A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. Part II: Transient simulation using space-time separated representations

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 modelling of complex fluids
Part II: Transient simulation using space-time separated representations

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 described within the Fokker–Planck formalism involve high-dimensional spaces (including physical and conformation spaces and time). One appealing strategy for treating this kind of problems lies in the use of separated representations and tensor product approximations basis. This technique that was introduced in a former work [A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, J. Non-Newtonian Fluid Mech. 139 (2006) 153–176] for treating steady state kinetic theory models is extended here for treating transient models.

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

1. Introduction

This paper constitutes the progression of a previous work [3] which focused on the steady solution of some classes of multidimensional partial differential equations (PDE) defined in spaces of dimension N (with $N \gg 1$), and that is extended in the present work for treating multidimensional parabolic PDE.

In [3] a new solution technique based on a tensor product representation was introduced. In this technique the solution defined in multidimensional domains is built by adding a certain number of products of N one-dimensional functions, each one defined in a different space dimension. These functions are not known a priori, but they are constructed during the solution procedure. The resulting technique was evaluated by solving some multidimensional elliptic problems as well as in some kinetic theory models.

The solution of multidimensional transient Fokker–Planck equations could be performed within an incremental time discretization. However, the separated representation could be extended for representing the space-time solution. The main difficulties related to such approach lies in the fact that the initial condition being non-zero the proposed procedure cannot be applied in a direct manner. Another difficulty lies in the fact that space approximations are built using standard piece-wise functions, and when this kind of approximation is retained to construct the time interpolation, the well known instability related to centered differences of time derivatives is encountered. This paper proposes alternatives to circumvent both difficulties.

Despite the fact that tensor product spaces allow a significant reduction of the number of degrees of freedom, i.e. $N \times n_n$ instead of $(n_n)^N$ (where n_n is the number of degrees of freedom on each coordinate axis and N is the dimension of the problem), the resulting

* Corresponding author. Tel.: +33 4 76 82 52 94; fax: +33 4 76 82 51 64.
E-mail addresses: Amine.Ammar@ujf-grenoble.fr (A. Ammar), Bechir.Mokdad@ujf-grenoble.fr (B. Mokdad), francisco.chinesta@paris.ensam.fr (F. Chinesta), rk@inma.ucl.ac.be (R. Keunings).

discrete system is strongly non-linear. In this work, we propose an efficient alternating directions strategy for solving those non-linear problems.

The novelty of the proposed technique justifies that a lot of questions are still open: (i) the treatment of non-linear Fokker–Planck equations; (ii) optimal basis enrichment; (iii) analysis of complex flows involving non-homogeneous solution in the physical space; (iv) general initial conditions; (v) analysis of convergence; (vi) stabilization of advection operators; . . ..

We would like to emphasize that we are not looking for a general numerical procedure for solving multidimensional PDE's. In this context, the sparse grids or sparse tensor product basis [6] are excellent candidates, even if it was argued in [1] its inability for treating highly multidimensional problems ($N \gg 20$). The sparse grid technique has been applied for solving parabolic equations in moderate high-dimensional spaces [13](where the numerical examples concerned $N \leq 20$) as well as kinetic theory models also involving moderate high-dimensional spaces [8]. Moreover, some recent works [9] on sparse grids prove its ability for integrating functions defined in really high-dimensional spaces (an example in dimension 256 was reported in that paper), enlarging the perspectives towards the solution of multidimensional partial differential equations. The technique proposed in our former work [3] and extended in the present one to transient simulations, is, in our opinion, a suitable choice when dealing with highly multidimensional parabolic PDE's with homogeneous boundary conditions. Despite the apparent loss of generality induced by these assumptions, usual molecular descriptions make use of a configuration distribution function that is defined in a bounded domain and vanishes at its boundary.

2. Illustrating the solution strategy

Let us define the following heat problem in a space of dimension N:

$$\frac{\partial T}{\partial t} - \triangle T = f(x_1, x_2, \ldots, x_N), \tag{1}$$

where T is a scalar function of space and time coordinates, i.e. $T(\underline{x}, t) = T(x_1, x_2, \ldots, x_N, t)$. Problem (1) is assumed defined in the domain $\Omega = \Omega_x \times \Omega_t =]-L, +L[^N \times]0, t_{\max}]$ and T vanishes at the space boundary, i.e. $T(\underline{x} \in \partial\Omega_x, t) = 0$, as well as at the initial time, i.e. $T(\underline{x}, t = 0) = 0$.

The problem solution is assumed in the form:

$$T(x_1, x_2, \ldots, x_N, t) = \sum_{j=1}^{\infty} \alpha_j \left[\left(\prod_{k=1}^{N} F_{kj}(x_k) \right) F_{(N+1)j}(t) \right], \tag{2}$$

where F_{kj} is the jth basis function which only depends on the k th space coordinate. Thus, within the Galerkin framework the solution can be approximated in the form:

$$T(x_1, x_2, \ldots, x_N, t) \approx \sum_{j=1}^{J} \alpha_j \left[\left(\prod_{k=1}^{N} F_{kj}(x_k) \right) F_{(N+1)j}(t) \right], \tag{3}$$

where functions F_{kj} are not given in advance but they are computed adaptively using the algorithm described later.

The construction of solution (Eq. (3)) consists of an iteration procedure involving, at each iteration n, three steps (see [3] for additional details).

Step 1. Projection of the solution onto a discrete basis

If we assume the functions $F_{kj}(\forall j \in [1, \ldots, n]; \forall k \in [1, \ldots, N+1])$ known (verifying the boundary and the initial conditions, that is vanishing on $\partial\Omega_x$), the coefficients α_j can be computed by introducing the approximation of T into the Galerkin variational formulation associated with Eq. (1):

$$\int_{\Omega} T^* \frac{\partial T}{\partial t} \, \mathrm{d}\Omega + \int_{\Omega} \mathrm{Grad}\, T^* \cdot \mathrm{Grad}\, T \, \mathrm{d}\Omega = \int_{\Omega} T^* f \, \mathrm{d}\Omega. \tag{4}$$

Introducing the approximation of T and T^*:

$$T(x_1, x_2, \ldots, x_N, t) = \sum_{j=1}^{n} \alpha_j \left(\prod_{k=1}^{N} F_{kj}(x_k) \right) F_{(N+1)j}(t), \tag{5}$$

and

$$T^*(x_1, x_2, \ldots, x_N, t) = \sum_{j=1}^{n} \alpha_j^* \left(\prod_{k=1}^{N} F_{kj}(x_k) \right) F_{(N+1)j}(t), \tag{6}$$

we obtain

$$\int_\Omega \left(\sum_{j=1}^{n}\alpha_j^*\left(\prod_{k=1}^{N}F_{kj}(x_k)\right)F_{(N+1)j}(t)\right)\times\frac{\partial}{\partial t}\left(\sum_{j=1}^{n}\alpha_j\left(\prod_{k=1}^{N}F_{kj}(x_k)\right)F_{(N+1)j}(t)\right)\,\mathrm{d}\Omega$$
$$+\int_\Omega \mathrm{Grad}\left(\sum_{j=1}^{n}\alpha_j^*\left(\prod_{k=1}^{N}F_{kj}(x_k)\right)F_{(N+1)j}(t)\right)\times\mathrm{Grad}\left(\sum_{j=1}^{n}\alpha_j\left(\prod_{k=1}^{N}F_{kj}(x_k)\right)F_{(N+1)j}(t)\right)\,\mathrm{d}\Omega$$
$$=\int_\Omega\left(\sum_{j=1}^{n}\alpha_j^*\left(\prod_{k=1}^{N}F_{kj}(x_k)\right)F_{(N+1)j}(t)\right)f\,\mathrm{d}\Omega. \tag{7}$$

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

$$f(x_1,\ldots,x_N)=\sum_{h=1}^{m}\prod_{k=1}^{N}f_{kh}(x_k). \tag{8}$$

Eq. (7) involves integrals of a product of $N+1$ functions, each one defined in a different dimension. Let $(\prod_{k=1}^{N}g_k(x_k))g_{N+1}(t)$ be one of these functions to be integrated. The integral over Ω can be performed by integrating each function over its definition interval and then multiplying the $N+1$ computed integrals according to:

$$\int_\Omega\left(\prod_{k=1}^{N}g_k(x_k)\right)g_{N+1}(t)\,\mathrm{d}\Omega=\prod_{k=1}^{N}\int_{-L}^{L}g_k(x_k)\,\mathrm{d}x_k\times\int_0^{t_{\max}}g_{N+1}(t)\,\mathrm{d}t. \tag{9}$$

This makes possible the numerical integration in highly dimensional spaces.

Now, due to the arbitrariness of coefficients α_j^*, Eq. (7) allows us to compute the n approximation coefficients α_j, by solving the resulting linear system of size $n\times n$. This problem rarely exceeds the order of tens of degrees of freedom.

Step 2. Checking convergence

From the solution of T at iteration n just calculated and given by Eq. (5), we compute the residual Re related to Eq. (1):

$$Re=\frac{\sqrt{\int_\Omega((\partial T/\partial t)-\triangle T-f(x_1,\ldots,x_N))^2}}{\|T\|}. \tag{10}$$

where the norm of the solution is computed according to:

$$\|T\|=\int_\Omega T^2\,\mathrm{d}\Omega, \tag{11}$$

with all the integrals are computed in the space-time domain.

If $Re<\epsilon$ (epsilon is a small enough parameter), the iteration process stops, the solution $T(x_1,\ldots,x_N,t)$ being given by Eq. (5). Otherwise, the iteration procedure continues.

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 $(\prod_{k=1}^{N}F_{k(n+1)}(x_k))F_{(N+1)(n+1)}(t)$. For this purpose, we solve the non-linear Galerkin variational formulation related to Eq. (1):

$$\int_\Omega T^*\frac{\partial T}{\partial t}\,\mathrm{d}\Omega+\int_\Omega\mathrm{Grad}\,T^*\cdot\mathrm{Grad}\,T\,\mathrm{d}\Omega=\int_\Omega T^*f\,\mathrm{d}\Omega, \tag{12}$$

using the approximation of T given by:

$$T(x_1,x_2,\ldots,x_N,t)=\sum_{j=1}^{n}\alpha_j\left(\prod_{k=1}^{N}F_{kj}(x_k)\right)F_{(N+1)j}(t)+\left(\prod_{k=1}^{N}R_k(x_k)\right)R_{N+1}(t), \tag{13}$$

and the test function

$$T^*(x_1,x_2,\ldots,x_N,t)=R_1^*(x_1)\times\cdots\times R_N(x_N)\times R_{N+1}(t)+\cdots+R_1(x_1)\times\cdots\times R_N^*(x_N)\times R_{N+1}(t)$$
$$+R_1(x_1)\times\cdots\times R_N(x_N)\times R_{N+1}^*(t). \tag{14}$$

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

Remark 1. Each one of these one-dimensional functions $R_k(x_k)$ are approximated in this work using a piecewise finite element interpolation built from n_{n_k} nodes. Other possibilities have being investigated, as for example the use of spectral approximations that seem an appealing choice for increasing the convergence rates of the method, or the use of wavelet basis in order to take advantage of its multiresolution character which could lead to efficient adaptive strategies.

Remark 2. The particular choice of the ansatz (14) was justified in the first part of this work [3] in the vartiational framework. Thus, in the steady case this ansatz leads to symmetric and positive definite linear systems (when the alternating direction strategy described in Section 3.2.1 applies) for the differential operators having these properties. In the general case (general differential operators or parabolic problems) this ansatz vanishes on the domain boundary and when one proceeds using the alternating direction strategy for solving the resulting non-linear discrete problem (as described in Section 3.2.1) this choice does not introduce additional numerical difficulties and it worked perfectly in all the problems until now considered. However, we are aware that a further analysis on this questions should be addressed in future works.

3. Matrix form and some numerical issues

3.1. Solving the heat equation

In the 2D case, Eq. (1) reduces to:

$$\frac{\partial T}{\partial t} - \triangle T = f(x, y), \tag{15}$$

where $T = T(x, y, t)$. Eq. (15) is solved in the domain $\Omega = \Omega_x \times \Omega_t =]-L, +L[^2 \times]0, t_{\max}]$ with zero boundary and initial conditions.

The solution is now sought in the form:

$$T(x, y) = \sum_{j=1}^{\infty} \alpha_j F_j(x) G_j(y) H_j(t). \tag{16}$$

As seen previously, the construction of such solution involves, at each iteration n, a projection, convergence checking and enrichment steps that we now describe.

In the enrichment step, the approximation basis is enriched by adding the new functions $F_{n+1}(x)$, $G_{n+1}(y)$ and $H_{n+1}(t)$ according to

$$T(x, y, t) = \sum_{j=1}^{n} \alpha_j F_j(x) G_j(y) H_j(t) + R(x) S(y) P(t), \tag{17}$$

where $F_{n+1}(x)$, $G_{n+1}(y)$, $H_{n+1}(t)$ are obtained by normalizing the functions $R(x)$, $S(y)$ and $P(t)$ using the L^2 norm.

