Journal of Computational Physics 231 (2012) 5016–5028

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Efficient spectral-Galerkin methods for systems of coupled second-order equations and their applications ☆

Feng Chen, Jie Shen *

Department of Mathematics, Purdue University, West Lafayette, IN 47907-1957, USA

ARTICLE INFO

Article history:
Received 23 November 2011
Received in revised form 16 March 2012
Accepted 20 March 2012
Available online 3 May 2012

Keywords:
Spectral-Galerkin method
System of coupled second-order equations
Cahn–Hilliard equations
High-order equations

ABSTRACT

We construct in this paper two efficient spectral-Galerkin algorithms for solving systems of n coupled second-order equations. The computational complexity of these algorithms is essentially n times the cost of solving one second-order equation. We present numerical results which illustrate the accuracy and flexibility of these algorithms, as well as several interesting and challenging applications, including in particular a number of high-order nonlinear parabolic type equations.

1. Introduction

Many mathematical models for scientific and engineering applications involve linear or nonlinear high-order equations/coupled systems of second-order equations. Examples include high-order Cahn–Hilliard systems in phase-field crystals (cf. [6,26]) and in polymer electrolyte membrane (PEM) in fuel cells (cf. [15]), phase-field models for strongly anisotropic systems (cf. [16,25]), rotating flows in geophysical fluid dynamics (cf. [13]), nonlinear Schrödinger equations and systems of reaction–diffusion equations. How to solve these equations efficiently and accurately present a great challenge, as they usually lead to very ill-conditioned systems due to the high-order derivatives and/or the coupling.

If the problem is of parabolic type, then under a suitable semi-implicit time discretization, these problems can often be reduced to solving a system of coupled second-order equations at each time step. On the other hand, if the problem is independent of time, a suitable iterative method may also lead to a system of coupled second-order equations at each iteration. Therefore, it is important to develop efficient and accurate algorithms for solving systems of coupled second-order equations. In this paper, we shall consider a class of such problems which are set in separable geometries with decoupled boundary conditions. We note that for problems with periodic boundary conditions in all directions or in all but one direction, these systems are easy to solve by using a Fourier spectral method (cf. [7,4]) in the periodic directions. Hence, we shall focus on the problems with non-periodic boundary conditions in all directions. While there are general software packages based on finite-elements available for solving such systems in a general setting, these software packages usually do not take into account the special geometries, and their accuracy is usually of limited lower-order.

☆ This work is partially supported by NSF Grant DMS-0915066 and AFOSR Grant FA9550-11-1-0328.
* Corresponding author
E-mail addresses: chen221@math.purdue.edu (F. Chen), shen@math.purdue.edu (J. Shen).

0021-9991/$ - see front matter © 2012 Published by Elsevier Inc.
http://dx.doi.org/10.1016/j.jcp.2012.03.001

While efficient spectral methods have been developed for multi-dimensional second- and fourth-order equations (cf. [17,18,3]) and for one-dimensional high-order equations (cf. [9]), not much efforts have been devoted to systems of second-order equations or high-order equations in multi-dimensions.

We shall construct two efficient spectral-Galerkin algorithms based on *a discrete separation of variable procedure* for solving systems of n coupled second-order equations. The first one is the full diagonalization method [12,11] which is easy to implement, while the second one is the so called partial diagonalization method [10,17] which is slightly more complicated to implement but is more efficient and can be applied to a wider class of problems. The computational complexity of these algorithms is CN^{d+1} (where d is the dimension, and C is a small number) which consists of mainly several matrix–matrix multiplications so that it is very efficient on current computer architectures. We note in particular that its computational complexity is essentially n times the cost of solving one second-order equation with the spectral-Galerkin method. Hence, these algorithms are extremely efficient and accurate for solving systems of coupled second-order equations.

The rest of the paper is organized as follows. In the next section, we recall some implementation detail of the Legendre–Galerkin method in the one-dimensional case. In Section 3, we construct the two spectral-Galerkin algorithms for a system of second-order equations in 2-D. We then discuss in Section 4 extensions of these two algorithms to various other situations, and present numerical results which illustrate the accuracy and flexibility of these algorithms. We present in Section 5 several interesting and challenging applications which can be solved by using the new algorithms. We conclude with several remarks in the last section.

2. Preliminaries

In this section, we briefly recall the Legendre–Galerkin method for 1-D Poisson type equations (cf. [17,21]). Our aim here is to introduce some basic notations and results that shall be used in later sections.

Consider

$$\alpha u - u'' = f, \quad \text{in } \Omega = (-1,1), \quad \mathcal{B}u|_{\pm 1} := (a_{\pm}u + b_{\pm}u_x)|_{\pm 1} = 0. \tag{2.1}$$

We assume that $a_{\pm}$ and $b_{\pm}$ are such that the above equation is well-posed. Examples of the operator $\mathcal{B}$ include the Dirichlet, Neumann, or Robin boundary condition.

Let us denote

$$\mathbb{X}_N = \{u \in \mathbb{P}_N : \mathcal{B}u|_{\pm 1} = 0\}, \tag{2.2}$$

where $\mathbb{P}_N$ is the polynomial space of degree N. Then the Legendre–Galerkin method for solving (2.1) consists of finding $u_N \in \mathbb{X}_N$, s.t.,

$$\alpha(u_N, w) - (u_N'', w) = (I_N f, w), \quad \forall w \in \mathbb{X}_N, \tag{2.3}$$

where $(\cdot,\cdot)$ denotes the L^2 inner product in Ω, and I_N is the interpolation operator based on the Legendre–Gauss–Lobatto points from $C(\Omega)$ to $\mathbb{P}_N$.

Let L_k be the Legendre polynomial of degree k, it is shown (cf. [21]) that there exists a unique pair (a_k,b_k) such that

$$\tilde{\phi}_k = L_k + a_k L_{k+1} + b_k L_{k+2} \in \mathbb{X}_N. \tag{2.4}$$

Consequently, we have $\mathbb{X}_N = \text{span}\{\tilde{\phi}_0, \tilde{\phi}_1, \cdots, \tilde{\phi}_{N-2}\}$. For example, in the case of the homogeneous Dirichlet boundary condition, we have $a_k \equiv 0$ and $b_k \equiv -1$. Thanks to the orthogonal properties of the Legendre polynomials, it is easy to see that $\left(\tilde{\phi}_j'', \tilde{\phi}_i\right) = 0$ for $j < i$. One can easily check by integration by part that $\left(\tilde{\phi}_j'', \tilde{\phi}_i\right) = \left(\tilde{\phi}_j, \tilde{\phi}_i''\right) = 0$ for $j > i$. Hence, by setting

$$\phi_k = d_k \tilde{\phi}_k \text{ with } d_k = \frac{1}{\sqrt{-\left(\tilde{\phi}_k'', \tilde{\phi}_k\right)}}, \quad 0 \leqslant k \leqslant N-2. \tag{2.5}$$

we have $S_{ij} := -\left(\phi_j'', \phi_i\right) = \delta_{ij}$. So the stiffness matrix S is the identity matrix. Obviously, we also have $M_{ij} := (\phi_j,\phi_i) = 0$ if $|i-j| > 2$, so the mass matrix M is symmetric and penta-diagonal.

Some special care is needed for the homogeneous Neumann boundary condition. The stiffness matrix S becomes singular since $\left(\tilde{\phi}_0'', \tilde{\phi}_0\right) = 0$. In that case, we revise the definition of d_k in (2.5) as follows,

$$d_0 = \frac{1}{\sqrt{(\tilde{\phi}_0, \tilde{\phi}_0)}}, \quad d_k = \frac{1}{\sqrt{-\left(\tilde{\phi}_k'', \tilde{\phi}_k\right)}}, \quad 1 \leqslant k \leqslant N-2. \tag{2.6}$$

Let $q = N-2$. Denote

$$u_N = \sum_{i=0}^{q} \hat{u}_i \phi_i, \quad \bar{u} = (\hat{u}_0, \hat{u}_1, \cdots, \hat{u}_q)^T;$$

$$\hat{f}_i = (I_N f, \phi_i), \quad \bar{f} = (\hat{f}_0, \hat{f}_1, \cdots, \hat{f}_q)^T.$$

The linear system (2.3) reduces to

$$(\alpha M + I)\bar{u} = \bar{f}, \tag{2.7}$$

which can be easily solved.

