



HAL
open science

A New Family of Solvers for Some Classes of Multidimensional Partial Differential Equations Encountered in Kinetic Theory Modeling of Complex Fluids

Amine Ammar, Béchir Mokdad, Francisco Chinesta, Roland Keunings

► **To cite this version:**

Amine Ammar, Béchir Mokdad, Francisco Chinesta, Roland Keunings. A New Family of Solvers for Some Classes of Multidimensional Partial Differential Equations Encountered in Kinetic Theory Modeling of Complex Fluids. *Journal of Non-Newtonian Fluid Mechanics*, 2006, 139 (3), pp.153-176. 10.1016/j.jnnfm.2006.07.007 . hal-01004909

HAL Id: hal-01004909

<https://hal.science/hal-01004909>

Submitted on 2 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids

A. Ammar^a, B. Mokdad^a, F. Chinesta^b, R. Keunings^c

^a *Laboratoire de Rhéologie, INPG, UJF, CNRS (UMR 5520), 1301 rue de la piscine, BP 53 Domaine Universitaire, F-38041 Grenoble Cedex 9, France*

^b *Laboratoire de Mécanique des Systèmes et des Procédés, UMR 8106 CNRS-ENSAM-ESEM, 151 Boulevard de l'Hôpital, F-75013 Paris, France*

^c *CESAME, Université Catholique de Louvain, Bat. Euler, Av. Georges Lemaitre 4, B-1348 Louvain-la-Neuve, Belgium*

Kinetic theory models involving the Fokker–Planck equation can be accurately discretized using a mesh support (finite elements, finite differences, finite volumes, spectral techniques, etc.). However, these techniques involve a high number of approximation functions. In the finite element framework, widely used in complex flow simulations, each approximation function is related to a node that defines the associated degree of freedom. When the model involves high dimensional spaces (including physical and conformation spaces and time), standard discretization techniques fail due to an excessive computation time required to perform accurate numerical simulations. One appealing strategy that allows circumventing this limitation is based on the use of reduced approximation basis within an adaptive procedure making use of an efficient separation of variables.

Keywords: Complex fluids; Kinetic theory; Model reduction; Multidimensional problems; Separation of variables; Numerical modeling

1. Introduction

Many natural and synthetic fluids are viscoelastic materials, in the sense that the stress endured by a macroscopic fluid element depends upon the history of the deformation experienced by that element. Notable examples include polymer solutions and melts, liquid crystalline polymers and fibre suspensions. Rheologists thus face a challenging non-linear coupling between flow-induced evolution of molecular configurations, macroscopic rheological response, flow parameters (such as the geometry and boundary conditions) and final properties. Theoretical modeling and methods of computational rheology have an important role to play in elucidating this coupling.

Atomistic modeling is the most detailed level of description that can be applied today in rheological studies, using techniques of non-equilibrium molecular dynamics. Such calculations require enormous computer resources, and they are currently limited to flow geometries of molecular dimensions. Consideration of macroscopic flows found in processing applications calls for less detailed mesoscopic models, such as those of kinetic theory.

Models of kinetic theory provide a coarse-grained description of molecular configurations wherein atomistic processes are ignored. They are meant to display in a more or less accurate fashion the important features that govern the flow-induced evolution of configurations. Over the last few years, different models related to dilute polymers have been evaluated in simple flows by means of stochastic simulation [9] or Brownian dynamics methods.

In this context, micro–macro methods of computational rheology that couple the coarse-grained molecular scale of kinetic theory to the macroscopic scale of continuum mechanics have an important role to play (for a review, see [10]). This approach is much more demanding in computer resources than more conventional continuum simulations that integrate a constitutive equation to evaluate the viscoelastic contribution of the stress tensor. On the other hand micro–macro techniques allow the direct use of kinetic theory models and thus avoid the introduction of closure approximations.

Since the early 1990s the field has developed considerably following the introduction of the CONNFESSIT method by Ottinger and Laso [14]. Being relatively new, micro–macro techniques have been implemented only for models of kinetic theory with few configurational degrees of freedom, such as non-linear dumbbell models of dilute polymer solutions and single-segment tube models of linear entangled polymers.

Kinetic theory provides two basic building blocks: the diffusion or Fokker–Planck equation that governs the evolution of the distribution function (giving the probability distribution of configurations) and an expression relating the viscoelastic stress to the distribution function. The Fokker–Planck equation has the general form

$$\frac{D\psi}{Dt} + \frac{\partial}{\partial \underline{X}}(\underline{A}\psi) = \frac{1}{2} \frac{\partial}{\partial \underline{X}} \frac{\partial}{\partial \underline{X}} : (\underline{D}\psi) \quad (1)$$

where D/Dt is the material derivative, vector \underline{X} defines the coarse-grained configuration and has dimensions N . Factor \underline{A} is a N -dimensional vector that defines the drift or deterministic component of the molecular model. Finally \underline{D} is a symmetric, positive definite $N \times N$ matrix that embodies the diffusive or stochastic component of molecular model. In general both \underline{A} and \underline{D} (and in consequence the distribution function ψ) depend on the physical coordinates \underline{x} , on the configuration coordinates \underline{X} and on the time t .

The second building block of a kinetic theory model is an expression relating the distribution function and the stress. It takes the form

$$\underline{\tau}_p = \int_C \underline{g}(\underline{X})\psi \, d\underline{X} \quad (2)$$

where C represents the configuration space and \underline{g} is a model-dependent tensorial function of configuration. In a complex flow, the velocity field is a priori unknown and stress fields are coupled through the conservation laws. In the isothermal and incompressible case the conservation of mass and momentum balance are then expressed (neglecting the body forces)

$$\begin{cases} \text{Div } \underline{v} = 0 \\ \rho \frac{Dv}{Dt} = \text{Div}(-p\underline{I} + \underline{\tau}_p + \eta_s \underline{d}) \end{cases} \quad (3)$$

where ρ is the fluid density, p the pressure and $\eta_s \underline{d}$ a purely viscous component (\underline{d} being the strain rate tensor). The set of coupled Eqs. (1)–(3), supplemented with suitable initial and boundary conditions in both physical and configuration spaces, is the generic multiscale formulation.

Kinetic theory models involving the Fokker–Planck equation, can be accurately discretized using a mesh support (finite elements, finite differences, finite volumes, spectral techniques, etc.). However these techniques involve a high number of approximation functions. In the finite element framework, widely used in complex flow simulations, each approximation function (also known as shape function) is related to a node that defines the associated degree of freedom. When the model involves high dimensional spaces (including physical and conformation spaces and time) standard discretization techniques fail due to an excessive computation time required to perform accurate numerical simulations.

To alleviate the computational effort some reduction strategies have been proposed, as is the case of the partial solution of the cyclic reduction (PSCR) [11,16] which solves linear systems by using techniques of separation of variables, partial solution and projection techniques [1,12,17] and [19]. Despite the significant reduction of the computational effort this technique can only be applied for solving particular partial differential equations and it fails also in the multidimensional case.

Another appealing strategy that allows alleviating the computational effort is based on the use of reduced approximation bases within an adaptive procedure. The new approximation functions are defined in the whole domain and they contain the most representative information of the problem solution. Thus, the number of degrees of freedom involved in the solution of the Fokker–Planck equation is drastically reduced. The construction of those new approximation functions is done with an ‘a priori’ approach, which combines a basis reduction (using the Karhunen–Loève decomposition) with a basis enrichment based on the use of some Krylov’s subspaces. This strategy has been successfully applied, in some of our former works, to solve kinetic theory models defined on the surface of the unit sphere (for simulating short fibers suspensions) as well as 3D models describing some molecular models [3,15]. The main conclusion of our former works was the fact that an accurate description of a complex system evolution can, in general, be performed from the linear combination of a reduced number of space functions (defined in the whole space domain), the coefficients of that linear combination evolving in time. Thus, during the solution of the evolution problem the coefficients of the approximation are computed at the same time that the numerical algorithm constructs the reduced approximation basis. An important drawback of one such approach is the fact that the approximation functions are defined in the space domain, and until now, the simplest form to represent one such function is given its value in some points of the domain of interest, being its value defined in any other point

by interpolation. Sometimes the treated model results highly multidimensional, as described later, and in this case the possibility of describing functions from their values at the nodes of a mesh or a grid in the domain of interest results simply prohibitory.

Thus, a new strategy able to operate in these conditions must be proposed, validated and finally applied. The main aim of this paper is to present a new numerical strategy able to circumvent some of the difficulties due to in the multidimensional character of kinetic theory models. The first results will be presented to illustrate its capability for treating highly dimensional spaces by considering different multidimensional Poisson's problems as well as some MBS molecular models consisting of S springs that are assumed with finite extensibility and located inside a one-dimensional physical space; that is, all springs are aligned in the same direction that coincides with the solicitation direction, which allows characterize the MBS from S coordinates related to the spring extensions.

2. Issues related to multidimensional models

As just mentioned, many physical problems need a multidimensional description which strongly affects the efficiency of standard numerical techniques. This difficulty is encountered especially in the case of microscopic description of complex fluids. The complexity of such models lies in the multidimensional character of their conformation. Thus, if the molecules are idealized as simple springs, the function that defined the distribution is defined in a space of dimension 7: the physical space that defines the position of the molecule centre of mass (3D), the time (1D) and three other coordinates to define the springs orientation and extension. If we assume that molecules consist of S connected springs, then the problem is defined in a $(3 + 1 + 3S)$ D space (the three space coordinates, the time and the three conformation coordinates related to each of the S springs defining the molecule).

To solve this kind of problems one can use: (i) deterministic discretization techniques [6,13,8,7,5,2] that become rapidly inefficient when the dimension of the space increases; (ii) decoupling techniques as used in [21] or more recently in [20] that are only valid for some classes of models; (iii) stochastic methods [18] where no multidimensional meshes are required, but the counterpart is that the number of realizations of the stochastic process increases with the dimensionality of the space (for a given precision), with the associated severe impact on the simulation time.

For these reasons, the use of a deterministic fixed mesh strategy (like the finite element or finite differences methods) seems to be the most convenient choice to solve the problems related to the multidimensional molecular configuration evolution, if one is able to circumvent its main difficulties: (i) the construction of a mesh (or a grid) in a multidimensional space cannot be contemplated because the extremely large number of degrees of freedom involved which makes the solution of the resulting linear systems impossible; (ii) even if the number of degrees of freedom can be drastically reduced by using some model reduction technique, the definition of a function in a multidimensional space requires the prescription of its value at some points from which an interpolation can be done at any point. However, the number of required points to define such interpolation is too large in multidimensional problems, which makes those problems untreatable. For example, if the problem is defined in a hyper-cube of dimension $N(N = 80)$ and we consider a grid defined by 10 nodes along each direction, the resulting number of nodes will be $N = 10^{80}$ (note that 10^{80} is the presumed number of elementary particles in the universe, and that $N = 80$ corresponds to a molecule composed of less than 30 bead connectors – remember that each spring has three conformation degrees of freedom in the 3D physical space – even if the real molecules are composed by several thousands).

Usual simulation techniques for kinetic theory models involve a first decoupling between the physical and the conformation spaces. Moreover, due to the novelty of the proposed approach, we will only consider molecular models defined in a 1D physical spaces (higher physical spaces will be briefly addressed in the last section and in a forthcoming paper). Thus, this work focuses on the solution of multidimensional problems (the problems related to the conformation coordinates) defined in a hyper-cube with homogeneous boundary conditions. In effect, if we assume that the S springs that compose the molecules have a total finite limit extension, L_{\max} , then the conformation problem can be defined in the hyper-cube $]-L, +L[^N$, N being the dimension of the space and $L = L_{\max}/S$. The boundary conditions are homogeneous because the probability of encountering a completely extended spring vanishes.

In this work, we propose the use of a separated representation which allows to define a tensor product approximation basis as well as to decouple the numerical integration in each dimension. The main aim of the present paper is to introduce a new family of numerical techniques able to treat multidimensional problems (as the ones encountered in kinetic theory).

3. Solution strategy: separated representation, tensor product approximation basis and separated discretization

For the sake of simplicity, we start by considering the Poisson problem defined in a space of dimension N

$$\Delta T = -f(x_1, x_2, \dots, x_N) \tag{4}$$

where T is a scalar function of (x_1, x_2, \dots, x_N) . Problem (4) is defined in the domain $(x_1, x_2, \dots, x_N) \in \Omega =]-L, +L[^N$ with vanishing boundary conditions.

The problem solution can be written in the form

$$T(x_1, x_2, \dots, x_N) = \sum_{j=1}^{\infty} \alpha_j \prod_{k=1}^N F_{kj}(x_k) \quad (5)$$

where F_{kj} is the j th basis function, with unit norm, which only depends on the k th coordinate.

It is well known that the solution of numerous problems can be accurately approximated using a finite (sometimes very reduced) number of approximation functions, i.e.

$$T(x_1, x_2, \dots, x_N) = \sum_{j=1}^Q \alpha_j \prod_{k=1}^N F_{kj}(x_k) \quad (6)$$

The previous expression implies the same number of approximation functions in each dimension, but each one of these functions could be expressed in a discrete form using different number of parameters (nodes of the 1D grids).

Now, an appropriate numerical procedure is needed for computing the coefficients α_j as well as the Q approximations functions in each dimension.

