Density-matrix renormalization for model reduction in nonlinear dynamics

Thorsten Bogner[*]

Condensed Matter Theory Group, Fakultät für Physik, Universität Bielefeld, 33615 Bielefeld, Germany

(Received 27 April 2007; revised manuscript received 25 May 2007; published 26 November 2007)

We present an approach for model reduction of nonlinear dynamical systems based on proper orthogonal decomposition (POD). Our method, derived from the density-matrix renormalization group, provides a significant reduction in computational effort for the calculation of the reduced system, compared to a POD. The efficiency of the algorithm is tested on the one-dimensional Burgers equations and a one-dimensional equation of the Fisher type as nonlinear model systems.

DOI: 10.1103/PhysRevE.76.056707 PACS number(s): 02.60.−x, 02.70.−c, 05.10.−a, 95.75.Pq

I. INTRODUCTION

Nonlinear dynamical systems arise in many fields of physics, e.g., turbulence [1], mathematics, and biology [2]. They often require a significant high number of degrees of freedom (DOF) for simulation with suitable accuracy. For a large class of systems the solutions are regular, nevertheless the influence of the nonlinearity is essential. Here model reduction (MR) can lead to an efficient description, if the dynamics is effectively confined to a lower dimensional attractor in phase space.

The aim of this work is to develop an algorithm that can find a reduced model for a given system, usually derived from a partial differential equation.

One method to obtain such a reduced description is the so-called proper orthogonal decomposition [1,3] (POD). It is obtained by calculating the eigenvectors of the spatial covariance matrix of the field over the phase space, typically by nusing a empirical spatial covariance matrix from a number of realization of the dynamic evolution. The method itself is linear in that the phase space is reduced to a subspace (in which the relevant attractor has to be embedded). Nevertheless it accounts for the nonlinearity and gives the "optimal" linear reduction possible. By definition the POD requires a simulation of the full, unreduced system and a diagonalization of a symmetric matrix of similar size. While the latter can be circumvented by the method of snapshots of Sirovich [1], the simulation in unavoidable.

Within our approach, we try to calculate "approximate" POD modes without simulating the full, unreduced system. This is achieved by following concepts from density matrix renormalization. Effectively, we adapt a system of already low dimensionality, to reproduce the full dynamics optimally. Consequently, all calculations are performed on low-dimensional systems. In exchange, several calculation steps have to be performed, but their number is proportional to the size of the full system of interest. Practically this is one possible way to study large dynamical systems, although within this work we are still restricted to spatially one-dimensional systems. Beside from possible benefits for the efficiency, an interesting question is whether it is possible to reconstruct dynamic behavior of a system from the knowledge of subsystems only.

This paper is organized as follows. First, we introduce the formulation of the equations defining the dynamical system. This includes the use of higher order tensors to describe the nonlinear part of the generator of the time evolution. Further, the discretization of the three model equations, namely the linear diffusion equation, the Burgers equation, and a nonlinear diffusion equation, is presented. Then the type of orthogonal projection, which is used in this work, is introduced and the basic concept of the proper orthogonal decomposition is recapitulated. After a brief outline of the single particle density-matrix renormalization group approach, the method devised in this paper is presented. The numerical results, including a comparison of our method with standard techniques, are given consecutively. This is followed by a short discussion of our approach and the conclusions. The Appendix finally contains an analysis of the optimal reduction for the linear case.

II. THE PROBLEM

A. The dynamical system

We consider discretized versions of nonlinear evolution equations of the form

$$\partial_t \Phi_i = (G(\Phi)\Phi)_i = L_{ij}\Phi_j + Q_{ijk}\Phi_j\Phi_k + K_{ijkl}\Phi_j\Phi_k\Phi_l. \quad (1)$$

Here Φ is the field, $G(\Phi)$ is the nonlinear generator of evolution, and we make use of the sum convention. The contributions L, Q, and K are the linear, the quadratic, and the cubic part, respectively, of the generator of evolution. Higher order terms can also be considered, but the number of nonlinear terms should be finite for our approach. Note that Q and K are third and fourth order tensors and have the corresponding transformation properties. The dynamical system described in Eq. (1) is typically derived from a partial differential equation. The spatial discretization is then done by finite differences or equivalently by linear finite elements. The temporal discretization is done by the simple explicit Euler method, although this choice is not relevant for our method. Here, we restrict ourselves to the spatially one-dimensional case. In the following we exemplify our approach on simple toy problems.

