



ELSEVIER



Postprocessing Fourier spectral methods: The case of smooth solutions

Bosco García-Archilla^a, Julia Novo^{b,*}, Edriss S. Titi^c

^a *Departamento de Matemática Aplicada II, Universidad de Sevilla, Escuela Superior de Ingenieros,
Camino de los Descubrimientos, s/n 41092 Sevilla, Spain*

^b *Departamento de Matemática Aplicada y Computación, Universidad de Valladolid, Valladolid, Spain*

^c *Department of Mathematics, and Department of Mechanical and Aerospace Engineering, University of California,
Irvine, CA 92697-3875, USA*

Abstract

A postprocessing technique to improve the accuracy of Galerkin methods, when applied to dissipative partial differential equations, is examined in the particular case of very smooth solutions. Pseudospectral methods are shown to perform poorly. This performance is studied and a refined postprocessing technique is proposed.

© 2002 IMACS. Published by Elsevier Science B.V. All rights reserved.

Keywords: Dissipative equations; Spectral methods; Aliasing errors; Multilevel methods

1. Introduction

In recent years, a simple and computationally inexpensive algorithm to improve the accuracy of Galerkin methods, for dissipative partial differential equations (PDEs), has been developed. In this paper, we review this technique, the so-called “Postprocessing Galerkin Method”, and examine its performance when the solutions of the PDE are in a certain Gevrey class of regularity (see [4,5,12,14,18,33–36,42,43] and references therein for discussions and examples concerning Gevrey regularity).

The postprocessing Galerkin method is of value when dealing with dissipative PDEs. Such evolution equations can be written in the abstract form

$$\frac{du}{dt} + \nu Au + F(u) = f, \quad t \in (0, T], \quad (1)$$

* Corresponding author.

E-mail address: jnovo@mac.cie.uva.es (J. Novo).

in an appropriate Hilbert space H with norm $\| \cdot \|$ and inner product (\cdot, \cdot) . Usually H is $L^2(\Omega)$ for a bounded domain Ω with smooth enough boundary. The solutions of (1) are determined uniquely by the initial condition

$$u(0) = u_0. \tag{2}$$

In (1), the operator A is typically the Laplacian operator $-\Delta$ or the biharmonic operator Δ^2 under appropriate boundary conditions, F gathers the nonlinear terms and the lower order derivatives, f is a forcing term independent of u , and ν is a positive diagonal matrix, however, for simplicity we will consider here only the scalar case. In general, the operator A is a densely-defined, unbounded, self-adjoint positive operator with compact inverse (see, however, [42,43]). Therefore the Hilbert space H has an orthonormal basis $\{w_1, w_2, \dots\}$ of eigenfunctions of the operator A . That is, $Aw_j = \lambda_j w_j$, for $j = 1, 2, \dots$, with $\|w_j\| = 1$. The corresponding eigenvalues are ordered such that $0 < \lambda_1 \leq \lambda_2 \leq \dots$. As a result, for every $v \in H$ we have

$$v = \sum_{j=1}^{\infty} \tilde{v}_j w_j, \quad \tilde{v}_j = (v, w_j), \quad j = 1, 2, \dots$$

Furthermore, we consider for $s \in \mathbb{R}$

$$D(A^s) = \left\{ v = \sum_{j=1}^{\infty} \tilde{v}_j w_j \in H \mid \sum_{j=1}^{\infty} \lambda_j^{2s} |\tilde{v}_j|^2 < \infty \right\},$$

and

$$A^s v = \sum_{j=1}^{\infty} \lambda_j^s \tilde{v}_j w_j.$$

Dissipative equations are characterized by the following property: There exist constants $M_0 > 0$ and $M_1 > 0$ such that for any $R > 0$, there exists a positive time $T_0 = T_0(R, \nu)$ such that for $u_0 \in H$ with $\|u_0\| < R$, the solution u of (1), (2) satisfies,

$$\|u(t)\| < M_0, \quad \|A^{1/2}u(t)\| < M_1, \quad t \in [T_0, \infty). \tag{3}$$

That is, in dissipative evolution equations the solutions become eventually bounded, and in particular the possibility that the solutions blow up in finite time is excluded. Let us observe that depending on the regularity of the forcing term (and the nature of Eq. (1)), the solution $u(t)$ is usually more regular than what (3) suggests, and typically for $t > 0$, $u(t) \in D(A^s)$ for some $s \geq 1$ or even $u(t) \in D(e^{\gamma A^{\sigma/2}})$ for some $\gamma > 0$ and where $\sigma = 1$ when $A = -\Delta$ and $\sigma = \frac{1}{2}$ when $A = \Delta^2$ (see, e.g., [5,12,14,18,35]). Typical examples of dissipative PDEs include model equations like:

- (i) The reaction–diffusion (RD) equation (real forced Ginzburg–Landau equation)

$$u_t - \nu u_{xx} - u + u^3 = f, \tag{4}$$

in $\Omega = [0, \pi]$ with boundary conditions $u(0, t) = u(\pi, t) = 0$.

- (ii) The Kuramoto–Sivashinsky equation

$$u_t + 4u_{xxxx} + \lambda(u_{xx} + uu_x) = 0, \tag{5}$$

in $\Omega = [-\pi, \pi]$ with periodic boundary conditions. For simplicity we will consider here the case of odd periodic solutions, i.e., $u(-x, t) = -u(x, t)$.

(iii) The two-dimensional Navier–Stokes (NS) equations

$$\begin{cases} u_t + (u \cdot \nabla)u + \nabla p = \nu \Delta u + g, \\ \operatorname{div}(u) = 0 \end{cases} \tag{6}$$

in a domain $\Omega \subset \mathbb{R}^2$, with no slip boundary condition $u = 0$ on $\partial\Omega$.

Other well-known instances of dissipative PDEs that fall into the framework (1), are viscous Burgers’ equation, the Cahn–Hilliard equation, the complex Ginzburg–Landau equation, etc. (see, e.g., [45] for these and other dissipative equations).

Since we will present examples related to (4) and (5), we treat in detail the framework of (1) for these equations (for the NS equation, see, e.g., [6,45]). The Hilbert space H is $H = L^2(0, \pi) = \{v : [0, \pi] \rightarrow \mathbb{R} \mid \int_0^\pi |v(x)|^2 dx < \infty\}$, in the case of (4), and in the case of (5) we restrict ourselves to $H = L^2_{\text{per, odd}} = \{v : \mathbb{R} \rightarrow \mathbb{R} \mid v(x) = v(x + 2\pi) \text{ and } v(-x) = -v(x), \text{ for almost all } x \in \mathbb{R}, \int_0^\pi |v(x)|^2 dx < \infty\}$. In both cases we consider the norm

$$\|v\| = \left(\int_0^\pi |v(x)|^2 dx \right)^{1/2},$$

with the associated inner product

$$(v, w) = \int_0^\pi v(x)w(x) dx.$$

In the case of the RD equation, the operator A is $A = -\partial_{xx}$ under homogeneous Dirichlet boundary conditions, and thus $\lambda_j = j^2, j = 1, 2, \dots$, and in the case of the KS equation $A = \partial_{xxxx}$ restricted to $L^2_{\text{per, odd}}$ and $\lambda_j = j^4, j = 1, 2, \dots$. In both cases the set

$$w_j(x) = \sqrt{\frac{2}{\pi}} \sin(jx), \quad j = 1, 2, \dots,$$

forms a complete orthonormal set for H consisting of the eigenfunctions of the operator A . In the case of the RD equation, $V = D(A^{1/2})$ coincides with the Sobolev space $H_0^1(0, \pi)$ of square-integrable functions v with first derivative in $L^2(0, \pi)$ and $v(0) = v(\pi) = 0$ (in the trace sense). In the case of the KS equation, V is the Sobolev space $H^2_{\text{per, odd}}$, the space of functions $v \in L^2_{\text{per, odd}}$ with $\partial_x v \in L^2_{\text{per, even}}$ (defined similarly to $L^2_{\text{per, odd}}$) and $\partial_{xx} v \in L^2_{\text{per, odd}}$.