The proposed numerical scheme consists of an iteration procedure that solves at each iteration n the following three steps.

Step 1. Projection of the solution in a discrete basis

If we assume the functions $F_{kj}(\forall j \in [1, \dots, n]; \forall k \in [1, \dots, N])$ known (verifying the boundary conditions), the coefficients α_j can be computed by introducing the approximation of T into the Galerkin variational formulation associated with Eq. (4)

$$\int_{\Omega} \text{Grad} T^* \text{Grad} T \, d\Omega = \int_{\Omega} T^* f \, d\Omega \quad (7)$$

Introducing the approximation of T and T^*

$$T(x_1, x_2, \dots, x_N) = \sum_{j=1}^n \alpha_j \prod_{k=1}^N F_{kj}(x_k) \quad (8)$$

and

$$T^*(x_1, x_2, \dots, x_N) = \sum_{j=1}^n \alpha_j^* \prod_{k=1}^N F_{kj}(x_k) \quad (9)$$

we have

$$\int_{\Omega} \text{Grad} \left(\sum_{j=1}^n \alpha_j^* \prod_{k=1}^N F_{kj}(x_k) \right) \text{Grad} \left(\sum_{j=1}^n \alpha_j \prod_{k=1}^N F_{kj}(x_k) \right) \, d\Omega = \int_{\Omega} \left(\sum_{j=1}^n \alpha_j^* \prod_{k=1}^N F_{kj}(x_k) \right) f \, d\Omega \quad (10)$$

Now, we assume that $f(x_1, \dots, x_N)$ can be written in the form

$$f(x_1, \dots, x_N) = \sum_{h=1}^m \prod_{k=1}^N f_{kh}(x_k) \quad (11)$$

Eq. (10) involves integrals of a product of N functions each one defined in a different dimension. Let $\prod_{k=1}^N g_k(x_k)$ be one of these functions to be integrated. The integral over Ω can be performed by integrating each function in its definition interval and then multiplying the N computed integrals according to

$$\int_{\Omega} \prod_{k=1}^N g_k(x_k) \, d\Omega = \prod_{k=1}^N \int_{-L}^L g_k(x_k) \, dx_k \quad (12)$$

which makes possible the numerical integration in highly dimensional spaces.

Now, due to the arbitrariness of the coefficients α_j^* , Eq. (10) allows to compute the n -approximation coefficients α_j , solving the resulting linear system of size $n \times n$. This problem is linear and moreover rarely exceeds the order of tens of degrees of freedom. Thus, even if the resulting coefficient matrix is densely populated, the time required for its solution is negligible with respect to the one required for performing the approximation basis enrichment (Step 3).

Step 2. Checking convergence

From the solution of T at iteration n given by Eq. (8) we compute the residual Re related to Eq. (4)

$$Re = \frac{\sqrt{\int_{\Omega} (\Delta T + f(x_1, \dots, x_N))^2}}{\|T\|} \quad (13)$$

If $Re < \epsilon$ (epsilon is a small enough parameter) the iteration process stops, yielding the solution $T(x_1, \dots, x_N)$ given by Eq. (8). Otherwise, the iteration procedure continues.

The integral in Eq. (13) can be written as the product of one-dimensional integrals by performing a separated representation of the square of the residual.

Step 3. Enrichment of the approximation basis

From the coefficients α_j just computed the approximation basis can be enriched by adding the new function $\prod_{k=1}^N F_{k(n+1)}(x_k)$. For this purpose we solve the non-linear Galerkin variational formulation related to Eq. (4)

$$\int_{\Omega} \text{Grad} T^* \text{Grad} T \, d\Omega = \int_{\Omega} T^* f \, d\Omega \quad (14)$$

using the approximation of T given by

$$T(x_1, x_2, \dots, x_N) = \sum_{j=1}^n \alpha_j \prod_{k=1}^N F_{kj}(x_k) + \prod_{k=1}^N R_k(x_k) \quad (15)$$

The solution of Eq. (14) can be expressed from the stationarity of functional $J(T)$

$$J(T) = \int_{\Omega} \frac{1}{2} (\text{Grad} T)^2 \, d\Omega - \int_{\Omega} T f \, d\Omega \quad (16)$$

which results in

$$\delta J = \int_{\Omega} \text{Grad} \delta T \text{Grad} T \, d\Omega - \int_{\Omega} \delta T f \, d\Omega = 0 \quad (17)$$

that corresponds to Eq. (14) by putting $\delta T = T^*$.

Now, taking the variation of T according to its expression given by Eq. (15), where the variation of the known functions F_{kj} vanishes, we have

$$\delta T = \delta(R_1(x_1) \cdots R_N(x_N)) = \delta R_1(x_1) R_2(x_2) \cdots R_N(x_N) + \cdots + R_1(x_1) R_2(x_2) \cdots \delta R_N(x_N) \quad (18)$$

that can be written as

$$T^*(x_1, x_2, \dots, x_N) = R_1^*(x_1) R_2(x_2) \cdots R_N(x_N) + \cdots + R_1(x_1) R_2(x_2) \cdots R_N^*(x_N) \quad (19)$$

This leads to a non-linear variational problem, whose solution allows to compute the N functions $R_k(x_k)$. Functions $F_{k(n+1)}(x_k)$ are finally obtained by normalizing, after convergence of the non-linear problem, the functions R_1, R_2, \dots, R_N .

To solve this problem we introduce a discretization of those functions $R_k(x_k)$. Each one of these functions is approximated using a 1D finite element description. If we assume that p_k nodes are used to construct the interpolation of function $R_k(x_k)$ in the interval $[-L, L]$, then the size of the resulting discrete non-linear problem is $\sum_{k=1}^N p_k$. The price to pay for avoiding a whole mesh in the multidimensional domain is the solution of a non-linear problem. However, even in high dimensions the size of the non-linear problems remains moderate and no particular difficulties have been found in its solution.

Concerning the computation time, even when the non-linear solver converges quickly, this step consumes the main part of the global computing time.

Different non-linear solvers have been analyzed: fixed-point, Newton or one based on an alternating directions scheme. The numerical results presented in this paper have been computed using both the Newton scheme and the alternating directions method.

4. Illustrating the solution procedure: matrix formulation of the 2D Poisson's problem

In this case Eq. (4) reduces to

$$\Delta T = -f(x, y) \quad (20)$$

where T is a function of x, y . Eq. (20) is solved in the domain $x, y \in \Omega =]-L, +L[\times]-L, +L[$ with homogeneous boundary conditions.

The solution is now sought in the form

$$T(x, y) = \sum_{j=1}^{\infty} \alpha_j F_j(x) G_j(y) \quad (21)$$

The construction of such solution involves at each iteration n , as just discussed, three steps. In what follows we focus on the first and the last ones.

The approximation is now defined at iteration n by

$$T(x, y) = \sum_{j=1}^n \alpha_j F_j(x) G_j(y) \quad (22)$$

and the enrichment step consists of looking for the functions $R(x)$ and $S(y)$ such that, when introduced in the field approximation

$$T(x, y) = \sum_{j=1}^n \alpha_j F_j(x) G_j(y) + R(x)S(y) \quad (23)$$

they allow to verify the Galerkin variational formulation related to Eq. (20).

Finally, functions $F_{n+1}(x)$ and $G_{n+1}(y)$ are obtained by normalizing the functions $R(x)$ and $S(y)$.

From a practical point of view all the functions must be defined in a discrete form. Thus, functions $F_j(x)$ (respectively $G_j(y)$) and $R(x)$ (respectively $S(y)$) are defined using a 1D finite element interpolation that in our simulations is assumed piecewise linear. We denote by \underline{N} (resp. \underline{M}) the vector containing the value of the p (resp. q) shape functions $N_p(x)$ (resp. $M_q(y)$). Finally \underline{E}_j , \underline{G}_j , \underline{R} and \underline{S} are the nodal description of those functions.

4.1. Computing of the alpha coefficients: projection stage

At iteration n we can write T in the following matrix form

$$T(x, y) = [\underline{N}^T \underline{E}_1 \underline{M}^T \underline{G}_1 \quad \underline{N}^T \underline{E}_2 \underline{M}^T \underline{G}_2 \quad \dots \quad \underline{N}^T \underline{E}_n \underline{M}^T \underline{G}_n] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \quad (24)$$

from which the gradient reads

$$\begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} = \underline{B} \underline{\alpha} = \begin{bmatrix} d\underline{N}^T \underline{E}_1 \underline{M}^T \underline{G}_1 & d\underline{N}^T \underline{E}_2 \underline{M}^T \underline{G}_2 & \dots & d\underline{N}^T \underline{E}_n \underline{M}^T \underline{G}_n \\ \underline{N}^T \underline{E}_1 d\underline{M}^T \underline{G}_1 & \underline{N}^T \underline{E}_2 d\underline{M}^T \underline{G}_2 & \dots & \underline{N}^T \underline{E}_n d\underline{M}^T \underline{G}_n \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \quad (25)$$

where $d\underline{N}$ (resp. $d\underline{M}$) represents the vector containing the value of the shape functions derivatives with respect to the coordinate x (resp. y) at point x (resp. y). The weighting function related to the Galerkin's variational formulation T^* can be expressed in the same form

$$T^*(x, y) = [\alpha_1^* \quad \alpha_2^* \quad \dots \quad \alpha_n^*] \begin{bmatrix} \underline{F}_1^T \underline{N} \underline{G}_1^T \underline{M} \\ \underline{F}_2^T \underline{N} \underline{G}_2^T \underline{M} \\ \vdots \\ \underline{F}_n^T \underline{N} \underline{G}_n^T \underline{M} \end{bmatrix} \quad (26)$$

and its gradient reads

$$\begin{bmatrix} \frac{\partial T^*}{\partial x} & \frac{\partial T^*}{\partial y} \end{bmatrix} = [\alpha_1^* \quad \alpha_2^* \quad \dots \quad \alpha_n^*] \begin{bmatrix} \underline{F}_1^T d\underline{N} \underline{G}_1^T \underline{M} & \underline{F}_1^T \underline{N} \underline{G}_1^T d\underline{M} \\ \underline{F}_2^T d\underline{N} \underline{G}_2^T \underline{M} & \underline{F}_2^T \underline{N} \underline{G}_2^T d\underline{M} \\ \vdots & \vdots \\ \underline{F}_n^T d\underline{N} \underline{G}_n^T \underline{M} & \underline{F}_n^T \underline{N} \underline{G}_n^T d\underline{M} \end{bmatrix} = \underline{\alpha}^{*T} \underline{B}^T \quad (27)$$

Now, we consider the variational formulation related to Eq. (20)

$$\int_{\Omega} \text{Grad}T \text{Grad}T^* \, d\Omega = \int_{\Omega} f(x, y)T^* \, d\Omega \quad (28)$$

where the fact that T^* vanishes at the domain boundary has been introduced (both the trial and the test functions are in the Sobolev's space $H_0^1(\Omega)$). Moreover, we assume that $f(x, y)$ can be also written in the form

$$f(x, y) = \sum_{h=1}^m a_h(x)b_h(y) \quad (29)$$

This particular form will be justified later. Thus, Eq. (28) yields

$$\int_{\Omega} \left[\frac{\partial T^*}{\partial x} \frac{\partial T}{\partial y} \right] \left[\frac{\partial T}{\partial x} \frac{\partial T^*}{\partial y} \right] \, d\Omega = \sum_{h=1}^m \int_{\Omega} a_h(x)b_h(y)T^*(x, y) \, d\Omega \quad (30)$$

Introducing the discrete form of the trial and test functions as well as their derivatives, we obtain

$$\int_{\Omega} \underline{\alpha}^{*\text{T}} \underline{\underline{B}}^{\text{T}} \underline{\underline{B}} \underline{\alpha} \, d\Omega = \sum_{h=1}^m \int_{\Omega} [\alpha_1^* \alpha_2^* \dots \alpha_n^*] \begin{bmatrix} a_h(x) \underline{F}_1^{\text{T}} \underline{N} b_h(y) \underline{G}_1^{\text{T}} \underline{M} \\ a_h(x) \underline{F}_2^{\text{T}} \underline{N} b_h(y) \underline{G}_2^{\text{T}} \underline{M} \\ \vdots \\ a_h(x) \underline{F}_n^{\text{T}} \underline{N} b_h(y) \underline{G}_n^{\text{T}} \underline{M} \end{bmatrix} \, d\Omega \quad (31)$$

where the components of matrix $\underline{\underline{B}}^{\text{T}} \underline{\underline{B}}$ are given by

$$(\underline{\underline{B}}^{\text{T}} \underline{\underline{B}})_{ij} = \underline{F}_i^{\text{T}} \underline{dN} \underline{dN}^{\text{T}} \underline{F}_j \underline{G}_i^{\text{T}} \underline{M} \underline{M}^{\text{T}} \underline{G}_j + \underline{F}_i^{\text{T}} \underline{N} \underline{N}^{\text{T}} \underline{F}_j \underline{G}_i^{\text{T}} \underline{dM} \underline{dM}^{\text{T}} \underline{G}_j \quad (32)$$

Because all the terms in Eq. (31) are the product of functions of x and functions of y , the integral in the domain Ω can be separated as the product of integrals in x and y . Thus, we have

$$\underline{\alpha}^{*\text{T}} \underline{\underline{K}} \underline{\alpha} = \underline{\alpha}^{*\text{T}} \underline{V} \Rightarrow \underline{\underline{K}} \underline{\alpha} = \underline{V} \quad (33)$$

where

$$\begin{aligned} \underline{\underline{K}}_{ij} &= \underline{F}_i^{\text{T}} \left(\int_{x=-L}^{x=+L} \underline{dN} \underline{dN}^{\text{T}} \, dx \right) \underline{F}_j \underline{G}_i^{\text{T}} \left(\int_{y=-L}^{y=+L} \underline{M} \underline{M}^{\text{T}} \, dy \right) \underline{G}_j \\ &\quad + \underline{F}_i^{\text{T}} \left(\int_{x=-L}^{x=+L} \underline{N} \underline{N}^{\text{T}} \, dx \right) \underline{F}_j \underline{G}_i^{\text{T}} \left(\int_{y=-L}^{y=+L} \underline{dM} \underline{dM}^{\text{T}} \, dy \right) \underline{G}_j \end{aligned} \quad (34)$$

and

$$\underline{V}_i = \sum_{h=1}^m \underline{F}_i^{\text{T}} \left(\int_{x=-L}^{x=+L} \underline{N} a_h(x) \, dx \right) \underline{G}_i^{\text{T}} \left(\int_{y=-L}^{y=+L} \underline{M} b_h(y) \, dy \right) \quad (35)$$

The linear system (33) can be solved for computing the coefficients α . The inversion of matrix $\underline{\underline{K}}$ is very simple due its reduced dimension. Of course the iteration procedure stops when $\alpha_{n+1}/\max(\alpha_1, \dots, \alpha_n) < \epsilon$ (ϵ being a small enough parameter).