[*] URL: http://www.physik.uni-bielefeld.de/theory/cm/

B. The linear diffusion equation

The diffusion equation describes diffusive transport of a scalar field, e.g., heat transport, in a medium. For homogeneous media it is given by

$$\frac{\partial}{\partial t}\Phi(x,t) = d\Delta\Phi(x,t), \quad x \in [0,1], \tag{2}$$

with the diffusion constant d. Spatial discretization of the interval [0, 1] with N nodes gives for the discrete Laplace operator with second order accuracy in Δx the following $N \times N$ matrix:

$$\Delta_N = \frac{1}{\Delta x^2}\begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & -2 & 1 \\ & & & 1 & -1 \end{pmatrix}. \tag{3}$$

Here homogeneous Neumann conditions are assumed for $x = 0$ and $x = 1$, and the spatial discretization step size is $\Delta x = \frac{1}{N}$. The explicit Euler method gives for the discrete time evolution with time step size h_t the following equation:

$$\tilde{\Phi}(\tilde{x},t_{n+1}) = \tilde{\Phi}(\tilde{x},t_n) + dh_t\Delta_N\tilde{\Phi}(\widetilde{x,t}_n), \tag{4}$$

where $\tilde{\Phi}$ and $\tilde{x}$ are N-dimensional vectors, indicated by $\tilde{\cdot}$. Thus the linear part L in Eq. (1) is given by

$$L = dh_t\Delta_N. \tag{5}$$

The nonlinear contributions in Eq. (1) vanish for the linear diffusion equation.

C. The Burgers equation

As one nonlinear example we consider the Burgers equation [5]. It describes a diffusive as well as a convective transport of a scalar field Φ and is given by

$$\frac{\partial}{\partial t}\Phi = d\Delta\Phi + \nu(\Phi\,\nabla\,)\Phi. \tag{6}$$

This equation is similar to the linear diffusion equation (2) but with an additional term $\nu(\Phi\nabla)\Phi$ describing the convection. This term is quadratic in the field Φ and can be discretized in the form of Q in Eq. (1). For one space dimension, the ∇ operator is simply the spatial derivative. This can be discretized with second order accuracy in Δx as [6]

$$D_{x,N} = \frac{1}{\Delta x}\begin{pmatrix} -1 & 1 & & & \\ -1 & 0 & 1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & 0 & 1 \\ & & & -1 & 1 \end{pmatrix}. \tag{7}$$

The term $(\Phi\nabla)$ is also known as the convective derivative. In one dimensions the discretization is given by multiplying the rows of $D_{x,N}$ with the components of $\tilde{\Phi}$,

$$(\Phi\,\nabla\,)_{N,i,j} = \tilde{\Phi}_i D_{x,N,i,j}; \tag{8}$$

here i,j indicate the component of the matrix and/or vector. Choosing

$$Q_{i,j,k} := \nu D_{x,N,j,k}\delta_{ij} \tag{9}$$

gives a discretization of the convection term, as defined in Eq. (8),

$$\begin{aligned}\sum_{j,k} Q_{i,j,k}\tilde{\Phi}_j\tilde{\Phi}_k &= \nu\sum_{j,k} D_{x,N,j,k}\delta_{ij}\tilde{\Phi}_j\tilde{\Phi}_k \\ &= \nu\sum_{k} \tilde{\Phi}_i D_{x,N,i,k}\tilde{\Phi}_k \;\; \nu(\Phi\,\nabla\,)_N\tilde{\Phi}.\end{aligned} \tag{10}$$