3. Coupled systems of two equations

In this section, we construct two efficient algorithms to solve a coupled system of two second-order equations in a two-dimensional rectangular domain. Extensions to more general cases will be considered in the next section.

3.1. Full diagonalization

Let $(x,y) \in \Omega \triangleq (-1,1)^2$, we consider the following coupled system of two second-order equations:

$$\begin{cases} \alpha u + \beta v - \gamma \Delta u = f, \ \text{in } \Omega, \\ \tilde{\alpha} u + \tilde{\beta} v - \tilde{\gamma} \Delta v = g, \ \text{in } \Omega \end{cases} \tag{3.1}$$

subject to the following homogeneous boundary conditions,

$$\begin{aligned} &\mathcal{B}_x u = \mathcal{B}_x v = 0, \ \text{on } \partial\Omega \cap \{y = \pm 1\}, \\ &\mathcal{B}_y u = \mathcal{B}_y v = 0, \ \text{on } \partial\Omega \cap \{x = \pm 1\}, \end{aligned} \tag{3.2}$$

where $\mathcal{B}_x$ and $\mathcal{B}_y$ are any set of suitable boundary conditions in the x and y directions, respectively. We assumed that the boundary conditions are homogeneous as non-homogeneous boundary conditions can be easily lifted.

We shall construct efficient Legendre–Galerkin methods for solving (3.1) and (3.2). To simplify the notations, we shall use the same number of modes, N, in each direction.

Let $\phi_k(x)$ and $\psi_j(y)$ be in the form of (2.4) such that $\mathcal{B}_x \phi_k|_{x=\pm 1} = 0$ and $\mathcal{B}_y \psi_j|_{y=\pm 1} = 0$, and $-\left(\phi_j'', \phi_i\right) = \delta_{ij}$, and $-\left(\psi_j'', \psi_i\right) = \delta_{ij}$.

We define

$$\mathbb{W}_N = \mathbb{X}_N \otimes \mathbb{Y}_N \text{ with } \mathbb{X}_N = \text{span}\{\phi_k(x), \quad 0 \leqslant k \leqslant q\}, \quad \mathbb{Y}_N = \text{span}\{\psi_j(y), \quad 0 \leqslant j \leqslant q\}. \tag{3.3}$$

Then, the Legendre–Galerkin method for (3.1) with (3.2) is as follows:

$$\begin{cases} \text{find } u_N, v_N \text{ in } \mathbb{W}_N \text{ s.t.}, \\ \alpha(u_N, w_1) + \beta(v_N, w_1) - \gamma(\Delta u_N, w_1) = (\mathcal{I}_N f, w_1), \quad \forall w_1 \in \mathbb{W}_N, \\ \tilde{\alpha}(u_N, w_2) + \tilde{\beta}(v_N, w_2) - \tilde{\gamma}(\Delta v_N, w_2) = (\mathcal{I}_N g, w_2), \quad \forall w_2 \in \mathbb{W}_N, \end{cases} \tag{3.4}$$

where $\mathcal{I}_N = I_N \otimes I_N$ is the interpolation operator from $C(\Omega)$ to $\mathbb{P}_N \otimes \mathbb{P}_N$.

Let $\xi_{kj}(x,y) = \phi_k(x)\psi_j(y)$ and write

$$u_N = \sum_{k=0}^{q}\sum_{j=0}^{q} \tilde{u}_{kj}\xi_{kj}(x,y), \quad v_N = \sum_{k=0}^{q}\sum_{j=0}^{q} \tilde{v}_{kj}\xi_{kj}(x,y). \tag{3.5}$$

Plugging in the above in (3.4) and taking $w_1, w_2 = \xi_{kj}$, we find

$$\begin{cases} \alpha M_x U M_y + \beta M_x V M_y + \gamma(U M_y + M_x U) = F, \\ \tilde{\alpha} M_x U M_y + \tilde{\beta} M_x V M_y + \tilde{\gamma}(V M_y + M_x V) = G, \end{cases} \tag{3.6}$$

where

$$\begin{aligned} &U_{kj} = \tilde{u}_{kj}, \quad (M_x)_{kj} = (\phi_j, \phi_k), \quad F_{kj} = (\mathcal{I}_N f, \xi_{kj}), \\ &V_{kj} = \tilde{v}_{kj}, \quad (M_y)_{kj} = (\psi_j, \psi_k), \quad G_{kj} = (\mathcal{I}_N g, \xi_{kj}). \end{aligned} \tag{3.7}$$

Note that we have used the fact that the mass matrices M_x and M_y are symmetric and that the stiffness matrices are the identity matrix.

The key idea in the full diagonalization procedure is to mimic the procedure of separation of variables in the continuous case. This procedure has been used successfully to solve a single second-order equations under different contexts (cf. [12,11,1,5]).

Let E_x and Λ_x (resp. E_y and Λ_y) be the eigenvectors and eigenvalues of M_x (resp. M_y), i.e.

$$M_x E_x = E_x \Lambda_x, \quad M_y E_y = E_y \Lambda_y, \tag{3.8}$$

where E_x and E_y are ortho-normal matrices, and Λ_x and Λ_y are diagonal matrices. Note that the above eigenvalue problems can be solved very efficiently and stably since M_x and M_y are symmetric, penta-diagonal matrices.

Then, we plug in the following substitutions

$$U = E_x P E_y^T, \quad V = E_x Q E_y^T, \tag{3.9}$$

into (3.6). Multiplying the resultant equations by E_x^T on the left and E_y on the right, and using the fact that $E_x^T E_x = I = E_y^T E_y$, one ends up with

$$\begin{cases} \alpha \Lambda_x P \Lambda_y + \beta \Lambda_x Q \Lambda_y + \gamma(\Lambda_x P + P \Lambda_y) = A, \\ \tilde{\alpha} \Lambda_x P \Lambda_y + \tilde{\beta} \Lambda_x Q \Lambda_y + \tilde{\gamma}(\Lambda_x Q + Q \Lambda_y) = B, \end{cases} \tag{3.10}$$

where

$$A = E_x^T F E_y, \quad B = E_x^T G E_y. \tag{3.11}$$

Because Λ_x and Λ_y are diagonal, (3.10) can be decoupled completely. More precisely, one only needs to solve, for each index (i,j), an 2×2 linear system as follows:

$$\begin{cases} \alpha \lambda_i^x \lambda_j^y p_{ij} + \beta \lambda_i^x \lambda_j^y q_{ij} + \gamma(\lambda_i^x p_{ij} + \lambda_j^y p_{ij}) = a_{ij}, \\ \tilde{\alpha} \lambda_i^x \lambda_j^y p_{ij} + \tilde{\beta} \lambda_i^x \lambda_j^y q_{ij} + \tilde{\gamma}\left(\lambda_i^x q_{ij} + \lambda_j^y q_{ij}\right) = b_{ij}, \end{cases} \tag{3.12}$$

where p_{ij}, q_{ij}, a_{ij}, b_{ij} are entries of P, Q, A, B, and λ_i^x, λ_i^y are diagonal entries of Λ^x, Λ^y.

Once we obtain $\{P,Q\}$ from the above, $\{U,V\}$ can be recovered from (3.9).

We summarize below the algorithm for solving (3.4).

Algorithm 1. Full diagonalization

(1) Precompute and store E_x, Λ_x, E_y, Λ_y in (3.8);
(2) Compute $\{F,G\}$ in (3.6) using the definitions in (3.7) from the values of f, g at the Gauss–Lobatto points;
(3) Compute A,B according to (3.11);
(4) Solve the 2×2 linear system for each (i,j) in (3.12) to obtain $\{P,Q\}$;
(5) Calculate $\{U,V\}$ according to (3.9);
(6) Compute the values of $\{u_N, v_N\}$ at the Gauss–Lobatto points from $\{U,V\}$.