4.2. Enrichment of the approximation basis

At this stage, we are looking for a solution such that

$$\begin{cases} T(x, y) = \sum_{j=1}^{j=n} \alpha_j F_j(x) G_j(y) + R(x) S(y) \\ T^*(x, y) = R^*(x) S(y) + R(x) S^*(y) \end{cases} \quad (36)$$

In order to compute both R and S from the associated variational formulation (Eq. (28)), we start by writing both functions as well their gradients using a matrix form

$$\begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} = \sum_{j=1}^n \alpha_j \begin{bmatrix} \underline{dN}^{\text{T}} \underline{F}_j \underline{M}^{\text{T}} \underline{G}_j \\ \underline{N}^{\text{T}} \underline{F}_j \underline{dM}^{\text{T}} \underline{G}_j \end{bmatrix} + \begin{bmatrix} \underline{M}^{\text{T}} \underline{S} \underline{dN}^{\text{T}} & \underline{0}_{(1,q)} \\ \underline{0}_{(1,p)} & \underline{N}^{\text{T}} \underline{R} \underline{dM}^{\text{T}} \end{bmatrix} \begin{bmatrix} \underline{R} \\ \underline{S} \end{bmatrix} = \sum_{j=1}^n \alpha_j \underline{D}_j + \underline{E} \begin{bmatrix} \underline{R} \\ \underline{S} \end{bmatrix} \quad (37)$$

$$T^*(x, y) = [\underline{R}^{*\text{T}} \ \underline{S}^{*\text{T}}] \begin{bmatrix} \underline{S}^{\text{T}} \underline{M} \underline{N} \\ \underline{R}^{\text{T}} \underline{N} \underline{M} \end{bmatrix} \quad (38)$$

$$\left[\frac{\partial T^*}{\partial x} \quad \frac{\partial T^*}{\partial y} \right] = [\underline{R}^{*T} \quad \underline{S}^{*T}] \begin{bmatrix} \underline{S}^T \underline{M} \underline{dN} \quad \underline{S}^T \underline{dM} \underline{N} \\ \underline{R}^T \underline{dN} \underline{M} \quad \underline{R}^T \underline{N} \underline{dM} \end{bmatrix} = [\underline{R}^{*T} \underline{S}^{*T}] \underline{F}^T \quad (39)$$

Using these expressions the variational formulation becomes

$$\int_{\Omega} [\underline{R}^{*T} \quad \underline{S}^{*T}] \sum_{j=1}^n \alpha_j \underline{F}^T \underline{D}_j \, d\Omega + \int_{\Omega} [\underline{R}^{*T} \quad \underline{S}^{*T}] \underline{F}^T \underline{E} \begin{bmatrix} \underline{R} \\ \underline{S} \end{bmatrix} \, d\Omega = \sum_{h=1}^m \int_{\Omega} [\underline{R}^{*T} \quad \underline{S}^{*T}] \begin{bmatrix} \underline{S}^T \underline{M} b_h(y) \quad \underline{N} a_h(x) \\ \underline{R}^T \underline{N} a_h(x) \quad \underline{M} b_h(y) \end{bmatrix} \, d\Omega \quad (40)$$

where

$$\underline{F}^T \underline{D}_j = \begin{bmatrix} \underline{dN} \underline{dN}^T \underline{F}_j \underline{S}^T \underline{M} \underline{M}^T \underline{G}_j + \underline{N} \underline{N}^T \underline{F}_j \underline{S}^T \underline{dM} \underline{dM}^T \underline{G}_j \\ \underline{R}^T \underline{dN} \underline{dN}^T \underline{F}_j \underline{M} \underline{M}^T \underline{G}_j + \underline{R}^T \underline{N} \underline{N}^T \underline{F}_j \underline{dM} \underline{dM}^T \underline{G}_j \end{bmatrix} \quad (41)$$

$$\underline{F}^T \underline{E} = \begin{bmatrix} \underline{dN} \underline{dN}^T \underline{S}^T \underline{M} \underline{M}^T \underline{S} & \underline{N} \underline{N}^T \underline{R} \underline{S}^T \underline{dM} \underline{dM}^T \\ \underline{M} \underline{M}^T \underline{S} \underline{R}^T \underline{dN} \underline{dN}^T & \underline{R}^T \underline{N} \underline{N}^T \underline{R} \underline{dM} \underline{dM}^T \end{bmatrix} \quad (42)$$

Again, each integral in Eq. (40) can be separated in one integral in the coordinate x and another one in the coordinate y as illustrated in the following expressions:

$$\begin{aligned} \int_{\Omega} \underline{F}^T \underline{D}_j \, d\Omega &= \begin{bmatrix} \left(\int_{x=-L}^{x=+L} \underline{dN} \underline{dN}^T \, dx \right) \underline{F}_j \underline{S}^T \left(\int_{y=-L}^{y=+L} \underline{M} \underline{M}^T \, dy \right) \underline{G}_j \\ \underline{R}^T \left(\int_{x=-L}^{x=+L} \underline{dN} \underline{dN}^T \, dx \right) \underline{F}_j \left(\int_{y=-L}^{y=+L} \underline{M} \underline{M}^T \, dy \right) \underline{G}_j \end{bmatrix} \\ &+ \begin{bmatrix} \left(\int_{x=-L}^{x=+L} \underline{N} \underline{N}^T \, dx \right) \underline{F}_j \underline{S}^T \left(\int_{y=-L}^{y=+L} \underline{dM} \underline{dM}^T \, dy \right) \underline{G}_j \\ \underline{R}^T \left(\int_{x=-L}^{x=+L} \underline{N} \underline{N}^T \, dx \right) \underline{F}_j \left(\int_{y=-L}^{y=+L} \underline{dM} \underline{dM}^T \, dy \right) \underline{G}_j \end{bmatrix} \end{aligned} \quad (43)$$

the integrals of the four submatrices of $\underline{F}^T \underline{E}$ being given by

$$\int_{\Omega} (\underline{F}^T \underline{E})_{11} \, d\Omega = \left(\int_{x=-L}^{x=+L} \underline{dN} \underline{dN}^T \, dx \right) \underline{S}^T \left(\int_{y=-L}^{y=+L} \underline{M} \underline{M}^T \, dy \right) \underline{S} \quad (44)$$

$$\int_{\Omega} (\underline{F}^T \underline{E})_{12} \, d\Omega = \left(\int_{x=-L}^{x=+L} \underline{N} \underline{N}^T \, dx \right) \underline{R} \underline{S}^T \left(\int_{y=-L}^{y=+L} \underline{dM} \underline{dM}^T \, dy \right) \quad (45)$$

$$\int_{\Omega} (\underline{F}^T \underline{E})_{21} \, d\Omega = \left(\int_{y=-L}^{y=+L} \underline{M} \underline{M}^T \, dy \right) \underline{S} \underline{R}^T \left(\int_{x=-L}^{x=+L} \underline{dN} \underline{dN}^T \, dx \right) \quad (46)$$

$$\int_{\Omega} (\underline{F}^T \underline{E})_{22} \, d\Omega = \underline{R}^T \left(\int_{x=-L}^{x=+L} \underline{N} \underline{N}^T \, dx \right) \underline{R} \left(\int_{y=-L}^{y=+L} \underline{dM} \underline{dM}^T \, dy \right) \quad (47)$$

and the last term in Eq. (40)

$$\sum_{h=1}^m \begin{bmatrix} \underline{S}^T \left(\int_{y=-L}^{y=+L} \underline{M} b_h(y) \, dy \right) \left(\int_{x=-L}^{x=+L} \underline{N} a_h(x) \, dx \right) \\ \underline{R}^T \left(\int_{x=-L}^{x=+L} \underline{N} a_h(x) \, dx \right) \left(\int_{y=-L}^{y=+L} \underline{M} b_h(y) \, dy \right) \end{bmatrix} \quad (48)$$

Eq. (40) can then be expressed in the compact form

$$\underline{V}_1(\underline{R}, \underline{S}) + \underline{K}(\underline{R}, \underline{S}) \begin{bmatrix} \underline{R} \\ \underline{S} \end{bmatrix} = \underline{V}_2(\underline{R}, \underline{S}) \quad (49)$$

After imposing the vanishing of both functions R and S on their respective domain boundaries, the non-linear system (49) can be solved by applying a Newton strategy, the size of the linearized systems being $p + q$. As the solution of such problem is not unique, we can impose for example that the norm of S is one by introducing the associated Lagrange multiplier. To accelerate the convergence the first approximated field T is constructed by using the functions a_h and b_h .

When the solution R and S is obtained (after convergence of the Newton procedure) the new approximation functions can be determined by normalizing

$$\begin{cases} \underline{F}_{n+1} = \frac{R}{\sqrt{R^T \left(\int_{x=-L}^{x=L} N N^T dx \right) R}} \\ \underline{G}_{n+1} = \frac{S}{\sqrt{S^T \left(\int_{y=-L}^{y=L} M M^T dy \right) S}} \end{cases} \quad (50)$$

If the normality of S has been used, then the second expression of Eq. (50) reduces to $\underline{G}_{n+1} = \underline{S}$.

5. Numerical results

In this section we present different results that are compared with the exact or the reference solutions, the last ones obtained from the finite element method using very fine meshes.

- Fig. 1 illustrates the solution of the 2D Poisson's problem defined in $\Omega =]-1, 1[\times]-1, 1[$, with $f(x, y) = \cos(2\pi x) \sin(2\pi y)$, assuming that the solution vanishes at the domain boundary. The solution is reached in only one iteration, and it consist of a function in the x -coordinate and the other one in the y -coordinate, both represented in Fig. 1, with $\alpha = 0.2243$.
Fig. 2 depicts similar results for $f(x, y) = x^2 - y^2$. Convergence is reached in two iterations. We obtain $\alpha^T = (0.373, 0.378)$.
The consideration of $f(x, y) = 2x^2 + x + y^2 - 0.2y + 3xy$ increases the number of iterations, and then the number of functions needed to accurately describe the solution (depicted in Fig. 3). In this case five approximation functions in each direction are needed. The resulting alpha vector is $\alpha^T = (3.96, 1.30, 0.17, 0.053, 0.019)$.
- The second test concerns a 3D Poisson's problem related to the source term $f(x, y, z) = 2(1 - y^4)(1 - z^6) + (1 - x^2)12y^2(1 - z^6) + (1 - x^2)(1 - y^4)30z^4$. The exact solution of this problems consists of: $T = (1 - x^2)(1 - y^4)(1 - z^6)$. The proposed algorithm converges in one iteration and gives a second degree polynomial in the x -coordinate, a fourth degree in the y -coordinate and a 6th degree in the z -coordinate as depicted in Fig. 4. We obtain $\alpha = 133.8$.
- Fig. 5 depicts the solution of the 6D Poisson's problem related to $f(x_1, \dots, x_6) = \prod_{k=1}^{k=6} \sin(k\pi x_k) + \prod_{k=1}^{k=6} \sin((7 - k)\pi x_k)$. The solution is obtained in two iterations, being $\alpha^T = (27.9, 27.9)$.
- Finally, Fig. 6 depicts the solution of the 10D Poisson's problem associated with $f(x_1, \dots, x_{10}) = \sum_{k=1}^{k=10} Re(x_k^{k/2})$. The solution is obtained in four iterations, being $\alpha^T = 10^5 \times (3.86, 0.5, 0.15, 0.2)$.

In all these examples we have considered uniform 1D grids with the same number of nodes in all space dimensions. We have analyzed the order of convergence, i.e. the evolution of the error norm with the mesh size, which seems to be of third order (one order higher than the second order expected for linear finite elements).