D. Nonlinear diffusion

We consider here a diffusion equation with a nonlinearity that resembles the action-potential part of the one-dimensional FitzHugh-Nagumo (FN) [7,8] equation. In particular the dynamics is defined by

$$\frac{\partial}{\partial t}\Phi = \Delta\Phi - \Phi(1-\Phi)(a-\Phi), \tag{11}$$

where a is a constant. Equation (11) has stable equilibria at $\Phi \equiv 0$ and $\Phi \equiv 1$ and an instable equilibrium at $\Phi \equiv a$. The nonlinear term is cubic in the field. It can be rewritten as $-\Phi(1-\Phi)(a-\Phi) = -\Phi^3 + (1+a)\Phi^2 - a\Phi$. Here the powers of Φ are defined component wise. The cubic part $-\Phi^3$, e.g., is discretized by

$$K_{i,j,k,l} = -\,\delta_{ij}\delta_{ik}\delta_{il} \tag{12}$$

since

$$\sum_{j,k,l}(-\,\delta_{ij}\delta_{ik}\delta_{il})\tilde{\Phi}_j\tilde{\Phi}_k\tilde{\Phi}_l = -\,\tilde{\Phi}_i^3. \tag{13}$$

Similarly, the quadratic part becomes

$$Q_{i,j,k} = (1+a)\,\delta_{ij}\delta_{ik} \tag{14}$$

and the linear part together with the contribution from the diffusive term is

$$L_{i,j} = dh_t\Delta_{N,i,j} - a\,\delta_{ij}. \tag{15}$$

Since we deal only with discretized fields $\tilde{\Phi}$ in the following, we drop the notation $\tilde{\cdot}$ for discrete variables.

E. The reduction

To obtain a reduced model, we will project the phase space to a lower dimensional subspace. Thus the reduction is linear, which simplifies the problem significantly. For nonlinear reduction see, e.g., [9,10]. For a linear projection we only need a basis of the relevant subspace, e.g., given by the column vectors of an $N \times M$ matrix B, where N is the dimensionality of the phase space and M that of the subspace ($N > M$). We will always assume an orthonormal basis in the following since B can always be brought to this form. This basis spans the range of the projection operator P, which is defined by

$$P = BB^{\dagger}. \tag{16}$$

The reduced dynamics is given by

$$\partial_t P\Phi = PLP\Phi + P(QP\Phi P\Phi) + P(KP\Phi P\Phi P\Phi). \tag{17}$$

One can write this equation directly for the reduced phase space which is only M dimensional by using

$$\hat{\Phi} = B^{\dagger}\Phi, \tag{18}$$

$$\hat{L} = B^{\dagger}LB, \tag{19}$$

$$\hat{Q}_{i,j,k} = \sum_{a,b,c} B^{\dagger}_{i,a}Q_{a,b,c}B_{b,j}B_{c,k}, \tag{20}$$

$$\hat{K}_{i,j,k,l} = \sum_{a,b,c,d} B^{\dagger}_{i,a}K_{a,b,c,d}B_{b,j}B_{c,k}B_{d,l}. \tag{21}$$

This gives the reduced equations which are still in the form of Eq. (1) as

$$\partial_t\hat{\Phi} = \hat{L}_{ij}\hat{\Phi}_j + \hat{Q}_{ijk}\hat{\Phi}_j\hat{\Phi}_k + \hat{K}_{ijkl}\hat{\Phi}_j\hat{\Phi}_k\hat{\Phi}_l. \tag{22}$$