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

3.1.1. Computing of the coefficients α_j: projection stage

Now, we consider the variational formulation related to Eq. (15):

$$\int_{\Omega} T^* \frac{\partial T}{\partial t} \, \mathrm{d}\Omega + \int_{\Omega} \mathrm{Grad}\, T^* \mathrm{Grad}\, T \, \mathrm{d}\Omega = \int_{\Omega} T^* f(x, y) \, \mathrm{d}\Omega, \tag{18}$$

where the fact that T^* vanishes at the boundary of Ω_x has been introduced. Moreover, we assume that $f(x, y)$ can be also written in the form:

$$f(x, y) = \sum_{h=1}^{m} a_h(x) b_h(y). \tag{19}$$

Thus, Eq. (18) becomes:

$$\int_\Omega T^* \frac{\partial T}{\partial t}\,\mathrm{d}\Omega + \int_\Omega \begin{bmatrix} \frac{\partial T^*}{\partial x} & \frac{\partial T^*}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} \mathrm{d}\Omega = \sum_{h=1}^{m} \int_\Omega T^* a_h(x) b_h(y)\,\mathrm{d}\Omega. \tag{20}$$

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

$$T(x,y,t) = [\, \underline{N}^{\mathrm{T}} \underline{F}_1 \underline{M}^{\mathrm{T}} \underline{G}_1 \underline{L}^{\mathrm{T}} \underline{H}_1 \quad \cdots \quad \underline{N}^{\mathrm{T}} \underline{F}_n \underline{M}^{\mathrm{T}} \underline{G}_n \underline{L}^{\mathrm{T}} \underline{H}_n \,] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \underline{A}^{\mathrm{T}} \underline{\alpha}, \tag{21}$$

from which we obtain the time derivative and the gradient:

$$\frac{\partial T}{\partial t} = [\, \underline{N}^{\mathrm{T}} \underline{F}_1 \underline{M}^{\mathrm{T}} \underline{G}_1 \underline{\mathrm{d}L}^{\mathrm{T}} \underline{H}_1 \quad \cdots \quad \underline{N}^{\mathrm{T}} \underline{F}_n \underline{M}^{\mathrm{T}} \underline{G}_n \underline{\mathrm{d}L}^{\mathrm{T}} \underline{H}_n \,] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \underline{B}_t^{\mathrm{T}} \underline{\alpha}, \tag{22}$$

$$\begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} = \begin{bmatrix} \underline{\mathrm{d}N}^{\mathrm{T}} \underline{F}_1 \underline{M}^{\mathrm{T}} \underline{G}_1 \underline{L}^{\mathrm{T}} \underline{H}_1 & \cdots & \underline{\mathrm{d}N}^{\mathrm{T}} \underline{F}_n \underline{M}^{\mathrm{T}} \underline{G}_n \underline{L}^{\mathrm{T}} \underline{H}_n \\ \underline{N}^{\mathrm{T}} \underline{F}_1 \underline{\mathrm{d}M}^{\mathrm{T}} \underline{G}_1 \underline{L}^{\mathrm{T}} \underline{H}_1 & \cdots & \underline{N}^{\mathrm{T}} \underline{F}_n \underline{\mathrm{d}M}^{\mathrm{T}} \underline{G}_n \underline{L}^{\mathrm{T}} \underline{H}_n \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \underline{\underline{B}}_x \underline{\alpha}. \tag{23}$$

Here, $\underline{\mathrm{d}N}$, $\underline{\mathrm{d}M}$ and $\underline{\mathrm{d}L}$ represent the vectors containing the value of the shape functions derivatives with respect to the coordinates x, y and t at any point x and y and at any time t, respectively. The weighting function related to the Galerkin variational formulation T^* can be expressed in the same form (the stabilization of the time derivative –first term of Eq. (18)– will be addressed later):

$$T^*(x,y,t) = [\, \alpha_1^* \quad \cdots \quad \alpha_n^* \,] \begin{bmatrix} \underline{F}_1^{\mathrm{T}} \underline{N} \underline{G}_1^{\mathrm{T}} \underline{M} \underline{H}_1^{\mathrm{T}} \underline{L} \\ \vdots \\ \underline{F}_n^{\mathrm{T}} \underline{N} \underline{G}_n^{\mathrm{T}} \underline{M} \underline{H}_n^{\mathrm{T}} \underline{L} \end{bmatrix} = \underline{\alpha}^{*\mathrm{T}} \underline{A}, \tag{24}$$

and its gradient

$$\begin{bmatrix} \frac{\partial T^*}{\partial x} & \frac{\partial T^*}{\partial y} \end{bmatrix} = [\, \alpha_1^* \quad \cdots \quad \alpha_n^* \,] \begin{bmatrix} \underline{F}_1^{\mathrm{T}} \underline{\mathrm{d}N} \underline{G}_1^{\mathrm{T}} \underline{M} \underline{H}_1^{\mathrm{T}} \underline{L} & \underline{F}_1^{\mathrm{T}} \underline{N} \underline{G}_1^{\mathrm{T}} \underline{\mathrm{d}M} \underline{H}_1^{\mathrm{T}} \underline{L} \\ \vdots & \vdots \\ \underline{F}_n^{\mathrm{T}} \underline{\mathrm{d}N} \underline{G}_n^{\mathrm{T}} \underline{M} \underline{H}_n^{\mathrm{T}} \underline{L} & \underline{F}_n^{\mathrm{T}} \underline{N} \underline{G}_n^{\mathrm{T}} \underline{\mathrm{d}M} \underline{H}_n^{\mathrm{T}} \underline{L} \end{bmatrix} = \underline{\alpha}^{*\mathrm{T}} \underline{\underline{B}}_x^{\mathrm{T}}. \tag{25}$$

Introducing the discrete form of the trial and test functions as well as their derivatives in Eq. (20), we obtain:

$$\int_\Omega \underline{\alpha}^{*\mathrm{T}} (\underline{A} \underline{B}_t^{\mathrm{T}} + \underline{\underline{B}}_x^{\mathrm{T}} \underline{\underline{B}}_x) \underline{\alpha}\,\mathrm{d}\Omega = \sum_{h=1}^{m} \int_\Omega [\, \alpha_1^* \quad \cdots \quad \alpha_n^* \,] \begin{bmatrix} a_h(x) \underline{F}_1^{\mathrm{T}} \underline{N}\, b_h(y) \underline{G}_1^{\mathrm{T}} \underline{M} \underline{H}_1^{\mathrm{T}} \underline{L} \\ \vdots \\ a_h(x) \underline{F}_n^{\mathrm{T}} \underline{N}\, b_h(y) \underline{G}_n^{\mathrm{T}} \underline{M} \underline{H}_n^{\mathrm{T}} \underline{L} \end{bmatrix} \mathrm{d}\Omega, \tag{26}$$

where the components of matrix $\underline{A} \underline{B}_t^{\mathrm{T}} + \underline{\underline{B}}_x^{\mathrm{T}} \underline{\underline{B}}_x$ are given by:

$$\begin{aligned} (\underline{A} \underline{B}_t^{\mathrm{T}} + \underline{\underline{B}}_x^{\mathrm{T}} \underline{\underline{B}}_x)_{ij} &= \underline{F}_i^{\mathrm{T}} \underline{\mathrm{d}N} \underline{\mathrm{d}N}^{\mathrm{T}} \underline{F}_j \underline{G}_i^{\mathrm{T}} \underline{M} \underline{M}^{\mathrm{T}} \underline{G}_j \underline{H}_i^{\mathrm{T}} \underline{L} \underline{L}^{\mathrm{T}} \underline{H}_j + \underline{F}_i^{\mathrm{T}} \underline{N} \underline{N}^{\mathrm{T}} \underline{F}_j \underline{G}_i^{\mathrm{T}} \underline{\mathrm{d}M} \underline{\mathrm{d}M}^{\mathrm{T}} \underline{G}_j \underline{H}_i^{\mathrm{T}} \underline{L} \underline{L}^{\mathrm{T}} \underline{H}_j \\ &+ \underline{F}_i^{\mathrm{T}} \underline{N} \underline{N}^{\mathrm{T}} \underline{F}_j \underline{G}_i^{\mathrm{T}} \underline{M} \underline{M}^{\mathrm{T}} \underline{G}_j \underline{H}_i^{\mathrm{T}} \underline{L} \underline{\mathrm{d}L}^{\mathrm{T}} \underline{H}_j. \end{aligned} \tag{27}$$

Because all terms in Eq. (26) are the product of separable functions of x, y and t, the integral over the domain Ω can be separated into the product of integrals in x, y and t. Thus, it finally results:

$$\underline{\alpha}^{*\mathrm{T}} \underline{\underline{K}} \underline{\alpha} = \underline{\alpha}^{*\mathrm{T}} \underline{V}. \tag{28}$$

Accounting for the arbitrariness of $\underline{\alpha}^*$, we get

$$\underline{\underline{K}} \underline{\alpha} = \underline{V}, \tag{29}$$

where

$$\underline{\underline{K}}_{ij} = \underline{F_i}^{\mathrm{T}} \left(\int_{x=-L}^{x=+L} \underline{\mathrm{d}N}\underline{\mathrm{d}N}^{\mathrm{T}}\,\mathrm{d}x \right) \underline{F_j} \times \underline{G_i}^{\mathrm{T}} \left(\int_{y=-L}^{y=+L} \underline{M}\underline{M}^{\mathrm{T}}\,\mathrm{d}y \right) \underline{G_j} \times \underline{H_i}^{\mathrm{T}} \left(\int_{t=0}^{t=t_{\max}} \underline{L}\underline{L}^{\mathrm{T}}\,\mathrm{d}t \right) \underline{H_j}$$
$$+ \underline{F_i}^{\mathrm{T}} \left(\int_{x=-L}^{x=+L} \underline{N}\underline{N}^{\mathrm{T}}\,\mathrm{d}x \right) \underline{F_j} \times \underline{G_i}^{\mathrm{T}} \left(\int_{y=-L}^{y=+L} \underline{\mathrm{d}M}\underline{\mathrm{d}M}^{\mathrm{T}}\,\mathrm{d}y \right) \underline{G_j} \times \underline{H_i}^{\mathrm{T}} \left(\int_{t=0}^{t=t_{\max}} \underline{L}\underline{L}^{\mathrm{T}}\,\mathrm{d}t \right) \underline{H_j}$$
$$+ \underline{F_i}^{\mathrm{T}} \left(\int_{x=-L}^{x=+L} \underline{N}\underline{N}^{\mathrm{T}}\,\mathrm{d}x \right) \underline{F_j} \times \underline{G_i}^{\mathrm{T}} \left(\int_{y=-L}^{y=+L} \underline{M}\underline{M}^{\mathrm{T}}\,\mathrm{d}y \right) \underline{G_j} \times \underline{H_i}^{\mathrm{T}} \left(\int_{t=0}^{t=t_{\max}} \underline{L}\underline{\mathrm{d}L}^{\mathrm{T}}\,\mathrm{d}t \right) \underline{H_j}, \tag{30}$$

and

$$\underline{V_i} = \sum_{h=1}^{m} \underline{F_i}^{\mathrm{T}} \left(\int_{x=-L}^{x=+L} \underline{N}\, a_h(x)\,\mathrm{d}x \right) \times \underline{G_i}^{\mathrm{T}} \left(\int_{y=-L}^{y=+L} \underline{M}\, b_h(y)\,\mathrm{d}y \right) \times \underline{H_i}^{\mathrm{T}} \left(\int_{t=0}^{t=t_{\max}} \underline{L}\,\mathrm{d}t \right). \tag{31}$$

The linear system (29) can be solved for computing the coefficients α_j. The inversion of matrix $\underline{\underline{K}}$ is very simple due to its reduced dimension.

3.1.2. Enrichment of the approximation basis

At this stage, we are looking for a solution in the form:

$$\begin{cases} T(x,y,t) = \displaystyle\sum_{j=1}^{j=n} \alpha_j F_j(x) G_j(y) H_j(t) + R(x)S(y)P(t) \\ T^*(x,y,t) = R^*(x)S(y)P(t) + R(x)S^*(y)P(t) + R(x)S(y)P^*(t) \end{cases}. \tag{32}$$

In order to compute functions R, S and P from the associated variational formulation (Eq. (18)), we start by writing these functions as well as their time and space derivatives in a matrix form:

$$\frac{\partial T}{\partial t} = \sum_{j=1}^{n} \alpha_j [\underline{N}^{\mathrm{T}}\underline{F_j}\underline{M}^{\mathrm{T}}\underline{G_j}\underline{\mathrm{d}L}^{\mathrm{T}}\underline{H_j}] + [\underline{0}^{\mathrm{T}}\underline{0}^{\mathrm{T}}\underline{N}^{\mathrm{T}}\underline{R}\underline{M}^{\mathrm{T}}\underline{S}\underline{\mathrm{d}L}^{\mathrm{T}}] \begin{bmatrix} \underline{R} \\ \underline{S} \\ \underline{P} \end{bmatrix} = \underline{B_t}^{\mathrm{T}}\underline{\alpha} + \underline{E_t}^{\mathrm{T}} \begin{bmatrix} \underline{R} \\ \underline{S} \\ \underline{P} \end{bmatrix}, \tag{33}$$

$$\begin{bmatrix} \dfrac{\partial T}{\partial x} \\ \dfrac{\partial T}{\partial y} \end{bmatrix} = \sum_{j=1}^{n} \alpha_j \begin{bmatrix} \underline{\mathrm{d}N}^{\mathrm{T}}\underline{F_j}\underline{M}^{\mathrm{T}}\underline{G_j}\underline{L}^{\mathrm{T}}\underline{H_j} \\ \underline{N}^{\mathrm{T}}\underline{F_j}\underline{\mathrm{d}M}^{\mathrm{T}}\underline{G_j}\underline{L}^{\mathrm{T}}\underline{H_j} \end{bmatrix} + \begin{bmatrix} \underline{\mathrm{d}N}^{\mathrm{T}}\underline{M}^{\mathrm{T}}\underline{S}\underline{L}^{\mathrm{T}}\underline{P} & \underline{0}^{\mathrm{T}} & \underline{0}^{\mathrm{T}} \\ \underline{0}^{\mathrm{T}} & \underline{N}^{\mathrm{T}}\underline{R}\underline{\mathrm{d}M}^{\mathrm{T}}\underline{L}^{\mathrm{T}}\underline{P} & \underline{0}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \underline{R} \\ \underline{S} \\ \underline{P} \end{bmatrix} = \underline{\underline{B}}_x\underline{\alpha} + \underline{\underline{E}}_x \begin{bmatrix} \underline{R} \\ \underline{S} \\ \underline{P} \end{bmatrix}, \tag{34}$$

$$T^*(x,y,t) = [\,\underline{R}^{*\mathrm{T}} \quad \underline{S}^{*\mathrm{T}} \quad \underline{P}^{*\mathrm{T}}\,] \begin{bmatrix} \underline{N}\underline{S}^{\mathrm{T}}\underline{M}\underline{P}^{\mathrm{T}}\underline{L} \\ \underline{R}^{\mathrm{T}}\underline{N}\underline{M}\underline{P}^{\mathrm{T}}\underline{L} \\ \underline{R}^{\mathrm{T}}\underline{N}\underline{S}^{\mathrm{T}}\underline{M}\underline{L} \end{bmatrix} = [\,\underline{R}^{*\mathrm{T}} \quad \underline{S}^{*\mathrm{T}} \quad \underline{P}^{*\mathrm{T}}\,]\underline{C}, \tag{35}$$

$$\begin{bmatrix} \dfrac{\partial T^*}{\partial x} & \dfrac{\partial T^*}{\partial y} \end{bmatrix} = [\,\underline{R}^{*\mathrm{T}} \quad \underline{S}^{*\mathrm{T}} \quad \underline{P}^{*\mathrm{T}}\,] \begin{bmatrix} \underline{\mathrm{d}N}\underline{S}^{\mathrm{T}}\underline{M}\underline{P}^{\mathrm{T}}\underline{L} & \underline{N}\underline{S}^{\mathrm{T}}\underline{\mathrm{d}M}\underline{P}^{\mathrm{T}}\underline{L} \\ \underline{R}^{\mathrm{T}}\underline{\mathrm{d}N}\underline{M}\underline{P}^{\mathrm{T}}\underline{L} & \underline{R}^{\mathrm{T}}\underline{N}\underline{\mathrm{d}M}\underline{P}^{\mathrm{T}}\underline{L} \\ \underline{R}^{\mathrm{T}}\underline{\mathrm{d}N}\underline{S}^{\mathrm{T}}\underline{M}\underline{L} & \underline{R}^{\mathrm{T}}\underline{N}\underline{S}^{\mathrm{T}}\underline{\mathrm{d}M}\underline{L} \end{bmatrix} = [\,\underline{R}^{*\mathrm{T}} \quad \underline{S}^{*\mathrm{T}} \quad \underline{P}^{*\mathrm{T}}\,]\underline{\underline{F}}^{\mathrm{T}}. \tag{36}$$