In the case of 2D or 3D models, the advantages of the approach based on a separated representation because in these cases the solution of the non-linear problems related to the approximation basis enrichment is more expensive than the solution of the linear system related to the application of the standard finite element method. The separated representation becomes an appealing strategy when the dimension of the models increases, because in the separated representation the problem size increases linearly instead of the exponential increase of finite elements based discretizations. Thus, finite element models become quickly intractable when the space dimension increases. For this reason, we have only compared in this section the solutions of some lower-dimensional models in terms of accuracy but not in terms of the computing time.

All simulations stop when the residual Re is small enough according to Eq. (13). The computed solution and the exact or reference ones (these last computed using fine enough finite element meshes) are very close. The solution accuracy increases by decreasing the value of the stop criterion. In our simulations the stop criterion is set to $Re < \epsilon = 10^{-6}$.

6. Application to multi-bead-spring models

6.1. Governing equations

The technique described below can be applied for simulating the multi-bead-spring models of polymer chains. The MBS chain consists of $S + 1$ beads connected by S springs. The bead serves as an interaction point with the solvent and the spring contains the local stiffness information depending on local stretching (see [4] for more details).

The dynamics of the chain are governed by viscous drag, Brownian and connector forces. If we denote by \dot{r}_k the velocity of bead k and by \dot{q}_k the velocity of the spring connector q_k , we have

$$\dot{q}_k = \dot{r}_{k+1} - \dot{r}_k \quad \forall k = 1, \dots, S \quad (51)$$

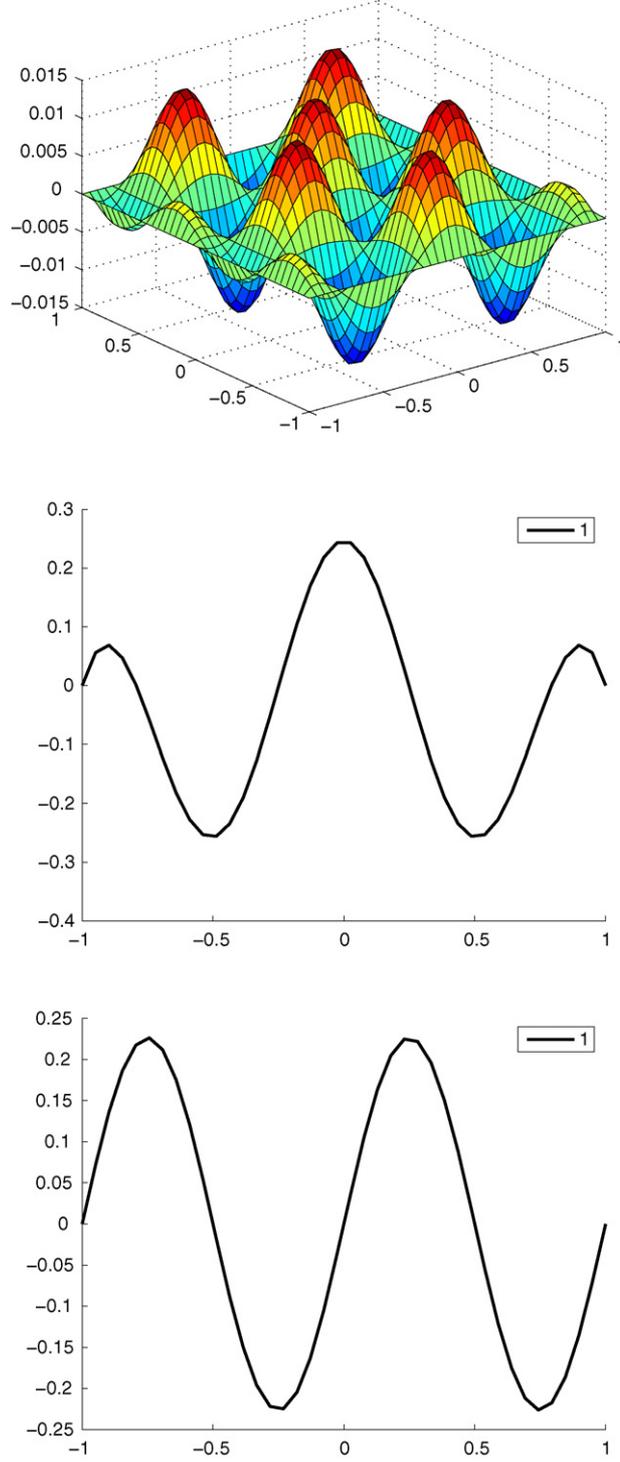


Fig. 1. 2D Poisson problem related to $f(x, y) = \cos(2\pi x) \sin(2\pi y)$: (top) computed solution; (middle) x -function; (bottom) y -function.

The dynamics of each bead can be written as

$$\underbrace{-\zeta(\dot{r}_k - v_0 - \text{Grad}v r_k)}_{\text{viscous drag}} - \underbrace{k_b T \frac{\partial}{\partial r_k} \ln(\psi)}_{\text{Brownian effects}} + \underbrace{F_k^c - F_{k-1}^c}_{\text{interaction forces}} = 0 \tag{52}$$

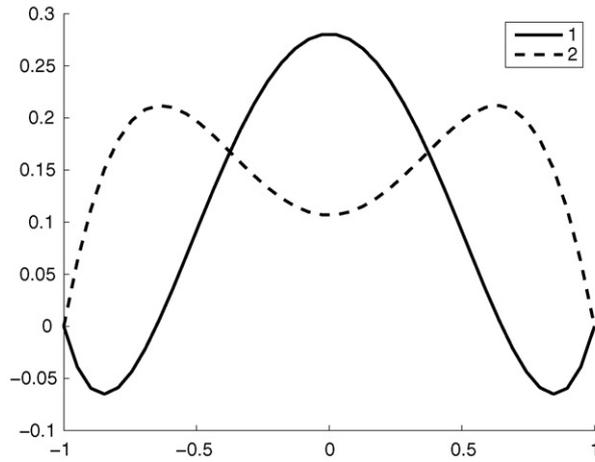
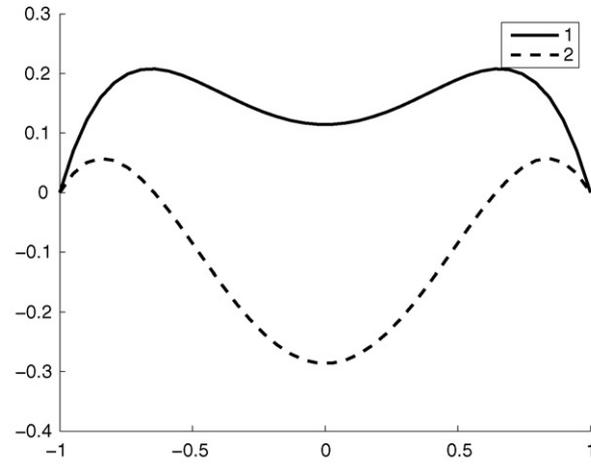
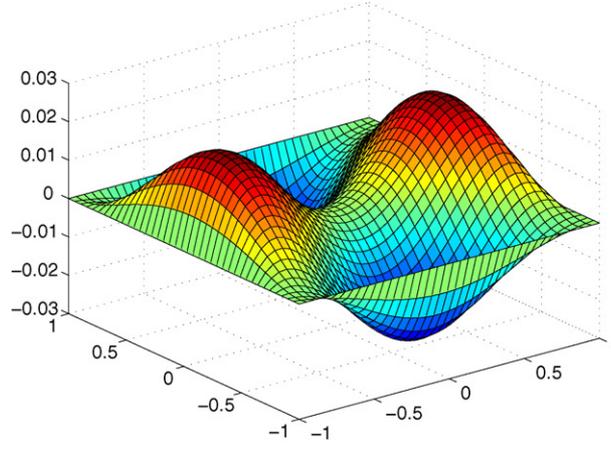


Fig. 2. 2D Poisson problem related to $f(x, y) = x^2 - y^2$: (top) computed solution; (middle) x -functions; (bottom) y -functions.

where ζ is the drag coefficient, v is the velocity field, v_0 is an average velocity, k_b is the Boltzmann constant, T is the absolute temperature and ψ is the probability distribution function. From Eqs. (51) and (52) we obtain

$$\dot{q}_k = \text{Grad} v q_j - \frac{1}{\zeta} \sum_{l=1}^S A_{kl} \left(k_b T \frac{\partial}{\partial q_l} \ln(\psi) + F_l^c \right) \quad (53)$$

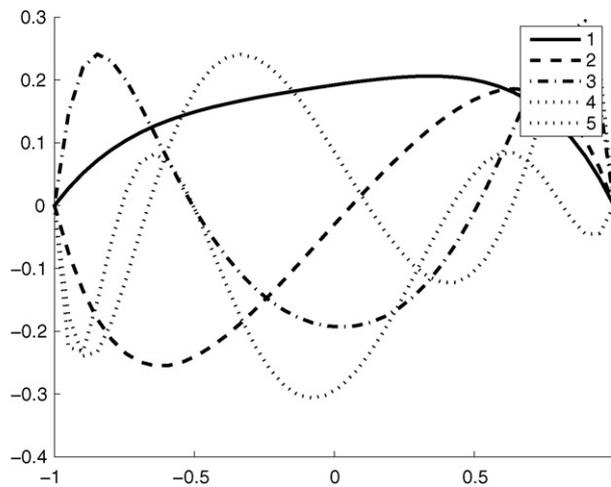
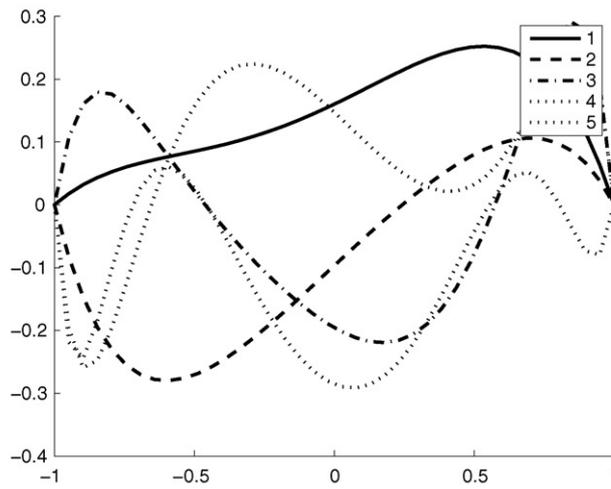
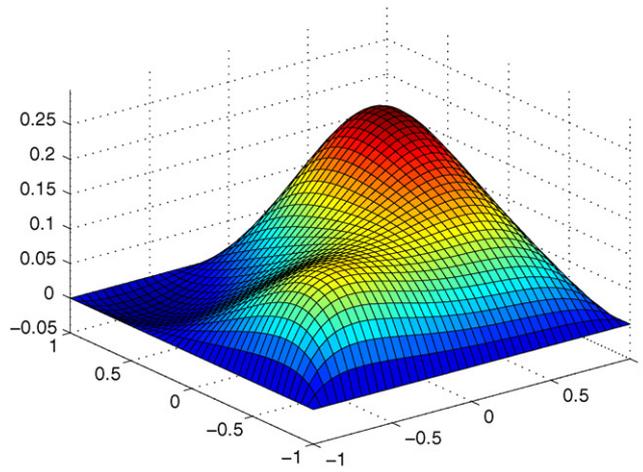


Fig. 3. 2D Poisson problem related to $f(x, y) = 2x^2 + x + y^2 - 0.2y + 3xy$: (top) computed solution; (middle) x -functions; (bottom) y -functions.

where A_{kl} is the Rouse matrix defined by

$$A_{kl} = \begin{cases} 2 & \text{if } k = l \\ -1 & \text{if } k = l \pm 1 \\ 0 & \text{otherwise} \end{cases} .$$

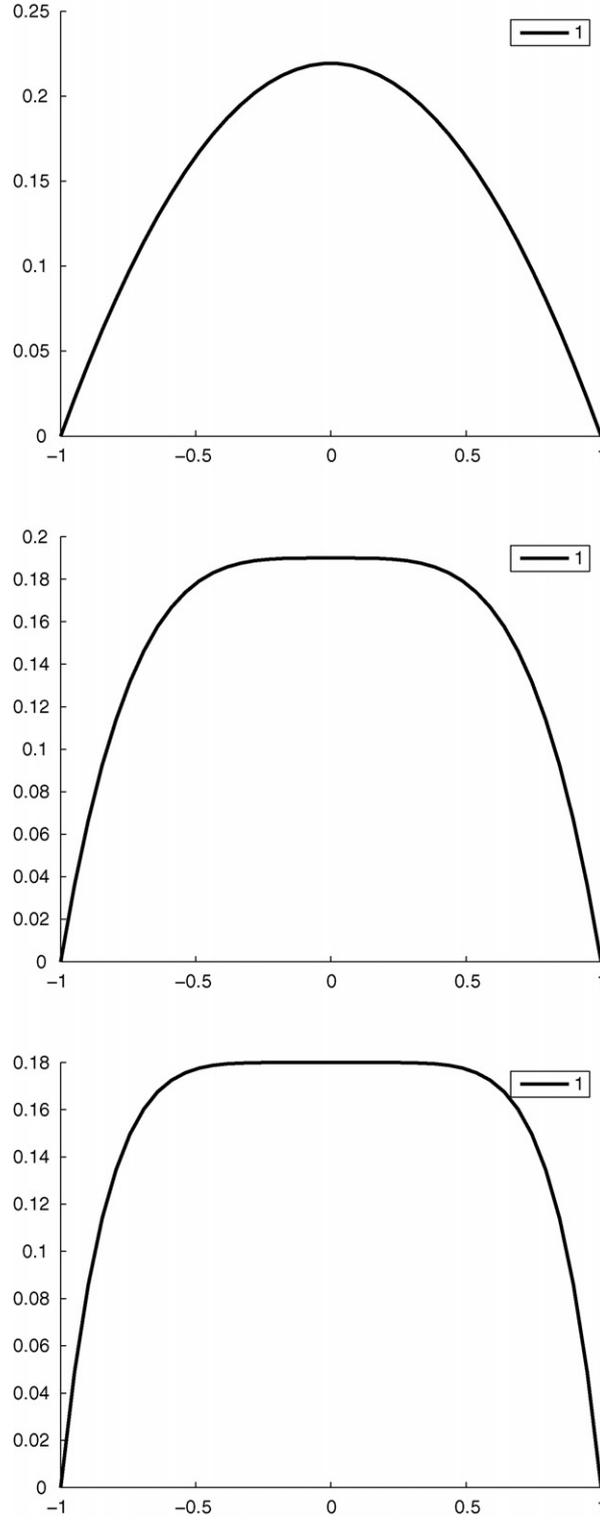


Fig. 4. 3D Poisson problem related to $f(x, y, z) = 2(1 - y^4)(1 - z^6) + (1 - x^2)12y^2(1 - z^6) + (1 - x^2)(1 - y^4)30z^4$: (top) x -function; (middle) y -function; (bottom) z -function.

