SIAM J. NUMER. ANAL.
Vol. 41, No. 2, pp. 695–714

THE POSTPROCESSING GALERKIN AND NONLINEAR GALERKIN METHODS—A TRUNCATION ANALYSIS POINT OF VIEW*

LEN G. MARGOLIN†, EDRISS S. TITI‡, AND SHANNON WYNNE§

Abstract. We revisit the postprocessing algorithm and give a justification from a classical truncation analysis point of view. We assume a perturbation expansion for the high frequency mode component of solutions to the underlying equation. Keeping terms to certain orders, we then generate approximate systems which correspond to numerical schemes. We show that the first two leading order methods are in fact the postprocessed Galerkin and postprocessed nonlinear Galerkin methods, respectively. Hence postprocessed Galerkin is a natural leading order method, more natural than the standard Galerkin method, for approximating solutions of parabolic dissipative PDEs. The analysis is presented in the framework of the two-dimensional Navier–Stokes equation (NSE); however, similar analysis may be done for any parabolic, dissipative nonlinear PDE.

The truncation analysis is based on asymptotic estimates (in time) for the low and high mode components. We also introduce and investigate an alternative postprocessing scheme, which we call the dynamic postprocessing method, for the case in which the asymptotic estimates (in time) do not hold (i.e., in the situation of long transients, nonsmooth initial data, or highly oscillatory time-dependent solutions).

Key words. dissipative equations, spectral methods, approximate inertial manifolds, nonlinear Galerkin methods, postprocessing algorithm, multigrid

AMS subject classification. 65P25

PII. S0036142901390500

1. Introduction.

We revisit the postprocessing algorithm for the Galerkin and nonlinear Galerkin methods. Postprocessing methods first evolved from the theory of approximate inertial manifolds (AIMs) (see, e.g., [3], [5], [6], [7], [15], [19], and [21]) and take advantage of the observation that, for dissipative evolution equations, the Galerkin and nonlinear Galerkin methods do better approximating the low modes of the exact solution u than approximating the solution itself. AIMs are used to "postprocess" the low modes in order to obtain a more accurate approximation for the high modes. For a variety of applications, the postprocessed Galerkin has been shown to be a very efficient algorithm for improving the accuracy of Galerkin/nonlinear Galerkin methods with very little extra computational cost (see, for example, [8], [10], [11], [12], and [17]). However, postprocessing is not simply a technique for improving efficiency. In this paper we show that postprocessing methods arise in a very natural way through a classical truncation analysis of the dissipative evolution equation. More specifically, we will show that, to leading order, the correct approximative scheme is

*Received by the editors June 17, 2001; accepted for publication (in revised form) August 20, 2002; published electronically May 6, 2003. This work was supported in part by National Science Foundation grants DMS-9706964 and DMS-9704632, by the Department of Energy under contract W-7405-ENG-36, and by the U.S. Air Force Office of Scientific Research under grants AFOSR-F49620-98-1-0180 and AFOSR-49620-01-1-0026.

http://www.siam.org/journals/sinum/41-2/39050.html

†Institute for Geophysics and Planetary Physics and Center of Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM 87545 (len@lanl.gov).

‡Department of Mathematics and Department of Mechanical and Aerospace Engineering, University of California, Irvine, CA 92697-3875 and Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel (etiti@math.uci.edu).

§Center for Research in Scientific Computing, North Carolina State University, Raleigh, NC 27695-8205 (snwynne@unity.ncsu.edu).

actually the postprocessed Galerkin method, and not the standard Galerkin method as is commonly believed.

We present this work in the context of the two-dimensional Navier–Stokes equations (NSE) in Ω, an open bounded set of $\mathbb{R}^2$, with smooth boundary $\partial\Omega$,

$$\begin{aligned} \frac{\partial u}{\partial t} - \nu\Delta u + (u\cdot\nabla)u + \nabla\pi &= f, \\ \nabla\cdot u &= 0, \\ u(0,x) &= u_0(x), \end{aligned} \tag{1.1}$$

where the unknowns are the vector velocity u and the scalar pressure π; $f(x,t)$ is a given body forcing, and $\nu > 0$ is the kinematic constant viscosity. The equations are subject to either nonslip Dirichlet boundary conditions for $\partial\Omega$ smooth enough, or periodic boundary conditions when Ω is a square. To this end, we define the Hilbert space H as

$$H = \{u \in L^2(\Omega)^2, \nabla\cdot u = 0, u\cdot\vec{n} = 0 \text{ on } \partial\Omega\}$$

in the case of nonslip Dirichlet boundary conditions, where $\vec{n}$ denotes the outward normal unit vector to $\partial\Omega$, or

$$H = \left\{u \in L^2_{\text{per}}(\Omega)^2, \nabla\cdot u = 0, \int_\Omega u\, dx = 0\right\}$$

in the case of periodic boundary conditions. The space H is a closed subspace of $L^2(\Omega)^2$ and is endowed with the scalar product and norm from $L^2(\Omega)^2$, denoted by $(\cdot,\cdot)$ and $\|\cdot\|$, respectively. We also define the Hilbert space V as $V = \{u \in H^1_0(\Omega)^2, \nabla\cdot u = 0\}$ or $V = \{u \in H^1_{\text{per}}(\Omega)^2, \nabla\cdot u = 0, \int_\Omega u\, dx = 0\}$, depending on the boundary conditions. Let P be the Leray orthogonal projection from $L^2(\Omega)^2$ onto H. Then (1.1) projected onto H may be written as an abstract functional differential equation of the form

$$\begin{aligned} \frac{du}{dt} + \nu Au + B(u,u) &= f, \\ u(0) &= u_0; \end{aligned} \tag{1.2}$$

see, e.g., [2] or [20].

The Stokes operator A is defined as $-P\Delta$ with the appropriate boundary conditions. The domain of A in H, denoted $D(A)$, is either $H^2(\Omega)^2 \cap V$ or $H^2_{\text{per}}(\Omega)^2 \cap V$, depending on the boundary conditions. The nonlinear term is $B(u,u)$ and is defined in general as $B(u,v) = P[(u\cdot\nabla)v]$. Finally, f (or $f = Pf$) is the forcing term and is assumed to be at least in H. The operator A is a positive, self-adjoint, densely defined, unbounded operator with compact inverse. The eigenfunctions of A, $\{\omega_1, \omega_2, \ldots\}$, form a complete orthonormal basis for the space H. The associated eigenvalues $\{\lambda_1, \lambda_2, \ldots\}$ satisfy $0 < \lambda_1 \le \lambda_2 \le \cdots$ and the asymptotic formula $\lambda_j \sim j$. Properties of the spaces H, $V = D(A^{1/2})$, and $D(A)$ may be found in [2], [18], or [20].

We decompose the solution u into low mode and high mode components by letting $H_N = \text{span}\{\omega_1, \omega_2, \ldots, \omega_N\}$, the span of the first N eigenfunctions of the Stokes operator A. Let P_N be the orthogonal projection of H onto H_N, and $Q_N = I - P_N$ be the projection onto the orthogonal complement space $H_N^\perp$. Then, for any $u \in H$, we can uniquely decompose $u = p + q$, where $p = P_N u$ and $q = Q_N u$. Projecting (1.2)

onto H_N and $H_N^{\perp}$, we get an equivalent system for the NSE

$$\frac{dp}{dt} + \nu Ap + P_N\left[B(p,p) + B(p,q) + B(q,p) + B(q,q)\right] = P_N f, \tag{1.3}$$

$$\frac{dq}{dt} + \nu Aq + Q_N\left[B(p,p) + B(p,q) + B(q,p) + B(q,q)\right] = Q_N f, \tag{1.4}$$

$$p(0) = P_N u_0 \quad \text{and} \quad q(0) = Q_N u_0.$$

The truncation analysis is accomplished by using estimates for the low modes p and high modes q of a solution u. We present the truncation analysis for the two-dimensional NSE; however, similar analysis may be done for general nonlinear parabolic evolution or elliptic equations, such as reaction-diffusion systems, the Bénard convection problem, etc. The key to the analysis is understanding the interaction of the low and high modes, and estimating the nonlinear term.

The truncation analysis is based on asymptotic (in time) estimates for the low and high mode components, as was done in [10] and [11] when developing the postprocessing algorithm. These asymptotic estimates hold when solutions of the NSE are on, or near, the attractor, i.e., for the case of autonomous systems (f time-independent) and provided that t is large enough. However, these estimates may not hold, for example, in the case of nonsmooth initial data, long transients, or nonautonomous systems with highly oscillatory (in time) forcing. In [15] the authors showed that, for a highly oscillatory time-dependent forcing function, the dominant balance in (1.4) is between the dq/dt term and the forcing term; hence the dq/dt term should not be dropped in the AIM construction. This case leads to and justifies an alternate/reform postprocessing method, proposed in [24] for integrating along transients, which we call here *dynamic postprocessing*.

