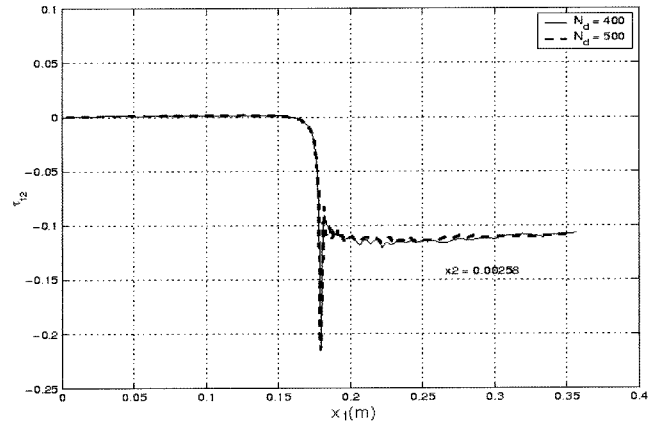


Fig. 18. The steady state planar 10:1 contraction flow problem using the FENE model fluid: the polymer shear stress on planes x_2=0.00258 for Deborah number De=4.4 , using N_d=400 and 200 dumbbells at each collocation point.

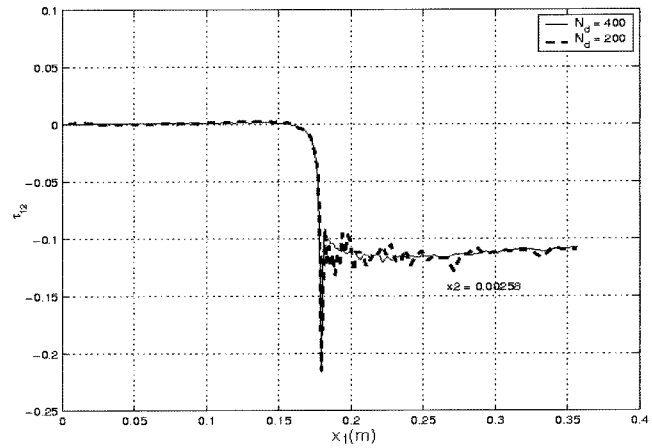


Fig. 19. The steady state planar 10:1 contraction flow problem using the FENE model fluid: the polymer shear stress on the plane x_2=0.00258 for Deborah number De=4.4, using N_d=400, 200 and 500 dumbbells at each collocation point.

polymer shear stresses and the first normal stress differences with Deborah numbers where the strength of the stress overshoots increases with increasing elasticity (higher Deborah number).

The problem is also solved using 200 and 500 dumbbells at each collocation point and the results show that there is no discernible difference between using 400 and 500 dumbbells at each collocation point (Fig. 18) but an increased oscillation of the stresses when reducing the number of dumbbells at each collocation point down to 200 (Fig. 19).

In order to estimate the efficiency of the parallel algorithm in conjunction with BCF-RBFN mesh-free method, a range of 8, 12, 16 and 20 sub-domains (CPUs) are also carried out using 1632 collocation points. Here, two ele-

Table 2. Planar 10:1 contraction flow problem: two sets of collocation points. N: the number of subdomains; N_t the total number of collocation points; N_s: the number of collocations points of a subdomain; N_{int}: the number of collocation points (range) on the interface of each subdomain; CM_u the convergence measure and E_n: the efficiency coefficient of the algorithm of the parallel DD technique

N	N_t	N_s	N_{int}	CM_u	E_n %
8	1632	210	10→51	1.12e-4	87.6
12	1632	140	10→48	1.20e-4	83.4
16	1632	110	10→42	1.82e-4	77.2
20	1632	90	10→38	2.12e-4	68.7

ments of interest studied to assess this algorithm: (i) the convergence measure or norm of error (CM) and (ii) the efficiency which is defined as the ratio of the execution time using one processor (T1) and the execution time using N processors. For this numerical example, due to the difficulties associated with solving a large scale problem, T1 is obtained from using 8 sub-domains but carrying out serially on only one processor. T1 and the required memory for the problem are 764 minutes and 430 Mbytes, respectively, on Pentium 3, 1.8 GHz, using Fortran 90. Table 2 shows the effect of the number of sub-domains on the CM and the number of CPUs on the efficiency of the parallel technique. Here, the method of calculating T1 is slightly different from that reported in Tran-Canh and Tran-Cong (2003). The results given in Table 2 show that the efficiency coefficient is quite high in comparison with the results reported on the efficiency of using the parallel techniques in the mesoscopic approach (for example in Laso *et al.*, 1997).

6. Concluding remarks

A parallel domain decomposition mesh-free TPS-RBFN-BCF method for dilute polymer solution is reported. The incorporation of a parallel domain decomposition technique makes the RBFN-BCF element-free methods more suitable for large scale problems. The iterative non-overlapping domain decomposition method, which is employed in both macro and micro components of the hybrid TPS-RBFN-BCF simulation scheme, converts a large problem into a number of smaller ones, facilitates coarse grained parallelization and decreases the wall time of a micro-macroscopic method. The experience gained from solving one benchmark problem shows the following (i) Although the convergence of the scheme can be affected by the number of subdomains, the results obtained are very good, judging by the convergence measure (Table 1); (ii) In spite of the complexity involved in parallelizing the mesoscopic approach as mentioned in Laso *et al.* (1997), the achieved efficiency is high (Table 2) (≥60% when using 20 CPUs); (iii) It is relatively simple to implement parallel domain decomposition using MPI for both macro and microscopic components.

