

Sparse dynamics for partial differential equations

Hayden Schaeffer^a, Russel Caflisch^a, Cory D. Hauck^b, and Stanley Osher^{a,1}

^aDepartment of Mathematics, University of California, Los Angeles, CA 90095-1555; and ^bComputer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831

Contributed by Stanley Osher, February 13, 2013 (sent for review November 30, 2012)

We investigate the approximate dynamics of several differential equations when the solutions are restricted to a sparse subset of a given basis. The restriction is enforced at every time step by simply applying soft thresholding to the coefficients of the basis approximation. By reducing or compressing the information needed to represent the solution at every step, only the essential dynamics are represented. In many cases, there are natural bases derived from the differential equations, which promote sparsity. We find that our method successfully reduces the dynamics of convection equations, diffusion equations, weak shocks, and vorticity equations with high-frequency source terms.

multiphysics | multiscale | optimization

In this work, we investigate the approximate dynamics of various partial differential equations (PDEs) whose solutions exhibit behaviors on multiple spatial scales. These scales may interact with one another in a nonlinear manner as they evolve. Many physical equations contain multiscale (as well as multiphysics) phenomena, such as the homogenization problems from material science and chemistry and multiscale systems in biology, computational electrodynamics, fluid dynamics, and atmospheric and oceanic sciences. In some cases, the physical laws used in the model can range from molecular dynamics on the fine scale to classical mechanics on the large scale. In other cases, the equations themselves contain high-wavenumber oscillations that separate into discrete scales, on top of the smooth underlying behavior of the system.

The main source of difficulty in multiscale computation is that accurate simulation of the system requires all phenomena to be fully resolved. The smaller spatial scales influence the global solutions; thus, they cannot be ignored in the numerical computation. In some cases, it is possible to derive an analytical equation for the effect of small scales on the solution (1, 2). In practice, however, it may not be possible to derive a simple expression that represents the fine-scale behavior. Many problem-dependent methods have been proposed in the literature, whereas a few provide a general methodology for modeling the macroscopic and microscopic processes that yield multiscale models. For example, some general methods include the heterogeneous multiscale method (3), the equation-free method (4), multiscale methods for elliptical problems (5), multiscale finite element methods (6, 7), and the sparse transform method (8). An overview of general multiscale approaches is provided in ref. 9. A key difference between our method and other methods (3–5) is that we are directly resolving all the significant scales in the solution. By contrast, the other methods (3–5) directly resolve only the coarse scales of the solution, and they separately “reconstruct” the fine-scale solution (as well as its effect on the coarse scales).

From the perspective of mathematics, multiscale methods began with representation of a function using a global basis, such as a Taylor series or Fourier series. More sophisticated bases have appeared, for example, any one of the many wavelet bases used in imaging and computational physics. The key to the basis approximation is that each basis element represents behavior on a specific scale; therefore, the coefficients of the basis provide complete information about the underlying function. This is also the principle behind multiresolution and decomposition methods.

As the methods of multiscale and multiphysics modeling developed over the past few decades, so did corresponding methods

in imaging and information science. One of the fundamental ideas in imaging is that of sparsity. Sparse data representation is used throughout imaging from compression to reconstruction. Early advances in sparse techniques (e.g., refs. 10, 11) presented a convex minimization approach to the computationally challenging sparse basis pursuit problem. Many models that use sparsity to produce both more efficient numerical methods and better quality solutions have been proposed. Some applications of sparsity to imaging include compressive sensing, reconstruction of images from sparse data (12, 13), and recovery of images using sparse regularization (14, 15). The underlying principle of sparsity is that images can be approximately represented by a small number of terms with respect to some basis. Inducing sparsity, creating effective bases, and developing efficient computational algorithms have been intensely active fields in information science.

For imaging and information science, one of the reasons for the success of sparse methods is their ability to resolve drastically different phenomena with a small amount of information. This is also a principal goal of multiscale modeling. In this work, we transfer sparsity methodology, which was developed for information science, to multiscale nonlinear differential equations and show that it can be an effective tool for accurately computing solutions using less information.

In particular, we propose solving PDEs with the constraint that the approximate solution resides on a sparse subspace of a basis. In this way, the complexity of the method will depend on the number of basis terms retained and will be (nearly) independent of the grid size. In the following sections, we will discuss the general problem and the optimization method used to induce sparsity in the solution. The general numerical method will also be explained, as well as results of numerical experiments. The method is tested on an advection equation with oscillatory velocity, a parabolic equation with oscillatory coefficients, a conservation law with oscillatory diffusion, and the vorticity equations with high-frequency source terms. We conclude with a discussion on the proposed work and implications.