Let us emphasize again that there is a basic difference between the nonlinear Galerkin methods and the postprocessing Galerkin method. Specifically, unlike the usual multigrid (in this case two-grid) and the nonlinear Galerkin methods, in the postprocessing Galerkin methods the evolution/integration on the coarse mesh, i.e., low frequencies, does not use at all the information on the fine mesh (small scales or high frequencies). Only at the end of the calculations does one use the solution on the coarse mesh to refine the solution. On the other hand, in standard two-grid methods, including the nonlinear Galerkin methods and their variants, one uses cycles in which one has to compute the solution on the fine mesh in order to update the time step integration on the coarse mesh and vice versa. In fact, this occasional updating of the solutions on the fine mesh is the major source of computational disadvantage of the nonlinear Galerkin method in comparison to the Galerkin method, as was demonstrated computationally in, for instance, [10] and [11].

In this paper we first present a classical truncation analysis of the NSE using established asymptotic (in time) estimates. In section 2 we present several approximate systems of varying orders of accuracy based on the truncation analysis results. We introduce a more general postprocessing algorithm in section 3 for the case in which the asymptotic estimates no longer hold (i.e., in the presence of long transients, nonsmooth initial data, or highly oscillatory forcing). In section 4 we analyze the accuracy of the various postprocessing methods for the case of a highly time-oscillatory solution. In section 5 we present some numerical experiments to support the analysis of sections 2, 3, and 4 and compare the computational efficiency of the standard and the more general postprocessing methods. Finally, we give some concluding remarks in section 6. Preliminary results of this study were reported in [23].

2. Near-attractor truncation analysis. We first present the truncation analysis based on asymptotic estimates for u, p, and q. It is well known (see, e.g., [2] or [18]) that for $f \in H$ and independent of time, (1.2) is dissipative in the spaces H, V, and $D(A)$. This means that any solution $u(t)$ of (1.2) will, after a certain time, enter and remain in a ball in H centered at 0 with radius ρ_0. The same is true for a ball in V of radius ρ_1, and a ball in $D(A)$ of radius ρ_2. The radii ρ_0, ρ_1, and ρ_2 depend on $\|f\|$, ν, and λ_1. Therefore, we will assume that for $t \geq T_0$, for some positive T_0 that depends on ν, $\|f\|$, λ_1, and the initial data $\|u_0\|$, we have

$$\|u(t)\| \leq \rho_0, \qquad \|A^{1/2}u(t)\| \leq \rho_1, \qquad \|Au(t)\| \leq \rho_2. \tag{2.1}$$

Notice that the global attractor for (1.2) is contained in these balls. For solutions on the attractor, $T_0 = 0$ and the uniform bounds apply for all time $t \in \mathbb{R}$, since the global attractor is invariant (see, e.g., [2] and [18]).

From the above bounds for u, we have that q is also bounded in H, $D(A^{1/2})$, and $D(A)$ for $t > T_0$. Using the bound $\|Aq\| \leq \rho_2$ and the fact that $\|A^{\alpha}q\| \leq \lambda_{N+1}^{-\alpha}\|q\|$, we quickly obtain estimates for q in terms of λ_{N+1}. We denote $\epsilon = (\lambda_1/\lambda_{N+1})^{1/2}$. Then for $t > T_0$ the following estimates for q and dq/dt are at hand:

$$\begin{aligned} \|q\| &\leq \lambda_{N+1}^{-1}\|Aq\| \leq \lambda_{N+1}^{-1}\rho_2 = O(\epsilon^2), \\ \|A^{1/2}q\| &\leq \lambda_{N+1}^{-1/2}\|Aq\| \leq \lambda_{N+1}^{-1/2}\rho_2 = O(\epsilon), \\ \|Aq\| &\leq \|Au\| \leq \rho_2 = O(1) \end{aligned} \tag{2.2}$$

as $\epsilon \to 0$. Using the fact that the solutions are analytic in time (see, e.g., [2] and [20]), one can apply the Cauchy formula for the derivatives of complex analytic functions to obtain an estimate for $\|dq/dt\|$ of the same order as $\|q\|$ (again, see, e.g., [2], [5], and [20]). We have

$$\left\| \frac{dq}{dt} \right\| = O(\epsilon^2) \quad \text{as } \epsilon \to 0. \tag{2.3}$$

Let us stress that the constant ρ_2, which depends on the physical parameters but not on N, is quite large in comparison with the constants ρ_0 or ρ_1 for small values of the viscosity ν or large values of $\|f\|$. It is preferable to avoid using the ρ_2 bound and to derive more delicate estimates for $\|q\|$, $\|dq/dt\|$, and $\|A^{1/2}q\|$ of the same orders as above involving only ρ_0 and ρ_1. Indeed, the authors of [5] derive bounds of the type given in (2.2) and (2.3) involving ρ_0 and ρ_1 but not ρ_2. However, this is done at the expense of adding a term of the order $|\log \epsilon|$. In practice, this is a more reasonable bound, since the best available bound for ρ_2 is many orders of magnitude larger than those for ρ_0 and ρ_1. Moreover, for practical computations, $|\log \epsilon|$ will be of order 1 even if ϵ is very small.

For the low mode component, we have only that p is bounded in H, $D(A^{1/2})$, and $D(A)$ for $t > T_0$. Hence, we set

$$\|p\|,\ \|A^{1/2}p\|,\ \|Ap\| = O(1) \quad \text{as } \epsilon \to 0. \tag{2.4}$$

For the truncation analysis we consider a perturbation expansion for q of the form

$$q = q_1 + q_2 + q_3 + q_4 + \cdots. \tag{2.5}$$

To leading order, we have estimates (2.2) and (2.3) for q. Hence, the corresponding estimates for the first expansion term q_1 are as follows:

$$\begin{aligned} \|q_1\|,\ \left\|\frac{dq_1}{dt}\right\| &= O(\epsilon^2), \\ \|A^{1/2}q_1\| &= O(\epsilon), \\ \|Aq_1\| &= O(1) \quad \text{as } \epsilon \to 0. \end{aligned} \tag{2.6}$$

Each successive term q_j is assumed to be of higher order in ϵ, i.e., $\|dq_j/dt\|, \|q_j\| = O(\epsilon^{j+1})$, $\|A^{1/2}q_j\| = O(\epsilon^j)$, and $\|Aq_j\| = O(\epsilon^{j-1})$, for $j = 1, 2, \ldots$. In principle, the initial value u_0 should also be decomposed accordingly, i.e., $u_0 = P_N u_0 + Q_N u_0$, with $Q_N u_0 = q_1^0 + q_2^0 + \cdots$ such that $O(q_j^0) = \epsilon O(q_{j-1}^0)$. In particular, for solutions on or near the attractor, we should have $\|q_j^0\| = O(\epsilon^{j+1})$, $\|A^{1/2}q_j^0\| = O(\epsilon^j)$, and $\|Aq_j^0\| = O(\epsilon^{j-1})$, for $j = 1, 2, \ldots$.

We substitute expansion (2.5) into system (1.3)–(1.4) above and estimate the order of each term in the system. By keeping terms up to order $\epsilon^{1/2}$, $\epsilon^{3/2}$, and so on, we generate approximate systems for NSEs of increasing orders of accuracy. The challenge comes with estimating the nonlinear terms. Substituting expansion (2.5) into the nonlinear terms results in the following:

$$\begin{aligned} B(p,q) &= B(p, q_1 + q_2 + q_3 + \cdots) \\ &= B(p,q_1) + B(p,q_2) + B(p,q_3) + \cdots, \\ B(q,p) &= B(q_1 + q_2 + q_3 + \cdots, p) \\ &= B(q_1,p) + B(q_2,p) + B(q_3,p) + \cdots, \\ B(q,q) &= B(q_1 + q_2 + q_3 + \cdots, q_1 + q_2 + q_3 + \cdots) \\ &= B(q_1,q_1) + B(q_1,q_2) + B(q_1,q_3) + \cdots \\ &\quad + B(q_2,q_1) + B(q_2,q_2) + B(q_2,q_3) + \cdots \\ &\quad + B(q_3,q_1) + B(q_3,q_2) + B(q_3,q_3) + \cdots \\ &\quad + \cdots. \end{aligned}$$

We majorize each term using inequalities for the nonlinear term given, for instance, in [2], [20], or [22] for inequalities (2.9) and (2.10). For convenience we recall the two-dimensional version of these inequalities. For any $u, v \in D(A)$,

$$\|B(u,v)\| \le c_1 \|u\|^{1/2} \|A^{1/2}u\|^{1/2} \|A^{1/2}v\|^{1/2} \|Av\|^{1/2} \tag{2.7}$$

$$\le c_2 \|u\|^{1/2} \|Au\|^{1/2} \|A^{1/2}v\| \tag{2.8}$$

$$\le c_3 \|A^{1/2}u\| \|A^{1/2}v\| \left(1 + \log \frac{\|Au\|^2}{\lambda_1 \|A^{1/2}u\|^2}\right)^{1/2} \tag{2.9}$$

and

$$\|B(u,v)\| \le c_4 \|u\|\ \|Av\| \left(1 + \log \frac{\|A^{3/2}v\|^2}{\lambda_1 \|Av\|^2}\right)^{1/2} \tag{2.10}$$

for $u \in D(A)$ and $v \in D(A^{3/2})$. The constants c_1–c_4 are independent of u, v, and the size of Ω, but might depend on its shape. For each nonlinear term we choose the inequality that results in the highest order of ϵ. Using estimates (2.4) for p and (2.6) for q_1, we obtain $\|B(p,p)\| = O(1)$, $\|B(p,q_1)\| = O(\epsilon)$, and $\|B(q_1,q_1)\| = O(\epsilon^2)$. For the $B(q_1,p)$ term, inequality (2.7) gives $\|B(q_1,p)\| = O(\epsilon^{3/2})$, and inequality (2.10)

gives $\|B(q_1,p)\| = O(\epsilon^2 L_\epsilon^{1/2})$, where $L_\epsilon = (1+2|\log\epsilon|)$. The $O(\epsilon^2 L_\epsilon^{1/2})$ estimate is "closer" to being of the order $O(\epsilon^2)$ than $O(\epsilon)$. However, in either case, the term is definitely of the order $O(\epsilon^{3/2})$. For simplicity of ordering the various terms, we will consider this term to be $O(\epsilon^{3/2})$.