Using these expressions the variational formulation results:

$$\int_{\Omega} [\,\underline{R}^{*\mathrm{T}} \quad \underline{S}^{*\mathrm{T}} \quad \underline{P}^{*\mathrm{T}}\,]\underline{C}\underline{B_t}^{\mathrm{T}}\underline{\alpha}\,\mathrm{d}\Omega + \int_{\Omega} [\,\underline{R}^{*\mathrm{T}} \quad \underline{S}^{*\mathrm{T}} \quad \underline{P}^{*\mathrm{T}}\,]\underline{C}\underline{E_t}^{\mathrm{T}} \begin{bmatrix} \underline{R} \\ \underline{S} \\ \underline{P} \end{bmatrix} \mathrm{d}\Omega + \int_{\Omega} [\,\underline{R}^{*\mathrm{T}} \quad \underline{S}^{*\mathrm{T}} \quad \underline{P}^{*\mathrm{T}}\,]\underline{\underline{F}}^{\mathrm{T}}\underline{\underline{B}}_x\underline{\alpha}\,\mathrm{d}\Omega$$
$$+ \int_{\Omega} [\,\underline{R}^{*\mathrm{T}} \quad \underline{S}^{*\mathrm{T}} \quad \underline{P}^{*\mathrm{T}}\,]\underline{\underline{F}}^{\mathrm{T}}\underline{\underline{E}}_x \begin{bmatrix} \underline{R} \\ \underline{S} \\ \underline{P} \end{bmatrix} \mathrm{d}\Omega = \sum_{h=1}^{m} \int_{\Omega} [\,\underline{R}^{*\mathrm{T}} \quad \underline{S}^{*\mathrm{T}} \quad \underline{P}^{*\mathrm{T}}\,]\underline{C}a_h(x)b_h(y)\,\mathrm{d}\Omega. \tag{37}$$

As previously, each integral in Eq. (37) can be separated into an integral in the coordinate x, a second one in the direction y and a third one in the time interval.

Eq. (37) can then be expressed in the compact form:

$$\underline{U}(\underline{R},\underline{S},\underline{P}) + \underline{\underline{K}}(\underline{R},\underline{S},\underline{P}) \begin{bmatrix} \underline{R} \\ \underline{S} \\ \underline{P} \end{bmatrix} = \underline{V}(\underline{R},\underline{S},\underline{P}), \tag{38}$$

which defines a non-linear problem of reduced dimension $(p+q+r)$. After imposing the nullity of both functions R and S at their respective domain boundaries and P at $t=0$, the non-linear system (38) can be solved by applying a Newton strategy, the size of the linearized systems being $p+q+r$. As the solution of such problem is not unique, we can impose for example that the norm of S is unity, by introducing the associated Lagrange multiplier. In order to accelerate the convergence, the first approximated field T is constructed by using the functions a_h and b_h.

When the solutions R, S and P are obtained (after convergence), the new approximation functions can be determined after normalization:

$$\begin{cases} \underline{F}_{n+1} = \dfrac{\underline{R}}{\sqrt{\underline{R}^{\mathrm{T}} \left(\displaystyle\int_{x=-L}^{x=+L} \underline{N}\underline{N}^{\mathrm{T}} \, \mathrm{d}x \right) \underline{R}}} \\ \underline{G}_{n+1} = \dfrac{\underline{S}}{\sqrt{\underline{S}^{\mathrm{T}} \left(\displaystyle\int_{y=-L}^{y=+L} \underline{M}\underline{M}^{\mathrm{T}} \, \mathrm{d}y \right) \underline{S}}} \\ \underline{H}_{n+1} = \dfrac{\underline{P}}{\sqrt{\underline{P}^{\mathrm{T}} \left(\displaystyle\int_{t=0}^{t=t_{\max}} \underline{L}\underline{L}^{\mathrm{T}} \, \mathrm{d}t \right) \underline{P}}} \end{cases}. \tag{39}$$

If normality of S has been set, then the second expression of Eq. (39) reduces to $\underline{G}_{n+1} = \underline{S}$.

3.2. Numerical issues

3.2.1. Efficient non-linear solvers

In high-dimensional spaces, the discrete system (38) is strongly non-linear, and the convergence of non-linear solvers becomes a delicate task. In our previous work [3] a Newton strategy was successfully applied. However, in this work we propose an alternative procedure based on the resolution of smaller linear systems within an iterative scheme, that from our numerical experiments seems to be faster, simpler and more robust.

The non-linear system can be transformed into a set of $N+1$ linear systems (3 in our example whose sizes are p, q and r) in the framework of the alternating directions technique. These linear systems are solved within an iteration procedure until convergence. To illustrate this technique we consider Eq. (38) written as:

$$\begin{pmatrix} \underline{U}_1(\underline{S},\underline{P}) \\ \underline{U}_2(\underline{R},\underline{P}) \\ \underline{U}_1(\underline{R},\underline{S}) \end{pmatrix} + \begin{pmatrix} \underline{\underline{K}}_{11}(\underline{S},\underline{P}) & \underline{\underline{K}}_{12}(\underline{R},\underline{S},\underline{P}) & \underline{\underline{K}}_{13}(\underline{R},\underline{S},\underline{P}) \\ \underline{\underline{K}}_{21}(\underline{R},\underline{S},\underline{P}) & \underline{\underline{K}}_{22}(\underline{R},\underline{P}) & \underline{\underline{K}}_{23}(\underline{R},\underline{S},\underline{P}) \\ \underline{\underline{K}}_{31}(\underline{R},\underline{S},\underline{P}) & \underline{\underline{K}}_{32}(\underline{R},\underline{S},\underline{P}) & \underline{\underline{K}}_{33}(\underline{R},\underline{S}) \end{pmatrix} \begin{pmatrix} \underline{R} \\ \underline{S} \\ \underline{P} \end{pmatrix} = \begin{pmatrix} \underline{V}_1(\underline{S},\underline{P}) \\ \underline{V}_2(\underline{R},\underline{P}) \\ \underline{V}_3(\underline{R},\underline{S}) \end{pmatrix}. \tag{40}$$

Using the matrix expressions previously introduced, it is easy to verify the following relations:

$$\begin{cases} \underline{\underline{K}}_{12}(\underline{R},\underline{S},\underline{P})\underline{S} \equiv \underline{\underline{K}}^*_{12}(\underline{S},\underline{P})\underline{R} \\ \underline{\underline{K}}_{13}(\underline{R},\underline{S},\underline{P})\underline{P} \equiv \underline{\underline{K}}^*_{13}(\underline{S},\underline{P})\underline{R} \\ \underline{\underline{K}}_{21}(\underline{R},\underline{S},\underline{P})\underline{R} \equiv \underline{\underline{K}}^*_{21}(\underline{R},\underline{P})\underline{S} \\ \underline{\underline{K}}_{23}(\underline{R},\underline{S},\underline{P})\underline{P} \equiv \underline{\underline{K}}^*_{23}(\underline{R},\underline{P})\underline{S} \\ \underline{\underline{K}}_{31}(\underline{R},\underline{S},\underline{P})\underline{R} \equiv \underline{\underline{K}}^*_{31}(\underline{R},\underline{S})\underline{P} \\ \underline{\underline{K}}_{32}(\underline{R},\underline{S},\underline{P})\underline{S} \equiv \underline{\underline{K}}^*_{32}(\underline{R},\underline{S})\underline{P} \end{cases}. \tag{41}$$

They lead to 3 linear systems that must be solved iteratively until reaching convergence:

(1) Assuming $\underline{S}$ and $\underline{P}$ known:

$$\underline{U}_1 + (\underline{\underline{K}}_{11} + \underline{\underline{K}}^*_{12} + \underline{\underline{K}}^*_{13})\underline{R} = \underline{V}_1, \tag{42}$$

(2) Assuming $\underline{R}$ and $\underline{P}$ known:

$$\underline{U}_2 + (\underline{\underline{K}}^*_{21} + \underline{\underline{K}}_{22} + \underline{\underline{K}}^*_{23})\underline{S} = \underline{V}_2, \tag{43}$$

(3) Assuming $\underline{R}$ and $\underline{S}$ known:

$$\underline{U}_3 + (\underline{\underline{K}}^*_{31} + \underline{\underline{K}}^*_{32} + \underline{\underline{K}}^*_{33})\underline{P} = \underline{V}_3. \tag{44}$$

Other alternatives for solving this kind of strongly non-linear problems are being investigated, as is the case of the so-called asymptotic numerical technique [7].

3.2.2. Consistent time discretization

Another important numerical issue is the discretization of the time derivative term in the variational formulation (18). If we consider the components of $\underline{L}$ as the usual piecewise linear shape functions, and perform the integral

$$\int_{\Omega_t} \underline{L}\mathrm{d}\underline{L}^{\mathrm{T}}\,\mathrm{d}t,$$

the i th-row of the resulting matrix contains two non-zero entries: $1/2$ at column $i+1$ and $-1/2$ at column $i-1$. This is equivalent (dividing by the time step h_t) to the central finite difference which is not consistent with the physics (justifying the use of backward time derivatives). As we are discretizing the space-time variational formulation we cannot apply backward finite difference, and for this reason we propose to use an "upwind" Petrov–Galerkin formulation which leads to a backward time discretization. Thus, matrix $\underline{L}\mathrm{d}\underline{L}$ is computed by considering $\underline{\tilde{L}}$ instead of the usual $\underline{L}$, which means that the test function T^* considered in the first term of the variational formulation (18) is in fact approximated using other approximation functions than the ones used to approximate T, i.e.:

$$\int_{\Omega} \tilde{T}^* \frac{\partial T}{\partial t}\,\mathrm{d}\Omega + \int_{\Omega} \mathrm{Grad}\,T^*\,\mathrm{Grad}\,T\,\mathrm{d}\Omega = \int_{\Omega} T^* f(x,\,y)\,\mathrm{d}\Omega. \tag{45}$$

Thus, using the analogy with the upwind Petrov–Galekin formulation widely used to discretize advection operators, we obtain:

$$T(x,\,y,\,t) = [\,\underline{N}^{\mathrm{T}}\underline{F}_1\underline{M}^{\mathrm{T}}\underline{G}_1\underline{L}^{\mathrm{T}}\underline{H}_1^{\mathrm{T}} \quad \cdots \quad \underline{N}^{\mathrm{T}}\underline{F}_n\underline{M}^{\mathrm{T}}\underline{G}_n\underline{L}^{\mathrm{T}}\underline{H}_n^{\mathrm{T}}\,] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}, \tag{46}$$

$$T^*(x,\,y,\,t) = [\,\underline{N}^{\mathrm{T}}\underline{F}_1\underline{M}^{\mathrm{T}}\underline{G}_1\underline{L}^{\mathrm{T}}\underline{H}_1^{\mathrm{T}} \quad \cdots \quad \underline{N}^{\mathrm{T}}\underline{F}_n\underline{M}^{\mathrm{T}}\underline{G}_n\underline{L}^{\mathrm{T}}\underline{H}_n^{\mathrm{T}}\,] \begin{bmatrix} \alpha^*_1 \\ \vdots \\ \alpha^*_n \end{bmatrix}, \tag{47}$$

and

$$\tilde{T}^*(x,\,y,\,t) = [\,\underline{N}^{\mathrm{T}}\underline{F}_1\underline{M}^{\mathrm{T}}\underline{G}_1\underline{\tilde{L}}^{\mathrm{T}}\underline{H}_1^{\mathrm{T}} \quad \cdots \quad \underline{N}^{\mathrm{T}}\underline{F}_n\underline{M}^{\mathrm{T}}\underline{G}_n\underline{\tilde{L}}^{\mathrm{T}}\underline{H}_n^{\mathrm{T}}\,] \begin{bmatrix} \alpha^*_1 \\ \vdots \\ \alpha^*_n \end{bmatrix} \tag{48}$$

with

$$\underline{\tilde{L}} = \underline{L} + \frac{h_t}{2}\,\mathrm{d}\underline{L}. \tag{49}$$

This leads to a matrix $[\int_0^{t_{\max}} \underline{\tilde{L}}\mathrm{d}\underline{L}^{\mathrm{T}}\,\mathrm{d}t]$ whose i th-row contains only two non-zero entries, at columns i (with a unit value) and $i-1$ where it takes a value -1. Thus, the time discretization is now fully consistent.