The postprocessing technique, which we developed in [19,25–27], applies to Galerkin methods for (1), (2). In a Galerkin method, the first requirement is a sequence of *finite-dimensional* subspaces $W_m, m = 1, 2, \dots$, of V , with $\bigcup_{m=1}^\infty W_m$ dense in H . Typical examples are the cases where W_m is composed of trigonometric polynomials of degree at most m in the case of Fourier spectral methods, of polynomials of degree at most m in the case of spectral methods based on Legendre polynomials, or of piecewise polynomials of degree at most r based on a partition of Ω into m elements, in the case of finite element methods. In this paper and in the case of (4) and (5) we will consider Galerkin methods based on

$$W_m = H_m = \operatorname{span}\{\sin(jx) \mid j = 1, \dots, m\} \tag{7}$$

(i.e., we will be considering Fourier spectral methods).

Let us denote by $P_m : H \rightarrow W_m$ the orthogonal projection of H onto W_m , which, for $v \in H$ is defined by

$$(v, w) = (P_m v, w), \quad \forall w \in W_m.$$

In the particular case of (4) and (5), notice that P_m is nothing else than the truncation of the Fourier series, that is, for

$$v(x) = \sqrt{\frac{2}{\pi}} \sum_{j=1}^{\infty} \tilde{v}_j \sin(jx), \quad (P_m v)(x) = \sqrt{\frac{2}{\pi}} \sum_{j=1}^m \tilde{v}_j \sin(jx).$$

In the Galerkin discretization of (1), (2) one considers the following family of approximating problems: find $u_m : [0, T] \rightarrow H_m$ such that

$$\frac{du_m}{dt} + v P_m A u_m + P_m F(u_m) = P_m f, \quad (8)$$

for $t \in (0, T]$ with the initial value

$$u_m(0) = P_m u_0. \quad (9)$$

The Postprocessing Algorithm. Next we describe the postprocessing technique introduced in [25–27]. Suppose we are interested in the solution of (1), (2) at a given time T , that is, $u(T)$.

- (i) Integrate first (8), (9) up to time T .
- (ii) Then solve the following *linear* elliptic problem: Find $\tilde{u} \in V$ such that

$$v A \tilde{u} = f(T) - F(u_m(T)) - \frac{d}{dt} u_m(T). \quad (10)$$

- (iii) The solution \tilde{u} found in the previous step is the new approximate solution.

The new approximation \tilde{u} is in general more accurate than the Galerkin approximation $u_m(T)$, in the sense that $\|u(T) - \tilde{u}\| \ll \|u(T) - u_m(T)\|$, as $m \rightarrow \infty$. The smallness of $\|u(T) - \tilde{u}\|$ with respect to $\|u(T) - u_m(T)\|$ depends on how large m is, the type of equation, and the regularity of the solution u of (1), (2). It is well-known that if the solution u of (1), (2) satisfies $\max_{0 \leq t \leq T} \|A^s u(t)\| < \infty$, then the Galerkin approximation u_m , for m sufficiently large, satisfies the bound

$$\max_{0 \leq t \leq T} \|u(t) - u_m\| \leq \frac{C}{\lambda_{m+1}^s}, \quad (11)$$

where $C = C(\max_{0 \leq t \leq T} \|A^s u(t)\|)$. For the approximation \tilde{u} the following result is proved in [25].

Theorem 1. *Let u be the solution of the RD equation or the KS equation with $u(0) = u_0$. Then, if $\max_{0 \leq t \leq T} \|A^s u(t)\| < \infty$, there exist positive constants $C = C(\max_{0 \leq t \leq T} \|A^s u(t)\|)$ and m_0 such that for $m \geq m_0$ the postprocessed Galerkin approximation \tilde{u} solution of (10) satisfies*

$$\|u(T) - \tilde{u}\| \leq \frac{C}{\lambda_{m+1}^{s+\beta}}, \quad (12)$$

where $\beta = 1$ in the case of the RD equation and $\beta = 3/4$ in the case of the KS equation.

Remark 1. Similar results are proved in [25] for other model equations such as Burgers’ equation, the Cahn–Hilliard equation and the two-dimensional Navier–Stokes equations under periodic boundary conditions, when discretized with Fourier spectral methods. For other boundary conditions we refer the reader to [26]. The postprocessing of spectral methods based on Chebyshev and Legendre polynomials is studied in [19,22], and that of finite-element methods in [27]. More refined postprocessing techniques are developed in [41]. The case of spectral elements is studied in [20,21].

Remark 2. Recently, it has been shown in [38] that the postprocessing Galerkin scheme, described above, arises in a very natural way through a classical truncation analysis of the underlying dissipative evolution equation. More specifically, it is shown that, to leading order, the correct approximative scheme is actually the postprocessed Galerkin method (8)–(10), and not the standard Galerkin method (8), (9), as is commonly believed. Consequently, the new approximation \tilde{u} can be expected to be more accurate than the Galerkin approximation $u_m(T)$.

Thus, solving the linear problem (10) brings a new approximation \tilde{u} with an improved convergence rate with respect to the Galerkin method. It must be noticed that in practice, (10) is never solved, since its solution belongs to the infinite-dimensional space V . Rather its solution \tilde{u} is approximated by $\tilde{u}_{m_1} \in W_{m_1}$ obtained by solving

$$\nu P_{m_1} A \tilde{u}_{m_1} = P_{m_1} (f(T) - F(u_m(T))) - \frac{d}{dt} u_m(T). \tag{13}$$

Notice that (13) is the Galerkin method, of order m_1 , applied to (10). Accordingly, the value of m_1 is chosen so that the error in solving (13) instead of (10) is not larger than the error of the postprocessed Galerkin method. In view of Theorem 1 and the error bound (11) a sensible choice is $m_1 = m^{l+\beta/s}$.

An idea of the better accuracy of the postprocessed Galerkin method can be seen in the results shown in Fig. 1. They correspond to the RD equation (4) with initial condition $u_0(x) = \sin(x)$ and forcing term f taken as

$$f(x, t) = \sin(t) f_0(x), \quad f_0(x) = \begin{cases} 0, & 0 \leq x \leq \pi/4, \\ (4/\pi)x - 1, & \pi/4 < x \leq \pi/2, \\ 3 - (4/\pi)x, & \pi/2 < x \leq 3\pi/4, \\ 0, & 3\pi/4 < x \leq \pi. \end{cases} \tag{14}$$

The purpose of choosing this forcing term is that the regularity of solution u is not very high. For this forcing term, $\max_{0 \leq t \leq T} \|A^s u(t)\| < \infty$ for $s < 7/4$ but not higher, so that the effect of the extra exponent $\beta = 1$ in the error bound (12) of the postprocessed Galerkin method is more noticeable with respect to the bound (11) of the Galerkin method. We make the forcing term time-dependent so that the solution u does not evolve to a steady state.

Fig. 1 is a convergence diagram depicting the errors committed by the methods when used with different values m of Fourier modes. In the horizontal axis the value $N = m + 1$ is represented. Each pair (N, error) is marked with an asterisk in the plot. Pairs corresponding to a given method are joined by segments, so that we get an idea of the results for values of N different from those represented in the diagram. Results corresponding to the standard Galerkin method are joined by a continuous line, and those of the postprocessed Galerkin method by a discontinuous line. It can be seen that, for a given value of N , the postprocessed method commits error substantially smaller than the Galerkin method (ten times smaller on average), and that the difference between the two methods increases as N grows.