2.1. Near-attractor approximate systems. To produce approximate schemes for the Navier–Stokes system, we keep only terms in (1.3)–(1.4) to certain orders in ϵ. Below, we list the approximate systems produced by keeping terms to order $\epsilon^{1/2}$ and $\epsilon^{3/2}$. We will set nonlinear terms of the order $O(\epsilon^2 L_\epsilon^{1/2})$ and $O(\epsilon^3 L_\epsilon^{1/2})$ to be of the order $O(\epsilon^{3/2})$ and $O(\epsilon^{5/2})$, respectively.

$\boxed{O(\epsilon^{1/2})\text{:}}$

$$\frac{dp}{dt} + \nu Ap + P_N\left[B(p,p)\right] \approx P_N f, \tag{2.11}$$

$$\nu Aq_1 + Q_N\left[B(p,p)\right] \approx Q_N f, \tag{2.12}$$

$$p(0) = P_N u_0. \tag{2.13}$$

Equation (2.11) is an evolution equation for the low mode component p. Equation (2.12) is coupled to (2.11); it defines q_1, the leading order approximation term of the high modes, in terms of the low modes and is therefore a postprocessing step. From (2.12) one can verify that $\|Aq_1\| = O(1)$, which is consistent with our assumptions.

$\boxed{O(\epsilon^{3/2})\text{:}}$

$$\frac{dp}{dt} + \nu Ap + P_N\left[B(p,p) + B(p,q_1) + B(q_1,p)\right] \approx P_N f, \tag{2.14}$$

$$\nu A(q_1+q_2) + Q_N\left[B(p,p) + B(p,q_1) + B(q_1,p)\right] \approx Q_N f, \tag{2.15}$$

$$\nu Aq_2 + Q_N\left[B(p,q_1) + B(q_1,p)\right] \approx 0, \tag{2.16}$$

$$p(0) = P_N u_0. \tag{2.17}$$

Equation (2.14) is the evolution equation for p with q_1 in the nonlinear term defined by (2.12); it is a nonlinear Galerkin method as defined in [5], [13], [14], and [16]. Equation (2.15) defines q_1+q_2, which is a higher order approximation of the high modes. Equation (2.16) defines q_2; it is derived from (2.15) and the definition of q_1 given in (2.12). From (2.16) one can show that $\|Aq_2\| \le \|B(p,q_1)\| + \|B(q_1,p)\| = O(\epsilon)$. Hence $\|q_2\| = O(\epsilon^2)$, which is consistent with our assumptions.

Similarly, we may obtain an approximate system for the NSE valid to order $O(\epsilon^{5/2})$, $O(\epsilon^{7/2})$, and in general, valid to order $O(\epsilon^{j+1/2})$. In the general case, the low mode equation is evolved with linear combinations of q_1 through q_j in the nonlinear term; the high mode equation involves linear combinations of q_1 through q_{j+1}, where $(q_1+q_2+\cdots+q_{j+1})$ is used to approximate q. Thus the high modes of the solution u should be approximated to one order higher in ϵ than the order of the high mode terms used in the low mode equation. The term q_{j+1} is not used to evolve the low modes; it needs to be evaluated only once, at some final time T, and may therefore be considered a postprocessing step. The approximate systems above, produced with a classical truncation analysis, demonstrate that the postprocessing step is a very natural and significant part of approximating the original system.

2.2. Standard (near-attractor) postprocessing schemes. For each approximate system from the previous section, we may generate a postprocessed Galerkin

or nonlinear Galerkin scheme of increasing order of accuracy. From the truncation analysis we know that, to approximate the low and high modes of a solution u to the same order in ϵ, we must include the postprocessing step. In general the solution of the evolution equation is sought as an approximation of the low modes of the exact solution u, and the solution of the high mode equation is sought as an approximation of the high modes of the solution u. The goal for each $\epsilon^{j+1/2}$ postprocessing scheme is to produce a more accurate approximation of the low and high modes as j increases.

From system (2.11)–(2.12), i.e., keeping terms to order $O(\epsilon^{1/2})$, we obtain the postprocessed standard Galerkin method

$$\frac{du_N}{dt} + \nu A u_N + P_N\left[B(u_N, u_N)\right] = P_N f, \tag{2.18}$$

$$\nu A\phi_1 + Q_N\left[B(u_N, u_N)\right] = Q_N f, \tag{2.19}$$

$$u_N(0) = P_N u_0, \tag{2.20}$$

where $u_N \in H_N$ is the solution of the evolution equation and is an approximation of the low modes p, and $\phi_1 \in Q_N H$ is an approximation of the high modes q (i.e., $\phi_1 \approx q_1$). Note that $\phi_1(t) = \Phi^1(u_N(t))$, where Φ^1 is exactly the Foias–Manley–Temam (FMT) AIM first introduced in [5]. This is the same postprocessed Galerkin method originally defined in [10] and [11]. Solving for $u_N(t)$ does not depend on ϕ_1, and hence one does not need to evaluate $\phi_1(t) = \Phi^1(u_N(t))$ at all times, but only when an approximate solution is needed. This is typically done once at some final time T. It is therefore a postprocessing step. The approximate solution at time T is then $u_N(T) + \phi_1(T) = u_N(T) + \Phi^1(u_N(T))$, and not $u_N(T)$ as is traditionally used with the standard Galerkin method. This scheme indicates that, to leading order in ϵ, the correct approximation method is the postprocessed Galerkin method.

The approximation properties of the postprocessed Galerkin method ($j = 0$) are well understood. We know, for instance, that Φ_1 is Lipschitz continuous, $\|\phi_1(t)\| = O(\epsilon^2)$, $\|q(t) - \phi_1(t)\| = O(\epsilon^3)$, and $\|p(t) - u_N(t)\| = O(\epsilon^3)$. Proofs of the first three properties may be found, for example, in [4] and [5]. The fourth property is proven in [11], specifically for the two-dimensional NSE.

In general, keeping terms of order $\epsilon^{j+1/2}$ for $j \geq 1$, we obtain a postprocessed nonlinear Galerkin scheme with successively more accurate approximation properties as j increases. In particular, for the scheme generated from system (2.14)–(2.16), i.e., the case $j = 1$, we can show that the low and high mode approximation errors are of the order $O(\epsilon^4)$ using the techniques as in [11] and [17]. Though more accurate, the $O(\epsilon^{j+1/2})$ systems are not computationally competitive for $j \geq 1$. We know from numerical experiments presented in [10], [11], and [17] that the more computationally efficient schemes are the postprocessed Galerkin method (2.18)–(2.19) and variants thereof, such as the postprocessed filtered Galerkin method. Hence, for the purposes of this paper, we will concentrate on the postprocessed Galerkin method, system (2.18)–(2.20).

3. A more general truncation analysis. The standard postprocessing scheme and systems in the previous section were generated based on asymptotic (in time) estimates for the low and high mode components. These estimates hold for autonomous systems when solutions of the NSE are on or near the attractor, i.e., for t large enough. However, these estimates may no longer hold, for instance, in the case of nonsmooth initial data, long transients, or nonautonomous systems with highly oscillatory time-dependent forcing. For these cases, the leading order approximation for system (1.3)–(1.4) is no longer clear. In particular, the dq/dt term may no longer be small in

comparison with the other terms in (1.4). For instance, in the case of a highly oscillatory time-dependent force, the authors of [15] presented an analytic example showing that the dominant balance in (1.4) is between the dq/dt term and the forcing term, and not between the dissipative term and the forcing and nonlinear terms. In this case they concluded that the dq/dt term should not be dropped in the AIM construction. In this section we consider the special case in which the forcing f is a highly oscillatory time-dependent function.