3.3. Numerical example

We consider the 2D transient problem defined by

$$\frac{\partial T}{\partial t} - \triangle T = \max(0,\,x^5) \times \max(0,\,y^5) + \min(0,\,x^5) \times \min(0,\,y^5),$$

in $\Omega = \Omega_x \times \Omega_t =]-L,\,L[^2 \times]0,\,t_{\max}]$ with $L = 3$ and $t_{\max} = 2$ and the boundary and initial conditions given by: $T(\underline{x} \in \partial\Omega_x,\,t = 0)$ and $T(\underline{x},\,t = 0) = 0$.

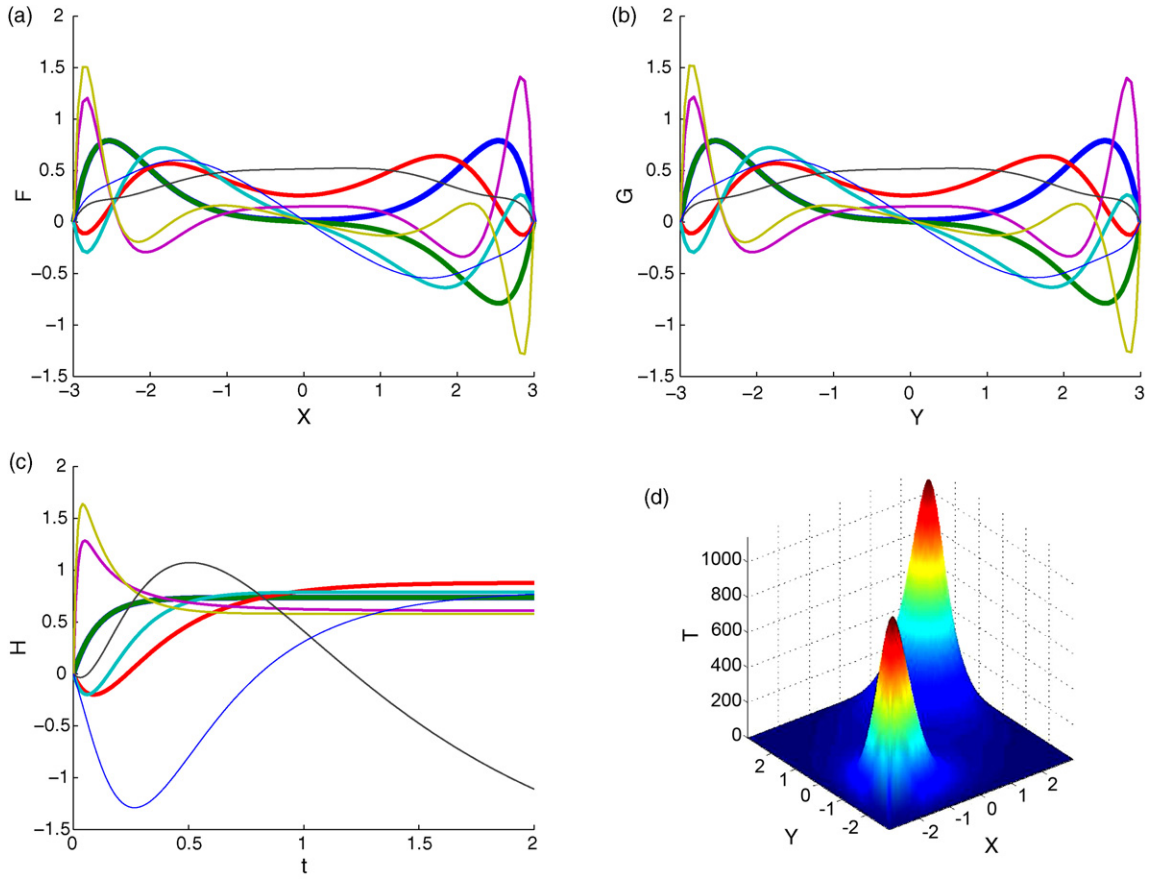

Fig. 1. 2D Poisson problem (a) functions in the direction x (b) functions in the direction y (c) time functions and (d) final solution distribution.

Fig. 1 illustrates the different space and time 1D functions as well as the final space distribution of the solution $T(\underline{x}, t = t_{\max})$. The width of each representation of the 1D approximation functions illustrates the weight of that function, which is given by the corresponding α coefficient. The computed solution is in perfect agreement with the one obtained by using a classical finite element method.

3.4. Convergence analysis

In order to conclude about the convergence rates of the proposed strategy we are considering a steady problem as well as a transient one whose large time solution coincides with the solution of the steady problem.

The steady problem is defined by:

$$\Delta T = -N \prod_{k=1}^{N} \sin(x_k) \tag{50}$$

in $\Omega =]-\pi, \pi[^N$, with $T(\underline{x} \in \partial\Omega) = 0$, whose exact solution reads:

$$T^{\text{ex}} = \prod_{k=1}^{N} \sin(x_k). \tag{51}$$

This problem is solved in different multidimensional spaces considering the separated representation and an increasing number of nodes for the functional approximation in each dimension (for the sake of simplicity we are considering that all the one-dimensional functional approximations are defined from the same number of nodes n_n).

The error E is quantified from the norm:

$$E = \frac{\int_\Omega (T^{\text{num}} - T^{\text{ex}})^2 \, \mathrm{d}\Omega}{|\Omega|}. \tag{52}$$

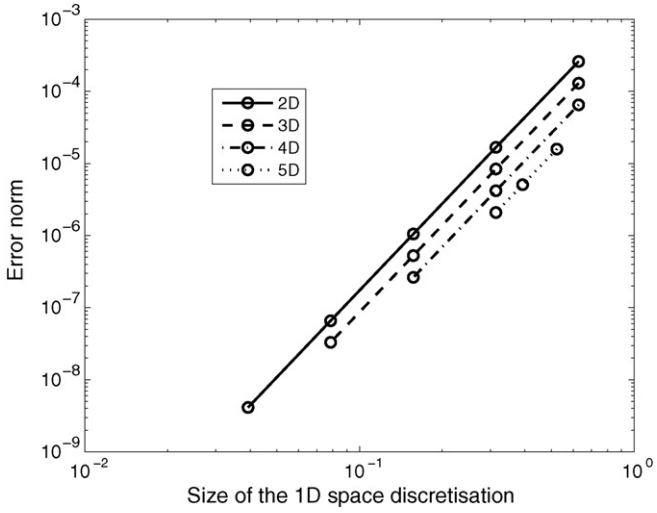

Fig. 2. Convergence analysis for the multidimensional heat equation.

Fig. 2 depicts the error versus the size of the one-dimensional space discretization, that in this example results $h = 2\pi/(n_n - 1)$, when different multidimensional spaces are considered. A convergence rate of 4 (slope in the log–log representation) is noticed, two times higher than the quadratic one characteristic of piecewise linear approximations.

Despite at present the code is not optimized, we have reported in Fig. 3 the CPU time associated to each solution. We can notice that even in 100 dimensions and using 1000 degrees of freedom in each dimension (note that the size of the resulting model is of 10^5 when the separated representation is used instead of 10^{300} required in the framework of the finite element or any other grid method) the CPU time remains in the order of 1000 s. A slope of two is noticed that seems independent of the space dimension. Thus, the computing time increases in a power of two with respect to the number of nodes considered for the one-dimensional functional approximations.

To check its dependence on the space dimension we depict in Fig. 4 the evolution of the CPU time versus the space dimension for two meshes consisting of $n_n = 401$ and $n_n = 1601$ nodes (in each direction). The same slope of two is noticed proving that the CPU time increases in a similar way with the space dimension and with the number of degrees of freedom considered for the one-dimensional functional approximations.

Now, we are considering the associated transient model:

$$\frac{\partial T}{\partial t} - \Delta T = N \prod_{k=1}^{N} \sin(x_k) \tag{53}$$

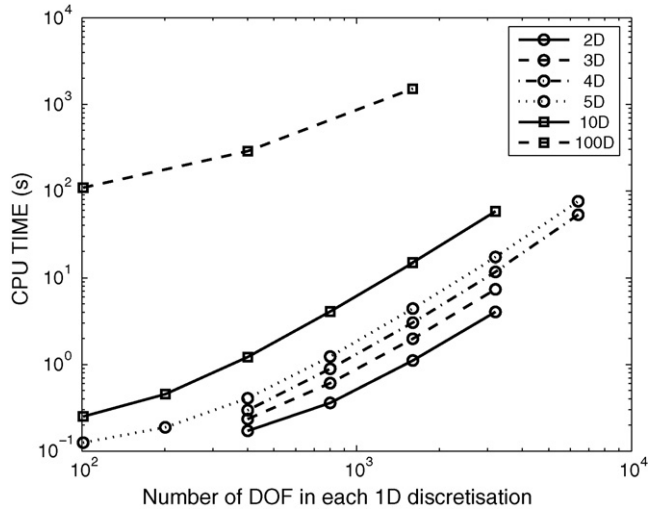

Fig. 3. Computing time against nodal density.

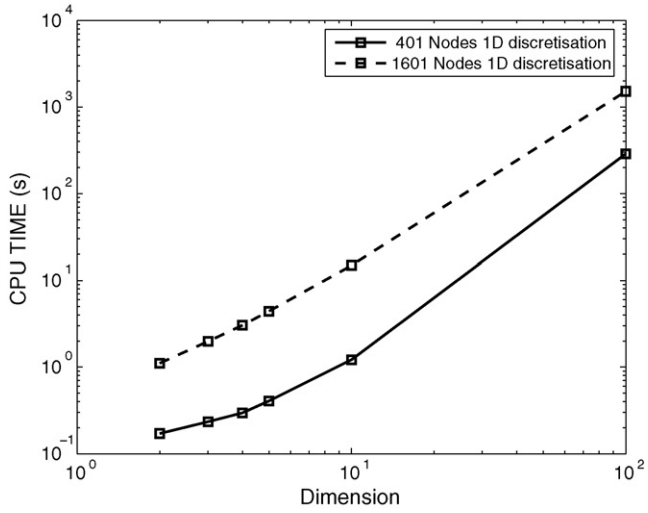

Fig. 4. Computing time against space dimension.

defined in $\Omega = \Omega_x \times \Omega_t =]-\pi, \pi[^N \times]0, 1]$, with $T(\underline{x} \in \partial\Omega_x, t) = 0$ and $T(\underline{x}, t = 0) = 0$, whose exact solution reads

$$T^{\text{ex}}(\underline{x}, t) = (1 - \mathrm{e}^{-Nt}) \prod_{k=1}^{N} \sin(x_k). \tag{54}$$

The error E is quantified from the norm:

$$E = \frac{\int_{\Omega_t} \int_{\Omega_x} (T^{\text{num}} - T^{\text{ex}})^2 \, \mathrm{d}\Omega_x \, \mathrm{d}\Omega_t}{|\Omega|}. \tag{55}$$

We solve the problem using a separated representation, with both the space dimension N and the number of nodes used in the space interpolation n_n fixed. In these solutions only the number of nodes involved in the time discretization is increased. The computed error is reported in Fig. 5 which proves that the proposed strategy is second order in time (one order higher than the expected order). We must notice that the usual stability conditions that apply in incremental solutions are no more concerned.

Finally we are summarizing the size of the problems to be solved as well as the memory needs associated with the application of the proposed strategy.

In the solution procedure one needs solve J fully populated linear systems of size J (J being of the order of ten) as well as J non-linear sparse systems of size $N \times n_n$ within an iterative procedure. In the case of using an alternating directions scheme at

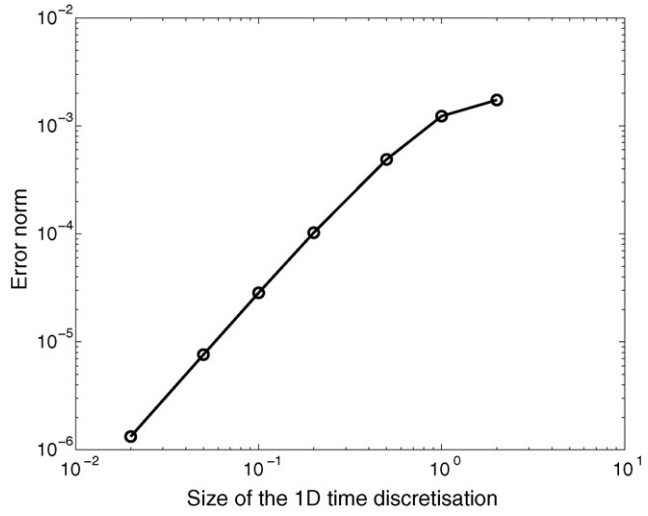

Fig. 5. Convergence analysis of time discretization.

each iteration of the non-linear solver N linear tri-diagonal systems of size n_n must be solved. It can be noticed that the problem complexity scales with $N \times n_n$. From the point of view of the memory needs, $N \times n_n \times J$ double precision numbers are needed to represent the solution, that is, the J products of N functions each one represented from its n_n nodal values.

4. Application to the time dependent 1D FENE dumbbell model

4.1. Model equations

We consider here the dimensionless form of the 1D FENE problem

$$\frac{\partial \Psi}{\partial t} = -\frac{\partial}{\partial q}\left[\left(\text{Grad}\, v\, q - \frac{1}{2}h(q)q\right)\Psi\right] + \frac{1}{2}\frac{\partial^2 \Psi}{\partial q^2}, \tag{56}$$

where $\Psi = \Psi(x, q, t)$, $v = v(x, t)$ and q is the dimensionless connector vector and the function h is given by

$$h(q) = \frac{1}{1 - q^2/b}, \tag{57}$$

where b is the dimensionless square of the maximum molecule stretch (see [3] for details). In addition, a normality condition is associated to Eq. (56)

$$\int_{\Omega_q} \Psi(q, t)\, \mathrm{d}q = 1, \quad \forall t, \tag{58}$$

where Ω_q refers to the configuration space. Finally, the link between the statistical distribution of dumbbell configurations and the polymer stress τ_p is provided, for the 1D molecular configuration considered in this section, by

$$\tau_p = \langle h(q)q^2 \rangle - 1 = \int \Psi(q)h(q)q^2\, \mathrm{d}q - 1. \tag{59}$$

Remark 3.