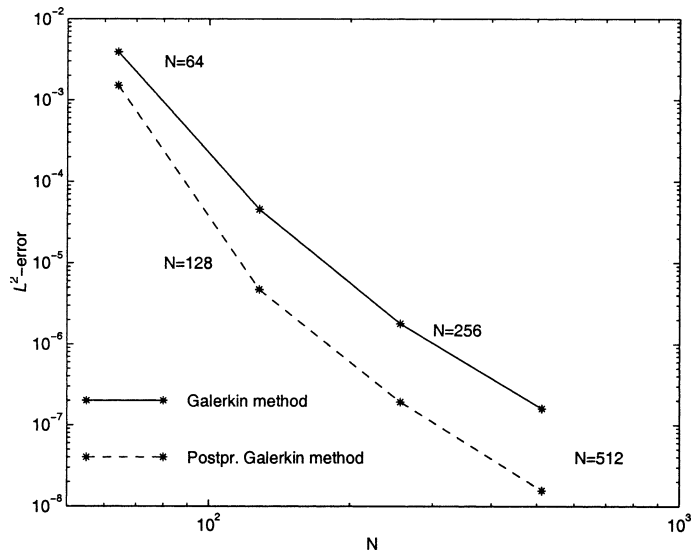


Fig. 1. Convergence diagram for the RD equation. Errors at $T = 20$.

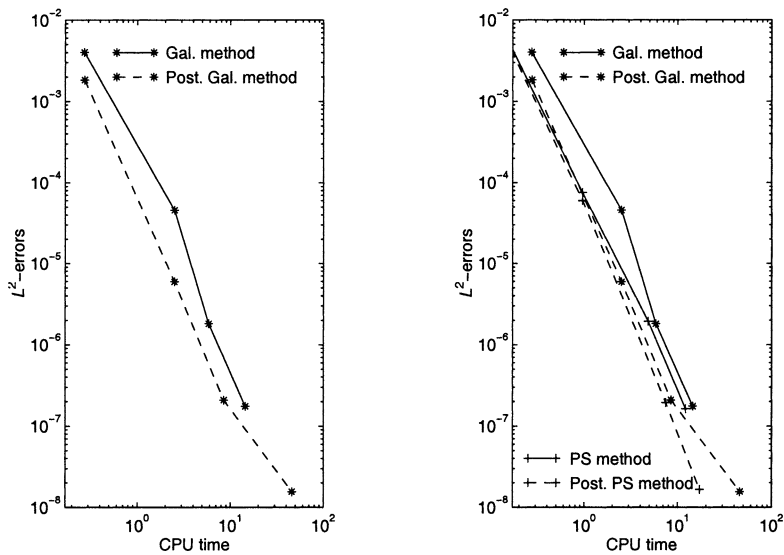


Fig. 2. Efficiency diagram for the RD equation. Errors at $T = 20$.

In Fig. 2 (left diagram) we have represented the same errors as in Fig. 1 but plotted against the smallest amount of computing time that it took to achieve them (on a SUN Ultra-1 Mod. 140). The purpose is to see that the accuracy enhancement brought by the postprocessing step (10) is done at negligible computational cost. As mentioned before, once $u_m(T)$ is available, the cost of obtaining \tilde{u} is that of solving the linear problem (10) (or, in practice, (13)) which is negligible as compared with the CPU time spent in obtaining $u_m(T)$. Observe that the system of ordinary differential equations (8) cannot in general

be solved exactly, and has to be approximated by using a numerical time integrator. We made sure that the error arising from time integrating was negligible as compared to the Galerkin error $\|u(T) - u_m(T)\|$. The time integrator used in all the experiments shown in this paper was VODE [2] with internally computed diagonal Jacobians (VODE option MF=23). The efficiency of time integrators like VODE when dealing with discretizations of dissipative equations was studied in [23] (see also [24]). The main parameter to VODE is the tolerance below which local *time* discretization errors are desired. For every value of m , we ran VODE with different values of the tolerance and measured errors to be sure that time-integration errors were not dominant. For example, the solution taken as exact was computed with $m = 8191$ and with VODE tolerances 10^{-11} and 10^{-13} . The norm of difference between the two computed solutions turned out to be smaller than 10^{-12} , much smaller than any of the errors shown in Figs. 1 and 2. For a given value of m , once the smallest error achieved with that value of m was determined, we looked for the largest tolerance (i.e., smaller cost) with which that error (with the same value of m) was achieved. The resulting cost was selected for Fig. 2.

It can be seen in Fig. 2 that the results of the postprocessed method (discontinuous line) are on the left of those of the Galerkin method. This means that for a given accuracy, the combination of Galerkin method plus postprocessing requires less computing cost to achieve that accuracy than the Galerkin method. The results show that the postprocessed Galerkin method is, for a given accuracy, between 2 and 3/2 times faster than the Galerkin method.

In Fig. 2 (right diagram) the results corresponding to the pseudospectral (PS) method and the postprocessed pseudospectral method are added to those of the Galerkin and postprocessed Galerkin method. We remark that in practice, the Galerkin or spectral method is hardly used, since computing the projection $P_m F(u_m)$ of the nonlinear terms in (8) is too costly. Rather, the PS method is used. In the PS method the projection $P_m F(u_m)$ is replaced by the trigonometric interpolant $I_m(F(u_m))$ based on the nodes $j\pi/N$, $j = 1, \dots, N - 1$, since this procedure is much cheaper than the projection P_m . Thus in the PS method, (8) is replaced by

$$\frac{du_m}{dt} + \nu P_m A u_m + I_m F(u_m) = P_m f. \quad (15)$$

The error committed in replacing $P_m F(u_m)$ by $I_m(F(u_m))$ is known as aliasing error (see, e.g., [3]). The influence of the aliasing error is in most cases negligible or so small that the smaller costs of the PS methods compensates for it, so that the PS method (although less accurate) is computationally more efficient than the Galerkin method. This can be seen in Fig. 2 where the results of the PS method, which are represented by crosses joined by continuous lines, are on the left of those of the Galerkin method (asterisks with continuous lines), meaning that the PS method requires less cost to achieve a given accuracy. It can also be seen that the postprocessed PS method (crosses joined by discontinuous lines), is the most efficient method of the four depicted. In fact, Theorem 1 is also valid when u_m is the PS approximation.

The rest of this paper is as follows. In Section 2 we review the postprocessed Galerkin method. Section 3 is devoted to study the performance of the methods when dealing with problems whose solutions are very smooth (real analytic with uniform radius of analyticity). The last section contains the conclusions.

2. Review of the postprocessing method

2.1. Defect-correction techniques

The postprocessed Galerkin method is related to classical defect-correction techniques for steady state problems. To better appreciate this fact, suppose that we are dealing with a steady state problem, so that instead of (1) and (8) we have

$$\nu Av + R(v) = f, \quad v \in V, \quad (16)$$

$$\nu P_m Av_m + P_m R(v_m) = P_m f, \quad v_m \in W_m. \quad (17)$$

Then, the postprocessed Galerkin method would be:

- (1) Find first the Galerkin approximation $v_m \in W_m$ by solving (17).
- (2) Find $\tilde{v} \in V$ by solving $\nu A\tilde{v} = f - R(v_m)$.

Thus, the whole technique amounts to solve in step (1) the full nonlinear problem in the simpler space W_m where the nonlinearities are more easily treated, and then compute a correction in step (2). Furthermore, once \tilde{v} is computed, a better approximation $\tilde{\tilde{v}}$ can be computed by iteration, i.e., by replacing v_m by \tilde{v} in the nonlinear terms R . The process can be further iterated until a prescribed accuracy is reached, or one can use Newton's iteration to converge faster to v . Again, \tilde{v} is never computed in practice but is replaced by its Galerkin approximation $\tilde{v}_{m_1} \in W_{m_1}$ found by solving $\nu P_{m_1} A\tilde{v}_{m_1} = P_{m_1}(f - R(v_m))$ for some adequately chosen $m_1 > m$.