Remark 3.1. The computational complexity of the above algorithm is CN^3 (where C is a small number) which mainly consists of the matrix–matrix multiplications in (3.11) and in (3.9), and the backward and forward Legendre transforms in Steps (2) and (6). In fact, the cost of solving a system of two coupled second-order equations is essentially twice of that for solving one second-order equation by the full diagonalization method.

3.2. Partial diagonalization

The full diagonalization procedure presented above relies on the fact that u and v satisfy the same boundary conditions in each direction. We describe below a partial diagonalization method which is more general and only requires that u and v have the same boundary conditions in one direction. Furthermore, it is also computationally more efficient. We note that the partial diagonalization procedure has been applied to solve a single second-order equation before (cf. [10,17]), we extend below the procedure to a system of two second-order equations.

To fix the idea, we assume that u and v have the same boundary conditions in the x-direction, but may have different boundary conditions in the y-direction. Namely, we consider (3.1) with

$$\begin{aligned} &\mathcal{B}_x u = \mathcal{B}_x v = 0, \ \text{on } \partial\Omega \cap \{y = \pm 1\}, \\ &\mathcal{B}_y^u u = \mathcal{B}_y^v v = 0, \ \text{on } \partial\Omega \cap \{x = \pm 1\}. \end{aligned} \tag{3.13}$$

Let $\phi_k(x)$, $\psi_j^u(y)$ and $\psi_j^v(y)$ be in the form of (2.4) such that $\mathcal{B}_x \phi_k|_{x=\pm 1} = 0$, $\mathcal{B}_y^u \psi_j^u|_{y=\pm 1} = 0$ and $\mathcal{B}_y^v \psi_j^v|_{y=\pm 1} = 0$, and $-\left(\phi_j'', \phi_i\right) = \delta_{ij}$, $-\left(\left(\psi_j^u\right)'', \psi_i^u\right) = \delta_{ij}$ and $-\left(\left(\psi_j^v\right)'', \psi_i^v\right) = \delta_{ij}$.

Let $\mathbb{X}_N = \text{span}\{\phi_k(x),\ 0 \leqslant k \leqslant q\}$, We define

$$\begin{aligned} &\mathbb{W}_N^u = \mathbb{X}_N \otimes \mathbb{Y}_N^u \text{ with } \mathbb{Y}_N^u = \text{span}\left\{\psi_j^u(y),\ 0 \leqslant j \leqslant q\right\}, \\ &\mathbb{W}_N^v = \mathbb{X}_N \otimes \mathbb{Y}_N^v \text{ with } \mathbb{Y}_N^v = \text{span}\left\{\psi_j^v(y),\ 0 \leqslant j \leqslant q\right\}. \end{aligned} \tag{3.14}$$

Then, the Legendre–Galerkin method for (3.1) with (3.13) is as follows:

$$\begin{cases} \text{find } u_N \text{ in } \mathbb{W}_N^u, \quad v_N \text{ in } \mathbb{W}_N^v \text{s.t.}, \\ \alpha(u_N, w_1) + \beta(v_N, w_1) - \gamma(\Delta u_N, w_1) = (\mathcal{I}_N f, w_1), \quad \forall w_1 \in \mathbb{W}_N^u, \\ \tilde{\alpha}(u_N, w_2) + \tilde{\beta}(v_N, w_2) - \tilde{\gamma}(\Delta v_N, w_2) = (\mathcal{I}_N g, w_2), \quad \forall w_2 \in \mathbb{W}_N^v, \end{cases} \tag{3.15}$$

where $\mathcal{I}_N = I_N \otimes I_N$ is the interpolation operator from $C(\Omega)$ to $\mathbb{P}_N \otimes \mathbb{P}_N$.

Writing

$$u_N = \sum_{k=0}^{q}\sum_{j=0}^{q} \tilde{u}_{kj}\phi_k(x)\psi_j^u(y), \quad v_N = \sum_{k=0}^{q}\sum_{j=0}^{q} \tilde{v}_{kj}\phi_k(x)\psi_j^v(y), \tag{3.16}$$

and plugging them in (3.15) with $w_1 = \phi_k(x)\psi_j^u(y)$ and $w_2 = \phi_k(x)\psi_j^v(y)$, we derive

$$\begin{cases} \alpha M_x U M_y^u + \beta M_x V M_y^{vu} + \gamma\left(U M_y^u + M_x U\right) = F, \\ \tilde{\alpha} M_x U M_y^{uv} + \tilde{\beta} M_x V M_y^v + \tilde{\gamma}\left(V M_y^v + M_x V\right) = G, \end{cases} \tag{3.17}$$

where U, V, F, G, M_x are defined in (3.7), and

$$\begin{aligned} &\left(M_y^u\right)_{kj} = \left(\psi_j^u, \psi_k^u\right), \quad \left(M_y^v\right)_{kj} = \left(\psi_j^v, \psi_k^v\right), \\ &\left(M_y^{uv}\right)_{kj} = \left(\psi_j^u, \psi_k^v\right), \quad \left(M_y^{vu}\right)_{kj} = \left(\psi_j^v, \psi_k^u\right). \end{aligned} \tag{3.18}$$

Once again, we let E_x and Λ_x be such that $M_x E_x = E_x \Lambda_x$, and set

$$U = E_x P, \quad V = E_x Q. \tag{3.19}$$

Next, by plugging (3.19) into (3.17), and multiplying the resultant equations by E_x^T from the left, we obtain

$$\begin{cases} \alpha \Lambda_x P M_y^u + \beta \Lambda_x Q M_y^{vu} + \gamma\left(P M_y^u + \Lambda_x P\right) = A, \\ \tilde{\alpha} \Lambda_x P M_y^{uv} + \tilde{\beta} \Lambda_x Q M_y^v + \tilde{\gamma}\left(Q M_y^v + \Lambda_x Q\right) = B, \end{cases} \tag{3.20}$$

where

$$A = E_x^T F, \quad B = E_x^T G. \tag{3.21}$$

Taking the transposes of (3.20) and decoupling the matrix system column by column, one ends up with a sequence of linear systems:

$$\begin{cases} \alpha\lambda_k^x M_y^u \bar{p}_k + \beta\lambda_k^x M_y^{vu} \bar{q}_k + \gamma\left(\lambda_k^x I + M_y^u\right)\bar{p}_k = \bar{a}_k, \\ \tilde{\alpha}\lambda_k^x M_y^{uv} \bar{p}_k + \tilde{\beta}\lambda_k^x M_y^v \bar{q}_k + \tilde{\gamma}\left(\lambda_k^x I + M_y^v\right)\bar{q}_k = \bar{b}_k, \end{cases} \quad k = 0, 1, \ldots, N-2, \tag{3.22}$$

where $\bar{p}_k, \bar{q}_k, \bar{a}_k$ and $\bar{b}_k$ are the vectors consisting the kth row of P, Q, A and B, respectively. Rearranging the above system as

$$\begin{cases} A\bar{p}_k + B\bar{q}_k = \bar{a}_k, \\ C\bar{p}_k + D\bar{q}_k = \bar{b}_k, \end{cases} \quad k = 0, 1, \ldots, N-2, \tag{3.23}$$

where

$$\begin{aligned} &A = (\alpha\lambda_k^x + \gamma)M_y^u + \gamma\lambda_k^x I, \quad B = \beta\lambda_k^x M_y^{vu}, \\ &C = \tilde{\alpha}\lambda_k^x M_y^{uv}, \quad D = \left(\tilde{\beta}\lambda_k^x + \tilde{\gamma}\right)M_y^v + \tilde{\gamma}\lambda_k^x I. \end{aligned} \tag{3.24}$$

In the general case, the above 1-D sparse systems can be solved directly. However, when u and v satisfy the same set of boundary conditions, i.e., $\left\{\psi_j^u = \psi_j^v\right\}$, we have $M_y^u = M_y^v = M_y^{uv} = M_y^{vu}$. In this case, A and C commute, so we can eliminate $\bar{p}_k$ from (3.23) to obtain

$$(-CB + AD)\bar{q}_k = A\bar{b}_k - C\bar{a}_k, \quad k = 0, 1, \ldots, N-2. \tag{3.25}$$

The above system is symmetric banded with nine non-zero diagonals so it can be solved in $O(N)$ operations. Then we recover $\bar{p}_k$ from (3.23). If B and D commute, we solve $\bar{p}_k$ first.