(i) Eq. (56) defines the time evolution of the molecule distribution function. An initial distribution must be prescribed. The equilibrium state Ψ_0 is defined for a null velocity gradient, being then solution of

$$0 = -\frac{\partial}{\partial q}\left[\left(-\frac{1}{2}h(q)q\right)\Psi_0\right] + \frac{1}{2}\frac{\partial^2 \Psi_0}{\partial q^2}, \tag{60}$$

given by

$$\Psi_0(q) = \frac{h(q)^{-b/2}}{\int h(q)^{-b/2}\, \mathrm{d}q}. \tag{61}$$

Thus, a natural initial condition consists of $\Psi(q, t = 0) = \Psi_0(q)$.

(ii) The derivative $\partial\Psi/\partial t$ refers to the material derivative. In this paper, the proposed method will be illustrated in homogeneous flows; the material derivative thus reduces to the partial derivative.

4.2. Problem discretization

In order to solve this problem using a separated representation, we can consider that the problem is defined in a 2D space, where the first dimension corresponds to the molecular stretch q and the second one to the time t, leading to $\Psi(q, t)$ with $(q, t) \in \Omega_q \times \Omega_t = \Omega =]-\sqrt{b}, \sqrt{b}[\times]0, t_{\max}]$. The use of the variable separation strategy requires null boundary and initial conditions. The molecular conformation distribution Ψ verifies naturally $\Psi(q = \pm\sqrt{b}, t) = 0$, however the initial condition cannot be set to zero. If for example we assume a system evolving from the equilibrium state, the initial condition is given by: $\Psi(q, t = 0) = \Psi_0(q)$. In order to define a problem with homogeneous boundary and initial conditions, we introduce the following change of variable:

$$\Psi(q, t) = \psi(q, t) + \Psi_0(q), \tag{62}$$

where $\psi(q = \pm\sqrt{b}, t) = 0$ and $\psi(q, t = 0) = 0$.

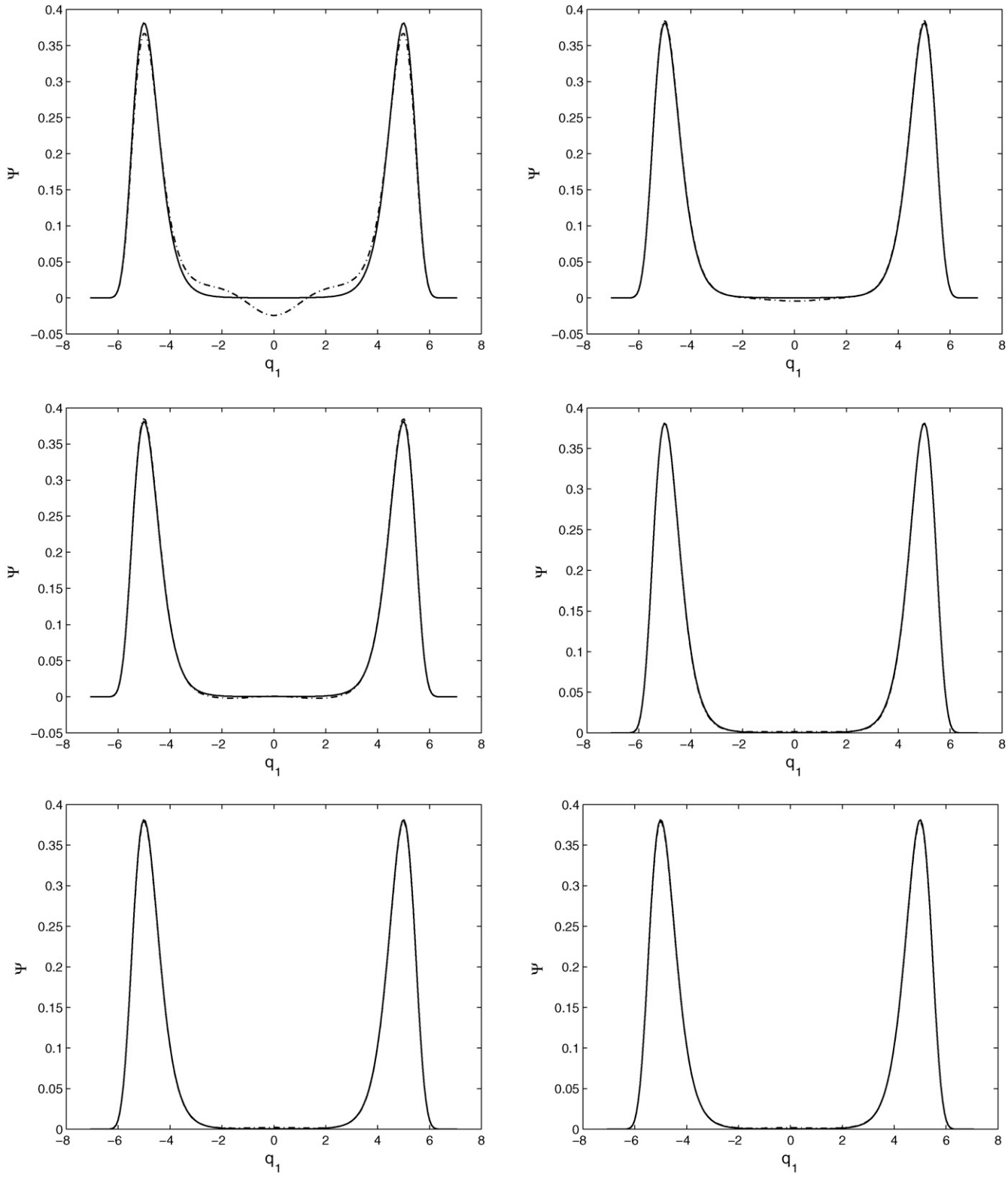

Fig. 6. Difference between computed solution at $t = t_{\max}$ and analytical steady state solution: (top-left) after one enrichment; (top-right) two enrichments; (middle-left) three enrichments; (middle-right) four enrichments; (bottom-left) five enrichments and (bottom-right) six enrichments.

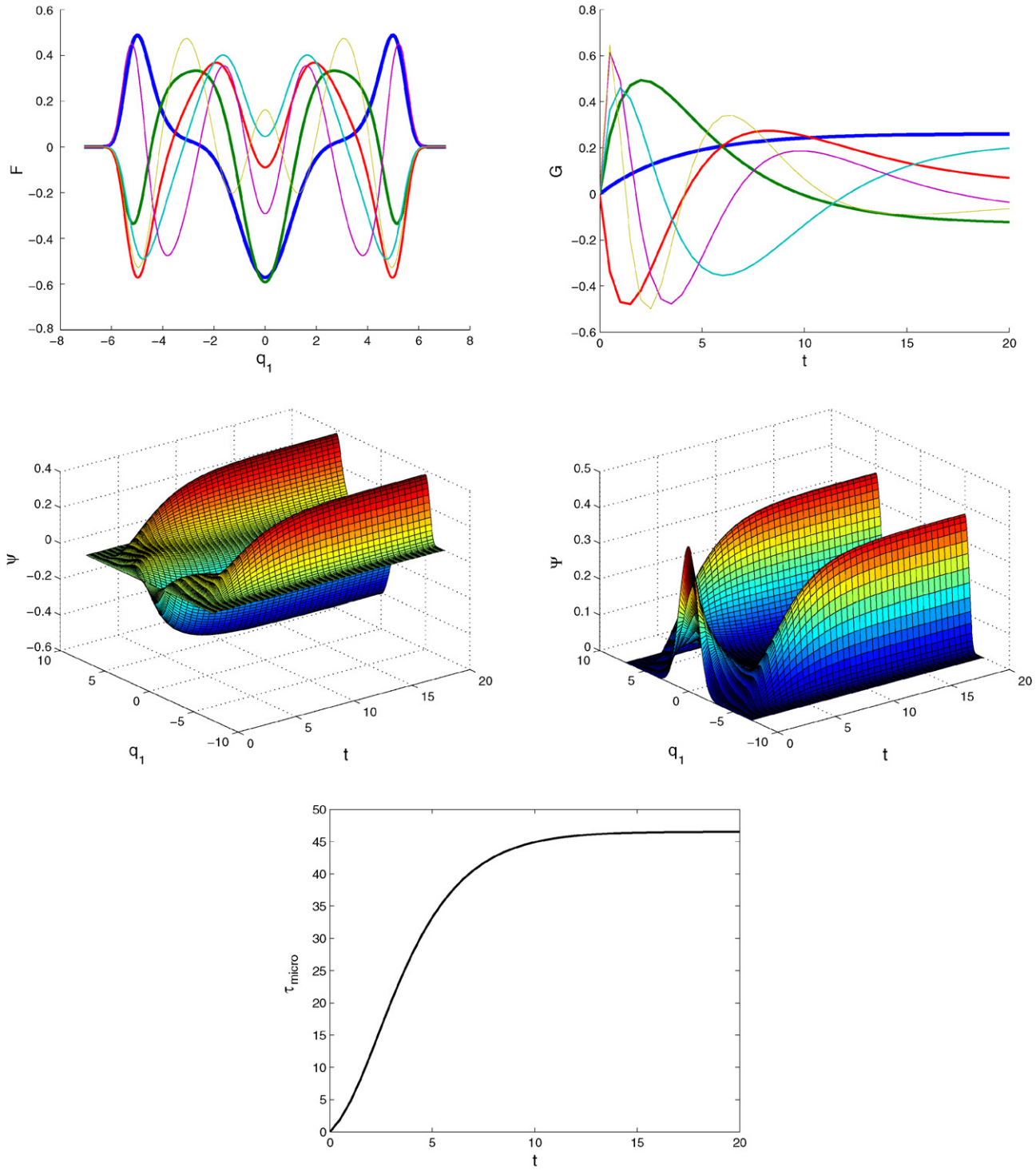

Fig. 7. 1D FENE model consisting of one spring ($N = 1$), $We = 1$ and $b = 50$.

The introduction of Eq. (62) into the Fokker–Planck Eq. (56) results in:

$$\frac{\partial \psi}{\partial t} + f(q)\psi + e(q)\frac{\partial \psi}{\partial q} - \frac{1}{2}\frac{\partial^2 \psi}{\partial q^2} = -f(q)\Psi_0 - e(q)\frac{\partial \Psi_0}{\partial q} + \frac{1}{2}\frac{\partial^2 \Psi_0}{\partial q^2}, \tag{63}$$

where

$$f(q) = \frac{\partial}{\partial q}\left(\mathrm{Grad}\, v\, q - \frac{1}{2}h(q)q\right),$$
$$e(q) = \left(\mathrm{Grad}\, v\, q - \frac{1}{2}h(q)q\right).$$

The weighted residual formulation (before integrating by parts) is given by:

$$\int_\Omega \psi^* \frac{\partial \psi}{\partial t}\, \mathrm{d}\Omega + \int_\Omega \psi^* f(q)\psi\, \mathrm{d}\Omega + \int_\Omega \psi^* e(q)\frac{\partial \psi}{\partial q}\, \mathrm{d}\Omega - \int_\Omega \frac{1}{2}\psi^* \frac{\partial^2 \psi}{\partial q^2}\, \mathrm{d}\Omega = \sum_{h=1}^{3} \int_\Omega \psi^* a_h(q)\, \mathrm{d}\Omega, \tag{64}$$

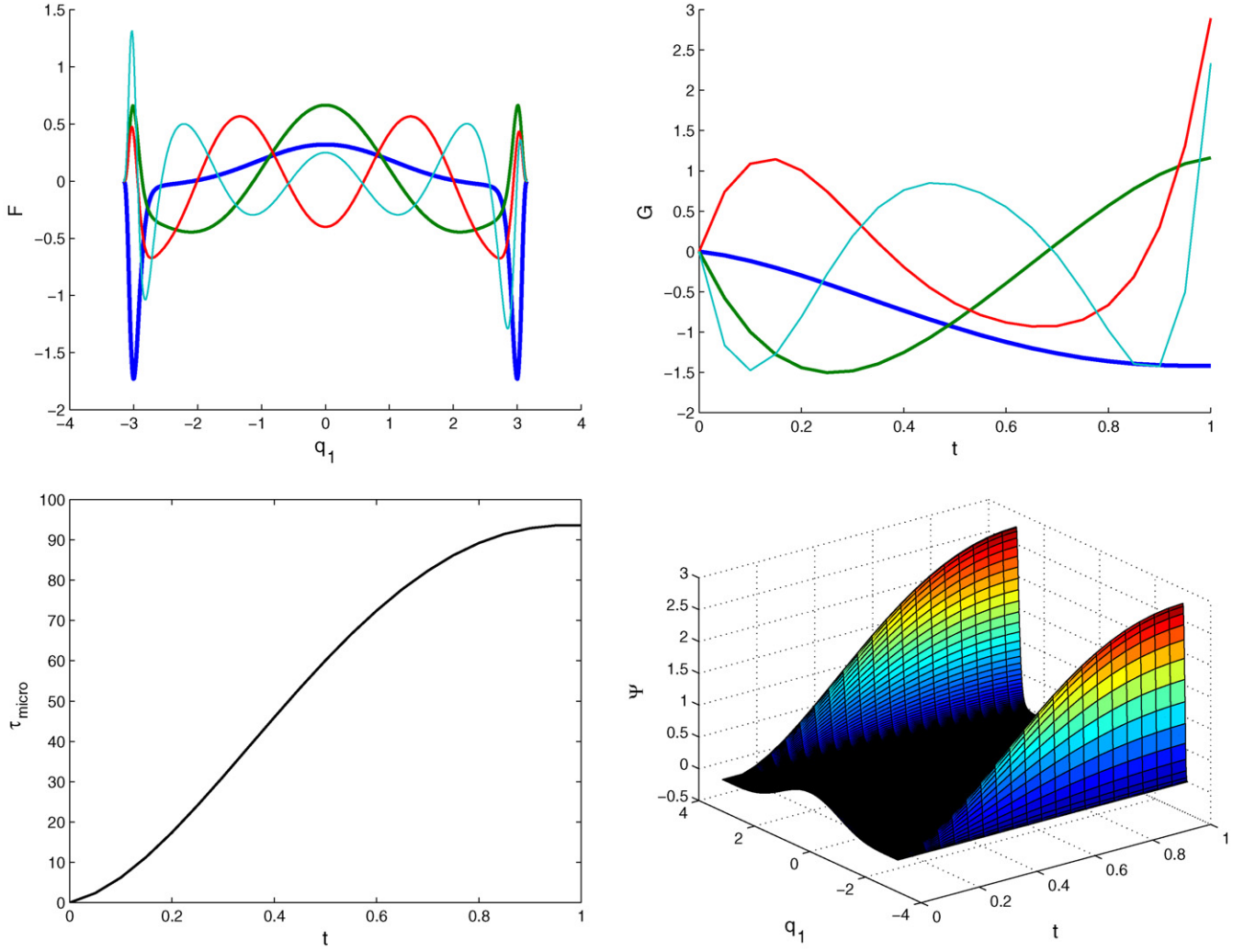

Fig. 8. 1D FENE model consisting of one spring ($N = 1$), $We = 5$ and $b = 10$.

