JOURNAL OF COMPUTATIONAL PHYSICS **142,** 562–580 (1998)
ARTICLE NO. CP985954

Fourier Collocation Splittings for Partial Differential Equations

Bruno Costa* and Lucia Dettori†

**Institute of Applied Mathematics, Indiana University, Bloomington, Indiana 47406, and Departamento de Matemática Aplicada, IM-UFRJ, Caixa Postal 68530, Rio de Janeiro, RJ, C.E.P. 21945-970, Brazil; and †Department of Mathematics, Southern Methodist University, P.O. Box 750156, Dallas, Texas 75275-0156, and Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912*
E-mail: *bcosta@hamlet.ucs.indiana.edu and †lucia@golem.math.smu.edu

Received May 21, 1997; revised December 9, 1997

In this article we study computational issues related to a nonlinear Galerkin type splitting (NLG) of partial differential equations in the case of a Fourier collocation discretization. We present an extension of the method to two-dimensional problems and show that the sole separation of modes in NLG can bring precision and computational costs advantages to the standard collocation scheme. Numerical experiments with the Burgers and a reaction-diffusion equation for 1 and 2 dimensions are also shown. © 1998 Academic Press

1. INTRODUCTION

In this article we introduce a pseudospectral Fourier collocation splitting for two dimensional partial differential equations aimed at reducing the cost of computing spatial derivatives. The splitting was originally motivated by the Nonlinear Galerkin Method and we will therefore refer to it as NLG.

We extend the results of a previous article [3] on the one dimensional case to two space dimensions and point out the relevant numerical aspects of both cases in a comparison with the standard collocation method (SCM). The extension to two dimensional problems of the Chebyshev collocation case as in [4] is the subject of a forthcoming work.

The basic idea in the Nonlinear Galerkin method (and in the theory of inertial manifolds) is the decomposition of the unknown u into its large scale and small scale components, y and z:

$$u = y + z.$$

In the case of a Fourier expansion it is clear that y corresponds to the low modes and z to the high modes (see [3]).

0021-9991/98 $25.00

When a collocation method is used (as opposed to a Galerkin spectral method) we need to find a decomposition of the kind above in the physical space. In one space dimension such a decomposition is accomplished via the splitting of the fine grid into two coarse grids based on half of the points. In the case of two space dimensions, the fine grid is split into four coarse grids each one having one fourth of the points of the fine grid. The high modes component is decomposed into three components z^1, z^2, and z^3. Each component can be represented by any one of the coarse grids, allowing most of the computations to be performed using only one fourth of the number of points of the original fine grid. This will result in a significant reduction of the number of operations necessary to calculate spatial derivatives. The treatment of nonlinear terms, however, requires a re-projection of all modes on the fine grid at the end of each iteration. In certain cases like, for example, the Burgers equation, this extra projection may compensate the advantage in computational costs of NLG when treating derivatives and one may end up with the same number of operations per iteration for NLG and SCM. However, as shown in Section 4, this is not the case when NLG is applied to equations with a more complicated structure and a higher number of derivatives.

Numerical results show how the NLG splitting produces an approximate solution which is as accurate as the one obtained with a SCM based on the fine grid. Other numerical and theoretical studies have been done to compare various versions of NLG methods and, more recently, the Postprocessed Galerkin method to a standard Galerkin or collocation approximation and different conclusions have been reached regarding the accuracy and efficiency of the methods, see, for example, [7–14] and references therein. It should be pointed out that in all of those cases the comparison was with a Galerkin or collocation scheme based on the low modes or coarse grid only.

Several numerical results in the one dimensional case suggest the choice of a NLG splitting of the equation into high and low modes that keeps all the linear and nonlinear terms, without disregarding any. Also in light of this fact we believe that the right comparison is with a SCM based on the fine grid.

The main focus of this article is not the complete study of the computational efficiency of the NLG. Further aspects of this issue are addressed in a forthcoming article (see [2]) where we show how the efficiency of the NLG approach can be significantly improved by considering different time integration schemes for the low modes and the high modes equations. By introducing a modified Runge–Kutta scheme for the high modes equation, one can choose a larger time step reducing the overall computational cost of the algorithm.

The article is organized as follows. In Section 2 we define the grids and the related spaces of functions and associated projections that will be used throughout the article. In Section 3 we present the decomposition into high and low modes and the construction of the interpolating polynomials. In Section 4 we apply the NLG to various equations in one and two space dimensions and discuss the computational cost of the method. Numerical results comparing NLG and the SCM are shown in Section 5.

2. PRELIMINARIES

Throughout this article we will use the following spaces of trigonometrical polynomials. Let $M = 2N + 1$; we define

$$U_N = \{e^{ikx}; |k| \leq N\}, \qquad V_M = \{\sin jx, \cos kx; 1 \leq j \leq M - 1, 0 \leq k \leq M\},$$

of dimensions $2N + 1$ and $2M = 2(2N + 1)$, respectively. We also define the space U_N^M, the orthogonal complement of U_N in V_M.

A generic function $f(x)$ can be projected on V_M by interpolation on the following set of points,

$$x_j^M = \frac{\pi j}{M}, \qquad 0 \le j \le 2M - 1,$$

and to project on U_N we can use either one of the set of points

$$\xi_j^N = \frac{2\pi j}{2N+1}, \qquad 0 \le j \le 2N,$$

$$\eta_j^N = \frac{(2j+1)\pi}{2N+1}, \qquad 0 \le j \le 2N.$$

For details on the construction and approximating properties of the projection operators corresponding to these grids we refer the reader to [3].

The 2-dimensional trigonometrical spaces that follow are formed by cross products of V_M, U_N, and U_N^M:

$$U_{NN} = U_N \times U_N, \qquad U_{NM} = U_N \times U_N^M, \qquad U_{MN} = U_N^M \times U_N, \qquad U_{MM} = U_N^M \times U_N^M \tag{1}$$

and

$$V_{MM} = V_M \times V_M.$$

Note that $V_{MM} = U_{NN} \oplus U_{NM} \oplus U_{MN} \oplus U_{MM}$.

A generic function $f(x, y)$ in V_{MM} can be decomposed in 4 components $f = w + z^1 + z^2 + z^3$, each one belonging to a different space above, i.e.,

$$w \in U_{NN}, \qquad z^1 \in U_{NM}, \qquad z^2 \in U_{MN}, \qquad z^3 \in U_{MM}.$$

We can represent these spaces by means of rectangular areas in the space of the 2-dimensional wave number $\mathbf{k} = (k_1, k_2)$ as it is shown in Fig. 1. Note that all four spaces in (1) occupy one fourth of the area occupied by V_{MM} in Fig. 1. This is an indication that we just need one fourth the number of points to represent each space in (1).

We define $\mathbf{k}$ as a *low* mode if $|k_1|, |k_2| \le N$, a *high* mode if $|k_1|, |k_2| > N$, and a *mixed* mode otherwise.

To project a function $f(x, y)$ onto the 2-dimensional spaces above we need to define the following 2-dimensional grids, which are cross products of the grids previously defined,

$$a_{i,j}^N = \left(\xi_i^N, \xi_j^N\right), \qquad b_{i,j}^N = \left(\eta_i^N, \eta_j^N\right), \qquad c_{i,j}^N = \left(\xi_i^N, \eta_j^N\right), \qquad d_{i,j}^N = \left(\eta_i^N, \xi_j^N\right), \tag{2}$$

and to project onto the space $V_{MM} = V_M \times V_M$ we use the fine grid

$$p_{i,j}^M = \left(x_i^M, x_j^M\right).$$

Figure 2 shows the spatial disposition of the several grids on the $x - y$ plane.