While this dynamics is defined on a smaller, M-dimensional phase space it has to be noted that the operators are now typically dense, i.e., most entries in the tensors L, Q, and K are nonzero. To define the reduction we have to make a choice for the relevant degrees of freedom that span the range of P, i.e., the orthonormal basis (ONB) B. Natural criteria for the determination of B would be based on the difference

$$\mathbf{E}(t) = \Phi(t) - BB^{\dagger}\hat{\Phi}(t) = (\mathbb{1} - P)\Phi(t), \tag{23}$$

e.g., the L^2-error

$$E(t) := \|\mathbf{E}(t)\|_{\mathbf{2}}. \tag{24}$$

For linear systems, as, e.g., the linear diffusion equation, it can be shown (see the Appendix) that the projector onto the eigenstates with lowest absolute eigenvalue of the generator of the time evolution [i.e., L in Eq. (1)] leads to a minimal L^2-error for long and short enough times. In the long time limit this approximation becomes even arbitrary accurate as long as the discarded eigenvalues are smaller than zero. This can be extended to nonlinear systems using the proper orthogonal decomposition (POD).

III. PROPER ORTHOGONAL DECOMPOSITION

The proper orthogonal decomposition is a linear projection method which is widely used in model reduction. On this topic an extensive literature exists. Some examples are [1,11–14]. A short explanation of POD together with the method of snapshots is also given in [15]. One of the advantages of this method is the possibility to incorporate information from the nonlinear dynamics to obtain a linear reduction. The basic idea is to generate sample trajectories by simulating the dynamical system of interest. Then the spatial two point correlation matrix C is calculated,

$$C_{i,j} := \langle\Phi(x_i,t)\Phi(x_j,t)\rangle_{\mathbf{T}}. \tag{25}$$

Here $\langle\ \rangle_{\mathbf{T}}$ denotes an average over all sample trajectories. The eigenvectors of this symmetric matrix, which correspond to the highest eigenvalues, span an "optimal" subspace in the sense that the average least square truncation error

$$\epsilon := \langle\|\Phi(x,t) - P\Phi(x,t)\|^2\rangle_{\mathbf{T}} \tag{26}$$

is minimal, see, e.g., [1,16]. The (orthonormal) basis vectors of this subspace constitute the columns of the matrix B which defines the projection operator P, see Eq. (16). The practical application in the following algorithm is simple: Once the operators L, Q, and K are calculated and the initial conditions are given, we can simulate the dynamics of the field Φ, e.g., by Eq. (1). For the reduced system Eq. (22) is used instead. During the simulation the data for the covariance matrix C is accumulated; if necessary several simulation runs are performed using different initial conditions with appropriate weighting. The eigenvectors of C are calculated using standard methods [6]. Then B is constructed from those eigenvectors corresponding to the highest eigenvalues. The discarded eigenvalues can also provide information on the quality of the reduction. The POD can be applied relatively independent from the actual blocking method. It should be noted that such a reduction is optimal for describing the generated dataset, but not necessarily optimal in reproducing the underlying dynamics [17].

IV. BLOCKING METHOD

Blocking methods were considered already earlier, e.g., [18], mainly because in many problems not all spatial regions are of similar interest. Our motivation is different, we aim to decompose a calculation into more feasible subproblems. In the linear case the basis B can be calculated in principle by simply diagonalizing L. Technically this is the same problem as determining the eigenstates of the Laplace operator, which also describes the single quantum mechanical 1D particle in a box. White [4,19,20] used this toy model for introducing the so-called density-matrix renormalization group (DMRG). This approach has been then applied most successfully to quantum many-body problems. In the following we want to carry this analogy further. Instead of approximating the eigenvalues and/or states of a linear operator we use a similar algorithm to obtain an approximate POD of a nonlinear system. To this end a few modifications are necessary, so we first summarize the original DMRG method.

A. Single particle DMRG

In the DMRG toy problem the system is split into blocks of size m. For each block a block-Laplace operator is stored, as well as the links T that define the interaction with the neighboring sites. The assembly of the superblock Laplace operator is pictorially presented in Fig. 1. This superblock operator is a $(2m+2)\times(2m+2)$ matrix and has to be diagonalized. From the eigenstates the so-called target states ϕ^i,

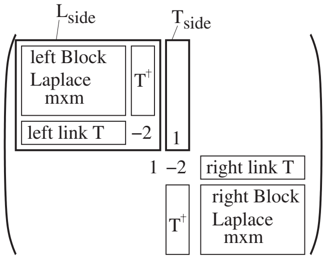

FIG. 1. Assembly of the superblock Laplace operator.