We summarize below the algorithm for solving (3.15).

Algorithm 2. Partial diagonalization

(1) Precompute and store E_x, Λ_x in (3.8);
(2) Compute $\{F,G\}$ in (3.6) using the definition in (3.7) from the values of f, g at the Gauss–Lobatto points;
(3) Compute A, B according to (3.21);
(4) Solve the sequence of 1-D equations (3.22) to obtain $\{P,Q\}$;
(5) Calculate $\{U,V\}$ according to (3.19);
(6) Compute the values of $\{u_N, v_N\}$ at the Gauss–Lobatto points from $\{U,V\}$.

Remark 3.2. The computational complexity of the above algorithm is also CN^3 (where C is a small number), but the number of matrix–matrix multiplications in (3.19) and (3.21) is only half of that in (3.11) and (3.9). Hence, **Algorithm 2** (the partial diagonalization method) is computationally more efficient than **Algorithm 1** (the full diagonalization method). Moreover, the partial diagonalization method can be applied to a larger class of problems since the two unknown functions only need to satisfy the same set of boundary conditions in one direction.

4. Miscellaneous extensions

In the last section, we described in detail two algorithms for solving a system of two second-order equations in a 2-D rectangular domain. We shall present in this section extensions of these algorithms to various other situations.

We emphasize that in order to apply the two algorithms, the boundary conditions can not be arbitrary. In fact, the full diagonalization method only applies to coupled systems with unknown functions satisfying the same set of boundary conditions in all directions, while the partial diagonalization method would work for coupled systems with unknown functions satisfying the same set of boundary conditions in all but one directions. To simplify the presentation, we shall only deal with problems with same set of boundary conditions in all directions below.

4.1. Three Dimensional Case

It is easy to see that the two algorithms presented in the last section can be generalized to three dimensions.

Consider, for example, the system (3.1) in $\Omega = (-1,1)^3$ with u and v satisfy the same set of boundary conditions $\mathcal{B}u|_{\pm 1} = 0$ in all directions. Let $\{\phi_k\}$ be constructed as before satisfying $\mathcal{B}\phi_k|_{\pm 1} = 0$ and $-\left(\phi_j'', \phi_i\right) = \delta_{ij}$. Let us denote

$$u_N = \sum_{i,j,k=0}^{q} \tilde{u}_{ijk}\phi_i(x)\phi_j(y)\phi_k(z), \quad v_N = \sum_{i,j,k=0}^{q} \tilde{v}_{ijk}\phi_i(x)\phi_j(y)\phi_k(z). \tag{4.1}$$

Similar to the two-dimensional case and by using the notation of Kronecker products, we can reduce the Legendre-Galerkin approximation to the system (3.1) in 3-D to:

$$\begin{aligned} &[\alpha M\otimes M\otimes M + \gamma(I\otimes M\otimes M + M\otimes I\otimes M + M\otimes M\otimes I)]\bar{u} + \beta(M\otimes M\otimes M)\bar{v} = \bar{f},\\ &\tilde{\beta}(M\otimes M\otimes M)\bar{u} + [\tilde{\alpha}M\otimes M\otimes M + \tilde{\gamma}(I\otimes M\otimes M + M\otimes I\otimes M + M\otimes M\otimes I)]\bar{v} = \bar{g}, \end{aligned} \tag{4.2}$$

where $M_{kj} = (\phi_j, \phi_k)$, $\{\bar{u}, \bar{v}\}$ are re-ordered vectors of three dimensional unknown coefficients $\tilde{u}_{ijk}$ and $\tilde{v}_{ijk}$, and $\{\bar{f}, \bar{g}\}$ are the corresponding right hand side vectors consisting of $(\mathcal{I}_N f, \phi_i(x)\phi_j(y)\phi_k(z))$ and $(\mathcal{I}_N g, \phi_i(x)\phi_j(y)\phi_k(z))$.

As before, let (E,Λ) be the eigen-pair of E, i.e., $ME = E\Lambda$. Setting

$$\bar{u} = (E\otimes E\otimes E)\bar{p}, \quad \bar{v} = (E\otimes E\otimes E)\bar{q} \tag{4.3}$$

plugging in the above into (4.2), and following the same procedure in [11] for the single equation in 3-D, we find that it reduces to the following sequence of 2×2 linear systems:

$$\begin{cases} \lambda_k\lambda_j\lambda_i(\alpha p_{ijk} + \beta q_{ijk}) + \gamma(\lambda_j\lambda_k + \lambda_i\lambda_k + \lambda_i\lambda_j)p_{ijk} = a_{ijk},\\ \lambda_k\lambda_j\lambda_i(\tilde{\alpha} p_{ijk} + \tilde{\beta} q_{ijk}) + \tilde{\gamma}(\lambda_j\lambda_k + \lambda_i\lambda_k + \lambda_i\lambda_j)q_{ijk} = b_{ijk}, \end{cases} \quad 0\leqslant i,j,k\leqslant N-2, \tag{4.4}$$

where a_{ijk} and b_{ijk} are corresponding elements of the vectors $[E^T\otimes E^T\otimes E^T]\bar{f}$ and $[E^T\otimes E^T\otimes E^T]\bar{g}$, respectively.

Therefore, the computational complexity in the 3D case is CN^4 (where C is a small number), and consists of mainly the matrix–matrix multiplications in the above and in (4.3).

On the other hand, by using the similar partial diagonalization procedure as in [17] for the single equation in 3-D, we can also extend the partial diagonalization procedure in the last section to the three dimensional case.

4.2. Systems of more than two coupled second-order equations

It is also easy to see that the two algorithms presented in the last section can be generalized to systems of more than two equations.

To illustrate the idea, we consider, for example, the following system of three coupled second-order equations in $\Omega = (-1,1)^2$:

$$\begin{cases} \alpha u + \beta v + \gamma w - \eta \Delta w = f, \\ \tilde{\alpha} u + \tilde{\beta} v + \tilde{\gamma} w - \tilde{\eta} \Delta v = g, \\ \tilde{\tilde{\alpha}} u + \tilde{\tilde{\beta}} v + \tilde{\tilde{\gamma}} w - \tilde{\tilde{\eta}} \Delta u = h, \end{cases} \tag{4.5}$$

with u, v, w satisfy the same set of boundary conditions $\mathcal{B}u|_{\pm 1} = 0$ in all directions.

Let ϕ_k be in the form of (2.4) such that $\mathcal{B}\phi_k|_{x=\pm 1} = 0$ and $-\left(\phi_j'', \phi_i\right) = \delta_{ij}$. We denote

$$u_N = \sum_{k=0}^{q}\sum_{j=0}^{q} \tilde{u}_{kj}\phi_k(x)\phi_j(y), \quad v_N = \sum_{k=0}^{q}\sum_{j=0}^{q} \tilde{v}_{kj}\phi_k(x)\phi_j(y), \quad w_N = \sum_{k=0}^{q}\sum_{j=0}^{q} \tilde{w}_{kj}\phi_k(x)\phi_j(y). \tag{4.6}$$

Let $M_{kj} = (\phi_j, \phi_k)$ and $ME = E\Lambda$ and set

$$\begin{aligned} &U = EXE^T, \quad V = EYE^T, \quad W = EZE^T, \\ &A = E^T FE, \quad B = E^T GE, \quad C = E^T HE, \end{aligned} \tag{4.7}$$

where U, V, W are the matrices with coefficients $\tilde{u}_{kj}, \tilde{v}_{kj}, \tilde{w}_{kj}$, and F, G, H are the corresponding right hand side vectors consisting of $(\mathcal{I}_N f, \phi_i(x)\phi_j(y))$, $(\mathcal{I}_N g, \phi_i(x)\phi_j(y))$ and $(\mathcal{I}_N h, \phi_i(x)\phi_j(y))$, respectively. Then, similar to the case of two equations, we find that the Legendre-Galerkin approximation of (4.5) can be reduced to a sequence of 3×3 linear system, i.e., for each index pair (i,j):

$$\begin{cases} \lambda_j\lambda_i(\alpha x_{ij} + \beta y_{ij} + \gamma z_{ij}) - \eta(\lambda_j z_{ij} + \lambda_i z_{ij}) = a_{ij}, \\ \lambda_j\lambda_i(\tilde{\alpha} x_{ij} + \tilde{\beta} y_{ij} + \tilde{\gamma} z_{ij}) - \tilde{\eta}(\lambda_j y_{ij} + \lambda_i y_{ij}) = b_{ij}, \\ \lambda_j\lambda_i(\tilde{\tilde{\alpha}} x_{ij} + \tilde{\tilde{\beta}} y_{ij} + \tilde{\tilde{\gamma}} z_{ij}) - \tilde{\tilde{\eta}}(\lambda_j x_{ij} + \lambda_i x_{ij}) = c_{ij}, \end{cases} \tag{4.8}$$

which can be solved easily. The above procedure is obviously applicable to systems of n coupled second-order equations.