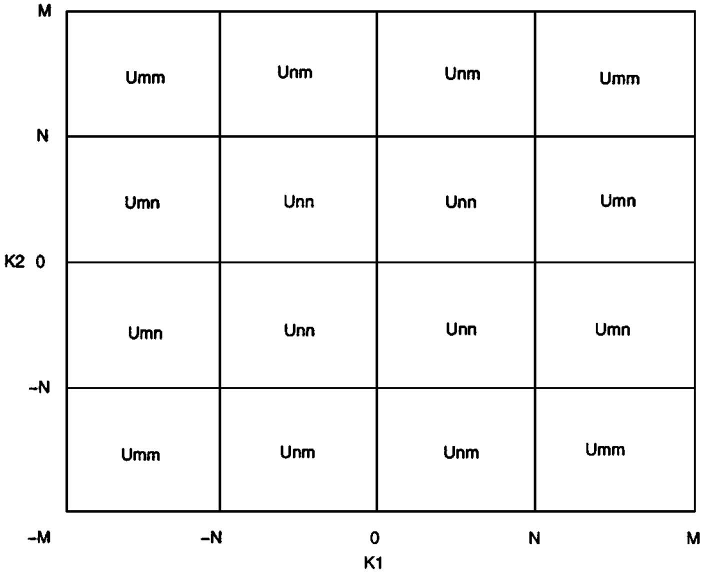


FIG. 1. The spaces U_{NN}, U_{NM}, U_{MN}, and U_{MM}.

Note that

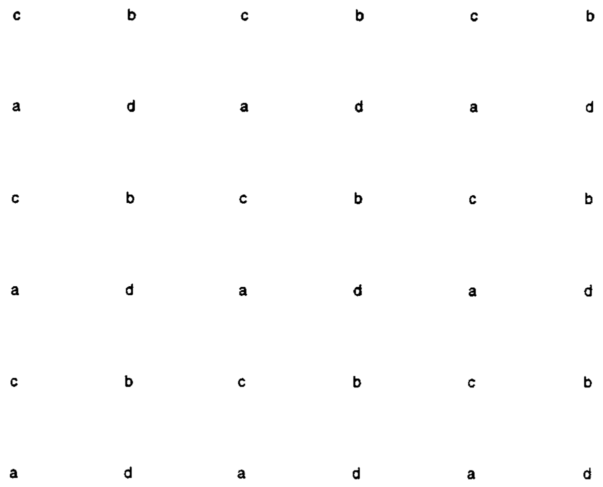
$$\{p^M_{i,j}\} = \{a^N_{i,j}\} \cup \{b^N_{i,j}\} \cup \{c^N_{i,j}\} \cup \{d^N_{i,j}\};$$

indeed, we have

$$p^M_{2i,2j} = a^N_{i,j}, \qquad p^M_{2i+1,2j+1} = b^N_{i,j}, \qquad p^M_{2i+1,2j} = c^N_{i,j}, \qquad p^M_{2i,2j+1} = d^N_{i,j}.$$

In order to construct projection operators that interpolate a function $f(x, y)$ at each of the grids mentioned above, let us consider the Dirichlet kernels

$$H^{2N+1}(x, y) = \frac{1}{M} \sum_{k=-N}^{N} e^{ik(x-y)}, \tag{3}$$



c	b	c	b	c	b
a	d	a	d	a	d
c	b	c	b	c	b
a	d	a	d	a	d
c	b	c	b	c	b
a	d	a	d	a	d

FIG. 2. The 2D grids for NLG.

and

$$H^{2M}(x,y)=\frac{1}{2M}\sum_{k=-M}^{M}\frac{1}{c_k^M}e^{ik(x-y)},\tag{4}$$

where $c_k^M=1+\delta_{|k|,M}$. It is well known that

$$H^{2N+1}\left(\xi_i^N,\xi_j^N\right)=H^{2N+1}\left(\eta_i^N,\eta_j^N\right)=\delta_{i,j}\tag{5}$$

and

$$H^{2M}\left(x_i^M,x_j^M\right)=\delta_{i,j}.\tag{6}$$

Definition 2.1. Let $f(x,y)$ be a function defined in $[0,2\pi]^2$. We define the trigonometric polynomial $Q_N^a f$ as

$$\begin{aligned}Q_N^a f(x,y)&=\sum_{j,l=0}^{2N} f\left(a_{i,j}^N\right)H^{2N+1}\left(x,\xi_j^N\right)H^{2N+1}\left(y,\xi_l^N\right)\\&=\sum_{j,l=0}^{2N} f\left(\xi_j^N,\xi_l^N\right)H^{2N+1}\left(x,\xi_j^N\right)H^{2N+1}\left(y,\xi_l^N\right).\end{aligned}$$

Equation (5) shows that $Q_N^a f$ interpolates the function $f(x,y)$ at the collocation grid $\{a_{i,j}^N\}$, i.e.,

$$Q_N^a f\left(a_{i,j}^N\right)=f\left(a_{i,j}^N\right),\qquad 0\le i,j\le 2N.$$

In a similar way we define the trigonometric polynomials $Q_N^b f$, $Q_N^c f$, and $Q_N^d f$ that interpolate $f(x,y)$ at the grids $\{b_{i,j}^N\}$, $\{c_{i,j}^N\}$, and $\{d_{i,j}^N\}$, respectively,

$$Q_N^b f(x,y)=\sum_{j,l=0}^{2N} f\left(\eta_j^N,\eta_l^N\right)H^{2N+1}\left(x,\eta_j^N\right)H^{2N+1}\left(y,\eta_l^N\right),\tag{7}$$

$$Q_N^c f(x,y)=\sum_{j,l=0}^{2N} f\left(\xi_j^N,\eta_l^N\right)H^{2N+1}\left(x,\xi_j^N\right)H^{2N+1}\left(y,\eta_l^N\right),\tag{8}$$

$$Q_N^d f(x,y)=\sum_{j,l=0}^{2N} f\left(\eta_j^N,\xi_l^N\right)H^{2N+1}\left(x,\eta_j^N\right)H^{2N+1}\left(y,\xi_l^N\right).\tag{9}$$

Finally, we define the trigonometric polynomial $Q_M^p f$ interpolating at the fine grid $\{p_{i,j}^M\}$:

Definition 2.2. Let $f(x,y)$ be a function defined in $[0,2\pi]^2$. We define the trigonometric polynomial $Q_M^p f$ as

$$Q_M^p f(x,y)=\sum_{j,l=0}^{2M-1} f\left(x_j^M,x_l^M\right)H^M\left(x,x_j^M\right)H^M\left(y,x_l^M\right).$$

It is easy to show that the polynomial $Q_M^p f$ interpolates $f(x,y)$ at the grid $\{x_i^M\}$:

$$Q_M^p f\left(x_i^M,x_j^M\right)=f\left(x_i^M,x_j^M\right),\qquad 0\le i,j\le 2M-1.$$

We can give alternate representations of the polynomials above by using (3) and (4),

$$Q_N^a f(x,y) = \sum_{k_1,k_2=-N}^{N} \hat{f}_{k_1,k_2}^{N} e^{i(k_1 x + k_2 y)}, \tag{10}$$

where

$$\hat{f}_{k_1,k_2}^{a} = \frac{1}{M^2} \sum_{j,l=0}^{2N} f\left(\xi_j^N, \xi_l^N\right) e^{-i\left(k_1 \xi_j^N + k_2 \xi_l^N\right)}; \tag{11}$$

and

$$Q_M^p f(x,y) = \sum_{k_1,k_2=-M}^{M} \hat{f}_{k_1,k_2}^{M} e^{i(k_1 x + k_2 y)}, \tag{12}$$

where

$$\hat{f}_{k_1,k_2}^{p} = \frac{1}{4M^2 c_{k_1,k_2}^M} \sum_{j,l=0}^{2M-1} f\left(x_j^M, x_l^M\right) e^{-i\left(k_1 x_j^M + k_2 x_l^M\right)}, \tag{13}$$

and $c_{k_1,k_2}^M = c_{k_1}^M c_{k_2}^M = (1 + \delta_{|k_1|,M})(1 + \delta_{|k_2|,M})$.