$i=1\ldots(m+1)$ are selected, usually the low lying spectrum. Since we have in this special case a single particle problem, the block truncation matrix R can be calculated simply by applying a Gram-Schmidt orthogonalization to the block part of the target states.

$$R_{i,j} = \text{Gram-Schmidt}(\phi^i_j), \quad i = 1 \ldots (M+1), \quad j \in \text{Block}. \tag{27}$$

For more general DMRG applications R would be calculated by diagonalizing the density matrix of the target state. R is a $(m+1)\times m$ matrix which projects the phase space of one system side to an effective block.

The effective block Laplacian L_{eff} and the effective block link T_{eff} are derived as follows:

$$L_{eff} = R^{\dagger} L_{side} R, \quad T_{eff} = R^{\dagger} T_{side}, \tag{28}$$

where the form of L_{side} and T_{side} are depicted in Fig. 1 for the left side. By this process effective blocks are calculated that describe a higher number of sites, but have numerically still m degrees of freedom.

In DMRG this "growing" of blocks is first used in an initialization step until the superblock describes a sufficiently large system, see Fig. 2. Then an iteration is carried out to increase accuracy. Here only one side is grown, while the other is replaced by an already calculated block so that the effective size of the superblock is constant, see Fig. 3.

B. DMRG POD

Basically three modifications are necessary to obtain a DMRG-POD algorithm.

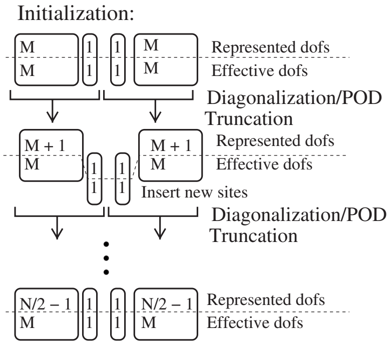

FIG. 2. Graphical illustration of the DMRG initialization (or warmup) scheme.

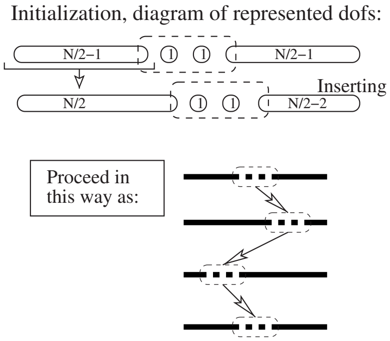

FIG. 3. Graphical illustration of the DMRG iteration (or sweeping) scheme.

First, instead of a diagonalization of the superblock operator, a POD on the superblock system has to be performed. This is composed of, first, a simulation of the superblock system, as defined in Eq. (22). Then the superblock correlation matrix from the generated data has to be diagonalized. This gives an orthonormal set of vectors which are the target vectors in the context of DMRG.

Second, to each subblock there exist not only a linear sub-block operator but also higher order operators, given by third and higher order tensors, see Eq. (17). These have to be updated in a similar way.

Third, for the POD the initial states for the sample trajectories are crucial. The initial states are defined for the full system. They have to be projected onto the superblock system which requires all truncation matrices explicitly.

Concerning the first point, this is no great difference, since the POD (simulation and diagonalization of the correlation matrix) returns also a orthonormal set of "relevant" states (the POD modes) that serve as target states, as described above.