Defect-correction techniques, or the related two-grid, two-level or multilevel methods are a well established technique for nonlinear *steady* problems (see, e.g., [1,13,37,47,48], and the references cited therein). That is, there is nothing new in the above steps (1), (2). For *evolution* problems, similar defect-correction techniques are used on the steady problems that arise when doing implicit time-stepping. For example, if the implicit Euler method is used to integrate in time (8), looking for approximations $u_m^{(n)} \approx u_m(t_n)$ with $t_n = n\delta t$, $n = 0, \dots, n_T = T/\delta t$, at every time step the following nonlinear problem should be solved: Find $u_m^{(n)}$ such that

$$u_m^{(n)} + \delta t P_m (\nu A u_m^{(n)} + F(u_m^{(n)})) = u_m^{(n-1)} + \delta t P_m f(t_n). \quad (18)$$

Thus, step (2) can be used to find an approximation $\tilde{u}^{(n)}$ to $u(t_n) \in V$ better than $u^{(n)} \in W_m$.

More recently, there has been renewed interest in multilevel methods for evolution problems (see, e.g., [7,39,40]). But so far, the two levels (or several levels) of discretization are maintained in the whole interval $(0, T]$ (i.e., defect-correction techniques are used at every time step like (18)). What the postprocessed Galerkin method suggests is that it is safe (up to a point) to work at the lower level of discretization (or coarse grid) in the interval $(0, T)$, and use the higher level of discretization only at the time $t = T$ when the time integration on the coarser grid is completed.

The relation between the postprocessed Galerkin method and defect-correction techniques can be better seen by considering the steady problem of finding $w \in V$ such that

$$\nu Aw + F(w) = f(T) - \frac{du(T)}{dt}, \quad (19)$$

which has an obvious solution $w = u(T)$. Notice, however, that the right-hand side of (19) is not known due to the term $du(T)/dt$ since it involves the unknown u that we want to approximate. The postprocessed Galerkin method replaces this problem by finding $v \in V$ such that

$$vAv + F(v) = f(T) - \frac{du_m(T)}{dt}. \tag{20}$$

That is, in the right-hand side, the unknown term $du(T)/dt$ is replaced by the already computed $du_m(T)/dt$. Now the postprocessed Galerkin method can be seen as nothing else than the step (2) of the defect-correction method applied to (20), since the Galerkin approximation v_m to the solution v of (20) satisfies $vP_mAv_m + P_mF(v_m) = P_mf(T) - du_m(T)/dt$, and, thus, $v_m = u_m(T)$ (recall Eq. (8) defining u_m).

2.2. The case of spectral methods

When the Galerkin method is based on the eigenfunctions w_j of the operator A in (1) the post-processing step (10) can be further simplified. We explain in this section this simplification as well as the convergence properties of the method.

If the spaces W_m where the Galerkin approximation u_m is sought are eigenspaces of A , that is,

$$W_m = H_m = \text{span}\{w_1, \dots, w_m\},$$

then, the projection P_m onto H_m commutes with A , that is

$$P_mA = AP_m. \tag{21}$$

Thus, since, as already shown $P_m\tilde{v} = u_m(T)$, it is only left to compute $\tilde{q} = (I - P_m)\tilde{v}$. Applying $I - P_m$ to (10), and taking into account (21), it follows that

$$vA\tilde{q} = (I - P_m)(f(T) - F(u_m(T))). \tag{22}$$

By decomposing

$$u = p + q, \quad p = P_mu, \quad q = (I - P_m)u, \tag{23}$$

the error $u(T) - \tilde{v}$ can be orthogonally decomposed as

$$u(T) - \tilde{v} = (p(T) - u_m(T)) + (q(T) - \tilde{q}). \tag{24}$$

For the first term on the right-hand side above, the following result is proved in [25].

Theorem 2. *Let u be the solution of the RD equation or the KS equation with $u(0) = u_0$. Then if $\max_{0 \leq t \leq T} \|A^s u(t)\| < \infty$, there exist a positive constants $C = C(\max_{0 \leq t \leq T} \|A^s u(t)\|)$ and m_0 such that for $m \geq m_0$ the Galerkin approximation or the pseudospectral approximation u_m solution of (9) and (8) or (15), respectively, satisfies that*

$$\max_{0 \leq t \leq T} \|p(t) - u_m(t)\| \leq \frac{C}{\lambda_{m+1}^{s+\beta}}, \tag{25}$$

where β is as in Theorem 1.

Comparing the bound (25) with (11), the result above states that u_m is a much better approximation to $p = P_mu$ than to u itself. Thus, if the error $u - u_m$ is orthogonally decomposed as $u - u_m = (p - u_m) + q$

(recall (23) above), then, at least as $m \rightarrow \infty$ the dominant component is q . Hence, in view of (24), the computation of \tilde{q} in the postprocessing step increases the accuracy by approximating the q part of u .

To study the error component $q(T) - \tilde{q}$, it is better to introduce $\Phi \in (I - P_m)H$ defined as the solution of

$$A\Phi = (I - P_m)(f(T) - F(p(T))). \quad (26)$$

Thus $q(T) - \tilde{q}$ can be further decomposed as

$$q(T) - \tilde{q} = (q(T) - \Phi) + (\Phi - \tilde{q}). \quad (27)$$

For the second term on the right-hand side above, it can be shown (see, e.g., [10]) that

$$\|\Phi - \tilde{q}\| \leq L \|p(T) - u_m(T)\| \quad (28)$$

with $L = L(M_1, \nu)$. Thus, with Theorem 2 in sight, we only have to bound the first term on the right-hand side of (27). For this term, it is shown in [16] that $\|q(T) - \Phi\| \leq cM_1\lambda_{m+1}^{-3/2} \log(\lambda_{m+1}/\lambda_1)$ in the particular case of the Navier–Stokes equations with time independent forcing term (see [32] for the case of time dependent forcing). Similar results for other equations can be found in [8,9,15,17,29,30,46]. Using the techniques in these references it is possible to show that

$$\|q(T) - \Phi\| \leq \frac{C}{\lambda_{m+1}^{s+\beta}} \max_{0 \leq t \leq T} \|A^s u(t)\| \quad (29)$$

(at least when in (3) $T_0 = 0$), where $C = C(\nu, M_1)$ (see also remarks in [25]). The value β depends on the particular equation. For example, it is $\beta = 1/2$ for the viscous Burgers' equation and the two-dimensional Navier–Stokes equation, $\beta = 2$ for the RD equation considered here, and for general reaction-diffusion (dissipative) equations with polynomial nonlinearities, and $\beta = 3/4$ for the KS equation.

3. Smooth solutions

We have seen that for spectral methods, the postprocessing step (10), or, rather, (22) computes a new approximation with improved convergence rate. The error bound is decreased by a factor $C\lambda_{m+1}^{-\beta}$ with the value β depending on Eq. (1). Obviously, this enhancement of the convergence rate with respect to the Galerkin method (or PS method) will be more significant the smaller the value of s in (11). In the example of Section 1, the error in the Galerkin and PS methods decay like (almost) $\lambda_{m+1}^{-7/4} = N^{-7/2}$, whereas in the postprocessed methods, the error decays like (almost) $\lambda_{m+1}^{-11/4} = N^{-11/2}$, that is N^{-2} smaller. As a consequence, for a given level of accuracy, the postprocessed methods can achieve that accuracy while saving between 30% and 50% of the cost that the standard methods, on which they are based, need to attain that accuracy.

If the solutions are smoother, however, the improvement is less significant. In [25], the experiments with the two-dimensional Navier–Stokes equations, with Galerkin and PS errors decaying like $N^{-13/2}$ the postprocessed methods, for a given error, save only between 20% to 40% of the cost of the standard methods (although the savings can be larger if errors are measured in the H^1 norm [26]).