Sparsity

Problem Statement. In general, the problem can be stated as follows. Assuming $\mathbf{x} \in \mathbb{R}^n$ and $t > 0$, let $u(\mathbf{x}, t): \Omega \rightarrow \mathbb{R}$ be the approximate solution of

$$\partial_t u = F(u) \quad [1]$$

subject to the constraint: $u(\mathbf{x}, t) = \sum_j \hat{u}_j(t) \phi_j(\mathbf{x})$, where the number of nonzero \hat{u}_j at a given time step is sparse. The operator $F(\cdot)$ can be nonlinear, nonlocal, and dependent on the derivative of u . The basis terms ϕ_j are assumed to exist on separate scales, which is true of most bases (e.g., Legendre polynomials, Fourier wavelets). In this way, the basis terms represent different global behaviors. The method involves two steps: Evolve the PDE forward in time, and project the updated solution onto

Author contributions: H.S., R.C., C.D.H., and S.O. performed research; and H.S. wrote the paper.

The authors declare no conflict of interest.

¹To whom correspondence should be addressed. E-mail: sjo@math.ucla.edu.

a sparse subset. We first address sparsity induction through soft thresholding.

Sparsity via Optimization: Soft Thresholding. At a given time step, the problem of projecting the updated solution onto a sparse subset is equivalent to fitting a solution \hat{u}^n with corresponding coefficients $\{\hat{u}_j^n\}$ at the n th time step to a solution u whose corresponding coefficients $\{u_j\}$ are sparse. This can be written as a constrained least squares fit as follows:

$$\min_u \|u - \hat{u}^n\|_{L^2}^2 \text{ s.t. } u = \sum_j \hat{u}_j \phi_j \text{ \& } \{\hat{u}_j\} \text{ is sparse.} \quad [2]$$

Expanding with respect to the basis and assuming that the basis is orthonormal, this constrained optimization problem is related to the following unconstrained problem:

$$(L0) \quad \hat{u} = \underset{\hat{u}}{\operatorname{argmin}} \lambda \|\hat{u}\|_0 + \frac{1}{2} \|\hat{u} - \hat{u}^n\|_{L^2}^2, \quad [3]$$

where \hat{u} is the vector of coefficients. The “norm” $\|\cdot\|_0$ is the number of nonzero coefficients in Eq. 2. This makes Eq. 3 both nonconvex and difficult to solve. By replacing the L^0 norm with the L^1 norm, we get the following convex relaxation of Eq. 3:

$$(L1) \quad \hat{u} = \underset{\hat{u}}{\operatorname{argmin}} \lambda \|\hat{u}\|_1 + \frac{1}{2} \|\hat{u} - \hat{u}^n\|_{L^2}^2. \quad [4]$$

Note that because $\hat{u} \in \mathbb{C}$, the L^1 norm is $\|\hat{u}\|_{L^1} = \sum_j |\hat{u}_j|$, where $|\hat{u}_j| = \sqrt{\hat{u}_j \bar{\hat{u}}_j}$. The solution of Eq. 4 is given by the following equation:

$$\hat{u}_j = \mathcal{S}_\lambda(\hat{u}_j^n) = \max(|\hat{u}_j^n| - \lambda, 0) \frac{\hat{u}_j^n}{|\hat{u}_j^n|}. \quad [5]$$

In general, this can be computed for a nonorthonormal basis, which is equivalent to a basis pursuit problem with the L^1 norm as a sparse regularizer. In that case, the solution must be found by an iterative method rather than the simple shrinkage provided here as an example. The resulting minimizer \hat{u} is a proximal solution that lies on a sparse subset of the original coefficient domain (16). This can be used to show that the solutions form a contraction map in the L^2 norm. Alternatively, we can simply apply the soft thresholding on the coefficients directly in order to induce sparsity in this way.

Numerical Method

Assuming $u(x, t)$ is periodic in the domain $\Omega \subset \mathbb{R}^n$, one natural basis is the Fourier basis, whose coefficients are the Fourier transform of $u(x, t)$. This is appropriate for the examples shown here. For the rest of this work, we will use the Fourier basis; however, the overall methodology presented here is independent of the corresponding basis.

Taking the Fourier transform of the PDE from Eq. 1 and discretizing the resulting differential equation in time yield a multistep scheme. Because our method does not depend on the choice of numerical updating, we can assume that the scheme takes the following form:

$$\hat{v} = Q(\hat{u}^{n-q}, \dots, \hat{u}^n). \quad [6]$$

The updated solution \hat{v} may be sparse depending on both the PDE and the update operator Q ; however, in general, it will have nontrivial values everywhere depending on the approximation and implementation. The auxiliary variable v is projected onto a sparse subspace by the shrinkage operator:

$$\hat{u}^{n+1} = \mathcal{S}_\lambda(\hat{v}). \quad [7]$$

Altogether, the update in the spatial domain is simply:

$$u^{n+1} = \sum_j \mathcal{S}_\lambda(Q(\hat{u}^{n-q}, \dots, \hat{u}^n)) \phi_j. \quad [8]$$

Unlike traditional projections, this is nonlinear and adaptive. Rather than sorting the coefficients and retaining a fixed number of large-amplitude terms or keeping terms whose wavenumbers are below some cutoff, the shrinkage allows the number and choice of nonzero coefficients to evolve over time. Also, this is not the same as hard-thresholding the solution at every step (i.e. keeping only the terms larger than a fixed value) because the coefficients that remain have decreased their magnitude by λ . Most importantly, the projection does not favor any particular part of the spectrum; instead, the amplitude of the coefficient determines if it remains. In terms of the Fourier basis, the importance is placed on the amplitude rather than the wavenumber.