Beside the linear operator [L in Eq. (1)] which is assembled identically as the superblock operator in DMRG, the higher order operators have to be assembled also. This is principally possible, but complex. Here we use a simple trick. For all our models it is sufficient to know the component-wise squaring operator $\Omega_{i,j,k} := \delta_{ij}\delta_{ik}$ (and in some cases the derivative operator which is linear and is also assembled similar to the superblock Laplace operator). Ω is purely diagonal, so no links have to be stored and assembled. The reduction with a truncation matrix R is straightforward,

$$\hat{\Omega}_{i,j,k} = \sum_{a,b,c} R_{i,a}\Omega_{a,b,c}R_{b,j}R_{c,k}. \tag{29}$$

From this the higher order tensors can be calculated directly, e.g., for the Burgers equation

$$\hat{Q}_{\text{Burgers},i,j,k} := \nu \sum_{l} \delta_{ij}\delta_{lj}\hat{D}_{x,N,l,k} = \sum_{l} \hat{\Omega}_{j,i,l}\hat{D}_{x,N,l,k}. \tag{30}$$

For fourth and higher order operators this procedure is a bit memory consuming. For example, for calculating Φ^3 it is more efficient to calculate first $\Phi_{\text{tmp}} := \hat{\Phi}^2 = \Omega\Phi\Phi$ and then $\hat{\Phi}^3 = \hat{\Phi}^2\hat{\Phi} = \Omega\Phi_{\text{tmp}}\Phi$.

The third point is a small disadvantage, since the projection operators all have to be stored, which is not necessary in DMRG if only the energy values are of interest. However, here as well as in DMRG it is possible to expand a superblock state to a state of the original system as well as project down a system state to the superblock if all truncation matrices are stored. The down projection of the N-dimensional state is in particular done by iteratively contracting the $m+1$ outermost sites of, e.g., Φ with the corresponding block truncation matrix R. Apart from the memory requirement this is simply a bookkeeping problem.

It should be noted that only $m+1$ most relevant states from the POD are used as target states. Thus only $m+1$ relevant states of the superblock are optimized although it represents $2m+2$ DOFs. This has to be considered in comparing the results. However, the DMRG POD is nevertheless faster than the full POD, see Sec. VI A.

To summarize: Apart from the POD itself, which is a standard technique, no fundamental changes have to be implemented to obtain a DMRG-POD method from the simple toy model DMRG. The assembly of linear operators has to be performed in any case, only our method requires several operators. The assembly of the Ω operator is even more simple, since all links vanish. Reconstruction of full system states is also possible in DMRG, only it is mandatory for our method for evaluating the correct initial conditions.

V. APPLICATION

For all applications we choose a finite differencing scheme of second order accuracy, homogeneous Neumann conditions at the boundaries, and the explicit Euler method for the calculations. The details are given in Sec. II, above. The boundary conditions as well as the time integration method can be chosen—more or less—arbitrarily. However, higher order finite elements in the spatial discretization lead to additional interactions between single DOFs, i.e., a form of nonlocality and do thereby complicate the problem. For the reduced system size always four DOFs were retained. This is mainly for convenience and easy comparison. The success of the method does not depend strongly on this choice.

As explained above, we measure the quality of a reduction by the L^2-error, see Eq. (24). It has the same units as the fields Φ which are not further specified. The time units are also arbitrary. The error calculations in the following are performed in a separate program which obtains the optimized bases from the various methods as input. Thus the simulation time do not have to coincide with the length of the POD simulations. Further we have chosen a different random seed for statistical initial conditions unless otherwise stated.

A. Linear diffusion

For this problem the dynamics is given by Eq. (2). The only nonzero contribution according to Eq. (1) is $L \equiv \Delta_N$. The eigenstates of L are the sine and/or cosine or Fourier modes whose contributions decay over time with characteristic lifetime inversely proportional to the frequency and/or energy. Standard DMRG can be viewed as an approximate diagonalization method for a linear operator. Therefore it is very effective to find the optimal reduction determined by the eigenstates, see the Appendix, in the linear case. In contrast to the diagonalization, POD as well as our method depends on the initial conditions for the sample trajectories over which the averaging is carried out. Both POD approaches cannot exploit the linearity of the evolution equation. This affects the quality of the results for linear problems compared to diagonalization-based methods. Nevertheless, restriction to a few sample trajectories can also be an advantage, since sometimes the interest lies on a certain region in phase space. However, for the diffusion equation we choose normally distributed initial conditions, i.e., the field $\Phi_0(x_i)$ is normally distributed. This is then also true for the Fourier modes. By this choice effectively the whole phase space will be sampled for a high enough number of realizations. This is also due to the invariance of Eq. (2) under multiplication with a constant factor.