where functions $a_h(q)$ are defined by:

$$\begin{cases} a_1(q) = -\Psi_0(q) f(q), \\ a_2(q) = -\dfrac{\partial \Psi_0(q)}{\partial q} e(q), \\ a_3(q) = \dfrac{1}{2} \dfrac{\partial^2 \Psi_0(q)}{\partial q^2}. \end{cases} \tag{65}$$

Integrating by parts (with respect to q) the last term of the left-hand-side of Eq. (64), we obtain:

$$\int_\Omega \psi^* \frac{\partial \psi}{\partial t} \, \mathrm{d}\Omega + \int_\Omega \psi^* f(q) \psi \, \mathrm{d}\Omega + \int_\Omega \psi^* e(q) \frac{\partial \psi}{\partial q} \, \mathrm{d}\Omega + \int_\Omega \frac{1}{2} \frac{\partial \psi^*}{\partial q} \frac{\partial \psi}{\partial q} \, \mathrm{d}\Omega = \sum_{h=1}^{3} \int_\Omega \psi^* a_h(q) \, \mathrm{d}\Omega, \tag{66}$$

where the nullity of ψ^* on $\partial\Omega_q$ ($\psi^*(q = \pm\sqrt{b}, t)$) has been accounted.

Due to both, the parabolic and convective character of Eq. (66), suitable stabilizations are required. For this purpose, we introduce an extra-diffusion to stabilize the advection term by analogy with the usual SUPG procedure (that was also considered in [3]) as well as the time derivative upwinding previously described in Section 3.2.2 for accounting for the parabolic character of the Fokker–Planck equation:

$$\int_\Omega \tilde{\psi}^* \frac{\partial \psi}{\partial t} \, \mathrm{d}\Omega + \int_\Omega \psi^* f(q) \psi \, \mathrm{d}\Omega + \int_\Omega \psi^* e(q) \frac{\partial \psi}{\partial q} \, \mathrm{d}\Omega + \int_\Omega \frac{\beta e(q) h_q}{2} \frac{\partial \psi^*}{\partial q} \frac{\partial \psi}{\partial q} \, \mathrm{d}\Omega + \int_\Omega \frac{1}{2} \frac{\partial \psi^*}{\partial q} \frac{\partial \psi}{\partial q} \, \mathrm{d}\Omega = \sum_{h=1}^{3} \int_\Omega \psi^* a_h(q) \, \mathrm{d}\Omega, \tag{67}$$

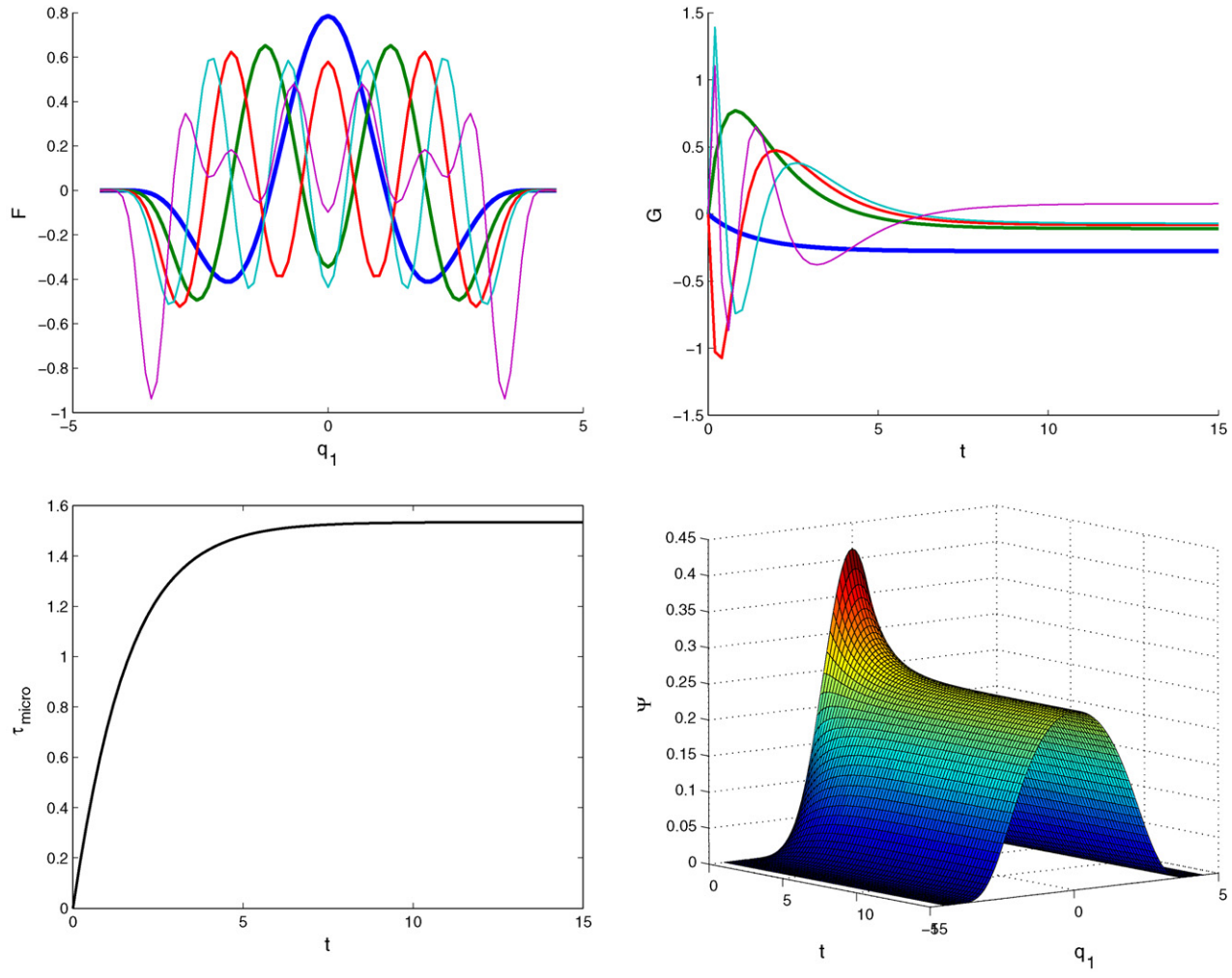

Fig. 9. MBS model consisting of $N = 1$ spring, $We = 0.4\sqrt{N}$ and $b = 20/N$.

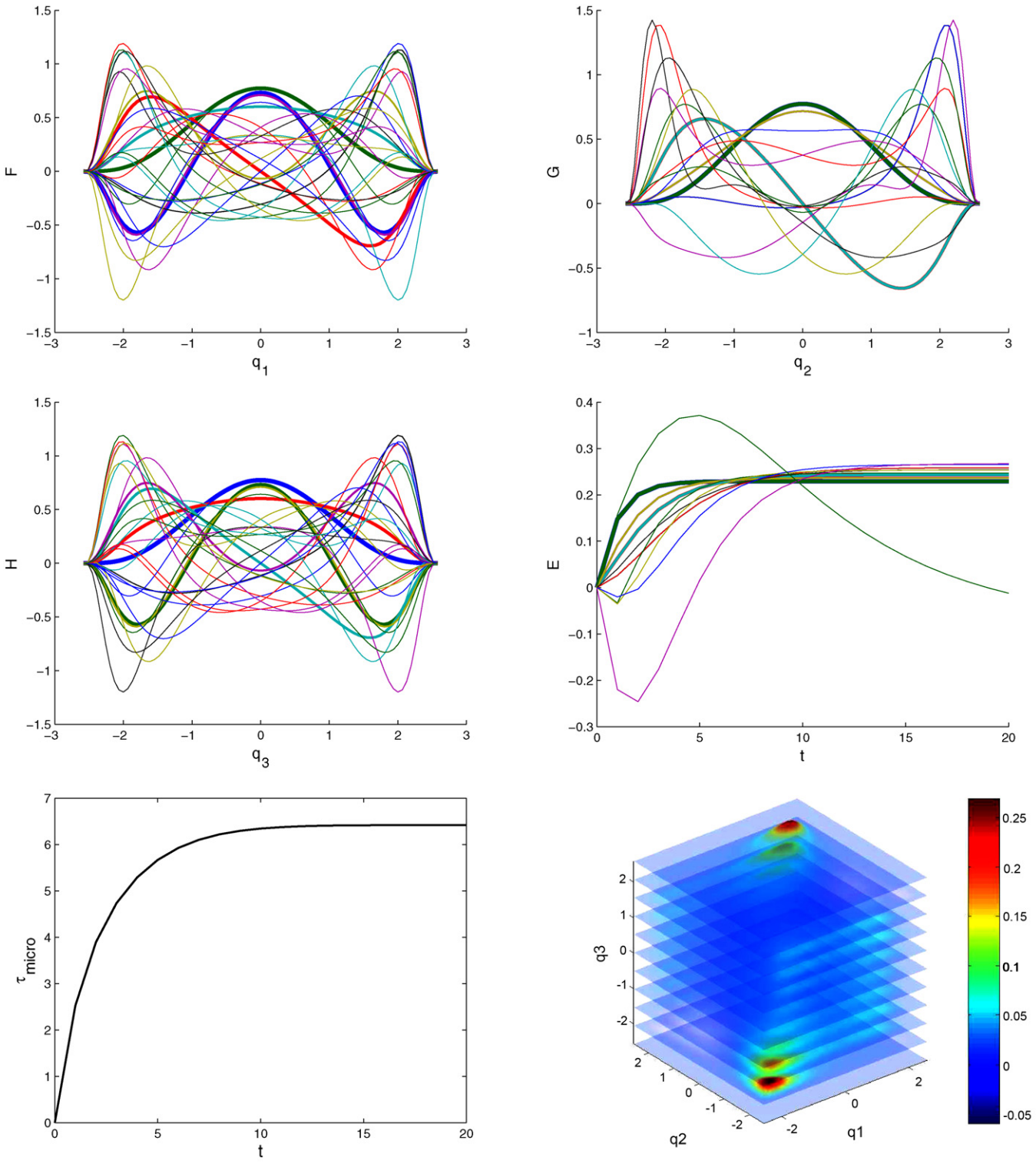

Fig. 10. MBS model consisting of $N = 3$ springs, $We = 0.4\sqrt{N}$ and $b = 20/N$.

where h_q is the size of discretization in the configuration space and β is calculated from

$$\beta = \coth(Pe) - \frac{1}{Pe}. \tag{68}$$

Here the Peclet number defined by

$$Pe = e(q)h_q. \tag{69}$$

The solution is now searched in the form:

$$\psi(q,t) = \sum_{j=1}^{\infty} \alpha_j F_j(q) G_j(t), \tag{70}$$

using the procedure described previously. The resulting non-linear system

$$\underline{U}(\underline{R},\underline{S}) + \underline{\underline{K}}(\underline{R},\underline{S}) \begin{bmatrix} \underline{R} \\ \underline{S} \end{bmatrix} = \underline{V}(\underline{R},\underline{S}), \tag{71}$$