In the Rouse model the connector force F^c is a linear function of the connector orientation vector. We shall use finitely extensible non-linear springs, with a dimensionless connector force given by

$$F^c(q_k) = \frac{1}{1 - q_k^2/b} q_k \quad (54)$$

where \sqrt{b} is the maximum length of each spring connector of the chain.

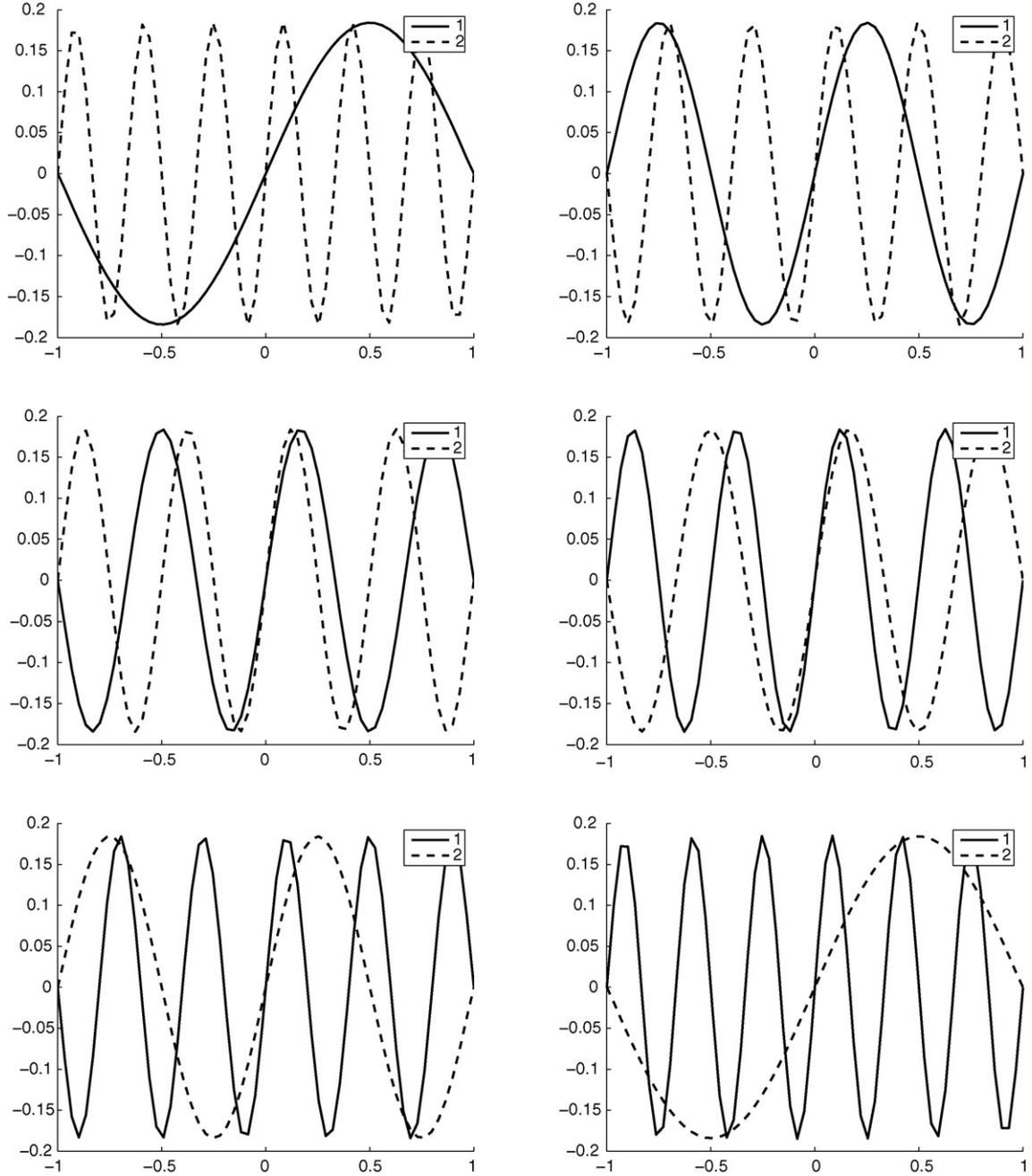


Fig. 5. 6D Poisson problem: (top-left) x_1 -functions; (top-right) x_2 -functions; (middle-left) x_3 -functions; (middle-right) x_4 -functions; (bottom-left) x_5 -functions; (bottom-right) x_6 -functions.

Introducing Eq. (53) in the equation governing the evolution of the distribution function, and considering homogeneous flows

$$\frac{\partial \psi(q_1, \dots, q_S, t)}{\partial t} = - \sum_{k=1}^S \left(\frac{\partial}{\partial q_k} (\dot{q}_k \psi(q_1, \dots, q_S, t)) \right) \quad (55)$$

we obtain

$$\frac{\partial \psi}{\partial t} = - \sum_{k=1}^S \left(\frac{\partial}{\partial q_k} \left((\text{Grad}_v q_k - \frac{1}{\zeta} \sum_{l=1}^S A_{kl} F_l^c) \psi \right) \right) + \frac{k_b T}{\zeta} \sum_{k=1}^S \sum_{l=1}^S A_{kl} \frac{\partial^2 \psi}{\partial q_k \partial q_l} \quad (56)$$

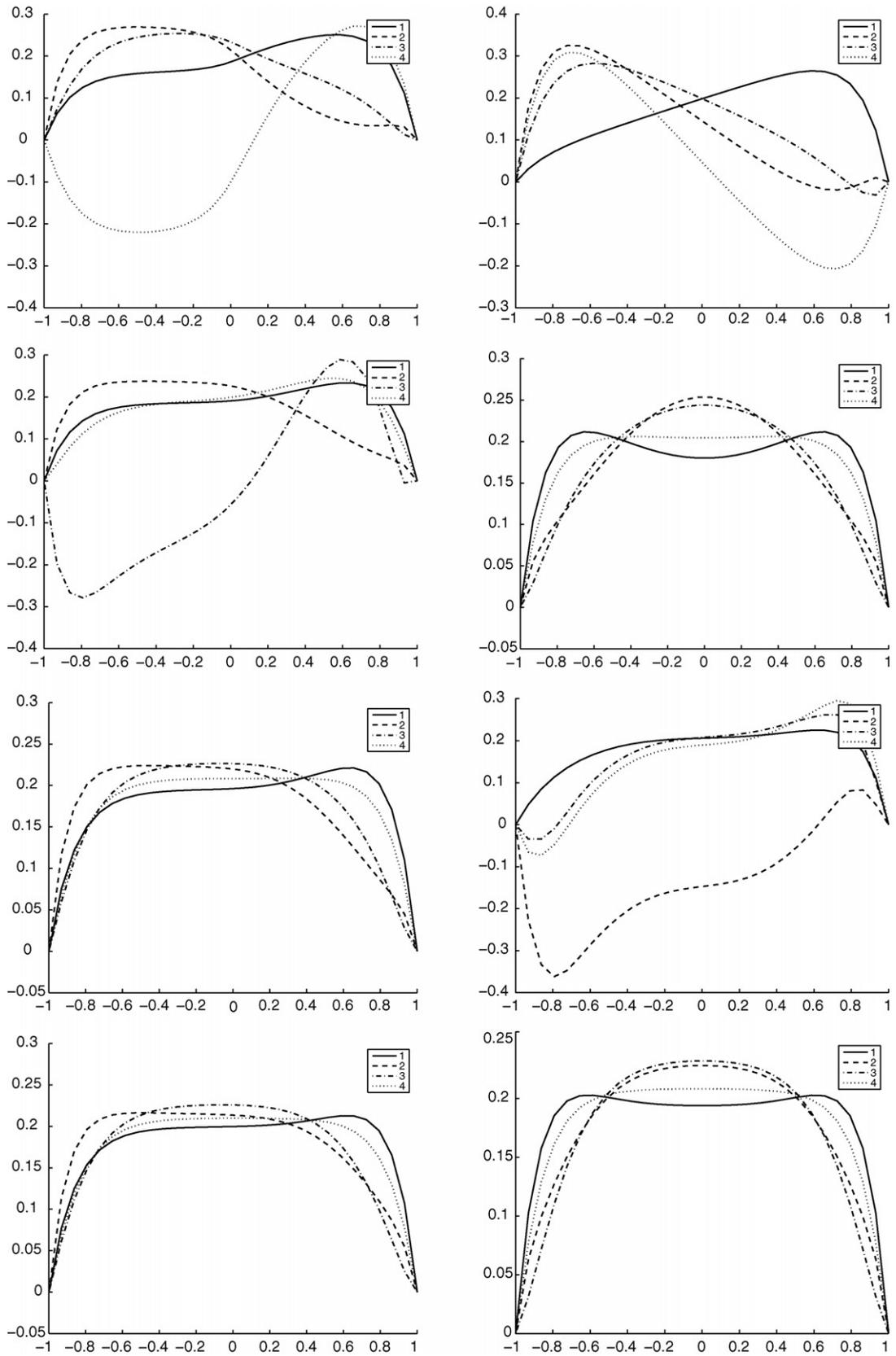


Fig. 6. 10D Poisson problem: (top-left) x_1 -functions; (top-right) x_2 -functions, . . . ; (bottom-left) x_9 -functions; (bottom-right) x_{10} -functions.

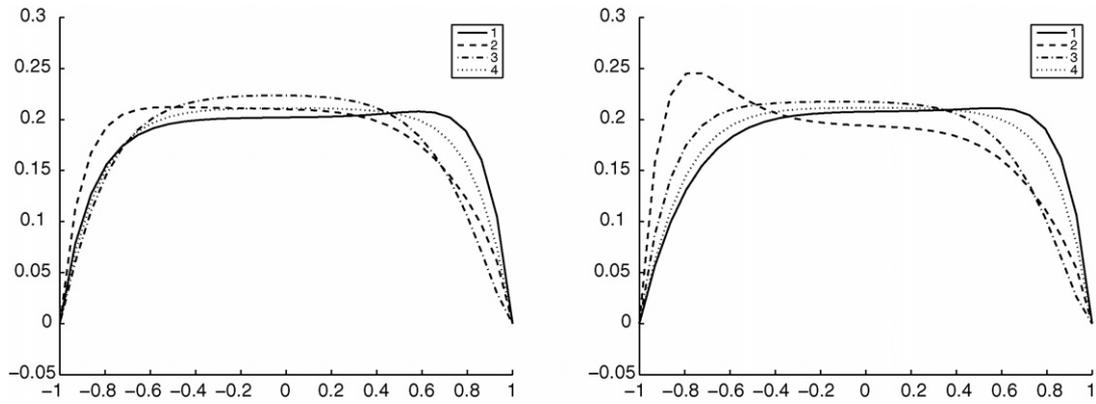


Fig. 6. (Continued).