For general convergence, as long as $\lambda = Cdt^p$ for p larger than the accuracy of the scheme used to update the variable in time, the shrinkage operation does not change the spatial accuracy of the original method and the method will still converge as $dt \rightarrow 0$. For example, discretize using the forward Euler method, and then expand the shrinkage operator to get

$$\hat{u}^{n+1} = \hat{u}^n + dt \widehat{F(u^n)} + \mathcal{O}(\lambda). \quad [9]$$

Therefore, to have convergence as $dt \rightarrow 0$, the shrinkage parameter must be $\lambda = Cdt^{1+\alpha}$. For linear PDEs, convergence can easily be shown. In general, the shrinkage operator is nonexpansive in each coefficient; hence, it is nonexpansive in coefficient norms. This may help with obtaining a general convergence result.

Numerical Results

In this section, we discuss the application of the proposed sparse method to several equations with different numerical schemes.

Convection. The convection equation we consider is the following:

$$\partial_t u = a(x) \partial_x u, \quad [10]$$

where the coefficient $a(x)$ is highly oscillatory.

Let k be the wavenumber, and use spectral Leap Frog as the updating for Eq. 10 to obtain

$$\hat{v}^{n+1} = \hat{u}^{n-1} + 2 dt \hat{a} * (i k \hat{u}^n), \quad [11]$$

in which $*$ is the convolution operator over frequency.

The time step is $\mathcal{O}(dx)$ to preserve the stability condition in Eq. 10. In Fig. 1, the coefficient is chosen as follows:

$$a(x) = \frac{1}{4} \exp\left(\frac{0.6 + 0.2 \cos(x)}{1 + 0.7 \sin(64x)}\right).$$

This choice of $a(x)$ exhibits both fast and slow modes, but the particular structure is not directly needed.

Fig. 1 illustrates the performance of the sparse solution method on this example by comparison of the sparse solution, the true solution produced using a standard fully resolved method, and a “low-frequency solution” produced by solving Eq. 11 for wavenumbers k in the interval $|k| \leq K/2$ in which K is the number of modes in sparse solution. In Fig. 1 (Upper Left and Upper Right), the sparse solution produced by our method and the true solution at $t = 1$ are plotted in the spatial domain at a given time. In Fig. 1 (Lower Left), the true and low-frequency solutions are displayed. The low-frequency solution is unable to capture the correct speed. In Fig. 1 (Lower Right), the sparse and true spectra are plotted. The sparse spectrum captures the largest amplitude