The alternative representations for the other polynomials Q_N^b, Q_N^c, and Q_N^d are analogous to the one in (10) and (11).

3. HIGH AND LOW MODES DECOMPOSITION

In this section we construct the collocation operators J_{NN}, G_{NM}, G_{MN}, and G_{MM} that project $f(x,y)$ onto the spaces U_{NN}, U_{NM}, U_{MN}, and U_{MM}, respectively. We will then prove that each of these quantities can be expressed in terms of its values at only one of the coarse grids in Fig. 2.

THEOREM 3.1. *Let $M = 2N+1$ and Q_M^p, Q_N^a, Q_N^b, Q_N^c, Q_N^d be defined as in* (*10*) *and* (*12*). *We set*

$$J_{NN} = \frac{Q_N^a + Q_N^b + Q_N^c + Q_N^d}{4}, \tag{14}$$

$$G_{NM} = \frac{Q_N^a + Q_N^d - Q_N^b - Q_N^c}{4}, \tag{15}$$

$$G_{MN} = \frac{Q_N^a + Q_N^c - Q_N^b - Q_N^d}{4}, \tag{16}$$

$$G_{MM} = Q_M^p - J_{NN} - G_{NM} - G_{MN}. \tag{17}$$

Then any function $f \in V_M$ can be written uniquely as

$$f = J_{NN} f + G_{NM} f + G_{MN} f + G_{MM} f,$$

where $J_{NN} f \in U_{NN}$, $G_{NM} f \in U_{NM}$, $G_{MN} f \in U_{MN}$, and $G_{MM} f \in U_{MM}$.

Proof. We start by showing that the low modes ($|k_1|, |k_2| \leq N$) of f and $J_{NN} f$ are the same. We will use formula (13) and recall that for these values of k_1, k_2 we have $c^M_{k_1,k_2} = 1$,

$$\hat{f}^M_{k_1,k_2} = \frac{1}{4M^2} \sum_{j,l=0}^{2M-1} f\left(x^N_j, x^N_l\right) e^{-i\left(k_1 x^N_j + k_2 x^N_l\right)} \tag{18}$$

$$= \frac{1}{4M^2} \left(\sum_{j,l=0}^{2N} f\left(a^N_{i,j}\right) e^{-i\mathbf{k}\cdot a^N_{j,l}} + \sum_{j,l=0}^{2N} f\left(b^N_{i,j}\right) e^{-i\mathbf{k}\cdot b^N_{j,l}} + \sum_{j,l=0}^{2N} f\left(c^N_{i,j}\right) e^{-i\mathbf{k}\cdot c^N_{j,l}} + \sum_{j,l=0}^{2N} f\left(d^N_{i,j}\right) e^{-i\mathbf{k}\cdot d^N_{j,l}} \right). \tag{19}$$

Therefore, using (11) and the corresponding formulas for grids b, c, and d,

$$\hat{f}^M_{k_1,k_2} = \frac{1}{4}\left(\hat{f}^a_{k_1,k_2} + \hat{f}^b_{k_1,k_2} + \hat{f}^c_{k_1,k_2} + \hat{f}^d_{k_1,k_2}\right).$$

It remains to show that the components of f in U_{NM} and U_{MN} are $G_{NM} f$ and $G_{MN} f$, respectively. As we can see from Fig. 2, the components of f in U_{NM} are of the form

$$\hat{f}^p_{k_1,2N+1-k_2}, \qquad k_1 = -N, \ldots, N,\ k_2 = 0, \ldots, N$$

and

$$\hat{f}^p_{k_1,k_2-(2N+1)}, \qquad k_1 = -N, \ldots, N,\ k_2 = 0, \ldots, N.$$

We will consider the first case and write $\hat{f}^p_{k_1,2N+1-k_2}$ as

$$\hat{f}^p_{k_1,2N+1-k_2} = \frac{1}{4M^2} \left(\sum_{j,l=0}^{2N} f\left(a^N_{i,j}\right) e^{-i\left(k_1\xi^N_j + (2N+1-k_2)\xi^N_l\right)} + \sum_{j,l=0}^{2N} f\left(d^N_{i,j}\right) e^{-i\left(k_1\eta^N_j + (2N+1-k_2)\xi^N_l\right)} + \sum_{j,l=0}^{2N} f\left(b^N_{i,j}\right) e^{-i\left(k_1\eta^N_j + (2N+1-k_2)\eta^N_l\right)} + \sum_{j,l=0}^{2N} f\left(c^N_{i,j}\right) e^{-i\left(k_1\xi^N_j + (2N+1-k_2)\eta^N_l\right)} \right).$$

Using the identities

$$e^{\pm i(2N+1)\xi^N_j} = 1, \qquad e^{\pm i(2N+1)\eta^N_j} = -1 \tag{20}$$

we have

$$\hat{f}^p_{k_1,2N+1-k_2} = \frac{1}{4M^2} \left(\sum_{j,l=0}^{2N} f\left(a^N_{i,j}\right) e^{-i\left(k_1\xi^N_j - k_2\xi^N_l\right)} + \sum_{j,l=0}^{2N} f\left(d^N_{i,j}\right) e^{-i\left(k_1\eta^N_j - k_2\xi^N_l\right)} - \sum_{j,l=0}^{2N} f\left(b^N_{i,j}\right) e^{-i\left(k_1\eta^N_j - k_2\eta^N_l\right)} - \sum_{j,l=0}^{2N} f\left(c^N_{i,j}\right) e^{-i\left(k_1\xi^N_j - k_2\eta^N_l\right)} \right)$$

$$= \frac{1}{4}\left(\hat{f}^a_{k_1,-k_2} + \hat{f}^d_{k_1,-k_2} - \hat{f}^b_{k_1,-k_2} - \hat{f}^c_{k_1,-k_2}\right).$$

The proof for $\hat{f}^{p}_{k_1,k_2-(2N+1)}$ is analogous to this one and so is the proof for the component of f in U_{MN}.

Our goal is to show that we can express all the components of f in the above decomposition by using only the values of f at the collocation grid $a^N_{i,j}$. The next lemma will give an explicit formula for the Fourier coefficients of f in terms of its collocation values $f(a^N_{i,j})$. We will illustrate the details of the proof only in the case $f \in U_{MN}$, where all modes involved are a combination of a high mode in the first variable and a low mode in the second variable (see Fig. 2). The same result can be shown for the other components using a similar argument.