However, it is usually the case that errors in standard methods decay much faster than algebraically. This is the case when the solutions u of (1), (2) belong to a certain Gevrey class of regularity (real analytic), that is, $u(t)$ belongs not only to $D(A^s)$ for all $s > 0$ but to $D(e^{\gamma A^{\sigma/2}})$ for some $\gamma > 0$, and

where $\sigma = 1$ when $A = -\Delta$ and $\sigma = \frac{1}{2}$ when $A = \Delta^2$ (see, e.g., [14,18]). In this case, the errors in Galerkin and PS can be bounded as

$$\max_{0 \leq t \leq T} \|u(t) - u_m(t)\| \leq C e^{-\gamma \lambda_{m+1}^{\sigma/2}} \max_{0 \leq t \leq T} \|e^{\gamma A^{\sigma/2}} u(t)\|, \tag{30}$$

see [12,31,32]. One may then ask what will be the improvement of the postprocessing step (10) in this situation.

We first consider one of the components of the error, $\|q(T) - \Phi\|$ (recall (27)). As pointed out (and checked experimentally) in [28,31] (see also [25]) the bound (29) should be replaced by

$$\|q(T) - \Phi\| \leq C \frac{e^{-\gamma \lambda_{m+1}^{-\sigma/2}}}{\lambda_{m+1}^\beta} \max_{0 \leq t \leq T} \|e^{\gamma A^{\sigma/2}} u(t)\|. \tag{31}$$

Thus, when postprocessing, one can only expect an algebraic improvement in the error bound with respect to the exponential rate of decay of the error in Galerkin or PS methods (30).

Thus, when the solution u of (1), (2) belong to a Gevrey class of regularity, the practical usefulness of the postprocessing spectral methods becomes questionable. This is more so since spectral methods are most efficient when u is very smooth (i.e., Gevrey class). When this is not the case, other discretizations such as finite-difference methods or finite-element methods are usually more efficient from the computational point of view.

However, the outcome may not be as clear-cut as the previous paragraphs suggest. On the one hand if the geometry of the domain, software availability, etc. impose the use of finite-element methods, postprocessing may bring a substantial computational saving regardless of the smoothness of the solution. Numerical experiment with finite-element methods in [27] show an example where, while solutions belong to a Gevrey class of regularity, the postprocessed Galerkin method saves between 15% to 25% of the cost of the Galerkin method to achieve a predetermined accuracy. This is due to the fact that in finite-element methods, unlike spectral methods, errors decay algebraically with the mesh size, rather than exponentially with m , so that the algebraic improvement in the error bound brought by postprocessing is (almost) as noticeable as that shown in the example of Section 1.

On the other hand, the factor $\lambda_{m+1}^{-\beta}$ may be not that small with respect to $e^{-\gamma \lambda_{m+1}^{\sigma/2}}$ if $\gamma \ll 1$. In the rest of the section we consider two examples in which the effect of the size of γ and other issues pertinent to spectral methods are examined in relation to postprocessing.

We first test the postprocessing step with the KS equation with the parameter θ set to $\theta = 30.349489198$. For this value of θ , the solution u corresponding to certain initial conditions converges to a periodic (in time) solution. The initial condition (provided to us by R. Wallace) was obtained by the bifurcation package AUTO [11]. Fig. 3 shows a convergence diagram for $T = 12$. The symbols to represent the different methods are the same as in Section 1. The truncation value m_1 in (13) was set to $2(m + 1) - 1$, since in view of the nonlinear terms of the KS equation (5), $u_m(T) \in H_m$ implies that $F(u_m) \in H_{2m}$. We first notice the exponential convergence of the methods, that is, that errors decay exponentially with N rather than algebraically as in Fig. 1, which is the same conclusion as in [31]. We also observe that the PS method (crosses joined by continuous lines) commits errors significantly larger than the Galerkin method (asterisks joined by continuous lines). Taking into account that the error can be decomposed as

$$\|u(T) - u_m(T)\|^2 = \|P_m u(T) - u_m(T)\|^2 + \|(I - P_m)u(T)\|^2, \tag{32}$$

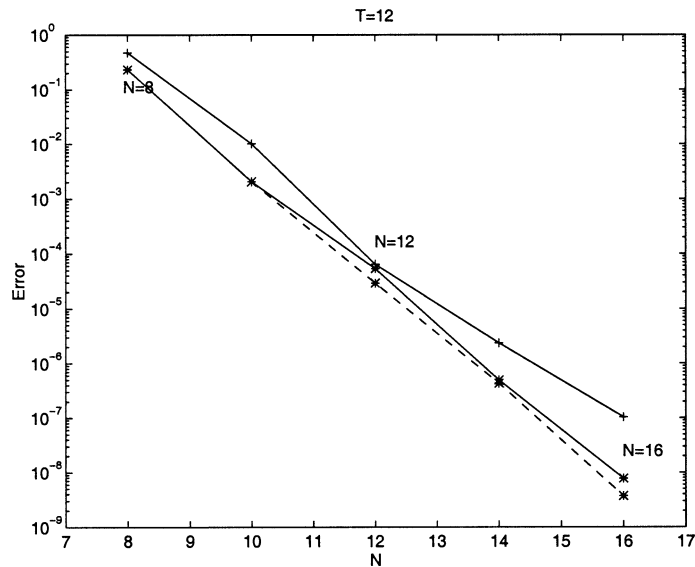


Fig. 3. Convergence diagram for the KS equation. Errors at $T = 12$.

and since the last term on the right-hand side above is shared by both the Galerkin and the PS methods, then the larger errors of the PS method are due to $\|P_m u(T) - u_m(T)\|$ being larger in the PS method than in the Galerkin method. Since this term is unaltered by the postprocessing step, it is to be expected then that postprocessing does not reduce the errors of the PS method. Indeed this is what happens: in Fig. 3, the results of the postprocessed PS method are exactly those of the PS method (i.e., postprocessing the PS method does not increase the accuracy in this example). Error bounds for the postprocessed PS method (based on Chebyshev polynomials) are proved in [22] for some dissipative equations, with special attention to the case of smooth solutions. The proofs there can be easily adapted to the present case.

In Fig. 3 we also observe that postprocessing increases the accuracy of the Galerkin method. But compared with Fig. 1, the postprocessed Galerkin method hardly improves on the Galerkin method. For a given value of N , the errors of the postprocessed Galerkin method are at most two times smaller than those of the Galerkin method (recall that in Fig. 1 they were typically ten times smaller than those of the Galerkin method).

We see then that when solutions are in a Gevrey class, the benefits of postprocessing may be hardly noticeable. In this example with the KS equation, Fourier coefficients of the solution (and, hence, errors in the Galerkin method) decay very fast with the wave number. An estimate of the decay rate γ is shown to be $\gamma \approx 1.3$ in Fig. 4. In Fig. 4 we also show the Fourier coefficients of the solution of the RD equation with $\nu = 10^{-4}$ and $u_0(x) = \sin(x)$ at time $T = 20$. In this example, we used as forcing term

$$f(x, t) = \sinh(\sin(t) \sin(x)). \quad (33)$$

Notice that, again, we chose a time-periodic forcing term so that the corresponding solutions do not fall into a steady state.

We see in Fig. 4 that for the RD equation forced with (33), the decay rate is $\gamma \approx 9.5 \times 10^{-3}$, more than a 100 times smaller than in the example of the KS equation. As mentioned before, this may give a chance to the postprocessing techniques to improve accuracy more noticeably. Thus in the rest of this

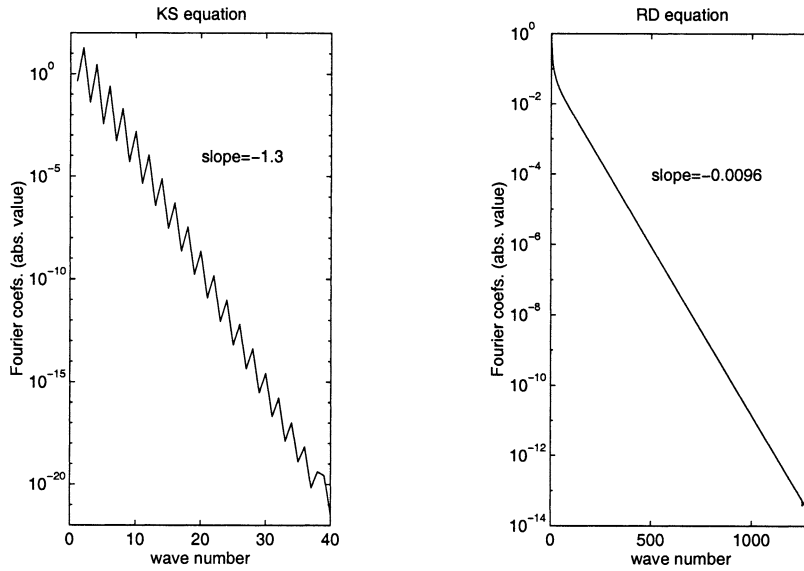


Fig. 4. Decay rates for the KS ($T = 12$) and RD ($T = 20$) equations. Slopes correspond to least-squares fit of points in the plots.

section, we comment on this example of the RD equation. We remark that, due to our experience in [24], the cutoff value m_1 in (13) was set to $m_1 = 3(m + 1)/2$.