must be solved subject to the normality of Ψ (Eq. (58)), which implies

$$\int_{\Omega_q} \psi(q,t)\,\mathrm{d}q = 0, \quad \forall t, \tag{72}$$

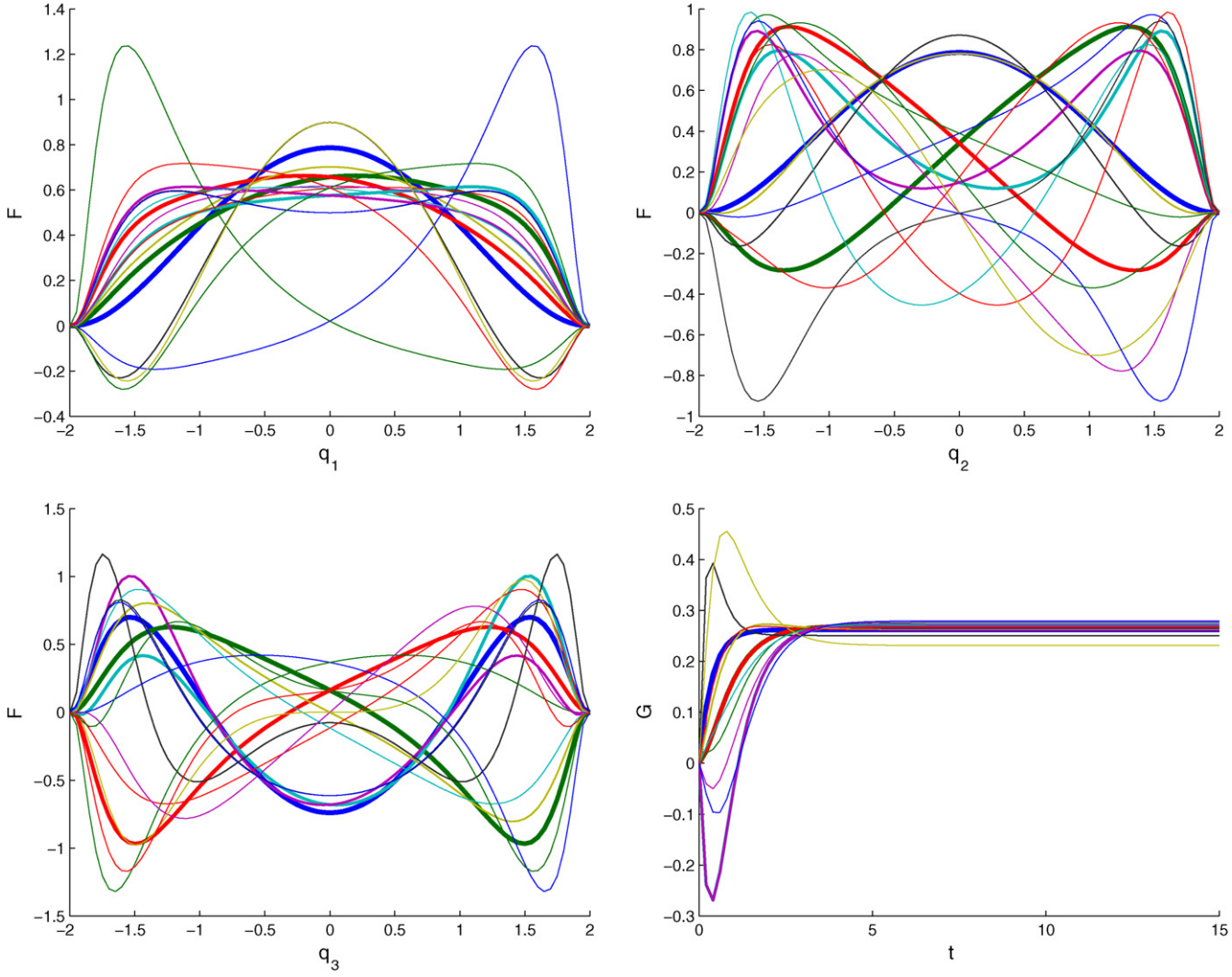

Fig. 11. MBS model consisting of $N = 5$ springs, $We = 0.4\sqrt{N}$ and $b = 20/N$.

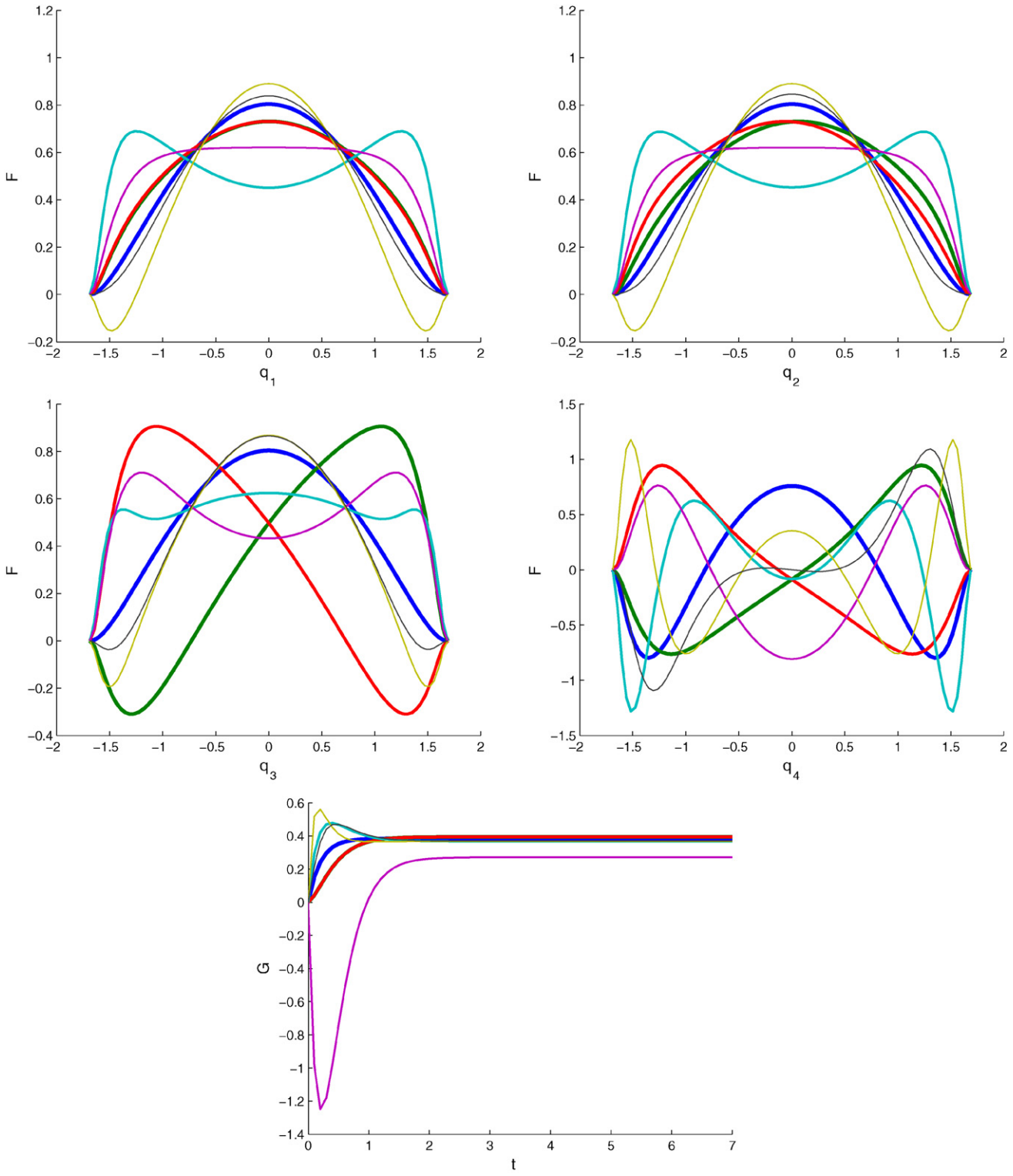

Fig. 12. MBS model consisting of $N = 7$ springs, $We = 0.4\sqrt{N}$ and $b = 20/N$.

that is

$$\int_{q=-\sqrt{b}}^{q=\sqrt{b}} \psi(q,t)\,\mathrm{d}q = 0, \quad \forall t \in]0, t_{\max}], \tag{73}$$

which is immediately verified by imposing

$$\int_{q=-\sqrt{b}}^{q=\sqrt{b}} R(q)\,\mathrm{d}q = 0, \tag{74}$$

using a Lagrange multiplier.

4.3. Extension to 1D multi-bead-spring models

The technique described above can be applied for simulating the Multi-Bead-Spring models of polymer chains. The MBS chain consists of $N+1$ beads connected by N springs. The bead serves as an interaction point with the solvent and the spring contains the local stiffness information depending on local stretching (see [5] for more details). In the case of a 1D physical description (all springs are located in a 1D physical space), the problem is described in a N-dimensional configuration space and time by the following dimensionless Fokker–Planck equation:

$$\frac{\partial \Psi}{\partial t} + \frac{\partial}{\partial \underline{q}}[(\mathrm{Grad}\, v\, \underline{q} - \underline{\underline{A}}\underline{g}(\underline{q}))\Psi] - \frac{1}{4}\frac{\partial}{\partial \underline{q}}\left(\underline{\underline{A}}\frac{\partial \Psi}{\partial \underline{q}}\right) = 0. \tag{75}$$

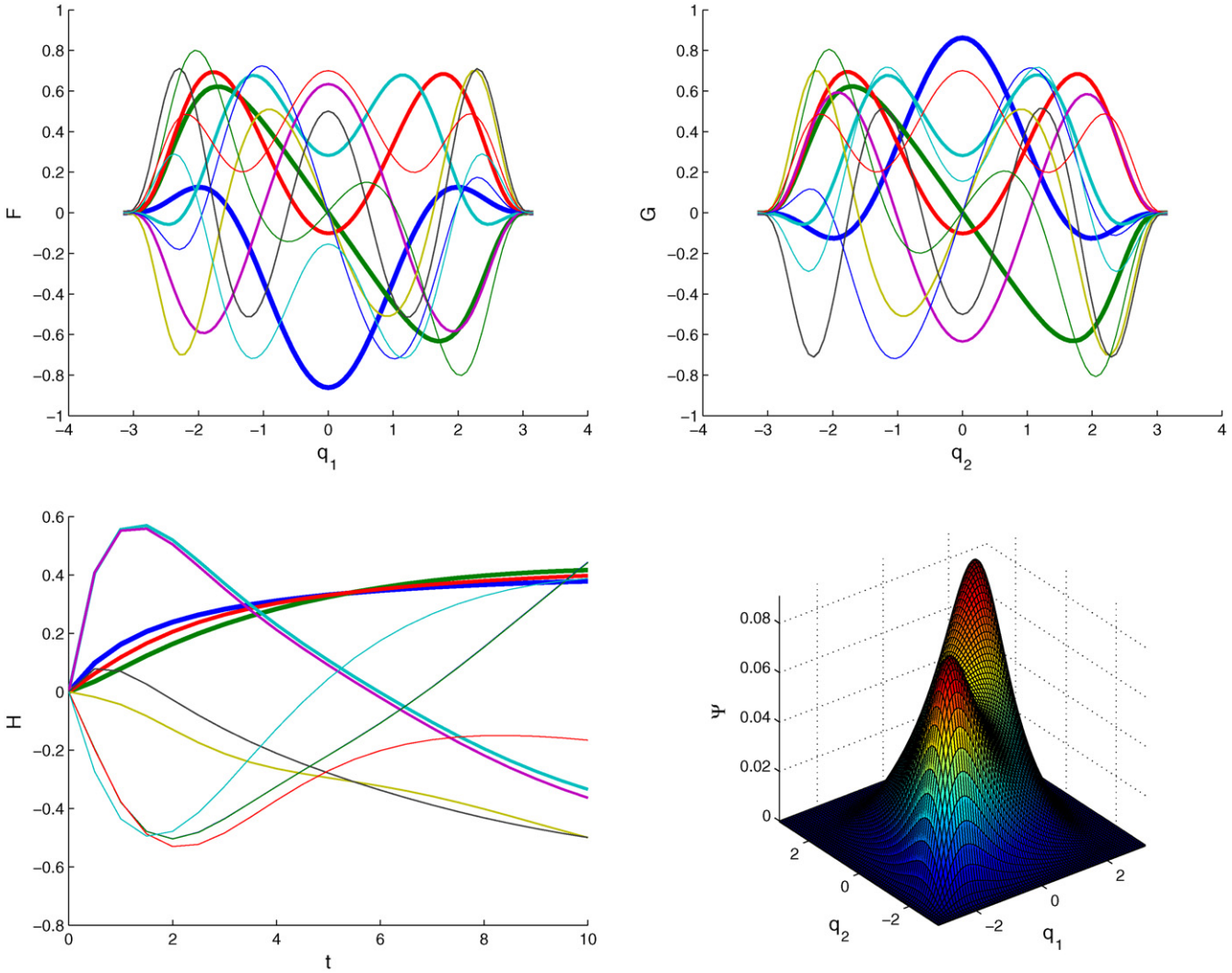

Fig. 13. MBS model consisting of $N = 2$ springs, validation of the 2D case.

Here q is the configuration vector $q = (q_1, q_2, \ldots, q_N)$ and the components of g are given by:

$$g_k = \tfrac{1}{4}\, q_k\, h(q_k), \tag{76}$$

where $\underline{\underline{A}}$ is the matrix

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

Eq. (75) can be solved using a separated representation.

5. Numerical results

The first example concerns a transient FENE problem consisting of a single dumbbell ($N = 1$) with $We = \mathrm{d}u/\mathrm{d}x = 1$, $b = 50$ discretized by using 200 nodes in the configuration space and 40 in the time interval $]0, t_{\max} = 20]$ (at $t_{\max}$ the steady state is almost reached). Fig. 6 shows the influence of the enrichment on the computed solution at $t_{\max}$.

Fig. 7 shows the involved functions allowing to reconstruct the distribution function Ψ after adding ψ_0 to the solution ψ. The polymer stress can then be calculated as a function of time using Eq. (59).