LEMMA 3.1. *Consider a function $f(x, y)$ in U_{MN} of the form*

$$f(x,y) = \sum_{k_1=1}^{N+1}\sum_{k_2=0}^{N}\Big(\hat{f}^{p}_{N+k_1,k_2}e^{i(N+k_1,k_2)\cdot\mathbf{z}} + \hat{f}^{p}_{N+k_1,-k_2}e^{i(N+k_1,-k_2)\cdot\mathbf{z}} + \hat{f}^{p}_{-(N+k_1),k_2}e^{i(-(N+k_1),k_2)\cdot\mathbf{z}} + \hat{f}^{p}_{-(N+k_1),-k_2}e^{i(-(N+k_1),-k_2)\cdot\mathbf{z}}\Big),$$

where $\hat{f}^{p}_{k_1,k_2}$ is given in (13) and $\mathbf{z} = (x, y)$. Defining the new Fourier coefficients,

$$F_{k_1,k_2} = \hat{f}^{p}_{k_1-(2N+1),k_2}, \qquad F_{k_1,-k_2} = \hat{f}^{p}_{k_1-(2N+1),-k_2}, \tag{21}$$

$$F_{-k_1,k_2} = \hat{f}^{p}_{2N+1-k_1,k_2}, \qquad F_{-k_1,-k_2} = \hat{f}^{p}_{2N+1-k_1,-k_2}, \tag{22}$$

we can rewrite $f(x, y)$ in a simpler form,

$$f(x,y) = \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{N}} F_{k_1,k_2}e^{i(k_1-(2N+1),k_2)\cdot\mathbf{z}} + F_{-k_1,-k_2}e^{i((2N+1)-k_1,-k_2)\cdot\mathbf{z}} + F_{k_1,-k_2}e^{i(k_1-(2N+1),-k_2)\cdot\mathbf{z}} + F_{-k_1,k_2}e^{i(2N+1-k_1,k_2)\cdot\mathbf{z}}, \tag{23}$$

where the coefficients $F_{\mathbf{k}}$ are given by the formula

$$F_{\mathbf{k}} = \frac{1}{M^2}\sum_{j,l=0}^{2N} f\left(a^N_{j,l}\right)\frac{e^{-i\mathbf{k}\cdot a^N_{j,l}}}{\beta^0_{k_1}}, \tag{24}$$

where $\beta^0_{k_1} = 1 + \delta_{k_1,0}$ and $\mathbf{k} = (k_1, k_2)$.

Proof. First, we express $f(a^N_{j,l})$ in terms of the coefficients $F_{\mathbf{k}}$ above. So, inserting $a^N_{j,l}$ into (23) and using the identities in (20) we have

$$f\left(a^N_{j,l}\right) = \sum_{\mathbf{k}=-\mathbf{N}}^{\mathbf{N}} \beta^0_{k_1}F_{\mathbf{k}}e^{i\mathbf{k}\cdot a^N_{j,l}}, \tag{25}$$

with $\mathbf{N} = (N, N)$.

Now we multiply this equation by $e^{-i\mathbf{q}\cdot a^N_{j,l}}$, where $\mathbf{q} = (q_1, q_2)$, $|q_1|, |q_2| \leq N$ and sum over the grid $\{a^N_{j,l}\}$ to obtain

$$\sum_{j,l=0}^{2N} f\left(a^N_{j,l}\right)e^{-i\mathbf{q}\cdot a^N_{j,l}} = \sum_{\mathbf{k}=-\mathbf{N}}^{\mathbf{N}} \beta^0_{k_1}F_{\mathbf{k}}\sum_{j,l=0}^{2N} e^{i(\mathbf{k}-\mathbf{q})\cdot a^N_{j,l}}.$$

But, since

$$\sum_{j,l=0}^{2N} e^{i(\mathbf{k}-\mathbf{q})\cdot a_{j,l}^N} = \begin{cases} M^2, & \mathbf{k}=\mathbf{q} \\ 0, & \text{otherwise,} \end{cases}$$

we find

$$\sum_{j,l=0}^{2N} f\left(a_{j,l}^N\right) e^{-i\mathbf{q}\cdot a_{j,l}^N} = \beta_{q_1}^0 F_{\mathbf{q}} M^2.$$

The next lemma shows how the Lagrange interpolation operator G_{NM} corresponds to formula Q_N^a defined before.

LEMMA 3.2. *Let* $f(x,y) \in U_{MN}$ *be as in* (23); *then we can represent* f *by*

$$f(x,y) = \sum_{j,l=0}^{2N} f\left(a_{j,l}^N\right) h_{j,l}^{MN}(x,y), \tag{26}$$

where

$$h_{j,l}^{MN}(x,y) = \left(2H^{2M}\left(x,\xi_j^N\right) - H^{2N+1}\left(x,\xi_j^N\right)\right) H^{2N+1}\left(y,\xi_l^N\right). \tag{27}$$

Proof. As in the previous lemma, we will use the expression with the Fourier coefficients of f,

$$\begin{aligned} f(x,y) = \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{N}} & F_{k_1,k_2} e^{i(k_1-(2N+1),k_2)\cdot\mathbf{z}} + F_{-k_1,-k_2} e^{i((2N+1)-k_1,-k_2)\cdot\mathbf{z}} \\ & + F_{k_1,-k_2} e^{i(k_1-(2N+1),-k_2)\cdot\mathbf{z}} + F_{-k_1,k_2} e^{i(2N+1-k_1,k_2)\cdot\mathbf{z}} \end{aligned} \tag{28}$$

with the $F_{\mathbf{k}}$'s defined as in (21) and (22). Substituting into (28) the values for the Fourier coefficients in

$$\begin{aligned} f(x,y) = \frac{1}{M^2} \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{N}} \Bigg(& e^{i(k_1-(2N+1),k_2)\cdot\mathbf{z}} \sum_{j,l=0}^{2N} f\left(a_{j,l}^N\right) e^{i\mathbf{k}\cdot a_{j,l}^N} \\ & + e^{i(2N+1-k_1,-k_2)\cdot\mathbf{z}} \sum_{j,l=0}^{2N} f\left(a_{j,l}^N\right) e^{-i\mathbf{k}\cdot a_{j,l}^N} \\ & + e^{i(k_1-(2N+1),-k_2)\cdot\mathbf{z}} \sum_{j,l=0}^{2N} f\left(a_{j,l}^N\right) e^{i(k_1,-k_2)\cdot a_{j,l}^N} \\ & + e^{i(2N+1-k_1),k_2)\cdot\mathbf{z}} \sum_{j,l=0}^{2N} f\left(a_{j,l}^N\right) e^{-i(k_1,-k_2)\cdot a_{j,l}^N} \Bigg), \end{aligned}$$

and using (20) we obtain

$$\begin{aligned} f(x,y) = \frac{1}{M^2} \sum_{j,l=0}^{2N} f\left(a_{j,l}^N\right) \Bigg(& \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{N}} e^{i(k_1-(2N+1),k_2)\cdot\left(\mathbf{z}-a_{j,l}^N\right)} + \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{N}} e^{i(2N+1-k_1,-k_2)\cdot\left(\mathbf{z}-a_{j,l}^N\right)} \\ & + \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{N}} e^{i(k_1-(2N+1),-k_2)\cdot\left(\mathbf{z}-a_{j,l}^N\right)} + \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{N}} e^{i(2N+1-k_1),k_2)\cdot\left(\mathbf{z}-a_{j,l}^N\right)} \Bigg). \end{aligned}$$

Now, we separate the sums in k_1 and k_2 and rearrange the summation in k_1 as

$$f(x,y)=\frac{1}{M^2}\sum_{j,l=0}^{2N} f\left(a_{j,l}^N\right)\sum_{N+1\le|N+k_1|\le M}\frac{e^{i(N+k_1)\left(x-\xi_j^N\right)}}{\beta_{k_1}^N}\sum_{k_2=-N}^{N} e^{ik_2\left(y-\xi_l^N\right)}.$$

It is easily seen from the definitions of the Dirichlet kernels $H^{2M}(x,y)$ and $H^{2N+1}(x,y)$, (3) and (4), that

$$\frac{1}{M}\sum_{N+1\le|N+k_1|\le M}\frac{e^{i(N+k_1)\left(x-\xi_j^N\right)}}{\beta_{k_1}^N}=2H^{2M}\left(x,\xi_j^N\right)-H^{2N+1}\left(x,\xi_j^N\right).$$

This concludes the proof of the lemma.