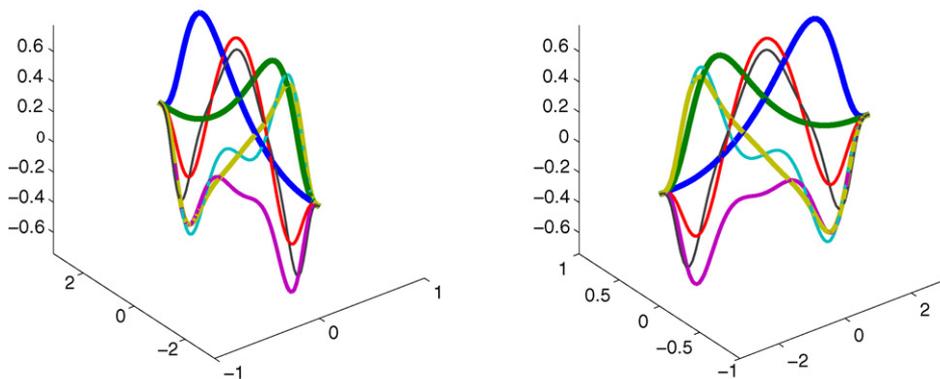
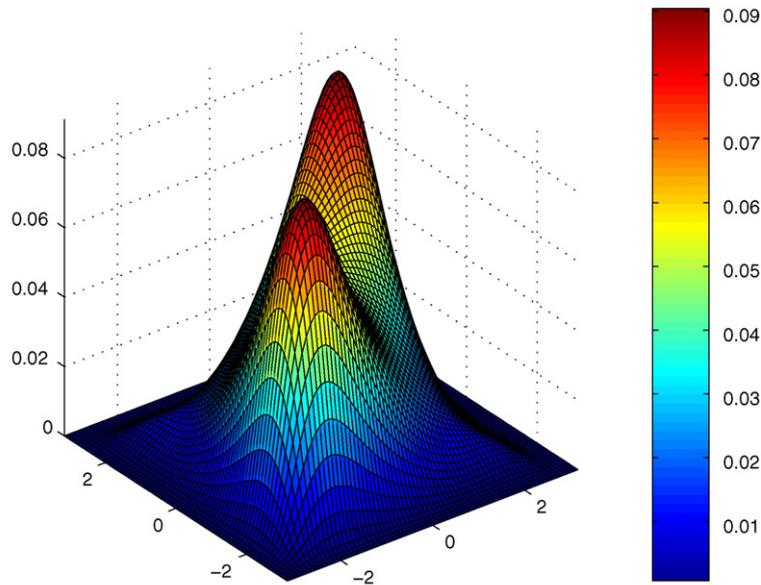


Fig. 7. MBS model consisting of $S = 2$ springs: (top) computed solution; (left) $F_{2j}(q_2)$ functions; (right) $F_{1j}(q_1)$ functions.

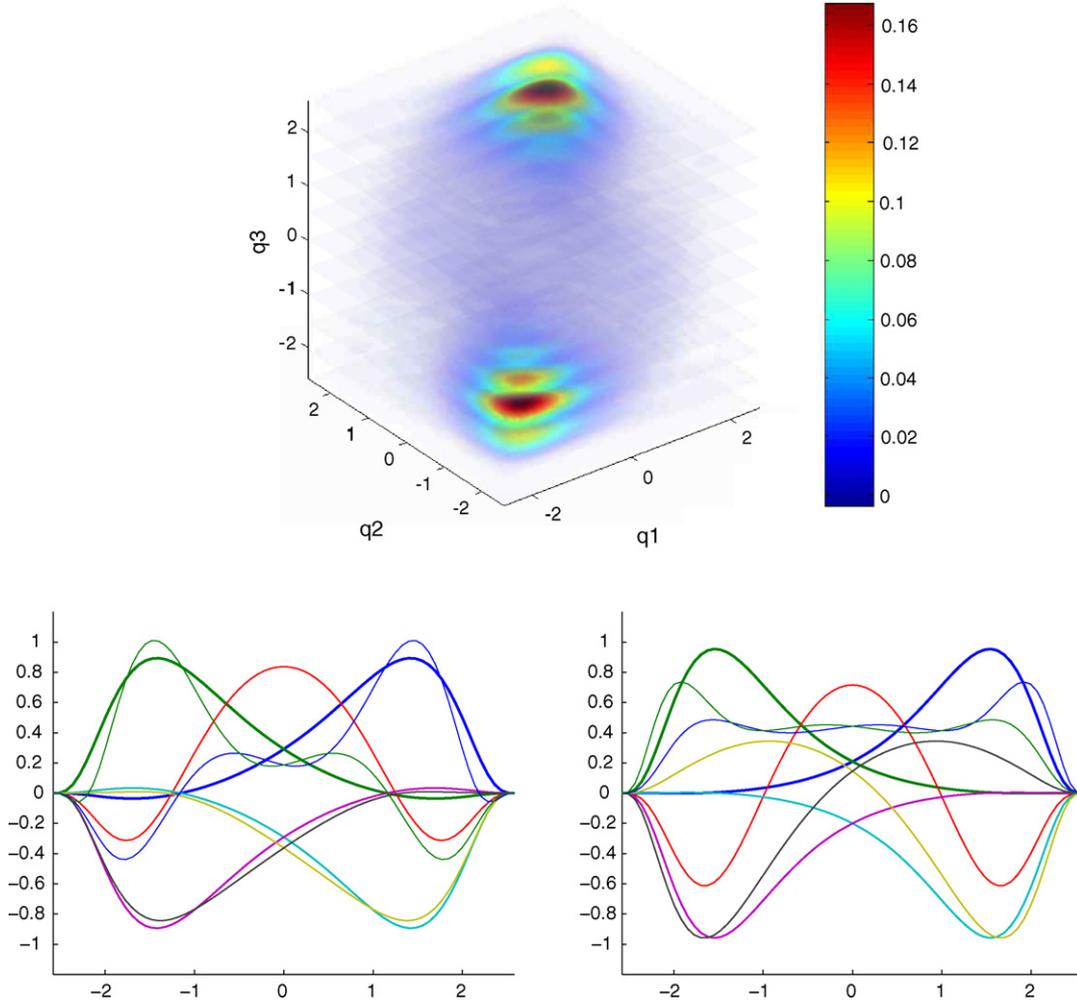


Fig. 8. MBS model consisting of $S = 3$ springs: (top) computed solution; (left) $F_{1j}(q_1)$ functions; (right) $F_{2j}(q_2)$ functions.

6.2. Numerical examples involving one-dimensional physical spaces

In the following examples, we consider a physical space of dimension one, which implies that the dimension of the conformation space coincides with the number of springs in the MBS chain. Moreover, the only component of the velocity field (denoted by u) depends on the single physical coordinate x . The conformation coordinates represent in this case the springs extension. The parameters involved in the simulations are $W_e = \text{Grad}v = du/dx = \sqrt{S}/4$ and $b = 20/S$.

Fig. 7 depicts the solution computed in the case of $S = 2$, which involves seven approximation functions in each coordinate q_1 and q_2 . Vector alpha results $\underline{\alpha}^T = (0.135, 0.135, 0.023, 0.013, 0.018, 0.0032, 0.0007)$. The computed solution is in perfect agreement with the one obtained using a 2D-FEM solution with a mesh fine enough to use it as reference solution. In the same figure the seven functions $F_{1j}(q_1)$ and $F_{2j}(q_2)$ (using the notation introduced in Section 3) are also depicted.

Fig. 8 shows the computed solution for a MBS consisting of three springs ($S = 3$). The computed approximation functions $F_{1j}(q_1)$ and $F_{3j}(q_3)$ coincide because of the symmetry of the chain. It can be noticed in the 3D representation that the central spring is more stretched than the ones located at the chain ends.

The solution for $S = 9$ (chain consisting of nine springs) is depicted in Fig. 9. It can be noticed that the approximation functions related to the central spring q_5 are sharper than the ones related to the springs located at the chain ends q_1 and q_9 , in agreement with the fact that the central springs are more stretched. These approximation functions are described from a one-dimensional FE approximation using 80 nodes. To compute this solution we must solve some seven non-linear problems (for determining the optimal approximation functions), each one involving 80×9 degrees of freedom (80 nodes describing each approximation function defined in each one of the nine dimensions of the conformation space).

In general, for a chain model consisting of S springs defined in a conformation space of dimension S (in the 1D physical space), where P nodes are considered to define the one-dimensional approximation functions in each conformation coordinate, we need to

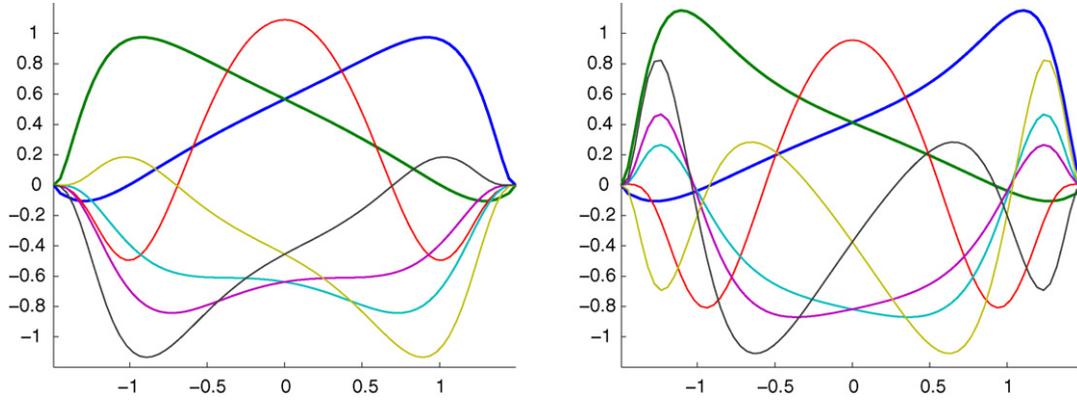


Fig. 9. MBS model consisting of $S = 9$ springs: (left) $F_{1j}(q_1)$ functions; (right) $F_{5j}(q_5)$ functions.

solve some non-linear problems of size $P \times S$, the solution accuracy being equivalent to the one obtained from the application of the FEM with P^S degrees of freedom. Thus, the quality of the solution depicted in Fig. 9 is of the same order than using $80^9 \sim 10^{16}$ degrees of freedom in the the FEM framework.

6.3. Numerical examples involving higher-dimensional physical spaces

Finally, even if the consideration of higher dimensional physical spaces is out the scope of the present paper, we would like to illustrate the applicability of the proposed technique for chain models with both orientation and extension conformation degrees of freedom. In some special cases (such as linear spring behavior or the one proposed in [20]), it is possible to decouple all the dimensions, writing the distribution function as the product of one-dimensional functions, as previously described. However, for

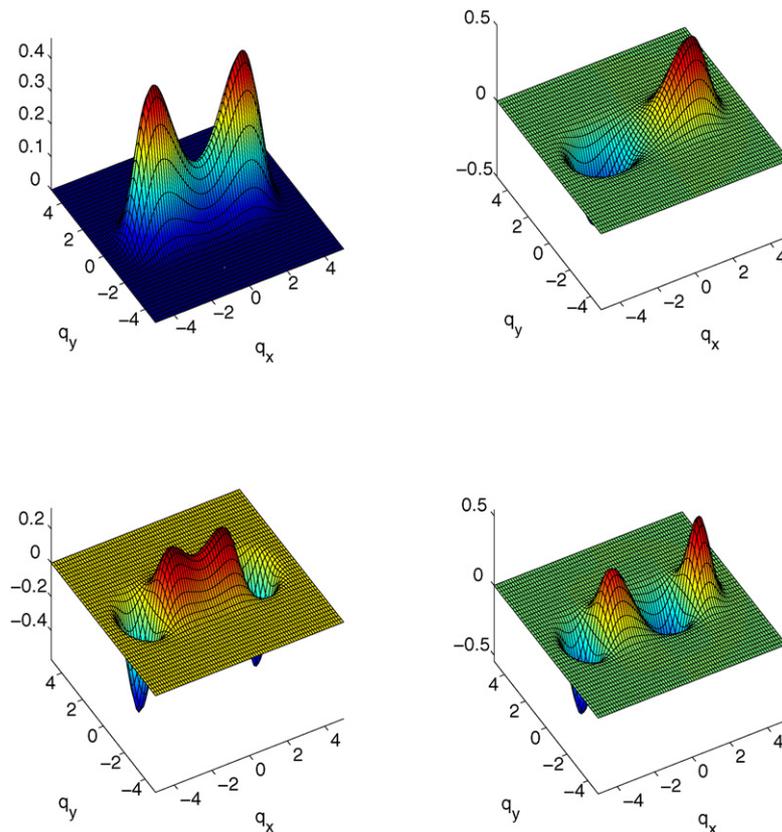


Fig. 10. MBS model consisting of two FENE springs in a two-dimensional physical space: reduced approximation functions in an elongational flow.

more realistic molecular models (FENE chains) two difficulties are found: (i) the molecular conformation function is defined over a bounded domain but no longer a hyper-cube; (ii) in the case of MBS models the advection terms related to each spring cannot be separated as a product of terms involving each of the physical space directions. The general way to deal with this problem consists in building up the solution as a product of functions each one defined in each 2D or 3D domain related to each spring connector

$$\psi(\underline{q}_1, \dots, \underline{q}_S) = \sum_{j=1}^{\infty} \alpha_j F_{1j}(\underline{q}_1) \cdots F_{Sj}(\underline{q}_S) \tag{57}$$

where the approximation of each function F_{kj} is built in a space of the same dimension as that of the physical space (a bounded circle in 2D or a bounded sphere in 3D).