We start with the nonautonomous Navier–Stokes system (1.3)–(1.4) with highly oscillatory forcing. We assume that the force remains bounded (i.e., $f \in L^\infty((0,\infty);H)$ but oscillatory in time (defined later in Theorem 4.4). Furthermore, we assume that the solution $u(t)$ is bounded in $D(A)$ for $t \geq 0$ and that the initial condition is smooth, i.e., $u_0 \in D(A)$. As before, we observe that $\|Au(t)\| = O(1)$ since $\|Au(t)\|$ is bounded uniformly. Then for all $t \geq 0$ we have

$$\|p(t)\|, \|A^{1/2}p(t)\|, \|Ap(t)\| = O(1) \qquad \text{and} \qquad \|Aq(t)\| = O(1).$$

Again using the fact that $\|q\| \leq \lambda_{N+1}^{-\alpha}\|A^\alpha q\|$, we obtain that $\|q(t)\| = O(\epsilon^2)$ and $\|A^{1/2}q(t)\| = O(\epsilon)$ as before. Since the forcing is highly oscillatory in time, we cannot assume that the time derivative of the solution u, and hence q, is necessarily small. In this situation we will suppose that $\|dq/dt\| = O(1)$, the same order as the $\|Aq(t)\|$ term or larger. We have the following bounds for q,

$$\begin{aligned} \|q(t)\| &= O(\epsilon^2), \\ \|A^{1/2}q(t)\| &= O(\epsilon), \\ \|Aq(t)\|, \|dq/dt\| &= O(1). \end{aligned} \tag{3.1}$$

Without assuming that the forcing term is real analytic in time with values in H, one could not show that the solution $u(t)$ is real analytic in time with values in $D(A)$. Therefore, it would not be possible to employ the techniques used in [5] to get tight estimates on the constants involved in the bounds given in (3.1).

3.1. More general approximate systems. For the truncation analysis, we again assume a perturbation expansion for q of the form $q = (q_1 + q_2 + q_3 + \cdots)$. Since q_1 is the leading order approximation for q, the above estimates hold for q_1 as well. We then substitute the perturbation expansion for q into system (1.3)–(1.4) and estimate the orders of the various terms as before. The only differences are the orders of the $d(q_1 + q_2 + \cdots)/dt$ terms.

Keeping terms up to order $\epsilon^{1/2}$, we have the following leading order approximate system:

$$\frac{dp}{dt} + \nu Ap + P_N\left[B(p,p)\right] \approx P_N f, \tag{3.2}$$

$$\frac{dq_1}{dt} + \nu Aq_1 + Q_N\left[B(p,p)\right] \approx Q_N f, \tag{3.3}$$

$$p(0) = P_N u_0, \tag{3.4}$$

$$q_1(0) = Q_N u_0. \tag{3.5}$$

Equation (3.2) is the usual evolution equation for the low mode component; it is the standard Galerkin method. Equation (3.3) is used to define q_1, the leading order approximation of the high modes, only now it is an evolution equation. Here q_1 is not needed for the evolution of the low modes; hence (3.3) may be considered a

postprocessing step. This is the same postprocessing step introduced in [24] for the case of nonsmooth initial data and long transients, and justifies the postprocessing method given therein. It is worth noting that one can think about the above system (3.2)–(3.5) as a two-level multigrid method, where one integrates (3.2) on the coarse mesh and then postprocesses on the fine mesh using (3.3).

Keeping terms up to order $\epsilon^{3/2}$, we have the following approximate system:

$$\frac{dp}{dt} + \nu Ap + P_N\left[B(p, p+q_1) + B(q_1, p)\right] \approx P_N f, \tag{3.6}$$

$$\frac{d(q_1+q_2)}{dt} + \nu A(q_1+q_2) + Q_N\left[B(p, p+q_1) + B(q_1, p)\right] \approx Q_N f, \tag{3.7}$$

$$p(0) = P_N u_0, \tag{3.8}$$

$$(q_1+q_2)(0) = Q_N u_0. \tag{3.9}$$

Equation (3.6) is the same evolution equation for p as in (2.14), but with q_1 now defined by (3.3). It is a nonlinear Galerkin method. Equation (3.7) defines $q_1 + q_2$, the high mode approximation. From (3.3), (3.6), and (3.7) one concludes

$$\frac{dp}{dt} + \nu Ap + P_N\left[B(p, p+q_1) + B(q_1, p)\right] \approx P_N f, \tag{3.10}$$

$$\frac{dq_1}{dt} + \nu Aq_1 + Q_N\left[B(p, p)\right] \approx Q_N f, \tag{3.11}$$

$$\frac{dq_2}{dt} + \nu Aq_2 + Q_N\left[B(p, q_1) + B(q_1, p)\right] \approx 0, \tag{3.12}$$

$$p(0) = P_N u_0, \tag{3.13}$$

$$q_1(0) = q_1^0, \tag{3.14}$$

$$q_2(0) = q_2^0. \tag{3.15}$$

Equation (3.12) is a postprocessing step since q_2 is not used in the evolution equation for the low mode component p. Here again one can think about the above scheme as a two-level multigrid method.

We may continue this process as before, keeping terms to higher and higher orders in ϵ to generate a general postprocessing scheme. However, for computational efficiency, we are interested only in the leading order postprocessing algorithms.

3.2. A dynamic postprocessing scheme. Motivated by the approximate system (3.2)–(3.5), we introduce the *dynamic postprocessing scheme*

$$\frac{du_N}{dt} + \nu Au_N + P_N\left[B(u_N, u_N)\right] = P_N f, \tag{3.16}$$

$$\frac{d\tilde{\phi}_1}{dt} + \nu A\tilde{\phi}_1 + Q_N\left[B(u_N, u_N)\right] = Q_N f, \tag{3.17}$$

$$u_N(0) = P_N u_0, \tag{3.18}$$

$$\tilde{\phi}_1(0) = Q_N u_0, \tag{3.19}$$

where the approximation for the high modes $\tilde{\phi}_1$ is obtained as the solution of evolution equation (3.17). Notice that $\tilde{\phi}_1 = \tilde{\phi}_1(t; u_N(t))$.

4. Error analysis. In the following, we will compare the accuracy of the standard postprocessing method, system (2.18)–(2.20), with the dynamic postprocessing method, system (3.16)–(3.19), in the case of a highly oscillatory forcing function.

Since the approximation for the low mode component is exactly the same in each case, namely the Galerkin approximation, we will compare only the postprocessing approximation of the high mode component.

For comparison purposes we will use the uniform bounds

$$\|Au(t)\| \le \rho_2, \qquad \|Au_N(t)\| \le \rho_2^*, \qquad t \ge 0, \tag{4.1}$$

where ρ_2 and ρ_2^* are constants which depend on the data of the problem (i.e., ν, f, $\|u_0\|$, and λ_1) but are independent of N. Let us observe that usually $\rho_2^* = \rho_2$. We will also utilize a low mode accuracy estimate, which we restate below without proof (see [11], Theorem 1).

THEOREM 4.1. *Let $T > 0$ be fixed. Let $u = p + q$ be the solution of* (1.2) *on* $[0, T]$ *such that the bounds in* (3.1) *and* (4.1) *hold. Then, there exists a constant* $C = C(T, \rho_1, \rho_2)$ *such that for any* $t \in [0, T]$ *the solution* $u_N(t)$ *of* (2.18) *and* (2.20) *satisfies*

$$\|p(t) - u_N(t)\| \le C \frac{L_\epsilon^2}{\lambda_{N+1}^{3/2}} = O(\epsilon^3 L_\epsilon^2), \tag{4.2}$$

where $L_\epsilon = 1 + 2|\log \epsilon| = 1 + \log(\lambda_{N+1}/\lambda_1)$.

The theorem is proven in the case of f time-independent. However (see [11, Remark 2]), f plays no role in the estimates, and hence the result is valid for $f = f(t)$ as well.

We first work with the leading order postprocessing method presented in section 2, whose corresponding scheme is given by (2.18)–(2.19). Here u_N is the Galerkin low mode approximation. The high mode approximation is given by $\phi_1 = \Phi^1(u_N)$, where Φ^1 is the FMT AIM introduced in [5] and is defined in general as

$$\Phi^1(v) = (\nu A)^{-1}(Q_N f - Q_N B(v, v)), \qquad v \in H_N.$$

A common approach for estimating the error $\|q(t) - \Phi^1(u_N(t))\|$ is to first bound $\|q - \Phi^1(p)\|$ using asymptotic estimates for p, q, and dq/dt, where $u(t) = p(t) + q(t)$ is the exact solution. Since we no longer assume that $\|dq/dt\|$ is small, we first reexamine the $\|q - \Phi^1(p)\|$ estimate in the case of a highly oscillatory forcing function. We have the following theorem.