On the other hand, it is easy to see that the partial diagonalization procedure can also be applied directly to systems of n coupled second-order equations, although the bandwidth of the reduced 1-D problems (see (3.25) for the case of two equations) will increase as n increases.

4.3. Problems with variable coefficients in one direction

In the above, we have only considered problems with constant coefficients. We shall now describe briefly how to extend the partial diagonalization procedure to problems with variable coefficients in one direction. We note that there are many situations where one needs to deal with such problems. We provide two important cases below:

- Cylindrical domains: Consider (3.1) with homogeneous Dirichlet boundary conditions in a cylinder $Q = \{(x,y,z): x^2 + y^2 \leqslant R^2, 0 \leqslant z \leqslant H\}$. Under the cylindrical polar coordinate transform

$$x = r\cos\theta, \quad y = r\sin\theta, \quad z = z$$

and a Fourier transform in the θ direction, each Fourier components of u and v, still denoted by u, v to simplify the notations (similarly for $\alpha, \beta, \tilde{\alpha}, \tilde{\beta}, f$ and g which now depend on the Fourier mode), satisfy the following system:

$$\begin{cases} \alpha u + \beta v - \gamma\left[\frac{1}{r}\partial_r(r\partial_r u) + \partial_{zz} u\right] = f, \text{ in } \Omega, \\ \tilde{\alpha} u + \tilde{\beta} v - \tilde{\gamma}\left[\frac{1}{r}\partial_r(r\partial_r v) + \partial_{zz} v\right] = g, \text{ in } \Omega, \end{cases} \tag{4.9}$$

with $u|_{\partial\Omega} = v|_{\partial\Omega} = 0$, except for the zeroth Fourier mode, where $\Omega = \{(r,z): 0 < r < R; 0 < z < H\}$. For the zeroth Fourier mode, a slighly different boundary condition should be used (cf. [19]).

- Large-scale ocean circulation model: Consider, for example, the following system for the wind-driven, double-gyre, quasi-geostrophic model of large-scale ocean circulation (cf. [23,13]):

$$\begin{cases} \boldsymbol{u}_t + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = \frac{1}{Re}\Delta\boldsymbol{u} - \nabla p - y\boldsymbol{k}\times\boldsymbol{u} + \boldsymbol{f}, \\ \nabla\cdot\boldsymbol{u} = 0, \end{cases} \tag{4.10}$$

in a 2-D rectangular domain Ω with $\boldsymbol{u}|_{\partial\Omega} = 0$. Let (u,v) be the two components of $\boldsymbol{u}$. In the above $\boldsymbol{k}\times\boldsymbol{u} \triangleq (0,0,\omega)^T \times (u,v,0)$. Then, under a suitable semi-implicit projection type discretization (cf. [20]) with an implicit treatment of the rotational term

to avoid stability constraint by the strong rotation, one need to solve, at each time step, the following system of two second-order equations:

$$\begin{cases} \alpha u - \beta \Delta u + \gamma y v = f, \text{ in } \Omega, \\ \tilde{\alpha} v - \tilde{\beta} \Delta v + \tilde{\gamma} y u = g, \text{ in } \Omega, \\ u|_{\partial\Omega} = v|_{\partial\Omega} = 0. \end{cases} \tag{4.11}$$

Note that the variable coefficients in the above two cases only depend on one variable (r in the first case and y in the second case). Hence, we can apply the partial diagonalization method to solve the systems (4.9) and (4.11). We leave the detail to the interested reader.

4.4. Chebyshev–Galerkin method

We can replace the Legendre polynomials by the Chebyshev-polynomials, and replace the L^2-inner product by the Chebyshev weighted inner product, leading to the so called Chebyshev–Galerkin method (cf. [18]). The only difference is that the stiffness matrix is no longer diagonal, instead, it is a upper triangular matrix. However, the main advantage of the Chebyshev–Galerkin method is that the fast Fourier transform (FFT) can be used to transform between the physical values of a function and the coefficients of its Chebyshev expansion.

The full diagonalization and partial diagonalization procedures described in Section 3 can be applied directly to the Chebyshev–Galerkin method. More precisely, we only have to replace the eigenvalue problem $ME = E\Lambda$ by the generalized eigenvalue problem $ME = SE\Lambda$, where M and S are the 1-D mass and stiffness matrices associated with the Chebyshev–Galerkin method, and replace E^T by E^{-1}, as E is no longer an ortho-normal matrix since S is not symmetric.

4.5. Convergence tests

We now present two numerical experiments to show the convergence properties of our algorithms.

In the first example, we consider the following general system of n second-order equations:

$$\begin{cases} \boldsymbol{A}\boldsymbol{u} - (\boldsymbol{B}_1 \partial_{xx}\boldsymbol{u} + \boldsymbol{B}_2 \partial_{yy}\boldsymbol{u} + \boldsymbol{B}_3 \partial_{zz}\boldsymbol{u}) = \boldsymbol{f}, \quad \boldsymbol{x} \in \Omega, \\ \boldsymbol{u}|_{\Gamma_1} = \boldsymbol{0}, \quad \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\big|_{\Gamma_2} = \boldsymbol{0}, \quad (a\boldsymbol{u} + b\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}})\big|_{\Gamma_3} = \boldsymbol{0}, \end{cases} \tag{4.12}$$

where $\Omega = (-1,1)^3$, $\boldsymbol{u} \in \mathbb{R}^n, \boldsymbol{x} = (x_1, x_2, x_3), \Gamma_i = \partial\Omega \cup \{x_i = \pm 1\}, i = 1,2,3$. In (4.12), $\{\boldsymbol{A}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3\}$ are given matrices of size $n \times n$, and $\{a, b\}$ are given constants.

Here we chose $\{\boldsymbol{A}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3\}$ to be symmetric tridiagonal matrices as follows,

$$\boldsymbol{A}^{jk} = \boldsymbol{A}^{kj} = \begin{cases} 3, & \text{if } k = j, \\ 1, & \text{if } k = j-1, \\ 0, & \text{else.} \end{cases} \quad \boldsymbol{B}_i^{jk} = i\boldsymbol{A}^{jk}, \quad i = 1,2,3 \tag{4.13}$$

and set $a = b = 1$. We chose each component, u_i, of the exact solution to be

$$u_i(\boldsymbol{x}) = p_i(x_1) q_i(x_2) r_i(x_3), \quad 1 \leqslant i \leqslant n, \tag{4.14}$$

where

Table 1
Convergence test: the first example with a system of 12 equations by using the Legendre–Galerkin method with the full diagonalization procedure.

$\|u_i - u_{i,N}\|_{l^\infty}$	N = 16	N = 32	N = 64	N = 128
i = 1	1.92E+02	1.31E+00	4.71E−11	1.35E−13
i = 2	1.69E+02	8.78E−01	8.55E−13	1.49E−13
i = 3	1.02E+02	4.40E−01	1.49E−13	1.04E−13
i = 4	4.92E+01	9.92E−02	2.13E−13	1.40E−13
i = 5	2.86E+01	1.04E−02	2.10E−13	1.16E−13
i = 6	1.45E+01	4.85E−04	4.28E−13	1.26E−13
i = 7	7.48E+00	5.43E−05	3.30E−13	1.56E−13
i = 8	1.61E+01	2.10E−03	2.79E−13	2.01E−13
i = 9	2.82E+01	4.49E−02	4.17E−13	2.03E−13
i = 10	6.85E+01	5.13E−01	5.54E−13	1.81E−13
i = 11	1.15E+02	3.43E+00	5.29E−13	5.22E−13
i = 12	8.70E+02	1.40E+01	2.49E−11	7.50E−13
max	8.70E+02	1.40E+01	4.71E−11	7.50E−13

Table 2
Convergence test: the second example by the Legendre–Galerkin method with the partial diagonalization procedure.