Remark 3.1. Given a function $f\in U_{MN}$, its derivative with respect to x can be computed by using the interpolation formula (23),

$$\frac{\partial}{\partial x}f(x,y)=\sum_{j,l=0}^{2N} f\left(a_{j,l}^N\right)\frac{\partial}{\partial x}h_{j,l}^{MN}(x,y),$$

where

$$\frac{\partial}{\partial x}h_{j,l}^{MN}(x,y)=\frac{\partial}{\partial x}\left[2H^{2M}\left(x,\xi_j^N\right)-H^{2N+1}\left(x,\xi_j^N\right)\right]H^{2N+1}\left(y,\xi_l^N\right).$$

4. COMPUTATIONAL COSTS COMPARISON

In this section we compare the effectiveness of the NLG splitting with respect to the SCM in the case of spatial dimensions 1 and 2 when considering explicit integration in time. We use the same time step for both methods, therefore it is only necessary to compare their respective computational cost per iteration. As it was said before the use of distinct time steps for the high and low modes is the subject of a forthcoming paper (see [2]). All derivatives below are carried out by matrix-vector multiplications.

In the case of one space dimension we will consider the Burgers equation, a reaction-diffusion equation, and the Kuramoto–Shivashinsky equation. The details of the construction and approximating properties of the projection operators J_N and G_M can be found in [3].

Consider the Burgers equation

$$\begin{cases} u_t-\nu u_{xx}+\frac{1}{2}(u^2)_x=f, & x\in(0,2\pi),\ t>0,\\ u(0,t)=u(2\pi,t), & t>0.\end{cases} \tag{29}$$

The standard collocation scheme is based on the fine grid which contains $2M$ points. At each iteration we have to perform two derivatives, i.e., two matrix-vector multiplications. Therefore the number of operations per iteration necessary to evaluate the derivatives is $2(2M)^2=8M^2$. Since $M=2N+1$, the total cost per iteration for the SCM is $8M^2\approx 8(2N)^2=32N^2$ operations per time step.

In the case of NLG, we solve the system

$$\begin{cases} y_t - \nu y_{xx} + \dfrac{1}{2}\dfrac{\partial}{\partial x} J_N(y+z)^2 = J_N f, \\ z_t - \nu z_{xx} + \dfrac{1}{2}\dfrac{\partial}{\partial x} G_M(y+z)^2 = G_M f. \end{cases} \tag{30}$$

The projections J_N and G_M are carried out on the grid ξ_j^N with only $2N$ points, therefore each derivative takes $(2N)^2$ operations and the cost of computing y_{xx} and z_{xx} is $2(2N)^2 = 8N^2$. The nonlinear part of each equation is evaluated in the following way. First, we decompose $(y+z)^2$ in high and low modes by doing

$$\begin{aligned} J_N(y+z)^2\big(\xi_j^N\big) &= \frac{1}{2}(Q_N + \tilde{Q}_N)(y+z)^2\big(\xi_j^N\big) \\ &= \frac{1}{2}\big[(y+z)^2\big(\xi_j^N\big) + \tilde{Q}_N(y+z)^2\big(\xi_j^N\big)\big]. \end{aligned}$$

For this step, we only need to compute $\tilde{Q}_N(y+z)^2\big(\xi_j^N\big)$. This quantity is also used in the computation of the high modes

$$\begin{aligned} G_M(y+z)^2\big(\xi_j^N\big) &= (I_M - J_N)(y+z)^2\big(\xi_j^N\big) \\ &= (y+z)^2\big(\xi_j^N\big) - \frac{1}{2}\big[(y+z)^2\big(\xi_j^N\big) + \tilde{Q}_N(y+z)^2\big(\xi_j^N\big)\big] \\ &= \frac{1}{2}\big[(y+z)^2\big(\xi_j^N\big) - \tilde{Q}_N(y+z)^2\big(\xi_j^N\big)\big]. \end{aligned}$$

The derivatives of the nonlinear terms are carried out by 2 matrix-vector multiplications and using the identities

$$\frac{\partial}{\partial x} J_N(y+z)^2\big(\xi_j^N\big) = \frac{\partial}{\partial x} Q_N\big(J_N(y+z)^2\big(\xi_j^N\big)\big), \tag{31}$$

$$\frac{\partial}{\partial x} G_M(y+z)^2\big(\xi_j^N\big) = \sum_{j=0}^{2N} G_M(y+z)^2\big(\xi_j^N\big)\frac{\partial}{\partial x} h_j\big(\xi_j^N\big). \tag{32}$$

This takes $2(2N)^2 = 8N^2$ more operations, bringing the treatment of the nonlinear terms to $12N^2$ operations per iteration.

At the end of each iteration we have to project y and z on the other coarse grid η_j^N, via the operator Q_N, yielding another $8N^2$ operations. The total cost per iteration for the NLG is $28N^2$ operations. Using the same time step for both methods we save $4N^2$ operations per iteration, i.e., 12% of the operations when using NLG.

Remark 4.1. Starting with the values of y and z at the fine grid, the NLG requires fewer operations than SCM to evaluate the linear and nonlinear derivatives at one of the coarse grids. The projection of the y and z on the other coarse grid at the end of each iteration involves an additional $12N^2$ operations. However, this cost is fixed and does not depend on the equation. This indicates that the overall gain due to the NLG splitting will increase with the number of derivatives in the specific equation.

For example, in the case of the Kuramoto–Shivashinsky equation

$$\begin{cases} u_t + \nu \dfrac{\partial^4}{\partial x^4} u + \dfrac{\partial^2}{\partial x^2} u + u u_x = f, & x \in (0, 2\pi), t > 0, \\ u(0,t) = u(2\pi, t), & t > 0, \end{cases} \tag{33}$$

the NLG splitting in high and low modes costs less, since we have one more derivative than in (29). The total number of operations per iteration for the Kuramoto–Shivashinsky equation is $40N^2$ for NLG against the $48N^2$ for SCM.

Let us consider the Reaction-Diffusion equation

$$\begin{cases} u_t - \nu u_{xx} + u^3 - u = f, & x \in (0, 2\pi), t > 0, \\ u(0,t) = u(2\pi, t), & t > 0, \end{cases} \tag{34}$$

and the corresponding NLG splitting,

$$\begin{cases} y_t - \nu y_{xx} + \dfrac{\partial}{\partial x} J_N((y+z)^3 - (y+z)) = J_N f, \\ z_t - \nu z_{xx} + \dfrac{\partial}{\partial x} G_M((y+z)^3 - (y+z)) = G_M f. \end{cases} \tag{35}$$

Since the nonlinear part does not involve any derivatives, in the SCM we can just evaluate the terms at the grid points. On the other hand, for the NLG we still need to separate the nonlinear term in high and low modes and re-project, making this approach computationally more expensive.

The test problem in spatial dimension two is the following scalar Burgers-like equation

$$u_t - \nu u_{xx} - \nu u_{yy} + (u^2)_x + (u^2)_y = f, \qquad (x,y) \in (0, 2\pi)^2, t > 0 \tag{36}$$

with periodic boundary conditions.

This equation is a simple extension of the Burgers equation to 2 dimensions and it was chosen due to its simplicity and because it contains the important features of our analysis, which are the linear and nonlinear terms. Again, we will compare a NLG scheme with the 2D SCM based on the fine grid. Each directional derivative involves $(2M)^2 \times 2M = 8M^3$ operations. Since we have 4 derivatives, we have a total cost of $32M^3$ operations per iteration, or $32M^3 \approx 32(2N)^3 = 256N^3$ operations per iteration.