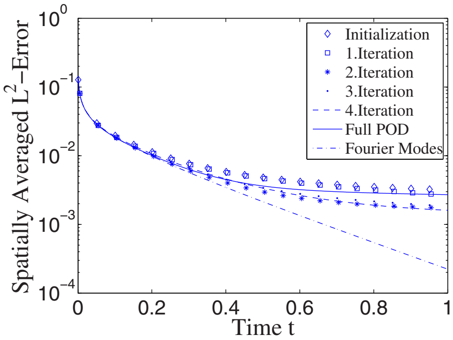

FIG. 4. (Color online) Reduced diffusion equation L^2-error $E(t)$ for the analytical reduction (Fourier modes), the full POD, and DMRG POD after initialization and several iterations, statistical initial conditions, $N=40$, $h_t=0.001$, and $d=0.05$. The error is expressed in units of Φ; for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

For the POD it is important to integrate over long enough times. For short times the state moves in the direction of the highest frequency modes which are decaying most rapidly. Thus POD would give the wrong relevant modes. The POD is in fact not a very appropriate tool to reduce the whole phase space of the diffusion equation. In Fig. 4 the error of the reduced fields $\hat{\Phi}$ is plotted in dependence of time. There the time step was $dt=10^{-3}$ and the diffusion constant $d=0.05$. The spatial resolution was 40 lattice sites within the interval $[0,1]$. In each POD step as well as for the error calculation the ensemble average, compare with Eq. (25), has been averaged over 50 realizations of the initial conditions. From this result we can state several things. First, all POD-based methods show a remaining error in the long time limit. Second, the initialization steps of DMRG POD give already reasonable results. An improvement due to the iteration is present, too. Third, our algorithm is able to compute the optimal reduction with even higher accuracy than the full POD. The last point is only a paradox on the first glance. The inaccuracy of the full POD is in this case influenced from the statistical initial conditions, in order to sample the full phase space. Within our algorithm, much more initial conditions

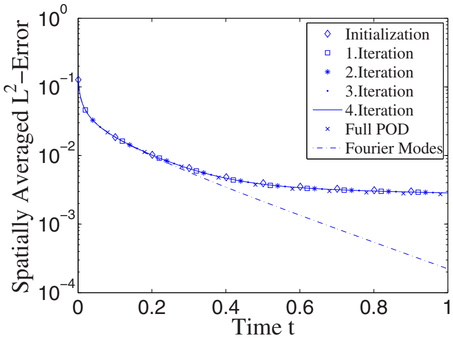

FIG. 5. (Color online) Reduced diffusion equation L^2-error $E(t)$, identical statistical initialization. The error is expressed in units of Φ, for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

are taken into account as the superblock POD is performed repeatedly. This leads to a better statistics. In Fig. 5 we have shown the same results but always using the same initialization for calculating all PODs (but of course not for the error calculation). It is clear that in this case, our method has no advantage over the full POD anymore. On the other hand, the results from our algorithm are not worse than that from the full system POD, which is not clear *a priori*.

B. Burgers equation

The Burgers equation is given by Eq. (6). The discretization used here is already described above. We begin our analysis with the choice of deterministic initial conditions for the calculation of all PODs. In particular, it is of the form

$$\Phi(t=0,x_i) = e^{-50(x_i-1)^2}, \quad x_i = 0 \ldots 1. \tag{31}$$

Figures 6 and 7 show the results for the L^2-error of the evolution. Here we have used two spatial resolutions, i.e., $N=40$ and $N=100$ nodes. The results are very similar. In contrast to the previous calculations the simulation runs for the error calculation are longer than the POD runs. The vertical line indicates the time interval of the POD runs. Here we have to state that the Fourier mode reduction is not optimal,