$\|u_i - u_{i,N}\|_\infty$	N = 16	N = 32	N = 64	N = 128
i = 1	1.07E+02	1.57E+00	6.07E−08	3.90E−14
i = 2	3.98E+02	2.45E+00	2.84E−09	3.33E−14
i = 3	1.02E+02	1.57E+00	6.07E−08	7.59E−13
max	3.98E+02	2.45E+00	6.07E−08	7.59E−13

$$p_i(\xi) = \sin(i\pi\xi), \quad q_i(\xi) = \cos((n+1-i)\pi\xi), \quad r_i(\xi) = i\left(e^{\xi} - \frac{1}{2}e\xi - e\right). \tag{4.15}$$

The above exact solution satisfies homogeneous Dirichlet boundary conditions in x_1, homogeneous Neumann boundary conditions in x_2 and homogeneous Robin boundary conditions in x_3.

In Table 1, we list the L^∞-error of each component of the numerical solution by using the Legendre–Galerkin method with the full diagonalization method. We observe clearly the spectral convergence. However, the accuracy does not reach the full double precision due to the round-off errors involved in the full diagonalization procedure. Note however that this kind of accuracy is still remarkable for such a large system in 3-D.

In the second example, we use the Legendre–Galerkin method with the partial diagonalization procedure to solve a system of three coupled equations in 2-D. We use the following exact solution:

$$u_i(x,y) = \cos((11+i)\pi x)\cos((14-i)\pi y), \quad i = 1,2,3. \tag{4.16}$$

And the coefficient matrices are chosen as

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 3 \\ 1 & -1 & 2 \\ -1 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{B}_1 = \boldsymbol{B}_2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -4 \end{bmatrix}. \tag{4.17}$$

The results are shown in Table 2. We observe once again the spectral convergence.

5. Applications

We present in this section several applications for which our algorithms can be directly applied.

5.1. Cahn–Hilliard Equations in 3-D

We consider first the three-dimensional isotropic Cahn–Hilliard equation

$$\begin{cases} \phi_t = \gamma\Delta\left(-\Delta\phi + \frac{1}{\epsilon^2}F'(\phi)\right), \\ \frac{\partial\phi}{\partial\boldsymbol{n}}|_{\partial\Omega} = \frac{\partial\Delta\phi}{\partial\boldsymbol{n}}|_{\partial\Omega} = 0, \end{cases} \tag{5.1}$$

where $\Omega = (-1,1)^3$ and $F(\phi) = (\phi^2-1)^2$.

To avoid solving a fourth-order equation at each time step, we reformulate (5.1) as a system of second-order equations

$$\begin{cases} \phi_t = \gamma\Delta\mu, \\ \mu = -\Delta\phi + \frac{1}{\epsilon^2}F'(\phi), \\ \frac{\partial\phi}{\partial\boldsymbol{n}}|_{\partial\Omega} = \frac{\partial\mu}{\partial\boldsymbol{n}}|_{\partial\Omega} = 0. \end{cases} \tag{5.2}$$

A very effective time discretization scheme is the stabilized scheme proposed in [22]. Its first-order version is as follows:

$$\begin{cases} \frac{\phi^{n+1}-\phi^n}{\delta t} = \gamma\Delta\mu^{n+1}, \\ \mu^{n+1} = -\Delta\phi^{n+1} + \frac{s(\phi^{n+1}-\phi^n)}{\epsilon^2} + \frac{1}{\epsilon^2}F'(\phi^n), \\ \left.\frac{\partial\phi^{n+1}}{\partial\boldsymbol{n}}\right|_{\partial\Omega} = \left.\frac{\partial\mu^{n+1}}{\partial\boldsymbol{n}}\right|_{\partial\Omega} = 0. \end{cases} \tag{5.3}$$

where s is a stabilization constant, it is shown in [22] that the above scheme is unconditionally stable with $s = 1$. On the other hand, with $s = 0$, the scheme will only be stable for $\delta t \leqslant C\epsilon^4$.

One observes that at each time step, (μ^{n+1}, ϕ^{n+1}) is the solution of a system of two coupled second-order equations so that our algorithms can be applied directly.

We now show a numerical example obtained by using this very efficient and stable algorithm. We start with two kissing balls at time $t = 0$, the evolution of the zero–isocontour of the phase function ϕ at several different times are shown in Fig. 1. We observe that the two kissing balls coalesce and merge gradually, and eventually turn into a single ball.

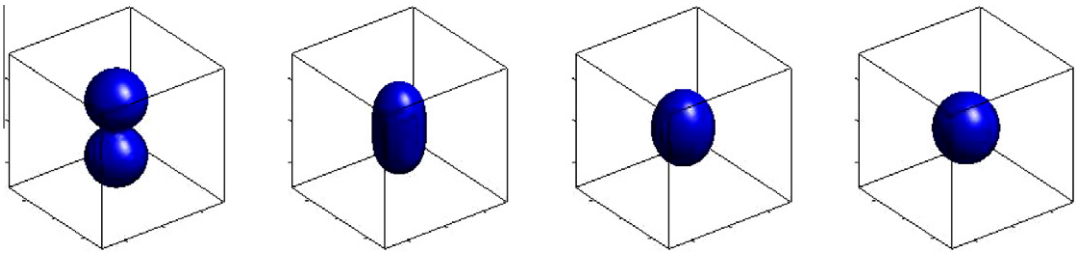

Fig. 1. The time evolution of the Cahn–Hilliard Eq. (5.2). The four plots are zero–isocontours of ϕ at $t = 0, \frac{T}{3}, \frac{2T}{3}, T$, with $T = 0.9$, $\delta t = 0.001$, $\epsilon = \gamma = 0.02$, $s = 1$. The Legendre–Galerkin method is used with $91 \times 91 \times 91$ Legendre–Gauss–Lobatto points.

Remark 5.1. We stress once again that it is essential that the boundary conditions are decoupled. Consider the bi-harmonic equation

$$\Delta^2 u = f \quad \text{in } \Omega = (-1,1)^d, \tag{5.4}$$

with the following boundary conditions:

$$u|_{\partial\Omega} = \left.\frac{\partial u}{\partial \boldsymbol{n}}\right|_{\partial\Omega} = 0. \tag{5.5}$$

We can rewrite (5.4) as a system of second-order equations by introducing $v = \Delta u$, but we do not have a separate boundary condition for v. Hence, our methods can not be applied in this case. We refer to [17,3] for some efficient, but much more complicated, spectral-Galerkin algorithms for solving (5.4) and (5.5).

5.2. A gradient–flow equation in functionalized polymers

As the second example, we consider the following gradient flow which models the polymer electrolyte membrane (PEM) in fuel cells [15],

$$\begin{cases} \phi_t = \Delta \mu, \\ \mu = [-\epsilon^2 \Delta + F''(\phi) - \eta]\omega, \ \text{in } \Omega = (-1,1)^2, \\ \omega = -\epsilon^2 \Delta \phi + F'(\phi), \end{cases} \tag{5.6}$$

where ϕ is the phase function which is set to 1 in water and -1 in polymer, with a small transitional layer of thickness ϵ, μ is the so called chemical potential, $F(\phi) = (\phi^2 - 1)^2$ is the double-well potential. The above system is subject to the following homogeneous Neumann boundary conditions:

$$\left.\frac{\partial \phi}{\partial \boldsymbol{n}}\right|_{\partial\Omega} = \left.\frac{\partial \mu}{\partial \boldsymbol{n}}\right|_{\partial\Omega} = \left.\frac{\partial \omega}{\partial \boldsymbol{n}}\right|_{\partial\Omega} = 0. \tag{5.7}$$

We observe that the system (5.6) is in fact a six-order equation for ϕ. However, similar as in the Cahn–Hilliard case in the last subsection, it is computational more efficient to write it as a system of three second-order equations.