In the numerical examples that follow, we first consider a FENE chain model defined in a 2D physical space, which consists of two FENE spring connectors ($S = 2$) characterized by $b = 25$. The distribution function is then written in the form

$$\psi(\underline{q}_1, \underline{q}_2) = \sum_{j=1}^{\infty} \alpha_j F_{1j}(\underline{q}_1) F_{2j}(\underline{q}_2) \tag{58}$$

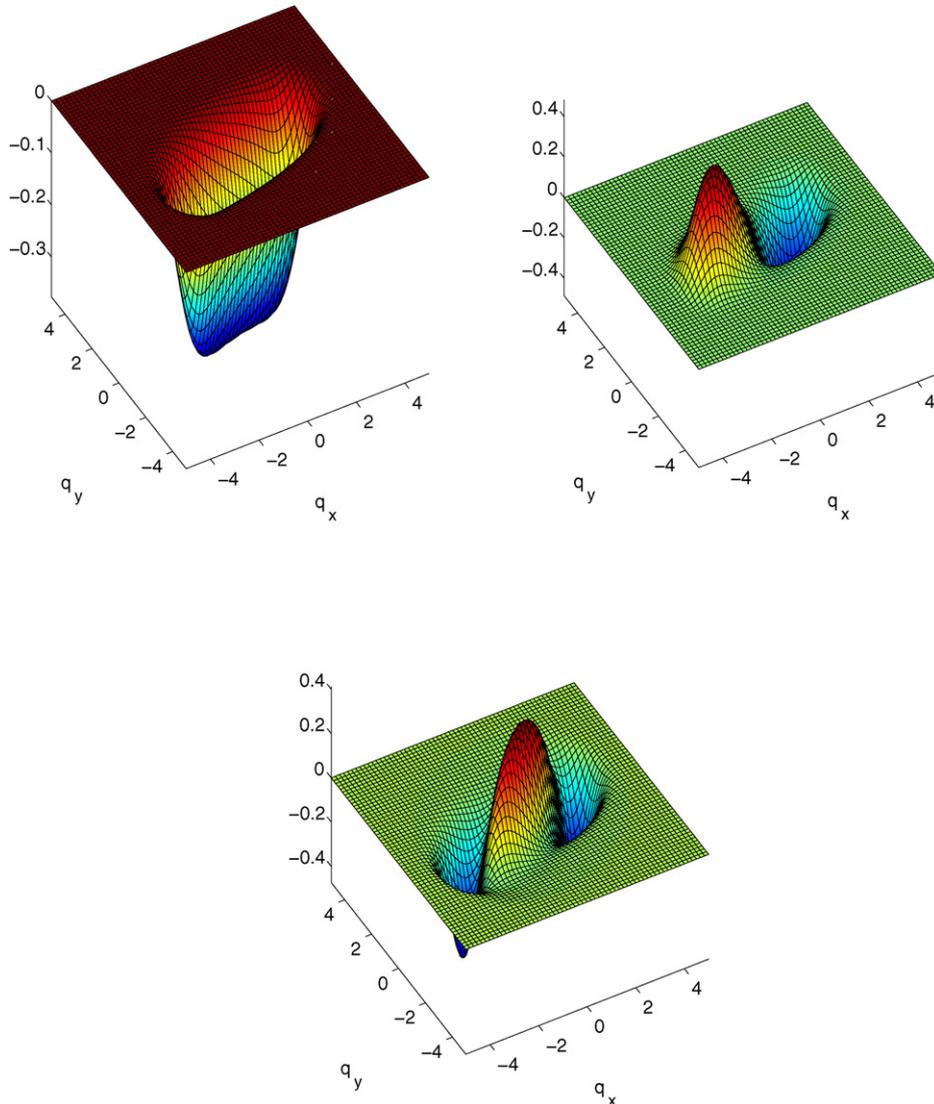


Fig. 11. MBS model consisting of two FENE springs in a two-dimensional physical space: Reduced approximation functions in a simple shear flow.

This defines a problem in a configuration space of dimension 4. The Fokker–Planck equation has been solved without considering any simplifying hypothesis, despite the expected equality $F_{1j}(q_1) = F_{2j}(q_2)$. Of course, the computed solution verifies this symmetry condition, which enables us to represent only one of those functions, $F_{1j}(q_1)$ for example.

The first flow considered was an elongation flow defined by $We = 0.25\sqrt{2}$, that is a flow characterized by the following velocity gradient

$$\text{Grad}\underline{v} = \begin{pmatrix} 0.25\sqrt{2} & 0 \\ 0 & -0.25\sqrt{2} \end{pmatrix} \quad (59)$$

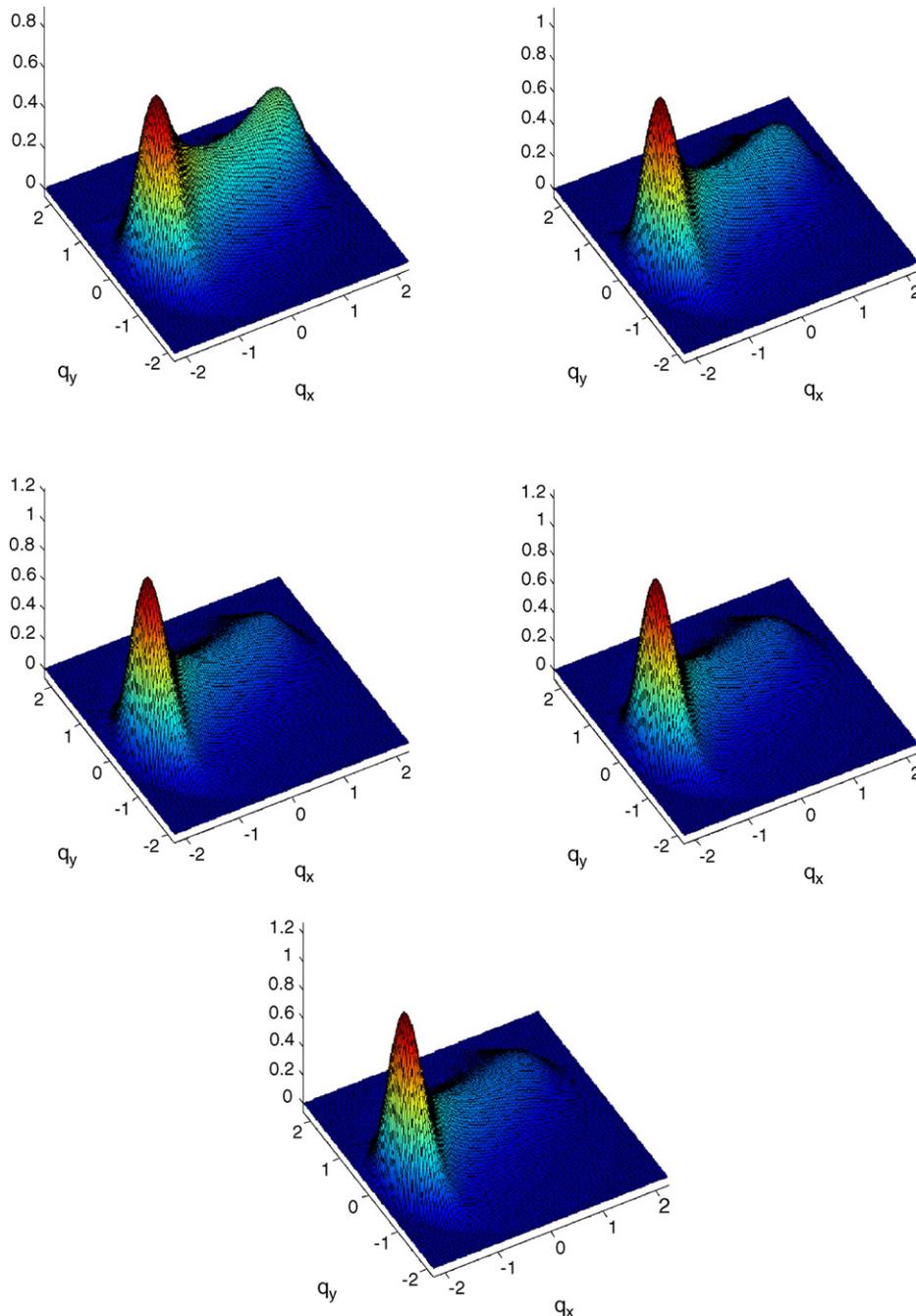


Fig. 12. MBS model consisting of 10 FENE springs in a two-dimensional physical space. First reduced approximation functions: (top-left) first spring; (top-right) second spring; (middle-left) third spring; (middle-right) fourth spring and (bottom) central spring.

Only four approximation functions are required to define accurately the distribution function which can be then written as

$$\psi(\underline{q}_1, \underline{q}_2) = \sum_{j=1}^4 \alpha_j F_{1j}(\underline{q}_1) F_{2j}(\underline{q}_2) \quad (60)$$

where $\alpha_1 = 0.058$, $\alpha_2 = 0.046$, $\alpha_3 = 0.014$ and $\alpha_4 = 0.006$. Fig. 10 depicts the four approximation functions $F_{1j}(\underline{q}_1)$.

In this solution, a mesh consisting of 1600 nodes was used to approximate the functions $F_{1j}(\underline{q}_1)$ and $F_{2j}(\underline{q}_2)$. Thus, the total number of degrees of freedom was $2 \times 1600 = 3200$. An equivalent solution using the FEM would have required 1600^2 dof's to discretize the four-dimensional space.

If we consider the shear flow defined for $We = \sqrt{2}$ by

$$\text{Grad} \underline{v} = \begin{pmatrix} 0 & \sqrt{2} \\ 0 & 0 \end{pmatrix} \quad (61)$$

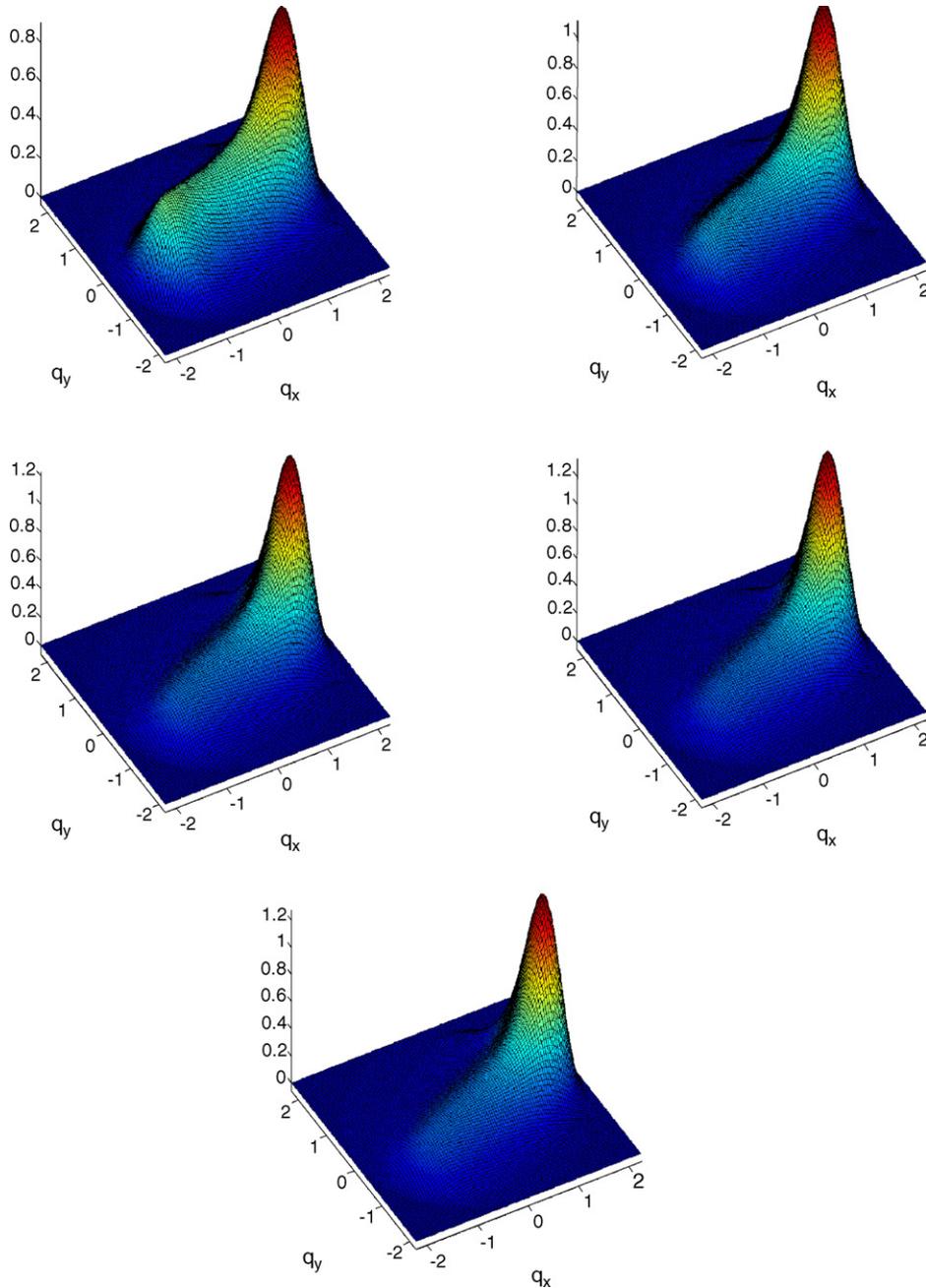


Fig. 13. MBS model consisting of 10 FENE springs in a two-dimensional physical space. Second reduced approximation functions: (top-left) first spring; (top-right) second spring; (middle-left) third spring; (middle-right) fourth spring and (bottom) central spring.

all the other simulation parameters being the same, only three approximation functions are required to describe the distribution function

$$\psi(\underline{q}_1, \underline{q}_2) = \sum_{j=1}^3 \alpha_j F_{1j}(\underline{q}_1) F_{2j}(\underline{q}_2) \quad (62)$$

where $\alpha_1 = 0.058$, $\alpha_2 = 0.026$ and $\alpha_3 = 0.001$. Fig. 11 depicts the three approximation functions $F_{1j}(\underline{q}_1)$.