For the NLG scheme, we split Eq. (36) in four equations, each one containing respectively the low mode w (not called y anymore due to the y direction), the mixed modes z^1, z^2, and the pure high mode z^3. This splitting results from applying the operators J_{NN}, G_{NM}, G_{MN}, and G_{MM} to (36) leading to the following system of equations:

$$\begin{cases} w_t - \nu w_{xx} - \nu w_{yy} + J_N \left(\dfrac{\partial}{\partial x} + \dfrac{\partial}{\partial y} \right) (w + z^1 + z^2 + z^3)^2 = J_N f \\ z^1_t - \nu z^1_{xx} - \nu z^1_{yy} + G_{NM} \left(\dfrac{\partial}{\partial x} + \dfrac{\partial}{\partial y} \right) (w + z^1 + z^2 + z^3)^2 = G_{NM} f \\ z^2_t - \nu z^2_{xx} - \nu z^2_{yy} + G_{MN} \left(\dfrac{\partial}{\partial x} + \dfrac{\partial}{\partial y} \right) (w + z^1 + z^2 + z^3)^2 = G_{MN} f \\ z^3_t - \nu z^3_{xx} - \nu z^3_{yy} + G_{MM} \left(\dfrac{\partial}{\partial x} + \dfrac{\partial}{\partial y} \right) (w + z^1 + z^2 + z^3)^2 = G_{MM} f. \end{cases} \tag{37}$$

Analogously to the 1 dimensional case, besides computing the derivatives we have to project the quantities w, z^1, z^2, and z^3 on the remaining coarse grids at the end of each iteration.

The cost of computing a second derivative is $8N^3$. Since we have 8 of them, we spend $64N^3$ operations.

To compute the nonlinear terms we have first to decompose the quantity $(w + z^1 + z^2 + z^3)^2$ in its 4 components. This decomposition takes four matrix-vector multiplications and therefore costs $32N^3$ operations. After this, we still have to compute eight derivatives, adding $64N^3$ more operations. So, the cost for the treatment of the nonlinear terms is $96N^3$ operations per iteration.

The projection of each of the four quantities on the remaining three coarse grids takes $24N^3$ operations and so we will need $96N^3$ to project all of them.

The total cost per iteration to solve the system of equations (37) is also $256N^3$ operation, which is the same cost as the SCM.

5. NUMERICAL RESULTS

In this section we compare the numerical solutions obtained by applying the NLG and the SCM to the Burgers equations (29), (36) and the Reaction-Diffusion equation (34). Numerical tests prove that the solutions produced by the two methods show the same behavior when considering temporal and spatial accuracy. In this article we do not address the comparison of actual CPU time because a more involved study including code optimization and computer architecture characteristics is necessary to offer reliable and useful conclusions about pros and cons of each method (see [5]). Instead, we want to give motivation to such study by showing that the NLG splitting of modes generates virtually the same numerical solution as the SCM involving an equal or lesser number of operations per iteration. This also suggests that a different choice of time integrator for the low modes and the high modes equations will increase significantly the efficiency of the NLG.

We start by showing that one does not lose spatial precision when differentiating after the splitting of the modes. Table 1 shows the L^2 error in evaluating the first and second derivatives of the modes $\cos(Kx)$ by using the SCM and NLG approaches. We considered

TABLE 1
L^2 Error of Computed Derivatives Using SCM and NLG

K	SCM 1st D	NLG 1st D	SCM 2nd D	NLG 2nd D
1	0.1264E-13	0.1678E-13	0.5729E-12	0.3352E-12
2	0.1369E-13	0.1716E-13	0.6046E-12	0.3425E-12
3	0.1708E-13	0.1667E-13	0.6111E-12	0.4267E-12
4	0.1601E-13	0.2075E-13	0.6359E-12	0.4049E-12
14	0.4815E-13	0.4987E-13	0.5724E-12	0.4846E-12
15	0.4874E-13	0.4729E-13	0.5315E-12	0.4756E-12
16	0.6153E-13	0.6082E-13	0.9003E-12	0.9166E-12
17	0.6392E-13	0.6335E-13	0.7647E-12	0.8203E-12
30	0.1265E-12	0.1200E-12	0.3062E-11	0.3114E-11
31	0.1649E-12	0.1603E-12	0.2834E-11	0.2839E-11
32	0.1569E-12	0.1545E-12	0.3393E-11	0.3400E-11
33	0.2950E-12	0.2917E-12	0.7266E-12	0.6221E-12

a grid with 66 points, corresponding to $N = 16$ and $M = 33$, meaning that values of $K \geq 17$ represent the high modes for this grid. Notice that NLG and SCM show the same spatial resolution.

It is well known that roundoff error in evaluating derivatives increases with the number of collocation points and the order of the derivative (see [5, 6]). This is the reason why NLG performs slightly better on the low modes (which are the high energy ones) for the second derivative results. The computations for these modes are done using a matrix based on half of the number of points of the corresponding matrix for the SCM. This difference will become more explicit in the case of a Chebyshev collocation method where roundoff error increases much more drastically with the number of points than in the Fourier case.

In the following numerical experiments we present a general view of the accuracy behavior of NLG splitting with respect to the SCM by solving the equations of Section 4 and comparing the numerical solutions obtained by both methods. We use a fourth order Runge–Kutta method for all time integrations with the time step Δt given by

$$\Delta t = \frac{CFL}{\nu M^2},$$

where CFL is the stability constant. We will always use the same time step for both NLG and SCM. In most of the examples below an exact solution $u(x, t)$ is chosen in advance and the right-hand side f is determined accordingly. The L^2 error is then measured between the exact solution and the numerical solution with initial condition $u(x, 0)$.

Let's consider the Burgers equation (29) with an exact solution of the form

$$u(x, t) = e^{-\frac{t}{100}} \cos x. \tag{38}$$

Figure 3 shows the L^2 errors of NLG and SCM plotted against the number of points in the fine grid for a final time $t = 100$. Here we took $\nu = 0.001$ and $CFL = 0.01$. The results for NLG are plotted with crosses (+) and SCM with circles (○).

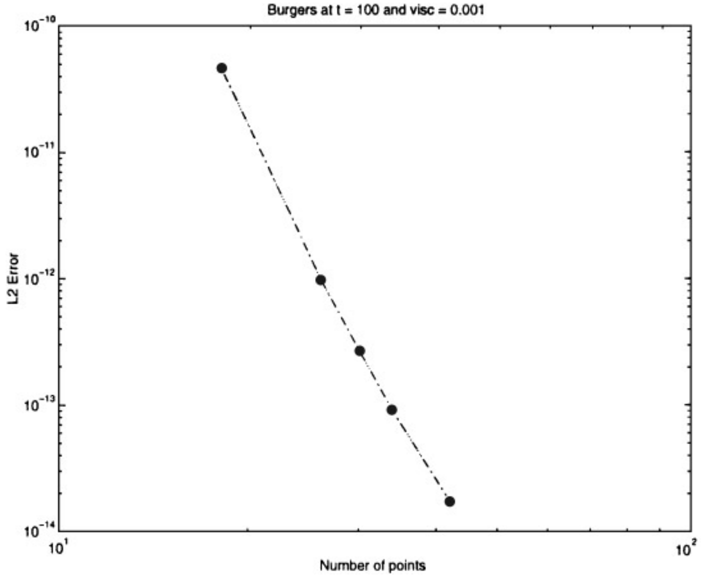


FIG. 3. L^2 errors for the Burgers equation with $\nu = 0.001$. $*$, SCM; $\circ$, NLG.