By using a similar idea as for the Cahn–Hilliard Eq. (5.2), we propose the following scheme for the system (5.6):

$$\begin{cases} \frac{\phi^{n+1} - \phi^n}{\delta t} = \Delta \mu^{n+1}; \\ \mu^{n+1} = -\epsilon^2 \Delta \omega^{n+1} + F''(\phi^n)\tilde{\omega}^n - \eta \omega^{n+1} + s_1(\omega^{n+1} - \tilde{\omega}^n), \\ \text{where } \tilde{\omega}^n = -\epsilon^2 \Delta \phi^n + F'(\phi^n); \\ \omega^{n+1} = -\epsilon^2 \Delta \phi^{n+1} + F'(\phi^n) + s_2(\phi^{n+1} - \phi^n), \\ \left.\frac{\partial \phi^{n+1}}{\partial \boldsymbol{n}}\right|_{\partial\Omega} = \left.\frac{\partial \mu^{n+1}}{\partial \boldsymbol{n}}\right|_{\partial\Omega} = \left.\frac{\partial \omega^{n+1}}{\partial \boldsymbol{n}}\right|_{\partial\Omega} = 0, \end{cases} \tag{5.8}$$

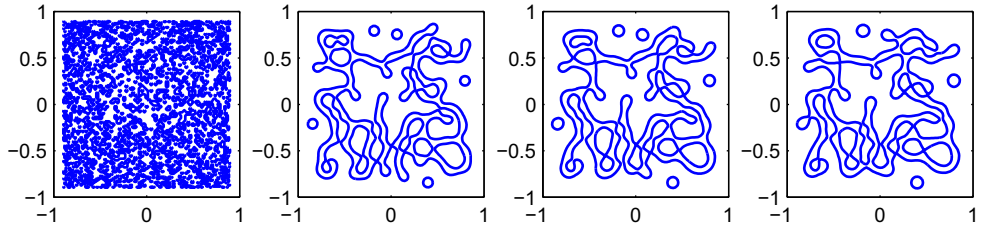

Fig. 2. The time evolution of the gradient–flow equation in PEM, implemented with scheme (5.8). The blue lines are interfaces drawn at four different time spots. It starts from the random initial condition. The Legendre–Galerkin method is used with 129×129 Gaussian points. The terminal time is $T = 12$. The time step size is $\delta t = 0.01$. $s_1 = s_2 = 2$ and $\epsilon = 0.02$.

where s_1 and s_2 are two suitable stabilization constants. Our numerical experiments show that (5.8) is unconditional stable with $s_1 = s_2 = 2$, though a rigorous proof of the stability is still elusive. On the other hand, with $s_1 = s_2 = 0$, the scheme will only be stable for extremely small δt.

At each time step, the scheme (5.8) is a system of three coupled second-order equations for $\{\phi^{n+1}, \mu^{n+1}, \omega^{n+1}\}$. Hence, the algorithms developed in the last section can be applied directly. We plot in Fig. 2 the time evolution of a typical pore network pattern which is qualitatively consistent with those presented in [15].

5.3. Phase-field-crystal equations

Consider the following energy functional that arises from the so called phase-field-crystal method [6,26],

$$\mathcal{E}(\phi) = \int_\Omega \left\{ \frac{\phi}{2}\left[-\epsilon + (\Delta + 1)^2 \left(\left(\Delta + Q_1^2 \right)^2 + R_1 \right) \right] \phi + \frac{\phi^4}{4} \right\} d\Omega, \tag{5.9}$$

where ϵ, R_1, and Q_1 are given constants depending on the material. One consider the gradient flow with respect to the free energy (5.9), leading to a 10th order nonlinear PDE in ϕ, which can be rewritten as a system of five coupled second-order equations:

$$\begin{cases} \phi_t = \Delta \mu, \\ \mu = f(\phi) + c_1 w_1 + c_2 w_2 + c_3 w_3 + c_4 \Delta w_3, \\ w_1 = \Delta \phi, \\ w_2 = \Delta w_1, \\ w_3 = \Delta w_2, \end{cases} \tag{5.10}$$

with homogeneous Neumann boundary conditions for ϕ,μ,w_1,w_2,w_3. In the above, $f(\phi) = c_0\phi + \phi^3$ and $\{c_i\}_{i=0}^4$ are constants depending on ϵ, R_1, and Q_1.

Similarly as in the previous two examples, we propose the following stabilized scheme for (5.10):

$$\begin{cases} \frac{\phi^{n+1} - \phi^n}{\delta t} = \Delta \mu^{n+1}, \\ \mu^{n+1} = f(\phi^n) + c_1 w_1^{n+1} + c_2 w_2^{n+1} + c_3 w_3^{n+1} + c_4 \Delta w_3^{n+1} + s(\phi^{n+1} - \phi^n), \\ w_1^{n+1} = \Delta \phi^{n+1}, \\ w_2^{n+1} = \Delta w_1^{n+1}, \\ w_3^{n+1} = \Delta w_2^{n+1}, \end{cases} \tag{5.11}$$

where s is a suitable stabilizing parameter. It can be shown, by using a similar procedure as in [22], that the above scheme is unconditionally stable with $s \geqslant s_0$ for some suitable parameter s_0.

It is clear that, at each time step, the scheme (5.11) is a system of five coupled second-order equations for $\{\phi^{n+1}, \mu^{n+1}, w_1^{n+1}, w_2^{n+1}, w_3^{n+1}\}$. Hence, this tenth-order nonlinear equation can be efficiently solved by using the above stabilized time discretization scheme coupled with the spectral-Galerkin algorithms developed in the last section.

5.4. Schrödinger equations

Next, we consider the following time dependent nonlinear Schrödinger equation,

$$i\frac{\partial}{\partial t}\psi(\boldsymbol{x}, t) = \left[-\frac{1}{2}\Delta + V(\boldsymbol{x}) + \beta|\psi(\boldsymbol{x}, t)|^2 \right] \psi(\boldsymbol{x}, t), \tag{5.12}$$

where $\psi(\boldsymbol{x}, t) \in \mathbb{C}$ is the unknown wave function, $V(\boldsymbol{x})$ a given potential function, and β the interaction intensity between atoms. The Eq. (5.12) is also known as the Gross–Pitaevskii equation [14], which describes the mean field dynamics of the Bose Einstein condensation (BEC). As an example, we consider $\Omega = (-1, 1)^3$ with

$$\psi(\boldsymbol{x}, t)|_{\partial\Omega} = 0. \tag{5.13}$$

A popular and efficient way to discretize (5.12) in time is to use a operator splitting scheme. For example, we can split the original equation into two sub-problems:

$$\begin{aligned} i\tfrac{\partial}{\partial t}\psi &= \mathcal{L}\psi \triangleq -\tfrac{1}{2}\Delta\psi, \\ i\tfrac{\partial}{\partial t}\psi &= \mathcal{N}\psi \triangleq [V(\boldsymbol{x}) + \beta|\psi^2|]\psi, \end{aligned} \tag{5.14}$$

and use the second–order Strang splitting method [24]:

$$\psi^{(1)} = e^{-i\frac{\delta t}{2}\mathcal{N}}\psi^n, \quad \psi^{(2)} = e^{-i\delta t\mathcal{L}}\psi^{(1)}, \quad \psi^{n+1} = e^{-i\frac{\delta t}{2}\mathcal{N}}\psi^{(2)}. \tag{5.15}$$

Note that the first and third substeps are ODEs which can be solved exactly (cf., for instance, [2]) so we focus on the second substep for the linear operator $\mathcal{L}$.