THEOREM 4.2. *Let $f(t) \in L^\infty((0, \infty); H)$ and $u_0 \in D(A)$. Then for any solution $u(t) = p(t) + q(t)$ of* (1.2), *and $u_N(t)$ the solution of* (2.18) *and* (2.20) *such that estimates* (3.1) *and* (4.1) *hold, we have*

$$\|q(t) - \Phi^1(p(t))\| \le \frac{C}{\nu \lambda_{N+1}} \left(\left\| \frac{dq}{dt} \right\| + \lambda_{N+1}^{-1/2} \|Aq\| \right), \tag{4.3}$$

$$\|q(t) - \Phi^1(u_N(t))\| \le L\|p(t) - u_N(t)\| + \frac{C}{\nu \lambda_{N+1}} \left(\left\| \frac{dq}{dt} \right\| + \lambda_{N+1}^{-1/2} \|Aq\| \right) \tag{4.4}$$

for every $t \ge 0$. Here L is the Lipschitz constant for Φ^1, which is known to be of the order $o(1)$ as $\lambda_{N+1} \to \infty$ (see, e.g., [4], [5], *and* [21]*).*

Proof. Let $t \ge 0$. Subtracting (2.11) from (1.4) and taking the L^2 norm, we obtain

$$\|\nu A(q(t) - \Phi^1(p(t)))\| \le \left\| \frac{dq}{dt} \right\| + \|B(p, q)\| + \|B(q, p)\| + \|B(q, q)\|.$$

Using inequalities (2.7)–(2.10) to bound the nonlinear terms, we have

$$\|\nu A(q-\Phi^1(p))\| \le \left\|\frac{dq}{dt}\right\| + \frac{c_1\|Ap\|}{\lambda_{N+1}^{1/2}}\|Aq\| + \frac{c_1\|Ap\|}{\lambda_{N+1}^{3/4}}\|Aq\| + \frac{c_1}{\lambda_{N+1}}\|Aq\|^2,$$

and hence

$$\|q-\Phi^1(p)\| \le (\nu\lambda_{N+1})^{-1}\left(\left\|\frac{dq}{dt}\right\| + \frac{C}{\lambda_{N+1}^{1/2}}\|Aq\|\right), \tag{4.5}$$

where $C = C(c_1, \rho_1, \rho_2)$. This proves estimate (4.3). To obtain estimate (4.4), first apply the triangle inequality,

$$\|q(t)-\Phi^1(u_N(t))\| \le \|q(t)-\Phi^1(p(t))\| + \|\Phi^1(p(t))-\Phi^1(u_N(t))\|.$$

Then use the Lipschitz continuity of Φ^1 (see [5]) and estimate (4.3). ∎

In this section we assume that $\|dq/dt\|, \|Aq\| = O(1)$, and thus $\|q-\Phi^1(p)\| = O(\epsilon^2)$, rather than of the order $O(\epsilon^3) = O(\lambda_{N+1}^{-3/2})$ for the solutions on or near the attractor in the case of autonomous systems. Different asymptotic estimates for the q and dq/dt terms result in different accuracy estimates for $\|q-\Phi^1(p)\|$ and for the total accuracy estimate of the standard postprocessing algorithm. In particular, using Theorem 4.1 to bound the $\|p(t)-u_N(t)\|$ term, the bounds for $\|dq/dt\|$ and $\|Aq\|$ dominate the error in estimate (4.4). The accuracy estimate for the high mode approximation using the standard postprocessing method is given below.

COROLLARY 4.3. *Let $f(t) \in L^\infty((0,\infty);H)$, $u_0 \in D(A)$, and $T > 0$. Then for any solution $u(t) = p(t) + q(t)$ of (1.2), and $u_N(t)$ the solution of (2.18) and (2.20) such that the estimates (3.1) and (4.1) hold, we have*

$$\|q(t)-\Phi^1(u_N(t))\| = O(\epsilon^2) \tag{4.6}$$

for $t \ge 0$.

Note that, without a better estimate for the $\|dq/dt\|$ term, we could easily have obtained the same estimate for $\|q-\Phi^1(p)\|$ by first applying the triangle inequality to get

$$\|q-\Phi^1(p)\| \le \|q\| + \|\Phi^1(p)\|.$$

Then, under the assumptions of Theorem 4.2, one can show that $\|\Phi^1(p)\| = O(\lambda_{N+1}^{-1}) = O(\epsilon^2)$. Since $\|q\| = O(\epsilon^2)$ as well, we obtain $\|q-\Phi^1(p)\| = O(\epsilon^2)$. Hence, Corollary 4.3 only indicates that q and $\Phi^1(p)$ are of the same order.

With additional assumptions on $f(t)$, the above estimate for $\|q-\Phi^1(p)\|$ may be improved. In particular, in [15] the authors show that $\|q-\Phi^1(p)\| = O(\epsilon^{1+2\theta})$ for f Hölder continuous in time (with exponent θ), with values in H for N large enough. For convenience we restate the theorem.

THEOREM 4.4. *Let $f(t)$ be Hölder continuous (i.e., $\|f(t_1)-f(t_2)\| \le L_1|t_1-t_2|^\theta$) and satisfy $\sup_{t\ge0}|f(t)| \le f_\infty < \infty$. Furthermore, impose sufficient conditions on $f(t)$ so that $\|Au\|$ is uniformly bounded and, hence, the solution $p(t)$ of (1.3) is uniformly Lipschitz in time, (i.e., $\|p(t_1)-p(t_2)\| \le L_2|t_2-t_1|$, where L_2 depends on ν, f_∞, and λ_1). Let $\|q(0)-\Phi^1(p(0))\| = O(\lambda_{N+1}^{(1/2+\theta)})$. Then, for N sufficiently large and $t \ge 0$, any solution $u(t) = p(t) + q(t)$ of (1.2) satisfies*

$$\|q(t)-\Phi^1(p(t))\| \le \frac{4\alpha_5}{\lambda_{N+1}^{1/2+\theta}},$$

where $\alpha_5 = \alpha_4(1+(1+e)^{-1})$, $\alpha_4 = \lambda_{N+1}^{-1/2}2\nu^{-1}(\alpha_3 L_2\lambda_N^{1/2}+L_1)+\alpha_2 L_2+\nu^{-1}L_1\lambda_{N+1}^{-1/2}$, L_1 *is the Hölder constant for* f, *and* L_2 *is the Lipschitz constant for* p.

Proof. We refer the reader to [15, Theorem 5.11] for specific conditions on N, definitions of α_2, α_3, α_4, and the proof of the above theorem. ∎

If $\theta > 1/2$, then Theorem 4.4 represents an improvement from the previous $O(\epsilon^2)$ estimate for $\|q-\Phi^1(p)\|$, where f was only assumed to be bounded. Using the Lipschitz continuity of Φ^1 and Theorem 4.1, we have an improved estimate for $\|q-\Phi^1(u_N)\|$ in the case in which f is Hölder continuous.

COROLLARY 4.5. *Let* $f(t)$ *satisfy the conditions of Theorem* 4.4, $u_0 \in D(A)$, $T > 0$, *and* N *sufficiently large. Then for any solution* $u(t) = p(t) + q(t)$ *of* (1.2), *and* $u_N(t)$ *the solution of* (2.18) *and* (2.20) *such that estimates* (3.1) *and* (4.1) *hold, we have*

$$\|q(t)-\Phi^1(u_N(t))\| \le L\|p(t)-u_N(t)\| + \frac{4\alpha_5}{\lambda_{N+1}^{1/2+\theta}} = O(\epsilon^{2+2(\theta-1/2)}) \tag{4.7}$$

for $t \in [0,T]$. *Here* $L = o(1)$ *is the Lipschitz constant for* Φ^1.

We now examine the dynamic postprocessing method, system (3.16)–(3.19), and obtain an estimate for $\|q(t)-\tilde{\phi}_1(t;u_N(t))\|$.

THEOREM 4.6. *Let* $f(t) \in L^\infty((0,\infty);H)$, $u_0 \in D(A)$, *and* $T > 0$. *Let* $u(t) = p(t)+q(t)$ *be a solution of* (1.2), *and* $u_N(t)$ *and* $\tilde{\phi}_1(t;u_N(t))$ *be a solution of system* (3.16)–(3.19) *such that estimates* (3.1) *and* (4.1) *hold. Then, for* $t \in [0,T]$, *we have*

$$\|q(t)-\tilde{\phi}_1(t;u_N(t))\| \le \frac{C}{\nu\lambda_{N+1}^{1/2}}\left(\max_{s\in[0,T]}\|p(s)-u_N(s)\| + \lambda_{N+1}^{-1}\max_{s\in[0,T]}\|Aq(s)\|\right), \tag{4.8}$$

where $C = C(\rho_2,\rho_2^*)$; ρ_2 *and* ρ_2^* *are defined in* (2.1) *and* (4.1), *respectively.*

Proof. We subtract (3.17) from (1.4). Letting $\Delta(t) = q(t)-\tilde{\phi}_1(t;u_N(t))$, we have

$$\frac{d\Delta}{dt} + \nu A\Delta = Q_N\left[B(p+q,q)+B(q,p)+B(p,p-u_N)+B(p-u_N,u_N)\right].$$

Taking the inner product of Δ and the above equation, we obtain

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\|\Delta\|^2 + \nu\|A^{1/2}\Delta\|^2 \le{}& |(B(p+q,q),\Delta)| + |(B(q,p),\Delta)| \\ &+ |(B(p,p-u_N),\Delta)| + |(B(p-u_N,u_N),\Delta)|.\end{aligned}$$

We apply the Cauchy–Schwarz inequality and Young's inequality to get a factor of $\nu\|A^{1/2}\Delta\|$ from each term on the right-hand side. Combining all $\|A^{1/2}\Delta\|$ terms with the left-hand side of the inequality and using the fact that $\lambda_{N+1}^{1/2}\|\Delta\| \le \|A^{1/2}\Delta\|$, we have

$$\begin{aligned}\frac{d}{dt}\|\Delta\|^2 + \nu\lambda_{N+1}\|\Delta\|^2 \le \frac{5}{\nu\lambda_{N+1}}\big(&\|B(p,q)\|^2+\|B(q,p)\|^2+\|B(q,q)\|^2 \\ &+\|B(p,p-u_N)\|^2+\|B(p-u_N,u_N)\|^2\big).\end{aligned}$$