We turn our attention first to the PS method. The errors in the postprocessed PS method (not shown here), as in the KS example, are hardly any better than those of the PS method (indeed they are practically the same). This, as in the example of the KS equation, should not be surprising in view of the convergence diagram in Fig. 5. Comparing the continuous lines of the standard methods, it can be seen that the PS method commits double error than the Galerkin method (recall that $N = m + 1$).

It is interesting to study in more detail the sources of error in the PS method. In Fig. 6 (top diagram) we have represented the size of the Fourier coefficients of the error $u(T) - u_m(T)$ for the Galerkin (continuous line) and the PS method (discontinuous line) for $N = 512$. It can be seen that for the PS method the error coefficients are much larger than those of the Galerkin method up frequency $j = 511$ (marked with an asterisk for the Galerkin method and with a cross for the PS methods). From $j = 512$ onwards the error coefficients of both method coincide since they are those of $(I - P_{511})u(T)$. Furthermore, the norms of $(P_{511}u(T) - u_m(T))$, for both methods, and of $(I - P_{511})u(T)$ are shown in the plot. We see that whereas for the Galerkin method $\|P_{511}u(T) - u_m(T)\|$ is more than 20 times smaller than $\|(I - P_{511})u(T)\|$, for the PS method they are comparable. So even in the (unlikely) case that, after postprocessing $(I - P_{511})(u(T) - \tilde{u}) = 0$, the error $u(T) - \tilde{u}$ would still be 2.8×10^{-6} since $P_{511}(u(T) - \tilde{u}) = P_{511}u(T) - u_m(T)$. Observe also that for the Galerkin method since $\|P_m u(T) - u_m(T)\| \ll \|(I - P_m)u(T)\|$, the error $\|u(T) - u_m(T)\|$ is roughly $\|(I - P_m)u(T)\|$, whereas this is not the case for the PS method.

In Fig. 6, it can be observed also that, specially for the PS method, the error grows with the wave number up to $j = 511$, and specially from $j = 350$ to $j = 511$ the growth rate is larger. This is not a particular feature of this example, but we have observed this phenomenon in the experimental work with Gevrey class of regularity solutions of [24]. Thus, in order to postprocess the PS method, the last modes

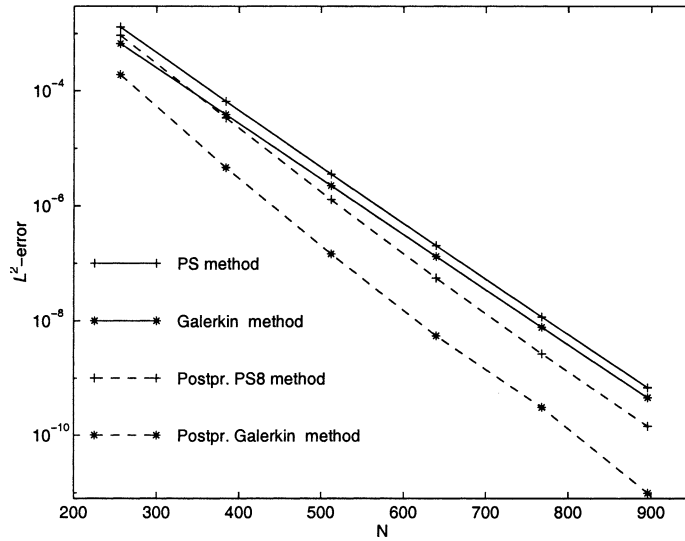


Fig. 5. Convergence diagram for the RD equation. Errors at $T = 20$.

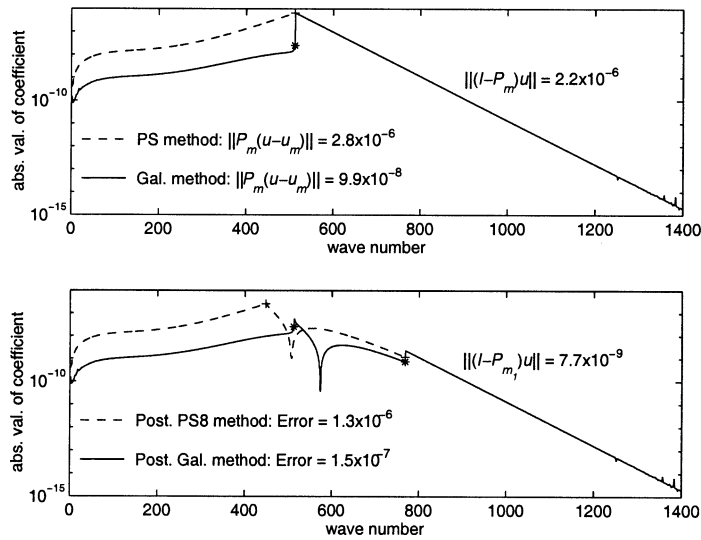


Fig. 6. Fourier coefficients of the error. Top: standard methods; bottom: postprocessed methods.

must be discharged (filtered out). This idea of removing the last modes of the solution and postprocessing the rest was first suggested in [25], and it has been used to postprocess with more refined techniques [41].

Accordingly, we replace the postprocessing step in the PS method by

$$\nu A \tilde{q} = (P_{m_1} - P_{m_0})(f(T) - F(P_{m_0}u_m(T))), \tag{34}$$

for some $m_0 < m$. In the experiments in this section, we took $m_0 = 7(m + 1)/8 - 1$, and the approximation $P_{m_0}u_m(T) + \tilde{q}$ will be denoted by *postprocessed PS8 method*. In Fig. 6 (bottom plot) the error coefficients of the postprocessed methods are shown. Comparing with the standard methods (top plot) it can be

seen that the extra coefficients \tilde{q} and $\tilde{\tilde{q}}$ do approximate the coefficients of the solution ranging from $j = m + 1 = 512$ ($j = m_0 + 1 = 448$ for the postprocessed PS8 method) to $j = m_1 = 767$. Observe now that for the postprocessed methods, the bulk of the error is $P_m u(T) - u_m(T)$ for the postprocessed Galerkin method and $P_{m_0}(u(T) - u_m(T))$ for the postprocessed PS8 method, and the smallest term in the error is $\|(I - P_{m_1})u(T)\|$.

As regards the convergence of the postprocessed methods, Fig. 5 clearly shows that postprocessed methods enjoy significantly smaller errors than the two standard methods. For errors below 10^{-5} , for a given value of N , the errors committed by the postprocessed PS8 method are on average four times smaller than those committed by the PS method, and this ratio becomes smaller as larger values of N are used. Observe also that the errors of the postprocessed Galerkin method are, on average, more than ten times smaller than those of the Galerkin method.