Writing $\psi = u + iv$ in the first equation of (5.14), and separating the real and imaginary parts, we find

$$\begin{cases} v_t = \frac{1}{2}\Delta u, \\ u_t = -\frac{1}{2}\Delta v. \end{cases} \tag{5.16}$$

Hence, we can discretize the above by the second-order Crank–Nicholson scheme in time

$$\begin{cases} \frac{v^{n+1}-v^n}{\delta t} = \frac{1}{2}\Delta(u^{n+1}+u^n), \qquad v^{n+1}|_{\partial\Omega} = 0, \\ \frac{u^{n+1}-u^n}{\delta t} = -\frac{1}{2}\Delta(v^{n+1}+v^n), \quad u^{n+1}|_{\partial\Omega} = 0, \end{cases} \tag{5.17}$$

which, at each time step, is a system of two coupled second-order equations in $\{u^{n+1}, v^{n+1}\}$. Thus, our spectral-Galerkin algorithms can be applied directly.

5.5. Rotational Navier–Stokes equations

We consider the following rotational Navier–Stokes equations

$$\begin{cases} \boldsymbol{u}_t + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = \nu\Delta\boldsymbol{u} - \nabla p - \boldsymbol{\Gamma}\times\boldsymbol{u}, \\ \nabla\cdot\boldsymbol{u} = 0, \end{cases} \tag{5.18}$$

in $\Omega = (-1,1)^3$ with homogeneous Dirichlet boundary conditions and $\boldsymbol{\Gamma} = (\gamma_1,\gamma_2,\gamma_3)$.

A typical semi-implicit time discretization for (5.18) is the following pressure-correction scheme (cf. [8]):

- In the first substep, we find $\tilde{\boldsymbol{u}}^{n+1}$ by solving:

$$\begin{cases} \frac{\tilde{\boldsymbol{u}}^{n+1}-\boldsymbol{u}^n}{\delta t} - \nu\Delta\tilde{\boldsymbol{u}}^{n+1} + \boldsymbol{\Gamma}\times\tilde{\boldsymbol{u}}^{n+1} = -\nabla p^n - \boldsymbol{u}^n\cdot\nabla\boldsymbol{u}^n, \\ \tilde{\boldsymbol{u}}^{n+1}|_{\partial\Omega} = 0. \end{cases} \tag{5.19}$$

- In the second substep, we find $(\boldsymbol{u}^{n+1}, \phi^{n+1})$ from

$$\begin{cases} \frac{\boldsymbol{u}^{n+1}-\tilde{\boldsymbol{u}}^{n+1}}{\delta t} + \nabla\phi^{n+1} = \boldsymbol{0}, \\ \nabla\cdot\boldsymbol{u}^{n+1} = 0, \\ \boldsymbol{u}^{n+1}\cdot\boldsymbol{n}|_{\partial\Omega} = 0. \end{cases} \tag{5.20}$$

- In the third substep, we update p^{n+1} by

$$p^{n+1} = \phi^{n+1} + p^n - \nu\nabla\cdot\tilde{\boldsymbol{u}}^{n+1}. \tag{5.21}$$

Note that the second substep can be reformulated as a Poisson equation for ϕ^{n+1} by applying the divergence operator on both side, while the first substep is a system of three coupled second-order equations so it can be solved efficiently by our spectral-Galerkin algorithms.

6. Conclusion

We constructed in this paper two efficient spectral-Galerkin algorithms for solving systems of n coupled second-order equations in separable geometries with decoupled boundary conditions. The first one is the full diagonalization method which is easy to implement, while the second one is the so called partial diagonalization method which is slightly more complicated to implement but is more efficient and can be applied to a wider class of problems. The computational complexity of these algorithms is CN^{d+1} (where d is the dimension, and C is a small number), and it is essentially n times the cost of solving one second-order equation. Hence, these algorithms are extremely efficient and accurate for solving systems of coupled second-order equations.

We presented several interesting and challenging applications that can be solved by using the new algorithms, as well as numerical results which demonstrate their accuracy and flexibility.

While the new algorithms presented in this paper only apply to a specific class of systems of coupled second-order equations, they are very efficient and accurate whenever applicable. We believe that they can be useful in dealing with many important scientific and engineering applications in materials science and fluid dynamics which are otherwise difficult to treat with conventional numerical methods.

References

[1] F. Auteri, L. Quartapelle, Galerkin–Legendre spectral method for the 3D Helmholtz equation, J. Comput. Phys. 161 (2) (2000) 454–483.
[2] Weizhu Bao, Shi Jin, Peter A. Markowich, On time-splitting spectral approximations for the Schrödinger equation in the semiclassical regime, J. Comput. Phys. 175 (2) (2002) 487–524.

[3] P.E. Bjørstad, B.P. Tjostheim, Efficient algorithms for solving a fourth order equation with the spectral-Galerkin method, SIAM J. Sci. Comput. 18 (1997).
[4] John P. Boyd, Chebyshev and Fourier Spectral Methods, second ed., Dover Publications Inc., Mineola, NY, 2001.
[5] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral methods, Scientific Computation, Fundamentals in single domains, Springer-Verlag, Berlin, 2006.
[6] K.R. Elder, Martin Grant, Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals, Phys. Rev. E 70 (5) (2004) 051605.
[7] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, SIAM–CBMS, Philadelphia, 1977.
[8] J.L. Guermond, P. Minev, Jie Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Engrg. 195 (2006) 6011–6045.
[9] Ben-Yu Guo, Jie Shen, Li-Lian Wang, Optimal spectral-Galerkin methods using generalized Jacobi polynomials, J. Sci. Comput. 27 (1–3) (2006) 305–322.
[10] D.B. Haidvogel, T.A. Zang, The accurate solution of Poisson's equation by expansion in Chebyshev polynomials, J. Comput. Phys. 30 (1979) 167–180.
[11] P. Haldenwang, G. Labrosse, S. Abboudi, M. Deville, Chebyshev 3-d spectral and 2-d pseudospectral solvers for the Helmholtz equation, J. Comput. Phys. 55 (1984) 115–128.
[12] R.E. Lynch, J.R. Rice, D.H. Thomas, Direct solution of partial differential equations by tensor product methods, Numer. Math. 6 (1964) 185–199.
[13] J. Pedlosky, Open Circulation Theory, Springer-Verlag, 1996.
[14] L. Pitaevskii, S. Stringari, Bose–Einstein Condensation, Oxford University Press, USA, 2003.
[15] Keith Promislow, Brian Wetton, PEM fuel cells: a mathematical overview, SIAM J. Appl. Math. 70 (2) (2009) 369–409.
[16] Andreas Ratz, Axel Voigt, Higher order regularization of anisotropic geometric evolution equations in three dimensions, J. Comput. Theor. Nanoscience 3 (2006) 560–564.
[17] Jie Shen, Efficient spectral-Galerkin method I: direct solvers of second-and fourth-order equations using Legendre polynomials, SIAM J. Sci. Comput. 15 (6) (1994) 1489–1505.
[18] Jie Shen, Efficient spectral-Galerkin method II.: direct solvers of second- and fourth-order equations using Chebyshev polynomials, SIAM J. Sci. Comput. 16 (1) (1995) 74–87.
[19] Jie Shen, Efficient spectral-Galerkin methods III: polar and cylindrical geometries, SIAM J. Sci. Comput. 18 (6) (1997) 1583–1604.
[20] Jie Shen, T. Tachim Mejdo, Shouhong Wang, On a wind-driven, double-gyre, quasi-geostrophic ocean model: numerical simulations and structural analysis, J. Comput. Phys. 155 (1999) 387–409.
[21] Jie Shen, Tao Tang, Li-Lian Wang, Spectral Methods: Algorithms, Analysis and Applications, Springer Series in Computational Mathematics, vol. 41, Springer, 2011.
[22] Jie Shen, Xiaofeng Yang, Numerical approximations of Allen–Cahn and Cahn–Hilliard equations, Discrete and Continuous Dynamical Systems 28 (4) (2010) 1669–1691.
[23] S. Speich, M. Ghill, Interannual variability of the mid-latitude oceans: a new source of climate variability?, Sistema Terra 3 (1994) 33
[24] Gilbert Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.
[25] Solmaz Torabi, John Lowengrub, Axel Voigt, Steven Wise, A new phase-field model for strongly anisotropic systems, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science 465 (2105) (2009) 1337–1359.
[26] Kuo-An Wu, Ari Adland, Alain Karma, Phase-field-crystal model for fcc ordering, Phys. Rev. E 81 (6) (2010) 061601.