Estimating the nonlinear terms using estimates (2.7)–(2.10) as before,

$$\begin{aligned}\frac{d}{dt}\|\Delta\|^2 + \nu\lambda_{N+1}\|\Delta\|^2 &\le \frac{5}{\nu\lambda_{N+1}}\Big(c_2^2\|p\|\|Ap\|\|A^{1/2}q\|^2 \\ &\quad + c_4^2(1+\log(\lambda_N/\lambda_1))\|Ap\|^2\|q\|^2 + c_2^2\|q\|\|A^{1/2}q\|^2\|Aq\| \\ &\quad + c_2^2\|p\|\|Ap\|\|A^{1/2}(p-u_N)\|^2 + c_4^2(1+\log(\lambda_N/\lambda_1))\|Au_N\|^2\|p-u_N\|^2\Big) \\ &\le \frac{5}{\nu\lambda_{N+1}}\left(\frac{c_2^2\|Ap\|^2}{\lambda_{N+1}}\|Aq\|^2 + \frac{c_4^2L_\epsilon\|Ap\|^2}{\lambda_{N+1}^2}\|Aq\|^2 + \frac{c_2^2\|Aq\|^2}{\lambda_{N+1}^2}\|Aq\|^2\right. \\ &\quad \left. + \lambda_N c_2^2\|p\|\|Ap\|\|p-u_N\|^2 + c_4^2L_\epsilon\|Au_N\|^2\|p-u_N\|^2\right) \\ &\le \frac{C}{\nu}\left(\|p-u_N\|^2 + \lambda_{N+1}^{-2}\|Aq\|^2\right),\end{aligned}$$

where $C = C(c_2, c_3, c_4, \rho_2, \rho_2^*)$. We then apply Gronwall's inequality to obtain

$$\|\Delta(t)\|^2 \le \|\Delta(0)\|^2 e^{-\nu\lambda_{N+1}t} + \frac{C}{\nu^2\lambda_{N+1}}\left(\max_{[0,T]}\|p-u_N\|^2 + \lambda_{N+1}^{-2}\max_{[0,T]}\|Aq\|^2\right)$$

for $t \in [0,T]$. Finally, by initializing $\tilde{\phi}_1(0;u_N) = Q_N u_0 = q(0)$, we have estimate (4.8). ∎

COROLLARY 4.7. *Let $f(t) \in L^\infty((0,\infty);H)$, $u_0 \in D(A)$, and $T > 0$. Let $u(t) = p(t) + q(t)$ be a solution of (1.2), and $u_N(t)$ and $\tilde{\phi}_1(t;u_N(t))$ be a solution of system (3.16)–(3.19) such that estimates (3.1) and (4.1) hold. Then, for $t \in [0,T]$, we have*

$$\|q(t) - \tilde{\phi}_1(t;u_N(t))\| = O(\epsilon^3). \tag{4.9}$$

Proof. Since $\|Aq\| = O(1)$ and $\|p(t) - u_N(t)\| = O(\epsilon^3)$ from Theorem 4.1, the $\|Aq\|$ term dominates the right-hand side of (4.8). Thus $\|q - \tilde{\phi}_1(t;u_N(t))\| = O(\epsilon^3)$ as $\epsilon \to 0$. ∎

In this case, the dynamic postprocessing method produces a more accurate high mode approximation. In particular, the dynamic postprocessing method produces a high mode approximation of the same order as the low mode approximation.

5. Numerical experiments. In this section we present some numerical experiments to support the above accuracy analysis and compare the efficiency of the two leading order methods. Opting for a one-dimensional calculation, we integrated Burgers equation with homogeneous Dirichlet boundary conditions on the interval $[0,\pi]$. That is, we used the equation

$$\begin{aligned}\frac{\partial u}{\partial t} - \nu\frac{\partial^2 u}{\partial x^2} + u\frac{\partial u}{\partial x} &= f(x,t), \\ u(0,t) = u(\pi,t) &= 0, \\ u(x,0) &= u_0(x).\end{aligned}$$

Using notation similar to the NSE, the above equation is equivalent to the functional differential equation

$$\frac{du}{dt} + \nu Au + B(u,u) = f,$$

where, in this case, $A = -\frac{\partial^2}{\partial x^2}$ with domain $D(A) = H^2(0,\pi) \cap H_0^1(0,\pi)$. The eigenfunctions of A are $\omega_k = \sqrt{2/\pi}\sin(kx)$, with corresponding eigenvalues $\lambda_k = k^2$, for $k = 1, 2, \ldots$. The bilinear term $B(u,u)$ is defined by $B(u,v) = \frac{2}{3}uv_x + \frac{1}{3}u_x v$ for every u, $v \in H_0^1(0,\pi)$. In particular, we have $B(u,u) = uu_x$ for every $u \in H_0^1(0,\pi)$.

We chose an exact solution $u_e(x,t)$ and then computed the "highly oscillatory" time-dependent forcing term from the exact solution. In this way we checked errors

without computing a large Galerkin approximation as an "exact" solution. We chose $u_e(x,t)$ as follows,

$$u_e(x,t) = \sum_{k=1}^{\infty} \frac{a_k(t)}{k^3} \sin kx, \qquad a_k(t) = \begin{cases} 1+\gamma \sin k^2 t, & 1 \geq k \geq 100, \\ 1, & k > 100, \end{cases} \tag{5.1}$$

and then calculated the forcing function as $f(x,t) := du_e/dt + \nu A u_e + B(u_e, u_e)$. The exact solution u_e is in $D(A)$ for $t \geq 0$ as assumed above in section 3. Actually, we can compute sharper estimates for $\|Aq\|$ and $\|dq/dt\|$ using expression (5.1) to obtain $\|Aq\| \leq \sqrt{2}/\lambda_N^{1/4} = O(\epsilon^{1/2})$, and similarly $\|dq/dt\| \leq \sqrt{\gamma}/\lambda_N^{1/4} = \gamma O(\epsilon^{1/2})$. Note that the $\|Aq\|$ and $\|dq/dt\|$ terms are of the same order, depending on the magnitude of γ. We then obtain $\|q\| = O(\epsilon^{5/2})$ and $\|A^{1/2}q\| = O(\epsilon^{3/2})$. These estimates are of slightly higher order in ϵ than those assumed in the theoretical section. However, using these estimates for the truncation analysis and keeping terms up to order ϵ, we obtain the same leading order approximate system as in system (3.2)–(3.3).

The experiments in this section were run on a Sun Ultra 5. The time integrator used was the VODE code [1] with computed diagonal Jacobians (VODE option MF=23). This code is a reliable and efficient tool for the time integration of systems of ODEs, especially for stiff problems like those arising from the spatial discretizations of dissipative PDEs. VODE consists of a backward differentiation formula (BDF) implemented with variable time step and variable order. Specifically, in the algorithm the time levels are unevenly spaced and the step sizes are produced by the code as the integration proceeds. Also, formulas of different orders (up to order six) are used, the order of the formula being selected by the code at every time step. For problems similar to those in this section, the superior efficiency of codes like VODE with respect to other frequently used time integrators was experimentally checked in [9].

For each value of N we sought the Galerkin approximation, the standard postprocessed Galerkin approximation (standard PP) from system (2.18)–(2.20), and the dynamic postprocessed Galerkin approximation (dynamic PP) from system (3.16)–(3.19). Each experiment was carried out with decreasing values of the time-integration tolerance (an input parameter to VODE) until additional reduction did not improve the accuracy of the solution any further. This means that the time discretization error is negligible in comparison with the spatial error that we are interested in examining.

Figure 5.1 shows the total errors for each of the approximations at time $t = 2.0$ units, with $\gamma = 0.1$ and $\nu = 1$. The initial condition was the projection of $u_e(x,0)$. The solid line represents errors from the Galerkin method, (2.18) and (2.20); dashed lines represent errors from the standard postprocessing method, system (2.18)–(2.20); and dotted lines represent errors from the dynamic postprocessing method, system (3.16)–(3.19). It is clear from Figure 5.1 that the dynamic postprocessing method achieves the best rate of convergence as indicated by the most negative slope. The improvement of the standard postprocessing method over the Galerkin method is only algebraic, and not a significant improvement in the rate of convergence. Thus, the addition of the dq_1/dt term in the high mode equation is beneficial in this case, at least in terms of accuracy. The low mode errors for each of the three methods are essentially the same. The high mode errors are very similar to the total errors shown in Figure 5.1, since the high mode error dominates the total Galerkin and standard postprocessing error. Only the dynamic postprocessing method produces high mode errors (and rate of convergence) that are approximately of the same order as those for the low mode errors. In the case being studied, i.e., highly oscillatory solutions, the dynamic postprocessing method is the most accurate.