TABLE 2
L^2 Error for the Burgers Equation

Pts	SCM	NLG	Order
18	4.601342004318910E-011	4.601225192272278E-011	
26	9.860888789122200E-013	9.859275305004013E-013	5.22
30	2.711686788383684E-013	2.712609741508736E-013	4.51
34	8.975752599721587E-014	8.955637530818245E-014	4.41
42	1.756242681346713E-014	1.755386246065911E-014	3.86

As we can see from Fig. 3, there is no difference in the numerical results of NLG and SCM and Table 2 shows that the order of decay of the error is the one of the Runge–Kutta method employed for the temporal integration. The decreasing in the order is due to a contamination of the results by roundoff errors (see Table 1).

This first experiment, however, involved only low modes for all the grids tested. A more significant example is Eq. (29) with the exact solution

$$u(x,t) = e^{-\frac{t}{1000}}(\cos x + \delta \cos kx),$$

where k will be set up to represent a high mode and δ its amplitude. For this second experiment, we consider a grid with 34 points, yielding $M = 17$ and $N = 8$, and values of $k \geq 9$ represent high modes for this grid. The viscosity and CFL values are taken as before with $k = 17$ and $\delta = 0.01$. This amplitude mimics a natural distribution of energy in realistic problems, where high modes carry less energy than low modes. Figure 4 shows the L^2 error as time evolves for both methods from $t = 0$ to $t = 1000$. Notice that the magnitude of the error is bigger in this case since the nonlinearity in (29) gives rise to the mode $\cos(34x)$ which cannot be well represented by the grid. Once again it is not possible to distinguish one result from the other throughout the whole process of temporal integration. It was to stress this fact that in Fig. 4 we plotted the error against the number of iterations.

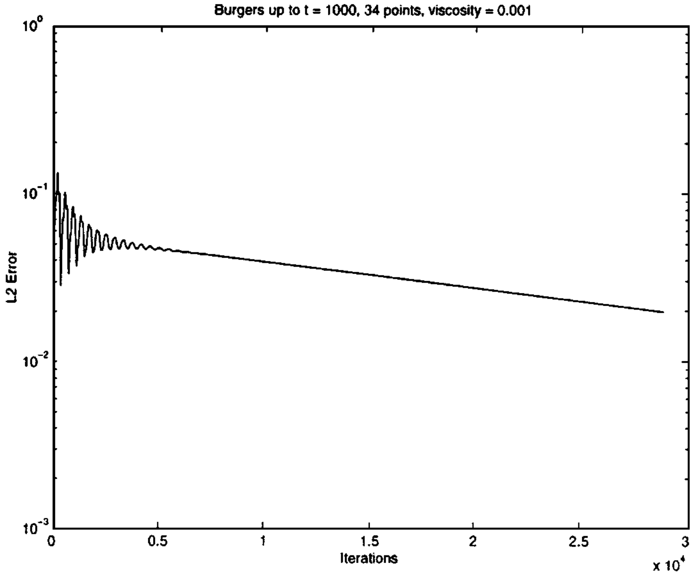


FIG. 4. L^2 error for Burgers with $\nu = 0.001$ and 34 points. The lines representing SCM and NLG are coincident.

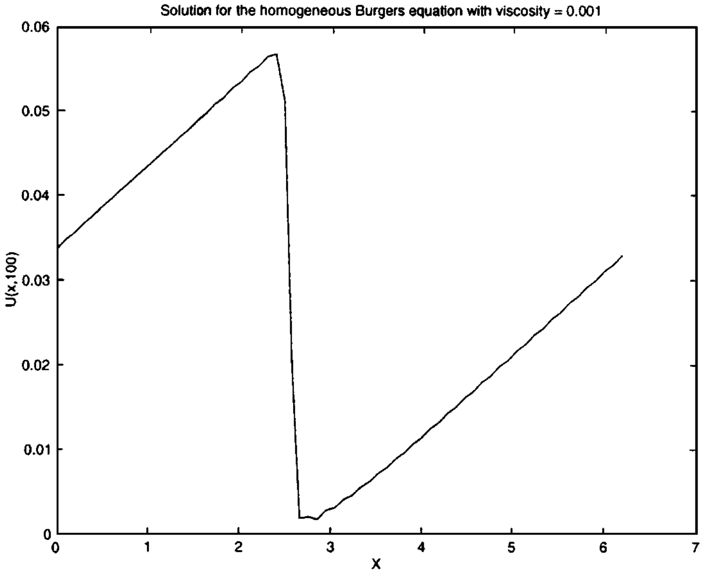


FIG. 5. Solution for the homogeneous Burgers at $t = 100$. Lines for SCM and NLG are coincident.

For a more realistic test, where one does not control which modes are, or are not, present in the numerical simulation, we consider Eq. (29) in its homogeneous form, i.e., with $f = 0$, and compare how both methods recover the classical sinusoidal shape of its solution from the Gaussian-like initial condition

$$u_0 = e^{-100(x-\pi)^2}$$

when $\nu = 0.001$ at $t = 100$. In this case a grid with 66 points was used in order to have sufficient spatial resolution to capture the steep gradients from the solution. The value for $CFL = 0.01$ still yielded stability. Once more the same numerical solutions were obtained and both are shown in Fig. 5.

The experiment for the Reaction-Diffusion equations (34) and (35) consists in considering an oscillatory exact solution of the form

$$u(x, t) = \cos(t)\cos(x),$$

with the corresponding forcing term determined by it. The numerical solutions were computed up to $t = 1$ using the parameters values $CFL = 0.01$ and $\nu = 0.001$. Figure 6 and Table 3 show that we achieved the same quality of results as for the Burgers equation, i.e., the NLG splitting generates the same numerical solution as the SCM.

TABLE 3

L^2 Error for the Reaction-Diffusion Equation

Pts	SCM	NLG	Order
18	2.685663324028776E-006	2.685663324295888E-006	
30	4.009815791896255E-008	4.009815786416866E-008	4.1153
42	2.702694205085399E-009	2.702694304005631E-009	4.0079
90	5.936846844181185E-012	5.936861229501006E-012	4.0156
126	4.075862407295771E-013	4.076214398280533E-013	4.2876

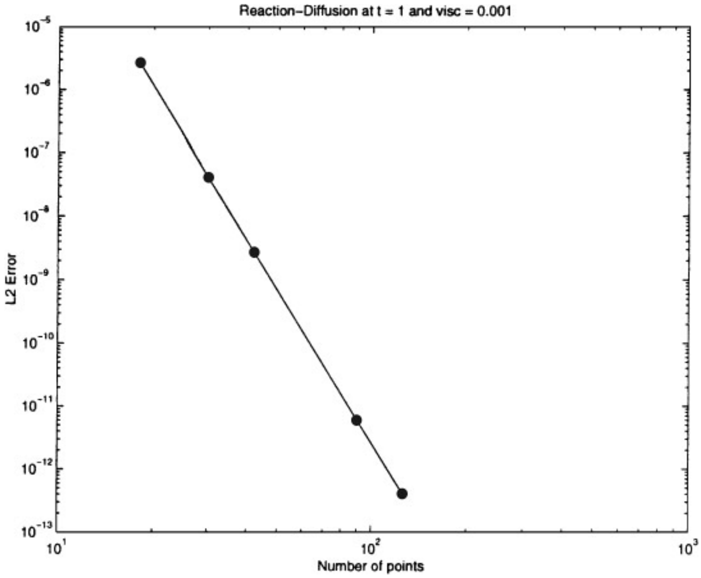


FIG. 6. L^2 errors for the Reaction-Diffusion equation. $*$, SCM; $\circ$, NLG.