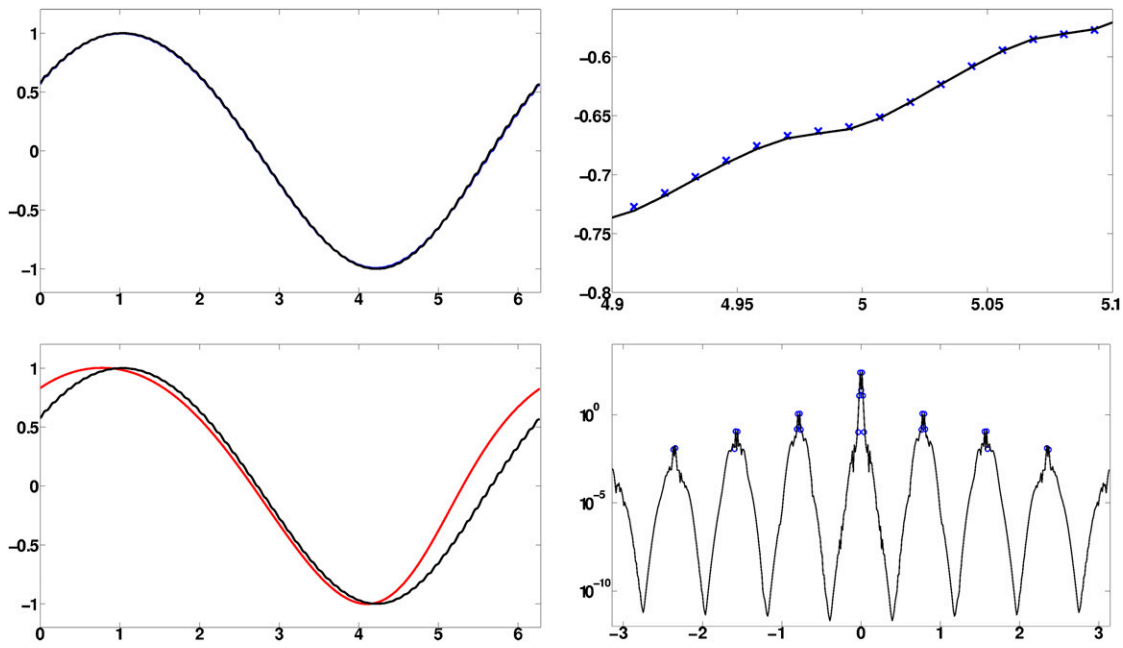


Fig. 1. Convection with highly oscillatory coefficients. The solution is shown in x space, in the zoomed-in version in x space, and in the k space. The solutions are shown on a 512 grid, with $dt = 2e - 3$, $dx = 1.23e - 2$, and $\lambda = 5e - 03$. (Upper Left) True (black) and sparse (blue) solutions in x space. (Upper Right) True (black) and sparse (blue "x") solutions in x space, zoomed in. (Lower Left) True (blue) and low-frequency (red) solutions in x space. (Lower Right) True (black) and sparse (blue) solutions in k space.

coefficients throughout the domain. In fact, of the 512 coefficients used in the true solution, only 27 are retained in the sparse solution (about 5.3%).

Parabolic. The parabolic equation we consider is the following:

$$\partial_t u = \partial_x (a(x) \partial_x u), \quad [12]$$

where the diffusion coefficient $a(x)$ is highly oscillatory. The coefficient is assumed to be bounded [i.e., $A_M \geq a(x) \geq A_m > 0$]. This is also related to the elliptical case $\partial_x(a(x) \partial_x u) = f$, because an elliptical equation can be solved by taking a parabolic scheme to steady state. Alternatively, the corresponding parabolic scheme can be iterated forward for a small number of time steps in order to find a partial solution to the elliptical problem. Then, by using the partial solution, the locations of the nonzero coefficients can be extracted and the elliptical problem can be solved by a Galerkin method on these coefficients (8).

The updated scheme we use for Eq. 12 is forward Euler method:

$$\hat{v}^{n+1} = \hat{u}^n + dt \, i \, k \, \hat{a} * (i \, k \, \hat{u}^n). \quad [13]$$

The time step is $\mathcal{O}(dx^2)$ to preserve the stability condition as well as the highly oscillatory nature of the coefficient $a(x)$ in Eq. 12. In Fig. 2, the coefficient is chosen as follows:

$$a(x) = \frac{1}{10} \exp \left(\frac{0.6 + 0.2 \cos(x)}{1 + 0.7 \sin(256x)} \right).$$

In Fig. 2 (Lower Right), the highly oscillatory diffusion coefficient $a(x)$ is plotted in space. In Fig. 2 (Upper Left and Upper Right), the sparse solution produced by our method and the true solution at $t = 1$ are plotted in the spatial domain at a given time and are nearly indistinguishable. The high-frequency information is near the scale of the grid size, which can be seen in the zoomed-in plot. In Fig. 2 (Lower Left), the true and sparse spectra are displayed. The sparse spectrum captures the largest coefficients throughout the domain

and not just the low wavenumbers. In fact, of the 2,048 coefficients used in the true solution, only 53 are retained in the sparse solution (about 2.6%). In time, the number of nonzero coefficients, as well as the identities of the nonzero coefficients, will change in order to capture various behaviors.

Viscous Burgers. To investigate the sparse dynamics of conservative laws with diffusion, we use the viscous Burgers-type equation:

$$\partial_t u + \frac{1}{2} \partial_x (u^2) = \partial_x (a(x) \partial_x u). \quad [14]$$

The left-hand side (LHS) of Eq. 14 is the standard Burgers advection term, and the right-hand side (RHS) is diffusion related to Eq. 12. The equation exhibits three separate phenomena: (1) smooth large-scale behavior from the diffusion, (2) small-scale oscillations induced from the high frequencies in the coefficient $a(x)$, and (3) nonlinear interactions between frequencies from the advection term. The update scheme in time is the standard total variational diminishing Runge-Kutta 2 method:

$$u_1 = \hat{u}^n + dt \, i \, k \left(\hat{a} * (i \, k \, \hat{u}^n) - \widehat{F(u^n)} \right)$$

$$v^{n+1} = \frac{1}{2} (\hat{u}^n + \hat{u}_1) + \frac{dt}{2} \, i \, k \left(\hat{a} * (i \, k \, \hat{u}_1) - \widehat{F(u_1)} \right),$$

where $F(u) = \frac{1}{2} u^2$. As before, we have the stability condition in which dt is $\mathcal{O}(dx^2)$.