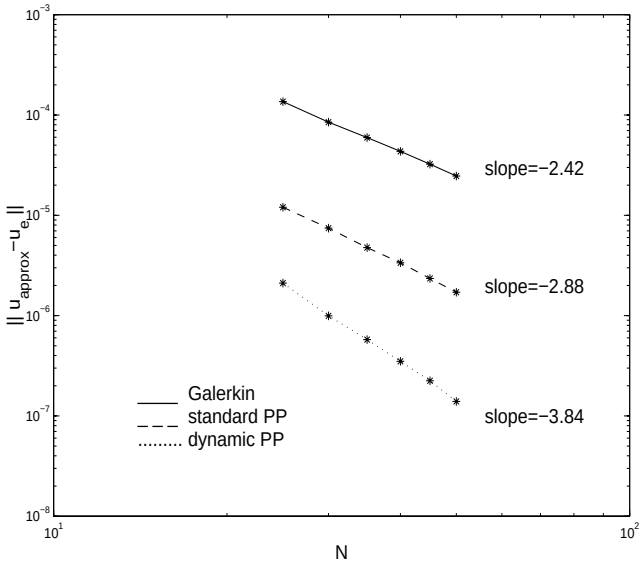

FIG. 5.1. *Total errors* $\|u_{approx} - u_e\|$.

The dynamic postprocessing method requires a numerical integration to obtain the high modes, rather than evaluating the high modes once at some final time, as with the standard postprocessing method [8], [10], [11], [17]. We next looked at the efficiency of the dynamic postprocessing method in the case of highly time-oscillatory solutions to determine whether the error improvement justifies any additional computational cost (CPU).

We again integrated Burgers equation with homogeneous Dirichlet boundary conditions on the interval $[0, \pi]$ as in section 5. However, this time we provided the forcing function defined by

$$f(x,t) = \sum_{k=1}^{\infty} \left(\frac{\dot{a}_k(t)}{k^3} + \frac{a_k(t)}{k} \right) \sin kx, \quad a_k(t) = \begin{cases} 1 + \gamma \sin k^2 t, & 1 \geq k \geq 100, \\ 1, & k > 100, \end{cases} \tag{5.2}$$

and used a large Galerkin run as an "exact" solution for computing errors. This way we did not accumulate the cost of computing the forcing function from an exact solution at each time step. The experiments in this section were run on an SGI Origin 2000 with $\gamma = 0.1$, $\nu = 1$, and initial condition $u_0(x) = \sum k^{-3} \sin kx$.

We first verified that we obtain accuracy results with this forcing function similar to those in the previous numerical experiment. Figure 5.2 shows the total error estimates at $t = 2.0$ units using the VODE time integrator. The rates of convergence are similar to those in Figure 5.1; the dynamic postprocessing method is again the most accurate method. In Figure 5.3 we plot the total errors $\|u_{approx} - u_e\|$ from Figure 5.2 versus the amount of computing time (in seconds) needed by each method to achieve those errors. We added results from larger mode standard postprocessing runs to better indicate any overlap. A horizontal line across the plot indicates, for a particular error, the CPU time needed by each method. Again, the solid line represents errors from the Galerkin method, dashed lines represent errors from the standard postprocessing method, and dotted lines represent errors from the dynamic

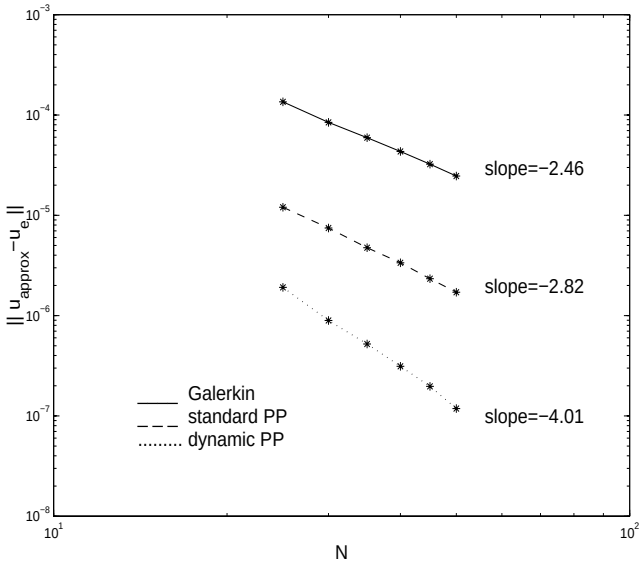

FIG. 5.2. *Total errors* $\|u_{approx} - u_e\|$.

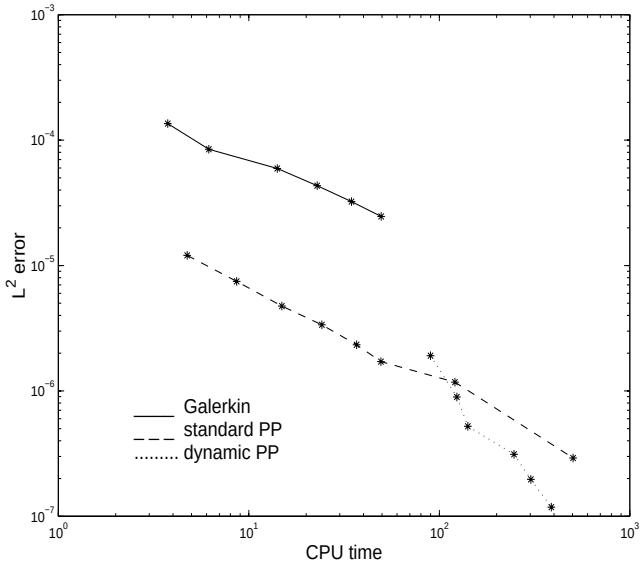

FIG. 5.3. *Error vs. CPU: VODE.*

postprocessing method. For the larger mode runs, the dynamic postprocessing method is slightly more efficient than the standard postprocessing method.

We also sought to take advantage of using a larger time step to integrate the high modes, subcycling the low mode integration within the high mode integration. For this experiment we used a semi-implicit backward Euler scheme. The low modes were integrated using the scheme

$$p^{n+1} - p^n + \Delta t \left(A p^{n+1} + P_N B(p^n, p^n) \right) = \Delta t P_N f^n,$$

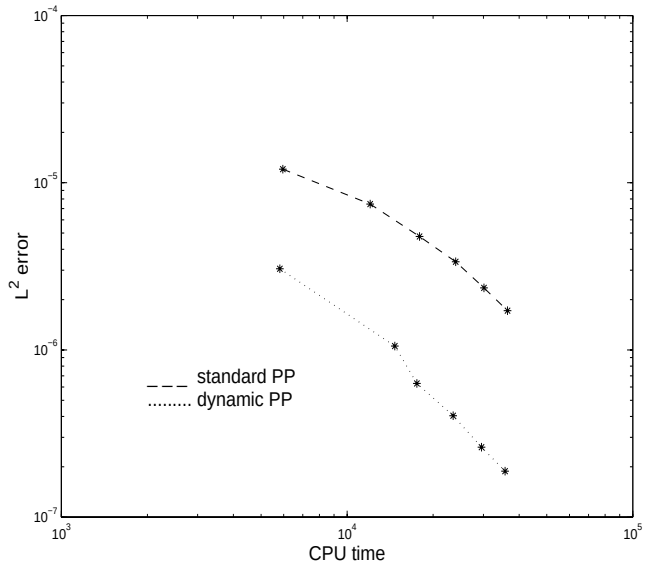

FIG. 5.4. *Error vs. CPU: backward Euler.*

and the high modes were integrated, in the dynamic postprocessing method, using

$$q^{n+1} - q^n + \Delta t \left(A q^{n+1} + Q_N B(p^n, p^n) \right) = \Delta t Q_N f^n.$$

Otherwise, the experimental setup was the same as with the VODE time integrator, i.e., we computed the errors at time $t = 2$ units with $\nu = 1$, $\gamma = 0.1$, and initial condition $u_0(x) = \sum k^{-3} \sin kx$. In each experiment we set Δt small enough so that the low mode error ($\|p - u_N\|$) was equivalent to the low mode errors from the VODE experiments. The effect of subcycling was to slightly increase the error with little improvement in CPU time. This is because the cost of evaluating the high modes at each time step is minimal compared to the cost of evaluating the nonlinear term within each subcycle in order to integrate the low modes. The CPU comparison for the semi-implicit backward Euler experiments without subcycling is given in Figure 5.4. We plot error versus CPU time for the standard and dynamic postprocessing methods. Again the dynamic postprocessing method is the more efficient method for the case of a *highly oscillatory* (in time) solution due to a highly oscillatory (in time) forcing function.