For the case of 2 dimensions in space we consider Eq. (36) with an exact solution of the form

$$u(x,t) = e^{-t}e^{-40((x-\pi)^2+(y-\pi)^2)}.$$

As before, this function provides a natural energy distribution throughout its spectrum with low modes carrying most of the energy, but still keeping the importance of the high modes for the accuracy of the numerical solution. We integrated (36) and (37) up to $t = 1$ with

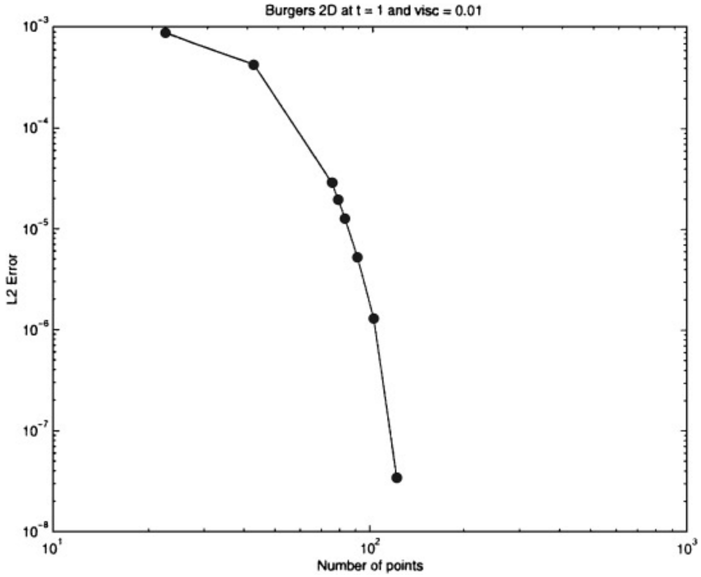


FIG. 7. L^2 errors for Burgers 2D with $\nu = 0.001$ at $t = 1$. $*$, SCM; $\circ$, NLG.

TABLE 4
L^2 Error for the 2D Burgers Equation

Pts	SCM	NLG	Order
22	8.635929773499972E-004	8.635929773499946E-004	
42	4.173275481643102E-004	4.173275481643098E-004	0.5623
74	2.889167206302692E-005	2.889167206302680E-005	2.3573
78	1.950972747131406E-005	1.950972747131406E-005	3.7292
82	1.272719803347214E-005	1.272719803347162E-005	4.2708
90	5.299905987531454E-006	5.299905987532374E-006	4.7054
102	1.288118941678376E-006	1.288118941678128E-006	5.6506
122	3.403030329596673E-008	3.403030329609838E-008	10.1472

$\nu = 0.01$ and $CFL = 0.01$. Figure 7 shows the norm of the L^2 error plotted against the number of points. Note that convergence is poor at the lower number of points when the grid is not fine enough to solve the steep gradients generated by the Gaussian solution and forcing terms. The refinement of the grid improves its ability to capture those gradients and leads to the fourth order convergence of the Runge–Kutta for the intermediate values of number of points, as we can see in Table 4. The further refinement of the grid yields a super-convergence order when Δt becomes small enough so that the spatial errors start to dominate. Nonetheless, the NLG splitting solution was the same as the SCM under all the different situations above described.

6. CONCLUSIONS

In this article a Nonlinear Galerkin type splitting in the case of a Fourier pseudospectral discretization in spatial dimension 2 was introduced and some of its numerical features as spatial precision and computational costs were analyzed by means of the Burgers and Reaction-Diffusion equations in 1 and 2 spatial dimensions.

From the numerical experiments here presented we conclude that the NLG splitting in high and low modes is a very attractive technique in solving partial differential equations via a pseudospectral discretization. It maintains the spatial precision and the computational cost of the standard method while opening the way to the application of more refined treatments that distinguishes between high and low energy quantities.

It clearly deserves further study with respect to its application to more realistic problems and the use of different basis of functions as the Chebyshev and Legendre ones. In subsequent works we will explore the NLG feature of distinguishing between low and high modes and also its potential for easy parallelization and reduction of the roundoff error.

ACKNOWLEDGMENTS

The authors thank David Gottlieb and Roger Temam for the fruitful discussions and valuable suggestions to this work. The first author was partly supported by the CNPq, Brazil, under Grant 200431/93-5; by the National Science Foundation under Grant NSF-DMS-9400615; and by the Research Fund of Indiana University. The second author was partly supported by AFOSR Grant F49620-96-1-0150 and NSF Grant DMS-9500814.

REFERENCES

1. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York/Berlin, 1988).

2. B. Costa, L. Dettori, D. Gottlieb, and R. Temam, Time marching techniques for the nonlinear Galerkin method, in preparation.
3. L. Dettori, D. Gottlieb, and R. Temam, A nonlinear Galerkin method: The two-level Fourier-collocation case, *J. Sci. Comput*. **10**(4), 371 (1995).
4. L. Dettori, D. Gottlieb, and R. Temam, A nonlinear Galerkin method: the two-level Chebyshev-collocation case, *Houston J. Math.*, 75 (1996).
5. W. S. Don and A. Solomonoff, Accuracy and speed in computing the Chebyshev collocation derivative, *SIAM J. Sci. Comput.* **16**(6), 1253 (1995).
6. W. S. Don and A. Solomonoff, Accuracy enhancement for higher derivatives using Chebyshev collocation and a mapping technique, *SIAM J. Sci. Comput*. **18**(4), (1997).
7. F. Jauberteau, C. Rosier, and R. Temam, A nonlinear Galerkin method for the Navier–Stokes equations, *Comput. Meth. Appl. Mech. Eng.* **8**, 245 (1990).
8. T. Dubois, F. Jauberteau, M. Marion, and R. Temam, Subgrid modelling the interaction of small and large wavelengths in turbulent flows, *Comput. Phys. Comm.* **65**, 100 (1991).
9. C. Devulder, M. Marion, and E. S. Titi, On the rate of convergence of nonlinear Galerkin methods, *Math. Comp.* **60**, 495 (1993).
10. A. Doelman and E. S. Titi, Regularity of solutions and the convergence of the Galerkin method in the Ginzburg–Landau equation, *Numer. Funct. Anal. Optim.* **14**, 299 (1993).
11. D. A. Jones, L. G. Margolin, and E. S. Titi, On the effectiveness of the approximate inertial manifolds—A computational study, *Theoret. Comput. Fluid Dynam.* **7**, 243 (1995).
12. B. García-Archilla and J. de Frutos, Time integration of the nonlinear Galerkin method, *IMA J. Numer. Anal.* **14**, 1 (1994).
13. B. García-Archilla, J. Novo, and E. S. Titi, Postprocessing the Galerkin method: A novel approach to approximate inertial manifolds, *SIAM J. Sci. Comput.*, in press.
14. R. Wallace and D. Sloan, Numerical solutions of a nonlinear dissipative system using pseudospectral method and inertial manifolds, *SIAM J. Sci. Comput.* **16**, 1049 (1995).
15. D. Gottlieb, M. Y. Hussaini, and S. A. Orzag, Theory and application of spectral methods, in *Spectral Methods for Partial Differential Equations*, edited by R. Voigt, D. Gottlieb, and M. Y. Hussaini (SIAM-CBMS, Philadelphia, PA, 1984), p. 1.
16. P. J. Davis and P. Rabinovitz, *Methods of Numerical Integration* (Academic Press, New York, 1975).