For Fig. 3, the diffusion coefficient is chosen as:

$$a(x) = 0.075 \exp \left(\frac{0.65 + 0.2 \cos(x)}{1 + 0.7 \sin(128x)} \right).$$

The convolutions in the diffusion and nonlinear terms are performed in the spectral domain rather than by other methods, such as the pseudospectral method. The various dynamics can be seen in the spatial and spectral plots (Fig. 3). The true, sparse, and low-frequency solutions are plotted in the space

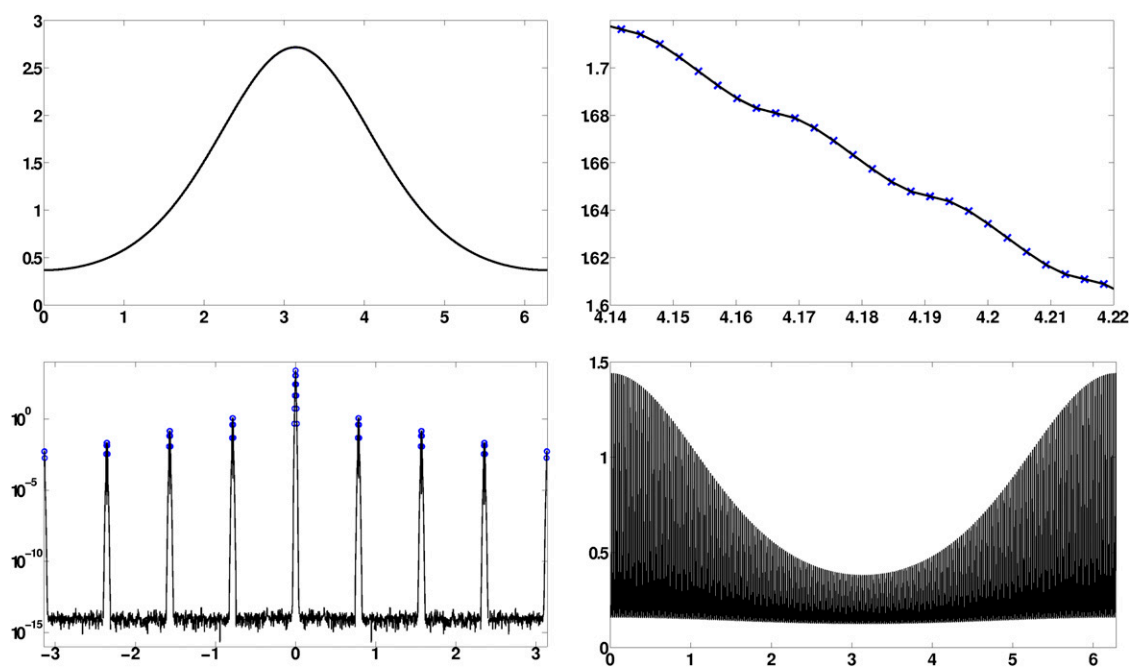


Fig. 2. Parabolic diffusion with highly oscillatory coefficients. The solution is shown in x space, in the zoomed-in version in x space, and in the k space. The solutions are shown on a 2,048 grid, with $dt = 1.5e-8$, $dx = 3.1e-3$, and $\lambda = 2.5e-06$. (Upper Left) True (black) and sparse (blue) solutions in x space. They are nearly indistinguishable. (Upper Right) True (black) and sparse (blue "x") solutions in x space, zoomed in. (Lower Left) True (black) and sparse (blue) solutions in k space. (Lower Right) Diffusion coefficient in x space.

in Fig. 3 (*Upper Left* and *Upper Right*). The low-frequency projection is done by thresholding any coefficients outside of a particular range. Specifically, the number of low wavenumbers retained is the same as in the sparse solution, although their identities are dramatically different. The sparse solution captures the local and global behaviors of the solution more

accurately than the low-frequency projected solution. In Fig. 3 (*Lower Left* and *Lower Right*), the spectrum of the true solution is compared with the sparse and low-frequency spectra, respectively. The local peaks in the spectra are related to the wavenumbers in the diffusion coefficient $a(x)$ and the harmonics induced by the nonlinear advection term. Notice that in this case,