We finally investigate computational efficiency. Fig. 7 shows errors vs cost. Before examining Fig. 7 we comment on how the pairs (CPU time, error) were selected for Fig. 7. We performed experiments with a few values of N . Since in our programs nonlinear terms are evaluated by collocation and FFT (see, e.g., [3]), and since FFT algorithms are more cost effective when the prime factors of N are small, we chose $N = 256, 348, 512, 640$ and 768 . We tried these values of N with different values of the time integrator tolerance. Fig. 8 shows all the results we obtained for the Galerkin and the postprocessed Galerkin methods, with results corresponding to a given value of N joined by segments. For the efficiency diagram in Fig. 7 we chose the leftmost values in the plots of Fig. 8 (they appear in Fig. 8 joined by a dotted line). In this way, it is reflected more accurately what happens in actual computations. Notice that in practice, one uses values of N with small prime factors.

Let us examine then Fig. 7. We see that the postprocessed PS8 method, for most of the error levels represented in the plot is slightly more advantageous than the standard PS method. We see, for example, that the postprocessed PS8 method, for errors around 10^{-7} , cost less than 1/3 of the cost of the PS method. However, this is the most favorable case: most of the pairs (CPU time, error) corresponding to

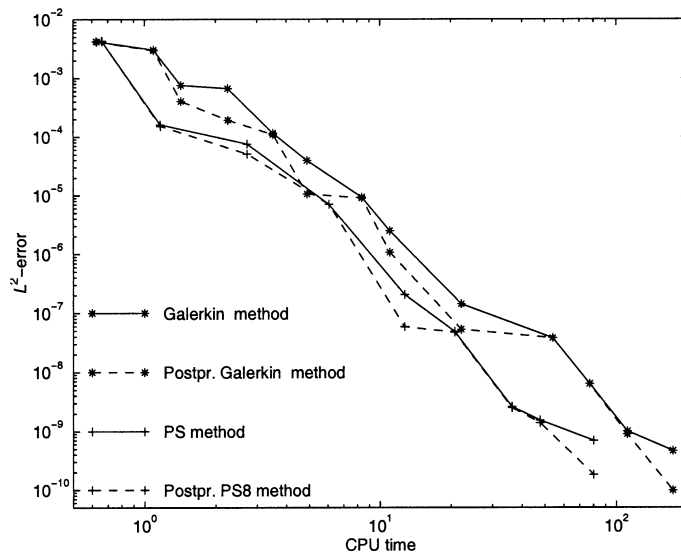


Fig. 7. Efficiency diagram for the RD equation. Errors at $T = 20$.

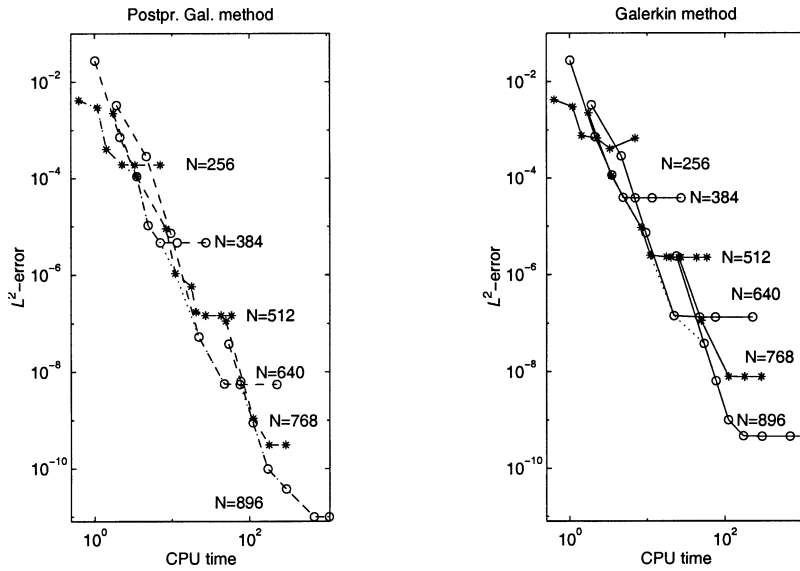


Fig. 8. Efficiency diagram for the RD equation. Errors at $T = 20$.

the postprocessed PS8 method are coincident or slightly left of the continuous line of the PS8 method, except for the pairs corresponding to errors 5×10^{-5} , 6×10^{-8} , and 1.8×10^{-10} . A similar situation can be observed when comparing the postprocessed Galerkin and Galerkin methods.

Hence, strictly speaking, the postprocessed methods in this example are not manifestly more efficient than the standard methods on which they are based. This would be a right conclusion if the postprocessed methods were independent of the standard methods, and one had to choose between postprocessed or standard method before doing the computation (as one has to when choosing between Galerkin or PS methods). With postprocessed methods, one has to use a standard method to obtain the approximation $u_m(T)$. Then, for a more accurate approximation, there are two options. If, when computing $u_m(T)$ we have been cautious with the time-integration tolerance and time-discretization errors are not dominant (a situation common in practice), we can compute \tilde{u} with the postprocessing step (10), which cost almost nothing since $u_m(T)$ is already computed, and the most likely outcome (as shown in Fig. 5) is an improved (more accurate) approximation. The other option is to start from the beginning and compute an approximation $u_{m'}(T)$ with $m' > m$, whose cost amounts to that of computing $u_m(T)$ and \tilde{u} again. Looked at from this point of view, it is clear that postprocessing is a computationally efficient procedure.

4. Conclusions and further remarks

We have seen that when dealing with very smooth solutions (Gevrey class of regularity), the postprocessed Galerkin and PS methods do not seem to be as computationally efficient as when dealing with “not very smooth” solutions. However, and specially when the value $\gamma = \max\{s \in \mathbb{R} \mid u(t) \in D(e^{sA^{\sigma/2}}), \forall t \in [0, T]\}$ is small, postprocessing brings a significant improvement of accuracy. Furthermore, since this is done at no cost, the real outcome is a more efficient method.

For the pseudospectral method, we see that although the influence of the aliasing error is negligible with respect to the error $u - u_m$, it is not so for the purposes of postprocessing. However, removing (filtering out) the last modes of the approximate solution and postprocessing the rest, allows the error of the postprocessed method to be significantly smaller than the error of the pseudospectral method.

Further investigation is needed to determine the optimal way to discard (filter out) the last modes of the pseudospectral method before postprocessing. Also, smaller decay rates need to be tested to better assess the practical value of the postprocessed method when dealing with Gevrey class of regularity solutions.

Finally, we mention that in practical situations, one hardly ever knows the level of accuracy achieved by a single computer run. The extended practice is to repeat the computations with a different value of m and compare the approximations to get an idea of the errors committed. We observe that postprocessing gives at a much smaller cost an estimation of the error. This is specially so when T is very large and computer runs are costly or lengthy.

Acknowledgements

This research was supported in part by DGICYT projects PB95-0216 and PB98-0074, and by the National Science Foundation grants number DMS-9706964 and DMS-9704632. E.S.T. acknowledges the kind hospitality of the Institute for Geophysical and Planetary Physics (IGPP) at the Los Alamos National Laboratory where this work was completed while he was the Orson Anderson Visiting Scholar.