Acknowledgements

This work is supported by a grant of computing time from Australia Partnership for Advanced Computing (APAC) National Facility for Tran-Cong. This support is gratefully acknowledged. The authors would like to thank Dr. Alfred Uhlherr (CSIRO, Molecular and Health technologies, Clayton, Melbourne) for critically reading the manuscript and correcting errors and the referees for their helpful comments.

References

Beatson, R. K., W. A. Light, and S. Billings, 2000, Fast solution of the radial basis function interpolation equations: domain decomposition methods, *SIAM J. Sci. Comput.* **22(5)**, 1717-1740.

Dissanayake, M. W. M. G. and N. Phan-Thien, 1994, Neural network based approximations for solving partial differential equations, *Commun. Numer. Meth. Engng.* **10**, 195-201.

Dubal, M. R., 1994, Domain decomposition and local refinement for multiquadratic approximations. I: Second order equations in one-dimension, *J. Appl. Sci. Comput.* **1**, 146-171.

Feigl, K. and H. C. Ottinger, 1996, A numerical study of the flow of a low-density polyethylene melt in a planar contraction and comparison to experiments, *J. Rheol.* **40**, 21-35.

Franke, R. 1982, Scattered data interpolation: test of some methods, *Math. Comput.* **48**, 181-200.

Funaro, D., A. Quarteroni and P. Zanolli, 1988, An iterative procedure with interface relaxation for domain decomposition methods, *SIAM J. Numer. Anal.* **25(6)**, 1213-1236.

Gropp, W., S. Huss-Lederman, A. Lumsdaine, E. Lusk, W. Saphir and M. Snir, 1998, MPI-The complete reference, vol 2, the MPI extension, London: The MIT Press.

Hornik, K., M. Stinchcombe and H. White, 1989, Multilayer feedforward networks are universal approximators, *Neural Networks* **2**, 359-366.

Hulsen, M. A., A. P. G. Van Heel and B. H. A. A. Van Den Brule, 1997, Simulation of viscoelastic flows using Brownian Configuration fields, *J. Non-Newt. Fluid Mech.* **70**, 79-101.

Laso, M., M. Picasso and H. C. Ottinger, 1997, Two dimensional time dependent viscoelastic flow calculations using CONNFFESSIT, *AIChE J.* **43(4)**, 877-892.

Laso, M. 1998, Two-dimension, time dependent viscoelastic flow

calculations using molecular models. In: Adam, M. J., Mashelkar, R. A., Pearson, J. R. A. and Rennie, A. R. (eds), Dynamics of complex fluids, Proceedings of the second Royal Society Unilever Indo-UK forum in material science and engineering, Chapter 6, 73-87, London: Imperial College Press.

Laso, M., Picasso, M. and H. C. Ottinger, 1999, Calculation of flows with large elongation components: CONNFFESSIT calculation of the flow of a FENE fluid in a planar 10:1 contraction, In: Nguyen, T. Q. and H. H. Kausch, Flexible polymer chains dynamics in elongational flow: theory and experiment, Chapter 6, 101-136, Berlin: Springer.

Marini, L. D. and A. Quarteroni, 1988, An iterative procedure for domain decomposition methods: a finite element approach. In Glowinski, R, G. H. Golub, G. A. Meurant and J. Periaux, (eds), Proceedings of first international symposium on domain decomposition methods for partial differential equations, 129-143, Philadelphia: SIAM.

Quarteroni, A. and A. Valli, 1999, Domain decomposition methods for partial differential equations. Oxford: Clarendon Press.

Raghay, S. and A. Hakim, 2001, Numerical simulation of White-Metzner fluid in a 4:1 contraction, *Int. J. Numer. Meth. Fluids* **35**, 559-537.

Ryssel, E. and P. O. Brunn, 1999, Comparison of quasi-Newtonian fluid with a viscoelastic fluid in planar contraction flow, *J. Non-Newt. Fluid Mech.* **86**, 309-335.

Smith, B. F., P. E. Bjorstad and W. D. Gropp, 1996, Domain decomposition parallel multilevel methods for elliptic partial differential equations, New-York: Cambridge University Press.

Snir, M., S. Otto, S. Huss-Lederman, D. Walker and J. Dongarra, 1998a, MPI-The complete reference, V1. London: The MIT Press.

Snir, M., S. Otto, S. Huss-Lederman, D. Walker and J. Dongarra, 1998b, MPI-The complete reference, the MPI core, V2. London: The MIT Press.

Tran-Canh, D. and T. Tran-Cong, 2002, BEM-NN computation of generalised Newtonian flows, *J. Engineering Analysis with Boundary Elements* **26**, 15-28.

Tran-Canh, D. and T. Tran-Cong, 2003, Parallel meshfree TPS-RBFN-BCF method for dilute polymer solution, The proceeding, of the 2nd Korean-Australia Rheology Conference, Gyeongju, 67-70, Gyeongju: The Korean Society of Rheology and The Australian Society of Rheology.

Tran-Canh, D. and T. Tran-Cong, 2004, Element-free simulation of dilute polymeric flows using Brownian configuration field, *Korea-Australia Rheology J.* **16(1)**, 1-15.

Xu, J. and J. Zou, 1998, Some nonoverlapping domain decomposition methods, *Siam Rev.* **40(4)**, 857-914.

Yang, D., 1996, A parallel iterative non-overlapping domain decomposition procedure for elliptic problems, *SIAM J. Numer. Anal.* **16**, 75-91.

Zerroukat, M., H. Power and C. S. Chen, 1998, A numerical method for heat transfer problems using collocation and radial basis functions, *Int. J. Numer. Meth. Engng.* **42**, 1263-1278.

Zhou, X., Y. C. Hon and J. Li, 2003, Overlapping domain decomposition method by radial basis functions, *Appl. Numer. Math.* **44(1-2)**, 241-255.