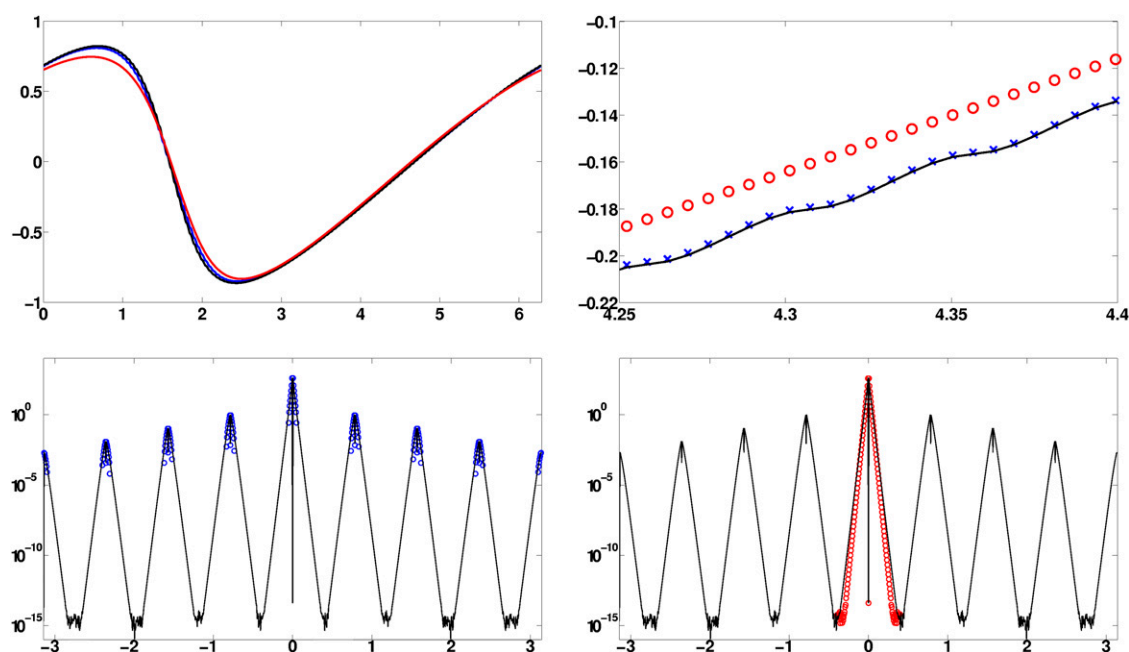


Fig. 3. Viscous Burgers equation with highly oscillatory diffusion. The solution is shown in x space, in the zoomed-in version in x space, and in the k space. The solutions are shown on a 1,024 grid, with $dt = 7.6e - 6$, $dx = 6.1e - 3$, and $\lambda = 6.3e - 5$. (*Upper Left*) True (black), sparse (blue), and low-frequency (red) solutions in x space. (*Upper Right*) True (black), sparse (blue), and low-frequency (red) solutions in x space, zoomed in. (*Lower Left*) True (black) and sparse (blue) solutions in k space. (*Lower Right*) True (black) and low-frequency (red) solutions in k space.

each of the distributions in the spectral domain does not decay as rapidly as in the parabolic case. The sparse solution contains 130 of a total 1,024 possible coefficients, which is about 12.7%.

Vorticity Equations. The vorticity equation we consider is derived from the (2D) incompressible Navier–Stokes equation (17):

$$\partial_t u + (\nabla^\perp \Delta^{-1} u) \cdot \nabla u = \gamma \Delta u + f, \quad [15]$$

where u is the vorticity (not the velocity). Similar to Eq. 14, Eq. 15 exhibits three separate phenomena: (1) smoothness from the diffusion term on the RHS, (2) small-scale oscillations induced from the high frequencies in the source term f , and (3) nonlinear interactions between frequencies from the advection term on the LHS. However, because the operator $\nabla^\perp \Delta^{-1}$ is smoothing (in some sense), the advection term can be viewed as less nonlinear than the one found in Eq. 14. In terms of the numerical method, the operator $\nabla^\perp \Delta^{-1}$ dampens the coefficients by a factor that acts as $|k|^{-1}$.

For the numerical implementation, the diffusion term is discretized using the Crank–Nicolson method, whereas the advection term is lagged. Because the operators are diagonalized in the coefficient basis, the steps can be invertible and lead to a simple updating scheme:

$$\hat{u}^{n+1} = \frac{2dt}{2 + \gamma dt |k|^2} \left(ik^\perp (|k|^{-2} \hat{u}^n) * ik \hat{u}^n + \hat{f} \right) + \frac{2 - \gamma dt |k|^2}{2 + \gamma dt |k|^2} \hat{u}^n.$$

For Fig. 4, the forcing term is chosen to be:

$$f(x, y) = 0.025 \frac{\sin(32x) + \sin(32y)}{1 + 0.25(\cos(64x) + \cos(64y))}.$$

The standard stability condition is used for choosing the time steps in order to ensure capture of all small-scale behaviors. In Fig. 4 (*Upper Left* and *Upper Center*), the true and sparse solutions are plotted in the spatial domain. Notice that the oscillations introduced by the source term interact with the two vortex patches, and thus contribute to the global behavior of the solution. The spectra of the true and sparse solutions are plotted in Fig. 4 (*Lower Left* and *Lower Center*), where the low wavenumbers are located in the middle of the domain. The sparse solution retains about 4.28% of the coefficients. In the sparse spectrum, the coefficients are located throughout the domain, including the highest frequency itself (seen on the boundary of the spectral domain). In Fig. 4 (*Lower Right*), the L^2 and L^∞ errors are shown to decrease as the resolution increases. This sparse solution, as well as the other examples presented here, converges as the spatial discretization goes to zero.

It was observed that as the dimension increases, the sparsity of the solution also increases (because it is proportional to the product of the sparsities in each dimension). Thus, the method scales well with dimension.

It is worth noting that in a related work, wavelet hard thresholding was used to separate coherent and incoherent structures in turbulent flows (18).