References

- [1] R.E. Bank, D.J. Rose, Analysis of multilevel iterative methods for nonlinear finite element equations, *Math. Comp.* 39 (1982) 453–465.
- [2] P.N. Brown, G.D. Byrne, A.C. Hindmarsh, VODE: A variable coefficient ODE Solver, *SIAM J. Sci. Statist. Comput.* 10 (1989) 1038–1051.
- [3] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods in Fluid Dynamics*, in: Springer Ser. Comput. Phys., Springer, Berlin, 1988.
- [4] C. Cao, M. Rammaha, E.S. Titi, The Navier–Stokes equations on the rotating 2-D sphere: Gevrey regularity and asymptotic degrees of freedom, *Z. Angew. Math. Phys.* 50 (1999) 341–360.
- [5] P. Collet, J.-P. Eckmann, H. Epstein, J. Stubbe, Analyticity for the Kuramoto–Sivashinsky equation, *Phys. D* 67 (1993) 321–326.
- [6] P. Constantin, C. Foias, *Navier–Stokes Equations*, in: Chicago Lectures in Mathematics, University of Chicago Press, Chicago, 1988.
- [7] C.N. Dawson, M.F. Wheeler, Two-grid methods for mixed finite element approximations of nonlinear parabolic problems, *Contemp. Math.* 180 (1994) 192–204.
- [8] A. Debussche, M. Marion, On the construction of families of approximate inertial manifolds, *J. Differential Equations* 100 (1992) 173–201.
- [9] C. Devulder, M. Marion, A class of numerical algorithms for large time integration: The nonlinear Galerkin method, *SIAM J. Numer. Anal.* 29 (1992) 462–483.
- [10] C. Devulder, M. Marion, E.S. Titi, On the rate of convergence of nonlinear Galerkin methods, *Math. Comp.* 60 (1993) 495–514.
- [11] E.J. Doedel, AUTO: A program for the bifurcation analysis of autonomous systems, *Congr. Numer.* 30 (1981) 279–309.
- [12] A. Doelman, E.S. Titi, Regularity of solutions and the convergence of the Galerkin method in the Ginzburg–Landau equation, *Numer. Funct. Anal. Optim.* 14 (1993) 299–321.
- [13] J. Douglas Jr., T. Dupont, A Galerkin method for a nonlinear Dirichlet problem, *Math. Comp.* 29 (1975) 696–698.

- [14] A. Ferrari, E.S. Titi, Gevrey regularity for nonlinear analytic parabolic equations, *Comm. Partial Differential Equations* 23 (1998) 1–16.
- [15] C. Foias, M.S. Jolly, I.G. Kevrekidis, G.R. Sell, E.S. Titi, On the computation of inertial manifolds, *Phys. Lett. A* 131 (1988) 433–436.
- [16] C. Foias, O. Manley, R. Temam, Modelling of the interaction of small and large eddies in two dimensional turbulent flows, *RAIRO Modél. Math. Anal. Numér.* 22 (1988) 93–114.
- [17] C. Foias, G.R. Sell, E.S. Titi, Exponential tracking and approximation of inertial manifolds for dissipative nonlinear equations, *J. Dynamics Differential Equations* 1 (1989) 199–243.
- [18] C. Foias, R. Temam, Gevrey class regularity for the solutions of the Navier–Stokes equations, *J. Funct. Anal.* 87 (1989) 359–369.
- [19] J. de Frutos, B. García-Archilla, J. Novo, A postprocessed Galerkin method with Chebyshev and Legendre polynomials, *Numer. Math.* 86 (2000) 419–442.
- [20] J. de Frutos, J. Novo, A spectral element method for the Navier–Stokes equations with improved accuracy, *SIAM J. Numer. Anal.* 38 (2000) 799–819.
- [21] J. de Frutos, J. Novo, A postprocess-based improvement of the spectral element method, *Appl. Numer. Math.* 33 (2000) 217–223.
- [22] J. de Frutos, J. Novo, An enhanced pseudospectral Chebyshev method for dissipative partial differential equations, *J. Comput. Appl. Math.* 115 (2000) 137–150.
- [23] B. García-Archilla, Some practical experience with the time integration of dissipative equations, *J. Comput. Phys.* 122 (1995) 25–29.
- [24] B. García-Archilla, J. de Frutos, Time integration of the nonlinear Galerkin method, *IMA J. Numer. Anal.* 14 (1994) 1–24.
- [25] B. García-Archilla, J. Novo, E.S. Titi, Postprocessing the Galerkin method: A novel approach to approximate inertial manifolds, *SIAM J. Numer. Anal.* 35 (1998) 941–972.
- [26] B. García-Archilla, J. Novo, E.S. Titi, An approximate inertial manifold approach to postprocessing Galerkin methods for the Navier–Stokes equations, *Math. Comp.* 68 (1999) 893–911.
- [27] B. García-Archilla, E.S. Titi, Postprocessing Galerkin methods: The finite-element case, *SIAM J. Numer. Anal.* 37 (2000) 470–499.
- [28] M.D. Graham, P.H. Steen, E.S. Titi, Computational efficiency and approximate inertial manifolds for a Bénard convection system, *J. Nonlinear Sci.* 3 (1993) 153–167.
- [29] M.S. Jolly, I.G. Kevrekidis, E.S. Titi, Approximate inertial manifolds for the Kuramoto–Sivashinsky equation: Analysis and computations, *Phys. D* 44 (1990) 38–60.
- [30] M.S. Jolly, I.G. Kevrekidis, E.S. Titi, Preserving dissipation in approximate inertial forms for the Kuramoto–Sivashinsky equation, *J. Dynamics Differential Equation* 3 (1991) 179–197.
- [31] D. Jones, L. Magolin, E.S. Titi, On the effectiveness of the approximate inertial manifolds—Computational study, *Theor. Comput. Fluid Dynamics* 7 (1995) 243–260.
- [32] D. Jones, E.S. Titi, A remark on quasi-stationary approximate inertial manifolds for the Navier–Stokes equations, *SIAM J. Numer. Anal.* 25 (1994) 894–914.
- [33] H.O. Kreiss, Fourier expansions of the Navier–Stokes equations and their exponential decay rate, in: *Analyse Mathématique et Applications*, Gauthier-Villars, Paris, 1988, pp. 245–262.
- [34] C.D. Levermore, M. Oliver, The complex Ginzburg–Landau equation as a model problem, in: *Lectures in Appl. Math.*, Vol. 31, American Mathematical Society, Providence, RI, 1996, pp. 141–190.
- [35] X. Liu, Gevrey class regularity and approximate inertial manifolds for the Kuramoto–Sivashinsky equations, *Phys. D* 50 (1991) 135–151.
- [36] H.V. Ly, E.S. Titi, Global Gevrey regularity for 3-D Bénard convection in porous medium with zero Darcy–Prandtl number, *J. Nonlinear Sci.* 9 (1999) 333–362.
- [37] L. Mansfield, On the solution of nonlinear finite element systems, *SIAM J. Numer. Anal.* 17 (1988) 752–765.
- [38] L.G. Margolin, E.S. Titi, S. Wynne, The post-processing Galerkin and nonlinear Galerkin methods—A truncation analysis point of view, submitted.
- [39] M. Marion, R. Temam, Nonlinear Galerkin methods, *SIAM J. Numer. Anal.* 26 (1989) 1139–1157.
- [40] M. Marion, J. Xu, Error estimates on a new nonlinear Galerkin method based on two-grid finite elements, *SIAM J. Numer. Anal.* 32 (1995) 1170–1184.

- [41] J. Novo, E.S. Titi, S. Wynne, Efficient methods using high accuracy approximate inertial manifolds, *Numer. Math.* 87 (2001) 485–502.
- [42] M. Oliver, E.S. Titi, Analyticity of the global attractor and the number of determining nodes for a weakly damped driven nonlinear Schrödinger equation, *Indiana Univ. Math. J.* 47 (1998) 49–73.
- [43] M. Oliver, E.S. Titi, Gevrey regularity for the attractor of a partially dissipative model of Bénard convection in a porous medium, *J. Differential Equation* 163 (2000) 292–311.
- [44] K. Promislow, Time analyticity and Gevrey regularity for solutions of a class of dissipative partial differential equations, *Nonlinear Anal. TMA* 16 (1991) 959–980.
- [45] R. Temam, *Infinite Dimensional Dynamical Systems in Mechanics and Physics*, in: *Appl. Math. Sci.*, Vol. 68, Springer, Berlin, 1988.
- [46] E.S. Titi, On approximate inertial manifolds to the Navier–Stokes equations, *J. Math. Anal. Appl.* 149 (1990) 540–557.
- [47] J. Xu, A novel two-grid method for semi-linear elliptic equations, *SIAM J. Sci. Comput.* 15 (1994) 231–237.
- [48] J. Xu, Two-grid discretization techniques for linear and nonlinear problems, *SIAM J. Numer. Anal.* 33 (1996) 1759–1777.