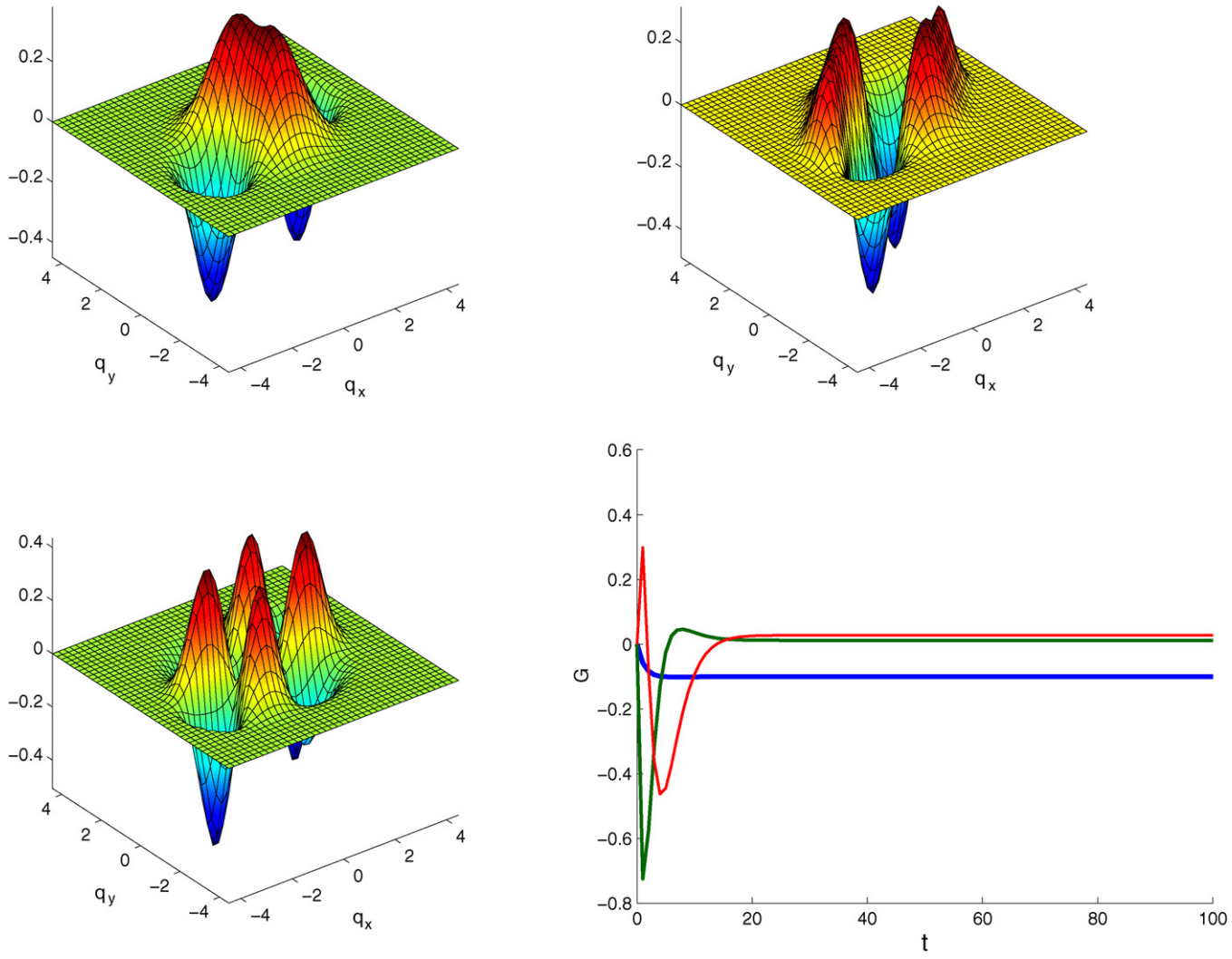

Fig. 14. Transient 2D FENE model. Space and time reduced approximation functions: (top-left) first space approximation function; (top-right) second space approximation function; (bottom-left) third space approximation function and (bottom-right) time functions.

In the second example (Fig. 8), we perform the simulation reported in [2]. The sharpness of the space evolution can be accurately reproduced using the proposed separated representation. In this example $We = 5$, $b = 10$, $t_{\max} = 1$ and we use 600 and 20 elements in the configuration and time coordinates, respectively.

Figs. 9–12 depict the solutions related to the multi-dumbbell models defined by $We = 0.4\sqrt{N}$, $b = 20/N$ for $N = 1, 3, 5$ and 7, respectively. The case of N = 2 can be observed in Fig. 13 and compared to the steady state simulation done in [3].

When richer physical spaces are considered, two difficulties are found: (i) the molecular conformation function is defined in a bounded domain (not necessary a hyper-cube); and (ii) in the case of MBS (multi-bead-spring) models the advection terms related to each spring cannot be separated as a product of terms involving each one of the physical space directions. In this case, a suitable approximation consists of:

$$\psi(\underline{q}_1, \ldots, \underline{q}_S, t) = \sum_{j=1}^{n} \alpha_j (F_{1j}(\underline{q}_1) \times \cdots \times F_{Nj}(\underline{q}_S) \times G_j(t)), \tag{77}$$

where S is the number of spring connectors. The approximation of each function F is built in a space of the same dimension than that of the physical space, that is, in the ball of radius $\sqrt{b}$. These functions can be accurately represented using spherical coordinates or rectangular coordinates combined with a penalty technique as considered in [3].

This technique was successfully applied in [3] for solving the steady state of multidimensional chain models involving both orientation and extension in the conformation space. We must mention that the direct computation of the steady state is an issue for stochastic simulation techniques. However, the deterministic technique based on a separated representation can be applied for solving both transient and steady state problems.

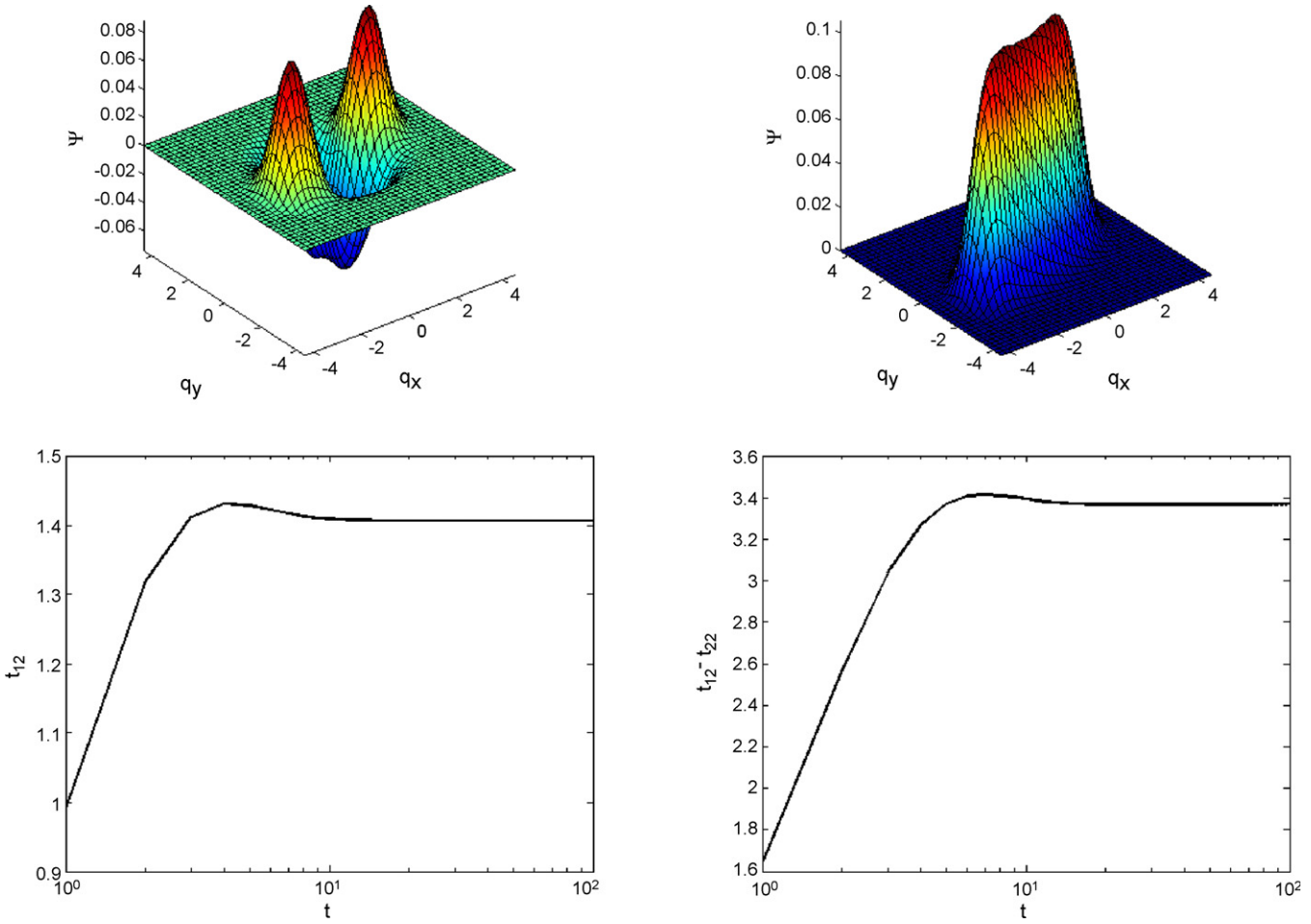

Fig. 15. Transient 2D FENE model: (top-left) final distribution function before introducing the initial condition; (top-right) real final distribution function; (bottom-left) shear stress evolution and (bottom-right) evolution of the first normal stress difference.

The last example that we are considering in the present work concerns the transient 2D FENE model in a starting shear flows. The problem is characterized by the following parameters $We = 2$ and $b = 20$. We consider the simple separated representation

$$\psi(\underline{q}, t) = \sum_{j=1}^{J} \alpha_j (F_j(\underline{q}) \times G_j(t)). \tag{78}$$

Functions $F_j(\underline{q})$ are approximated on a coarse mesh consisting of 1600 nodes. The accurate approximation of the distribution function only requires of 3 space-time functions $F_j(\underline{q}) \times G_j(t)$, allowing us to write:

$$\psi(\underline{q}, t) = \sum_{j=1}^{3} \alpha_j (F_j(\underline{q}) \times G_j(t)). \tag{79}$$

Fig. 14 depicts the three $2D$ space functions $F_j(\underline{q})$ as well as the three associated time functions.

Finally, the distribution function ψ can be reconstructed by multiplying each space function by its corresponding time function, and then performing the sum of these products. The actual distribution function is then computed from $\Psi = \psi + \Psi_0$, from which one computes the moments defining the stresses.

Fig. 15 depicts the reconstructed distribution functions at the final time $\psi(\underline{q}, t = t_{\max})$ and $\Psi(\underline{q}, t = t_{\max})$, as well as the evolution of both the shear stress and the first normal stress difference.

6. Conclusions and perspectives

This paper proves the ability of the proposed strategy to compute transient solutions of multidimensional parabolic PDE's with homogeneous boundary conditions admitting a separated representation.

The novelty of the proposed technique justifies that many questions remain open: (i) the treatment of non-linear Fokker–Planck equations; (ii) optimal basis enrichment; (iii) analysis of complex flows involving non-homogeneous solution in the physical space; (iv) general initial conditions; (v) convergence analysis; (vi) stabilization of advection operators; . . ..

The first point concerning the treatment of non-linear Fokker–Planck equations does not involve major difficulties. One could proceed by linearizing, and then apply the procedures proposed in [3] in the steady case, or the one just proposed for transient simulations. In a recent work [12] the separated representation was applied for treating the non-linear Fokker–Planck models related to advanced reptation models of polymer melts.

The second point concerns the optimal choice of the number of degrees of freedom to be used in the discretization of each dimension. The use of a wavelet approximation, could offer an optimal refinement via the multiresolution properties of wavelets (additional degrees of freedom are only required in regions where the wavelets coefficients are not small enough).

This procedure can be applied easily in complex flows simulations involving non-homogenous solutions in the physical space, by considering a micro–macro approach in the context of a splitting solution of the FP equation (see review [10]). For this purpose we consider that each node used in the discretization of the physical domain moves along its flow trajectory, allowing the integration of the advection operator of the Fokker–Planck equation using the method of characteristics. Then, from the just computed distribution, the conformation part of the FP equation is solved using the procedure described here during the time step used in the flow kinematics solution. The final distribution allows the calculation of the extra-stress tensor at each nodal position that will be used in the flow kinematics updating. The only remaining difficulty lies in the fact that nodes are assumed moving with the flow, which leads to highly distorted elements if one uses the finite element method (FEM) for solving momentum and mass balance equations. One possibility to circumvent this difficulty consists in using a meshless natural element method (NEM) [11] instead usual finite elements, because the accuracy of the NEM does not require any geometrical quality of the nodal distribution. Other fixed mesh techniques can be used, as the ones based on the characteristics-Galerkin scheme.

In the case of general initial conditions, we can compute a separated representation of that initial condition (as the sum of a product of terms involving the different coordinates) using for example an alternating least squares technique [4].

At present the stabilization of the advection operators has been performed by considering a SUPG technique. The results obtained in 2D or 3D simulations have been compared with the ones computed with more usual stabilized techniques in finite elements of finite differences, proving the ability of the proposed stabilization for accounting dominant advection effects. However, a deeper analysis remains to be performed.

References

[1] Y. Achdou, O. Pironneau, Computational methods for option pricing, Siam Front. Appl. Math. (2005).

[2] A. Ammar, D. Ryckelynck, F. Chinesta, R. Keunings, On the Reduction of Kinetic Theory Models Related to Finitely Extensible Dumbbells, J. Non-Newtonian Fluid Mech. 134 (2006) 136–147.

[3] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, J. Non-Newtonian Fluid Mech. 139 (2006) 153–176.

[4] G. Beylkin, M. Mohlenkamp, Algorithms for numerical analysis in high dimensions, SIAM J. Sci. Commun. 26/6 (2005) 2133–2159.
[5] R.B. Bird, C.F. Crutiss, R.C. Armstrong, O. Hassager, Dynamic of Polymeric Liquid, vol. 2: Kinetic Theory, John Wiley and Sons, 1987.
[6] H.J. Bungartz, M. Griebel, Sparse grids, Acta Numer. 13 (2004) 1–123.
[7] B. Cochelin, N. Damil, M. Potier-Ferry, The asymptotic numerical method: an efficient perturbation technique for non-linear structural mechanics, Eur. J. Finite Elements 3 (1994) 281–297.
[8] P. Delaunay, A. Lozinski, R.G. Owens, Sparse tensor-product Fokker–Planck-based methods for nonlinear bead-spring chain models of dilute polymer solutions, in: Workshop on High-Dimensional Partial Differential Equations in Science and Engineering, CRM Proceedings and Lecture Notes, Montral, 2005.
[9] T. Gerstner, M. Griebel, Dimension-Adaptive Tensor-Product Quadrature, Computing 71/1 (2003) 65–87.
[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] M.A. Martinez, E. Cueto, M. Doblaré, F. Chinesta, Natural element meshless simulation of injection processes involving short fiber suspensions, J. Non-Newtonian Fluid Mech. 115 (2003) 51–78.
[12] B. Mokdad, E. Pruliere, A. Ammar, F. Chinesta, On the simulation of kinetic theory models of complex fluids using the Fokker–Planck approach, Appl. Rheol., in press.
[13] T. von Petersdorff, C. Schwab, Numerical solution of parabolic equations in high dimensions, Math. Model. Numer. Anal., Isaac Newton Institute for Mathematical Sciences, 38, 2004, pp. 93–127.