In the examples from the previous section, the PDEs contained a mixture of multiscale properties with a diffusion term. The combination of nonlinear and oscillatory terms created a large range of wavenumbers in the solution, whereas the dynamics produced a range of amplitudes. This gives the necessary structure for sparsity with respect to the Fourier basis. In general, the highest

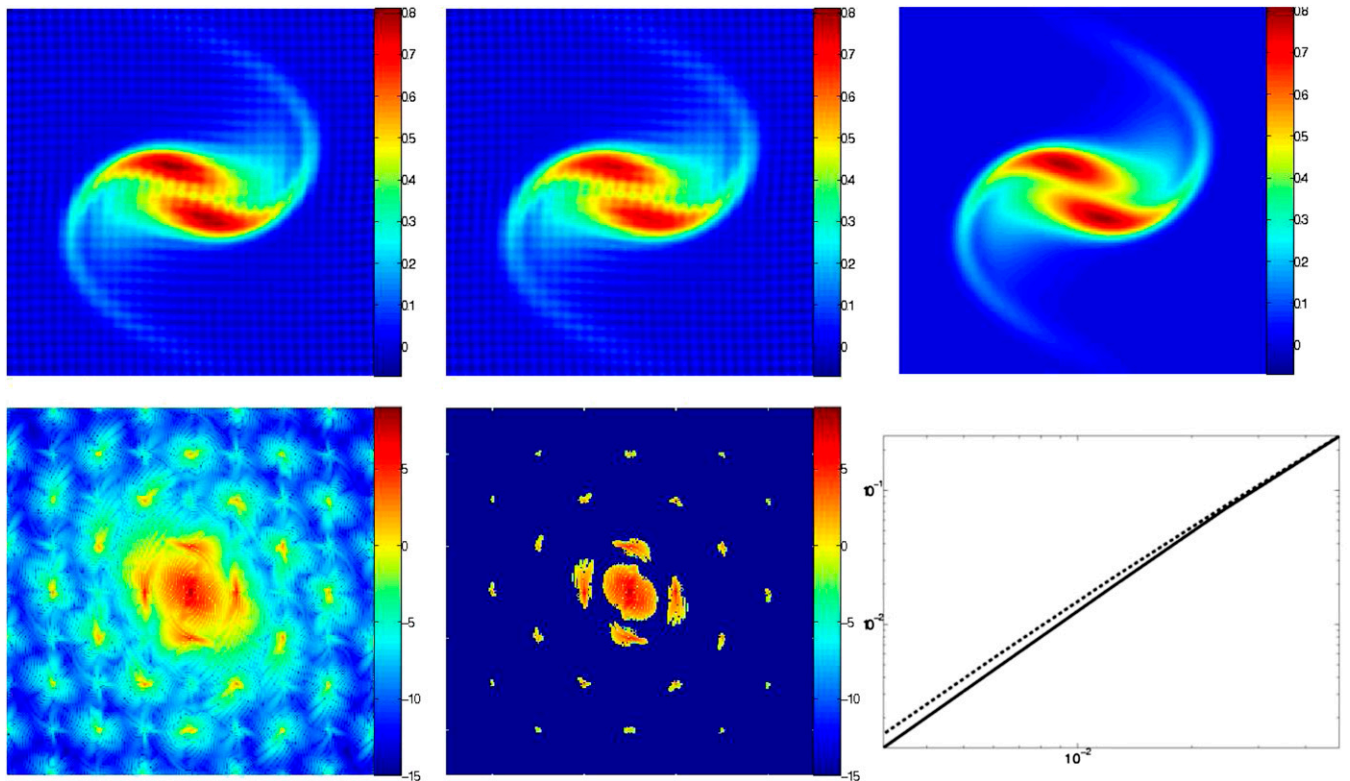


Fig. 4. Vorticity equations with a high-frequency source term. The solution is shown in x space, in the zoomed-in version in x space, and in the k space. The solutions are shown on a 256 by 256 grid, with $dt = 0.025$, $dx = 0.0245$, $\gamma = 0.001$, and $\lambda = 0.0497$. (*Upper Left*) True solution in x space. (*Upper Center*) Sparse solution in x space. (*Upper Right*) Low-frequency solution in x space. (*Lower Left*) True solution in k space. (*Lower Center*) Sparse solution in k space. (*Lower Right*) L^2 (solid) and L^∞ (dashed) errors vs. grid step size.

order derivative will determine the appropriate basis in which the solutions could be sparse.

If the spectrum is more localized (i.e. nonzero regions in the low-frequency regime), the proposed model can better condition the numerical method. Empirically, the shrinkage operator acts as a nonlinear filter on the coefficients. It was observed that for a fixed C and p , where $\lambda = Cdt^p$, the numerical updates presented here with dt larger than theoretically and numerically possible in the original scheme will converge. In the case of the vorticity example, dt can be taken much larger when soft thresholding is applied than in the standard scheme. Also, the nonlinear filter seems to reduce parasitic modes and spurious oscillations found in spectral approximations for linear and nonlinear slightly viscous hyperbolic equations.

One key point is that our method works by fully resolving the solution. Its efficiency is gained by omitting modes that are insignificant. This requires that λ be small enough so that the filter does not annihilate essential features. For example, if the initial data are smaller than λ for a particular unstable mode, our approximate solution will not match the true dynamics. As the grid is refined, the mode will be captured (because λ decreases as Δx decreases).