Performing the same experiments with a less time-oscillatory forcing function, we obtained different accuracy and CPU comparisons. In this final set of experiments, we used the forcing function

$$f(x,t) = \sum_{k=1}^{\infty} \left(\frac{\dot{a}_k(t)}{k^3} + \frac{a_k(t)}{k} \right) \sin kx, \quad a_k(t) = \begin{cases} 1 + \gamma \sin t, & 1 \geq k \geq 100, \\ 1, & k > 100. \end{cases} \tag{5.3}$$

In this case we expect the solution to be less oscillatory, and hence the $\|dq/dt\|$ term should be of a smaller order, and the standard postprocessing method should be as accurate as the dynamic postprocessing method. Figure 5.5 shows the total errors for the Galerkin, standard postprocessing, and dynamic postprocessing approximations using the VODE time integrator at time $t = 3.0$ units, again with $\gamma = 0.1$, $\nu = 1.0$, and $u_0(x) = \sum k^{-3} \sin kx$. Note that the standard and dynamic postprocessing methods

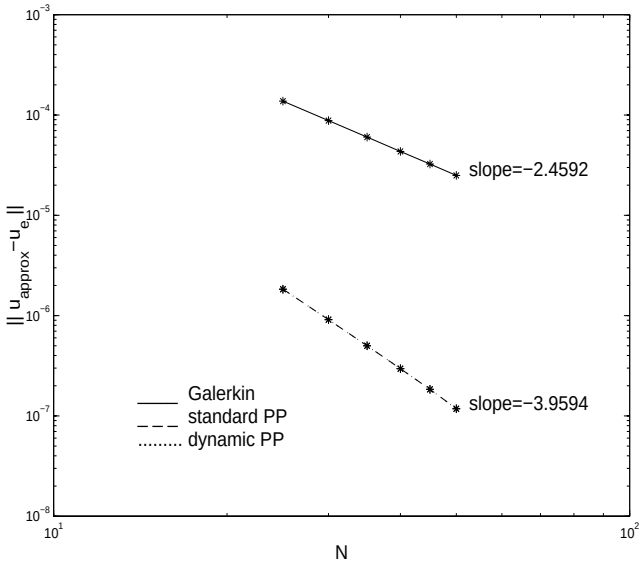

FIG. 5.5. *Total errors: f slowly oscillating in time.*

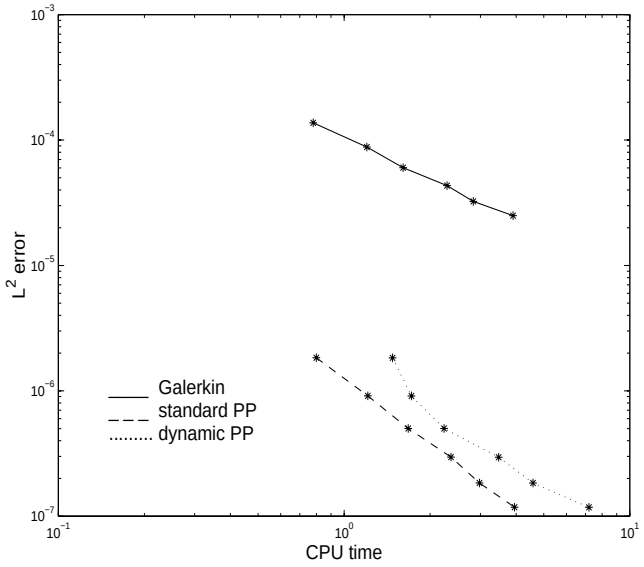

FIG. 5.6. *Error vs. CPU: VODE and f slowly oscillating in time.*

have the same rate of convergence, and there is no improvement with the dynamic postprocessing method.

In Figure 5.6 we plot the total error versus CPU time for the case of the slowly oscillating (in time) forcing function when using the VODE time integrator; the standard postprocessing method proves to be more efficient. Results using the backward Euler scheme are similar, though the differences are less pronounced.

6. Concluding remarks. Through a classical truncation analysis we have shown that postprocessing appears as a natural part of approximate systems and correspond-

ing schemes for numerically integrating the two-dimensional NSE. For autonomous systems, we generated a family of approximate systems (and schemes) of increasing orders of approximation using the asymptotic (in time) estimates from [5] for $\|Aq\|$ and $\|dq/dt\|$. Each system included a postprocessing step that resulted in a high mode approximation of the same order as the low mode approximation. We found that the leading order system is the standard Galerkin method with postprocessing as introduced in [11]. The standard Galerkin method alone uses a less accurate approximation for the high modes than for the low modes. Hence, the accuracy of the high mode approximation, or lack thereof, dominates the error.

By assuming different asymptotic (in time) estimates for $\|Aq\|$ and $\|dq/dt\|$, we obtained the *dynamic postprocessing* method as the leading order method. This was done for the case of a highly oscillatory (in time) solution; the algorithm applies to the case of nonsmooth initial data as well [24]. In the case of a highly oscillatory solution, the dynamic postprocessing method was more accurate and efficient than the standard postprocessing method. For nonautonomous systems with solutions that are not so oscillatory, both methods obtained the same accuracy; however, the standard postprocessing method was more efficient in this case.

The method of using truncation analysis with asymptotic estimates for the low and high modes may easily be extended to general nonlinear parabolic evolution or elliptic equations to obtain postprocessing systems and schemes which approximate the low and high modes to the same order of accuracy.

7. Acknowledgments. We would like to thank Bosco García-Archilla and Julia Novo for the stimulating and inspiring discussions. This work was initiated when E. S. T. and S. W. were visiting the Institute of Geophysical and Planetary Physics (IGPP) at the Los Alamos National Laboratory, while E. S. T. was the Orson Anderson Scholar from 1997–1998.

REFERENCES

[1] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh, *VODE: A variable-coefficient ODE solver*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1038–1051.

[2] P. Constantin and C. Foias, *Navier–Stokes Equations,* University of Chicago Press, Chicago, 1988.

[3] C. Devulder and M. Marion, *A class of numerical algorithms for large time integration: The nonlinear Galerkin methods*, SIAM J. Numer. Anal., 29 (1992), pp. 462–483.

[4] C. Devulder, M. Marion, and E. S. Titi, *On the rate of convergence of the nonlinear Galerkin methods*, Math. Comp., 60 (1993) pp. 495–514.

[5] C. Foias, O. Manley, and R. Temam, *Modelling of the interaction of small and large eddies in two dimensional turbulent flow*, RAIRO Modél. Math. Anal. Numér., 22 (1988), pp. 93–118.

[6] C. Foias, G. Sell, and R. Temam, *Inertial manifolds for nonlinear evolutionary equations*, J. Differential Equations, 73 (1988), pp. 309–353.

[7] C. Foias, G. Sell, and E. S. Titi, *Exponential tracking and approximation of inertial manifolds for dissipative nonlinear equations*, J. Dynam. Differential Equations, 1 (1989), pp. 199–244.

[8] J. de Frutos, B. García-Archilla, and J. Novo, *A postprocessed Galerkin method with Chebyshev and Legendre polynomials*, Numer. Math., 86 (2000), pp. 377–417.

[9] B. García-Archilla, *Some practical experience with the time integration of dissipative equations*, J. Comput. Phys., 122 (1995), pp. 25–29.

[10] B. Garciá-Archilla, J. Novo, and E. S. Titi, *Postprocessing the Galerkin method: A novel approach to approximate inertial manifolds*, SIAM J. Numer. Anal., 35 (1998), pp. 941–972.

[11] B. Garciá-Archilla, J. Novo, and E. S. Titi, *An approximate inertial manifolds approach to postprocessing the Galerkin method for the Navier–Stokes equations*, Math. Comp., 68 (1999), pp. 893–911.

[12] B. García-Archilla and E. S. Titi, *Postprocessing the Galerkin method: The finite-element case*, SIAM J. Numer. Anal., 37 (2000), pp. 470–499.
[13] M. S. Jolly, I. G. Kevrekidis, and E. S. Titi, *Approximate inertial manifolds for the Kuramoto–Sivashinsky equation: Analysis and computations*, Phys. D, 44 (1990), pp. 38–60.
[14] M. S. Jolly, I. G. Kevrekidis, and E. S. Titi, *Preserving dissipation in approximate inertial forms for the Kuramoto–Sivashinsky equation*, J. Dynam. Differential Equations, 3 (1991), pp. 179–197.
[15] D. A. Jones and E. S. Titi, *A Remark on quasi-stationary approximate inertial manifolds for the D Navier–Stokes equations*, SIAM J. Math. Anal., 25 (1994), pp. 894–914.
[16] M. Marion and R. Temam, *Nonlinear Galerkin methods*, SIAM J. Numer. Anal., 26 (1989), pp. 1139–1157.
[17] J. Novo, E. S. Titi, and S. Wynne, *Efficient methods using high accuracy approximate inertial manifolds*, Numer. Math., 87 (2001), pp. 555–595.
[18] R. Temam, *Infinite-Dimensional Dynamical Systems in Mechanics and Physics*, Appl. Math. Sci. 68, Springer-Verlag, New York, 1988.
[19] R. Temam, *Attractors for the Navier-Stokes equations, Localization and approximation*, J. Fac. Sci. Univ. Tokyo, Sect. IA Math., 36 (1989), pp. 629–647.
[20] R. Temam, *Navier–Stokes Equations and Nonlinear Functional Analysis,* 2nd ed. CBMS-NSF Regional Conf. Ser. in Appl. Math. 66, SIAM, Philadelphia, 1995.
[21] E. S. Titi, *On approximate inertial manifolds to the Navier-Stokes equations*, J. Math. Anal. Appl., 149 (1990), pp. 540–557.
[22] E. S. Titi, *On a criterion for locating stable stationary solutions to the Navier-Stokes equations*, Nonlinear Anal., 9 (1987), pp. 1085–1102.
[23] S. Wynne, *Efficient Numerical Algorithms for Simulating Evolution Equations*, Ph.D. Thesis, Department of Mathematics, University of California, Irvine, 1999.
[24] He Yinnian and R. M. M. Mattheij, *Stability and convergence for the reform postprocessing Galerkin method*, Nonlinear Anal. Real World Appl., 4 (2000), pp. 517–533.