Finally, to prove the potentiality of the proposed strategy for computing the steady state of multidimensional kinetic theory models, we are considering a FENE chain model defined in a 2D physical space and consisting of 10 FENE spring connectors, $S = 10$, characterized by $b = 25$. The distribution function, now defined in a 20-dimensional space, can be written in the form

$$\psi(\underline{q}_1, \dots, \underline{q}_{10}) = \sum_{j=1}^{\infty} \alpha_j F_{1j}(\underline{q}_1) \cdots F_{10j}(\underline{q}_{10}) \quad (63)$$

After the simulation, we have verified the symmetry conditions $F_{kj}(\underline{q}_k) = F_{10-k+1j}(\underline{q}_{10-k+1})$, $\forall k \in [1, 5]$. In this solution, we have used a mesh consisting of 10^4 nodes to approximate each function $F_{kj}(\underline{q}_k)$.

The previous shear flow was considered ($We = \sqrt{2}$) and three approximation functions were found sufficient to accurately describe the steady state of the distribution function. We depict in Fig. 12 the first function for the five first FENE spring connectors, i.e. $F_{k1}(\underline{q}_k)$, $\forall k \in [1, 5]$ (the other five being the same). The second approximation functions $F_{k2}(\underline{q}_k)$, $\forall k \in [1, 5]$, are depicted in Fig. 13. In Figs. 12 and 13, we notice that the results are sharper at the center of the chain, indicating that the central springs are more stretched than the ones located near the chain ends.

This simulation implied $10 \times 10^4 = 10^5$ dof's (10 spring connectors whose approximation functions were defined using a mesh of 10^4 nodes). A standard finite element solution would have required of the order of $(10^4)^{10} = 10^{40}$ dof's (nodes) for computing an equivalent solution.

7. Conclusions

In this work, we have proposed the use of a separated representation of the distribution function that allows to define a tensor product approximation basis as well as to decouple the integral in the whole domain as a product of integrals on each dimension. Moreover, the approximation functions in each direction will be computed during the simulation in order to keep a reduced number of degrees of freedom. First, this technique has been illustrated and tested in some multidimensional Poisson problems defined over a hyper-cube $]-L, +L[^N$ with homogeneous boundary conditions. This work opens new perspectives for the simulation of multidimensional kinetic theory models, for which we have presented encouraging early results. In this case, the resulting Fokker–Planck equation involves an advection term (which can also be written as the product of functions, each one involving a different conformation coordinate) that must be stabilized using an appropriate technique. In the last example, we have illustrated the applicability of this approach to chain models involving both orientation and extension conformational degrees of freedom.

Consideration of complex flow simulations is the subject of future work.

Appendix A. Multidimensional matrix form

We summarize in this appendix the matrix form in the case of multidimensional Poisson problems.

- *Projection stage*

In this case Eqs. (34) and (35) result

$$\underline{K}_{ij} = \sum_{k=1}^N \underline{F}_{ki}^T \left(\int_{-L}^{+L} d\underline{N}_k d\underline{N}_k^T dx_k \right) \underline{F}_{kj} \times \prod_{\substack{l=1 \\ l \neq k}}^N \underline{F}_{li}^T \left(\int_{-L}^{+L} \underline{N}_l \underline{N}_l^T dx_l \right) \underline{F}_{lj} \quad (64)$$

$$\underline{V}_i = \sum_{h=1}^m \prod_{k=1}^N \underline{F}_{ki}^T \left(\int_{-L}^{+L} \underline{N}_k a_{kh}(x_k) dx_k \right) \quad (65)$$

• *Basis enrichment*

We assume that matrix $\underline{\underline{K}}$ and vectors \underline{V}_1 and \underline{V}_2 can be written as

$$\underline{\underline{K}} = \begin{bmatrix} \underline{\underline{\chi}}_{11} & \underline{\underline{\chi}}_{12} & \cdots \\ \underline{\underline{\chi}}_{21} & \underline{\underline{\chi}}_{22} & \\ & & \ddots \\ & & & \underline{\underline{\chi}}_{NN} \end{bmatrix} \tag{66}$$

$$\underline{V}_1 = \begin{bmatrix} \underline{\lambda}_1 \\ \vdots \\ \underline{\lambda}_N \end{bmatrix} \tag{67}$$

$$\underline{V}_2 = \begin{bmatrix} \underline{\gamma}_1 \\ \vdots \\ \underline{\gamma}_N \end{bmatrix} \tag{68}$$

In those expressions $\underline{\underline{\chi}}_{k_1 k_2}$ is a matrix with dimensions (p_{k_1}, p_{k_2}) and $\underline{\lambda}_{k_1}$ and $\underline{\gamma}_{k_1}$ are vectors of dimension $(p_{k_1}, 1)$. Thus, is easy to verify that

$$\begin{cases} \underline{\underline{\chi}}_{k_1 k_1} = \left(\int_{-L}^{+L} \underline{dN}_{k_1} \underline{dN}_{k_1}^T dx_{k_1} \right) \prod_{\substack{l=1 \\ l \neq k_1}}^N \underline{R}_l^T \left(\int_{-L}^{+L} \underline{N}_l \underline{N}_l^T dx_l \right) \underline{R}_l \\ \underline{\underline{\chi}}_{k_1 k_2} = \left(\int_{-L}^{+L} \underline{N}_{k_1} \underline{N}_{k_1}^T dx_{k_1} \right) \underline{R}_{k_1} \underline{R}_{k_2}^T \left(\int_{-L}^{+L} \underline{dN}_{k_2} \underline{dN}_{k_2}^T dx_{k_2} \right) \times \prod_{\substack{l=1 \\ l \neq k_1 \\ l \neq k_2}}^N \underline{R}_l^T \left(\int_{-L}^{+L} \underline{N}_l \underline{N}_l^T dx_l \right) \underline{R}_l \\ k_1 \neq k_2 \end{cases} \tag{69}$$

$$\begin{aligned} \underline{\lambda}_{k_1} = \sum_{j=1}^n \alpha_j & \left(\left(\int_{-L}^{+L} \underline{dN}_{k_1} \underline{dN}_{k_1}^T dx_{k_1} \right) \underline{E}_{k_1 j} \times \prod_{\substack{l=1 \\ l \neq k_1}}^N \underline{R}_l^T \left(\int_{-L}^{+L} \underline{N}_l \underline{N}_l^T dx_l \right) \underline{E}_{lj} \right. \\ & \left. + \sum_{\substack{k_2=1 \\ k_2 \neq k_1}}^N \left(\left(\int_{-L}^{+L} \underline{N}_{k_1} \underline{N}_{k_1}^T dx_{k_1} \right) \underline{E}_{k_1 j} \times \underline{R}_{k_2}^T \left(\int_{-L}^{+L} \underline{dN}_{k_2} \underline{dN}_{k_2}^T dx_{k_2} \right) \underline{E}_{k_2 j} \times \prod_{\substack{l=1 \\ l \neq k_1 \\ l \neq k_2}}^N \underline{R}_l^T \left(\int_{-L}^{+L} \underline{N}_l \underline{N}_l^T dx_l \right) \underline{E}_{lj} \right) \right) \end{aligned} \tag{70}$$

$$\underline{\gamma}_{k_1} = \sum_{h=1}^m \left(\left(\int_{-L}^{+L} \underline{N}_{k_1} a_{k_1 h}(x_{k_1}) dx_{k_1} \right) \times \prod_{\substack{l=1 \\ l \neq k_1}}^N \underline{R}_l^T \left(\int_{-L}^{+L} \underline{N}_l a_{lh}(x_l) dx_l \right) \right) \tag{71}$$

The enrichment functions are then computed by

$$\underline{F}_{k(n+1)} = \frac{\underline{R}_k}{\sqrt{\underline{R}_k^T \left(\int_{-L}^{+L} \underline{N}_k \underline{N}_k^T dx_k \right) \underline{R}_k}} \tag{72}$$

References

- [1] A.A. Abakumov, A.Yu. Yeremin, Y.A. Kuznetsov, Efficient fast direct method for solving Poisson's equation on a parallelepiped and its implementation in an array processor, *Sov. J. Numer. Anal. Math. Model.* 3 (1988) 1–20.
- [2] A. Ammar, F. Chinesta, A particle strategy for solving the Fokker–Planck equation governing the fiber orientation distribution in steady recirculating flows involving short fiber suspensions, *Lectures Notes on Computational Science and Engineering*, vol. 43, Springer, 2005, pp. 1–16.
- [3] A. Ammar, D. Ryckelynck, F. Chinesta, R. Keunings, On the reduction of kinetic theory models related to finitely extensible dumbbells, *J. Non-Newton. Fluid Mech.* 134 (2006) 136–147.
- [4] R.B. Bird, C.F. Crutts, R.C. Armstrong, O. Hassager, *Dynamic of Polymeric Liquid, Kinetic Theory*, vol. 2, John Wiley and Sons, 1987.
- [5] H.J. Bungartz, M. Griebel, Sparse grids, *Acta Numerica* 13 (2004) 1–123.
- [6] C.V. Chaubal, A. Srinivasan, O. Egecioglu, L.G. Leal, Smoothed particle hydrodynamics techniques for the solution of kinetic theory problems, *J. Non-Newton. Fluid Mech.* 70 (1997) 125–154.
- [7] C. Chauviere, A. Lozinski, Simulation of dilute polymer solutions using a Fokker–Planck equation, *Comput. Fluids* 33 (2004) 687–696.
- [8] F. Chinesta, G. Chaidron, A. Poitou, On the solution of the Fokker–Planck equation in steady recirculating flows involving short fiber suspensions, *J. Non-Newton. Fluid Mech.* 113 (2003) 97–125.
- [9] R. Keunings, On the Peterlin approximation for finitely extensible dumbbells, *J. Non-Newton. Fluid Mech.* 68 (1997) 85–100.
- [10] R. Keunings, Micro–macro methods for the multiscale simulation viscoelastic flow using molecular models of kinetic theory, in: D.M. Binding, K. Walters (Eds.), *Rheology Reviews*, British Society of Rheology, 2004, pp. 67–98.
- [11] Y.A. Kuznetsov, A.M. Matsokin, On partial solution of systems of linear algebraic equations, *Sov. J. Numer. Anal. Math. Model.* 4 (6) (1989) 453–467.
- [12] Y.A. Kuznetsov, Block relaxation methods in subspaces their optimization and application, *Sov. J. Numer. Anal. Math. Model.* 4 (6) (1989) 433–451.
- [13] A. Lozinski, C. Chauviere, A fast solver for Fokker–Planck equation applied to viscoelastic flows calculations: 2D FENE model, *J. Comput. Phys.* 189 (2003) 607–625.
- [14] H.C. Ottinger, M. Laso, Smart polymers in finite element calculation, in: P. Moldanaers, R. Keunings (Eds.), *Theoretical and Applied Rheology*, vol. 1, Proceedings of the 11th International Congress on Rheology, Elsevier, 1992, pp. 286–288.
- [15] D. Ryckelynck, F. Chinesta, E. Cueto, A. Ammar, On the a priori model reduction: overview and recent developments, *State Art Rev.* 13 (1) (2006) 91–128.
- [16] T. Rossi, J. Toivanen, A non-standard cyclic reduction method, its variants and stability, *SIAM J. Matrix Anal. Appl.* 20 (3) (1999) 628–645.
- [17] T. Rossi, J. Toivanen, A parallel fast direct solver for block tridiagonal systems with separable matrices of arbitrary dimension, *SIAM J. Sci. Comput.* 20 (5) (1999) 1778–1796.
- [18] M. Somasi, B. Khomami, N.J. Woo, J.S. Hur, E.S.G. Shaqfeh, Brownian dynamics simulations of bead-rod and bead-spring chains: numerical algorithms and coarse-graining issues, *J. Non-Newton. Fluid Mech.* 108 (1–3) (2002) 227–255.
- [19] P.S. Vassilavski, Fast algorithm for solving a linear algebraic problem with separable variables, *C. R. Acad. Bulgare Sci.* 37 (3) (1984) 305–308.
- [20] G. Venkiteswaran, M. Junk, A QMC approach for high dimensional Fokker–Planck equations modelling polymeric liquids, *Math. Comput. Simul.* 68 (2005) 43–56.
- [21] J.M. Wiest, R.I. Tanner, Rheology of bead-non-linear spring chain macromolecules, *J. Rheol.* 33 (2) (1989) 281–316.