In terms of complexity, each iteration is dominated by the convolution step. The convolution in the coefficient domain (spectral domain) can be performed explicitly over the $n_s(t)$ -sparse vectors rather than transforming back onto the spatial grid, which is $\mathcal{O}(n_s(t)^2)$ at each step. When $n_s(t)^2 \ll N \log N$, convolving in the spectral domain rather than transforming back and forth between domains decreases the computational cost of the algorithm. Knowing *a priori* the maximum sparsity [i.e., $n_{s,max} = \max_t n_s(t)$], faster routines and transforms could be optimized for specific problems and applications. For example, one can optimize the transform between the spatial and coefficient domains knowing the given sparsity at the current step and the nontrivial coefficients' identities. In the linear cases, as in

Eq. 13, the operation \hat{a}^* can be stored as a large but sparse matrix, reducing the updates to a sparse matrix-sparse vector operation at every iteration. Our goal in this work is to formulate a PDE solver that promotes sparsity. In future work, we will present a study of the computational complexity and speed.

When the dynamics are dominated by a linear term, for example, high viscosity, the identities of the nontrivial coefficients settle over time. This was also observed in the nonlinear cases, but over a longer time period. This would enable creation of a sparse basis for elliptical equations (e.g., with oscillatory coefficients). Also, after this settling phase, the shrinkage operator can be replaced by a projection onto the known identities of the nontrivial coefficients.

In many of the cases here, it is possible to obtain hypersparse solutions (those with 1% or fewer coefficients) at the cost of accuracy.

Conclusion

In this work, we have proposed a method to resolve fully the solutions of multiscale PDEs while only retaining important modes. The reduced dynamics created by the sparse projection properly capture the true phenomena exhibited by the solution. The sparse projection amounts to a shrinkage of the coefficients of the updated solution at every time step. There exist many possibilities for using the sparsity to optimize individual algorithms and to create faster, more efficient computational routines.

ACKNOWLEDGMENTS. The research of H.S. was supported by the Department of Defense through the National Defense Science and Engineering Graduate Fellowship Program. The research of S.O. was supported by the Office of Naval Research (Grant N00014-11-1-719). The research of R.C. was supported by the Department of Energy (DOE; Grant DE-FG02-05ER25710). The research of C.D.H. was sponsored by the Office of Advanced Scientific Computing Research, DOE. C.D.H.'s work was performed at the Oak Ridge National Laboratory, which is managed by the University of Tennessee-Battelle under Contract DE-AC05-00OR22725.

1. Arnold VI (1989) *Mathematical Methods of Classical Mechanics* (Springer, New York), 2nd Ed.
2. Bensoussan A, Lions JL, Papanicolaou G (1978) *Asymptotic Analysis for Periodic Structures* (North-Holland, Amsterdam).
3. E W, Engquist B, Huang Z (2003) Heterogeneous multiscale method: A general methodology for multiscale modeling. *Physical Review B* 67(9):092101-1-092101-4.
4. Kevrekidis IG, et al. (2003) Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Commun Math Sci* 1(4):715-762.
5. Nolen J, Papanicolaou G, Pironneau O (2008) A framework for adaptive multiscale methods for elliptic problems. *Multiscale Model Simul* 7(1):171-196.
6. Efendiev Y, Hou TY (2009) *Multiscale Finite Element Methods: Theory and Applications* (Springer, New York).
7. Hou TY, Wu XH (1997) A multiscale finite element method for elliptic problems in composite materials and porous media. *J Comput Phys* 134(1):169-189.
8. Daubechies I, Runborg O, Zou J (2007) A sparse spectral method for homogenization multiscale problems. *Multiscale Model Simul* 6(3):711-740.
9. E W, Engquist B (2003) Multiscale modeling and computation. *Not Am Math Soc* 50(50):1062-1070.
10. Candès EJ, Romberg J, Tao T (2006) Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* 52(2):489-509.
11. Donoho DL (2006) Compressed sensing. *IEEE Transactions on Information Theory* 52(4):1289-1306.
12. Schaeffer H, Osher S (2013) A low patch-rank interpretation of texture. *SIAM J Imaging Sci* 6(1):226-262.
13. Zhou M, et al. (2009) Non-parametric Bayesian dictionary learning for sparse image representations. *Advances in Neural Information Processing Systems* 22, eds Bengio Y, Schuurmans D, Lafferty J, Williams CKI, Culotta A. Available at (<http://books.nips.cc/nips22.html>).
14. Goldstein T, Osher S (2009) The split Bregman method for L1-regularized problems. *SIAM J Imaging Sci* 2(2):323-343.
15. Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. *Physica D* 60(1):259-268.
16. Cai JF, Osher S, Shen Z (2009) Split Bregman methods and frame based image restoration. *Multiscale Modeling and Simulation* 8(2):337-369.
17. Majda AJ, Bertozzi AL (2001) *Vorticity and Incompressible Flow* (Cambridge Univ Press, Cambridge, UK), Vol 27.
18. Farge M, Schneider K (2001) Coherent vortex simulation (CVS), a semi-deterministic turbulence model using wavelets. *Flow, Turbulence and Combustion* 66(4):393-426.