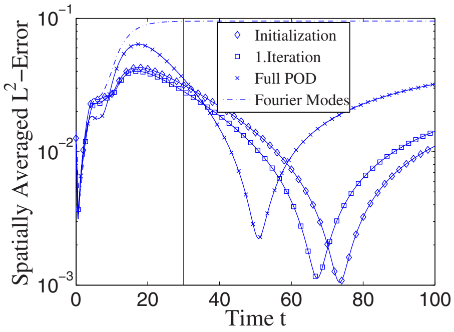

FIG. 6. (Color online) Reduced Burgers equation L^2-error $E(t)$, deterministic initial conditions, $N=40$, $h_t=0.02$, $d=0.01$, and $\nu=0.1$. The error is expressed in units of Φ; for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

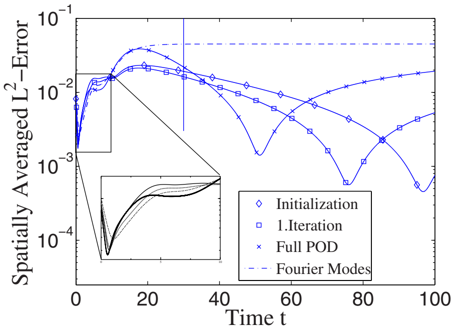

FIG. 7. (Color online) Reduced Burgers equation L^2-error $E(t)$, deterministic initial conditions, $N=100$, $h_t=0.005$, $d=0.01$, and $\nu=0.1$. The inset shows the begin of the error evolution enlarged. The error is expressed in units of Φ; for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

which is not surprising as we consider a nonlinear system and a very particular region of phase space. Further, we see that the error curves show a very pronounced minimum after which the approximation seemingly breaks down. The corresponding time point lies well after the POD time span. These minima correspond to the fact that after the passing of the wave front the profile becomes flat. The approximations do not reproduce the average value accurately, but show a spurious drift. The passing of the reduced (flat) states by the original (flat) state creates the minima in Fig. 6.

It is remarkable that our methods yield better results than the POD within the POD time, even for the initialization. Here it should be recalled that the POD is optimal only for reconstructing the states used in the calculation. As stated above, the reconstruction of the dynamics that created these states is a different thing, as can be directly seen from our results.

We continue our analysis of the Burgers equation by considering statistical initial conditions. In contrast to the calculations for the diffusion equation we have only three randomly sampled parameters in the initial condition. It is given by a peak of various height H, width W, and position X. In particular it is defined by the following equation:

$$\Phi(t=0,x_i) = He^{-50W(x_i-X)^2}. \tag{32}$$

Here, H and W are normally distributed whereas X is uniformly distributed.

The results are shown in Fig. 8. For all methods the error reaches a plateau very quickly. The performance of the full system POD is slightly better than that of the DMRG POD. However, the errors from our approach are of the same order as from the full POD and one magnitude better than that of the Fourier mode-based reduction. Also the iteration brings an improvement which reaches saturation already after the first step.

For deterministic initial conditions the evolution of the error is not monotonic in contrast to the case of statistical initial conditions. This is due to the fact that deterministic initial conditions can be considered more effectively by the POD. The statistical initial conditions were drawn from a

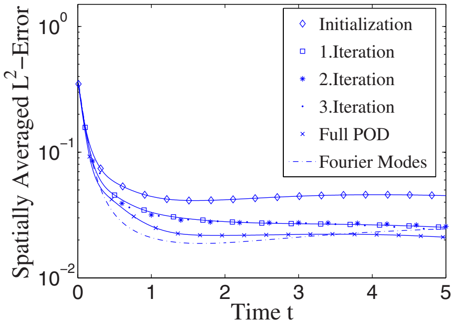

FIG. 8. (Color online) Reduced Burgers equation L^2-error $E(t)$, statistical initial conditions, $N=20$, $h_t=0.01$, $d=0.05$, and $\nu=0.1$. The error is expressed in units of Φ; for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

three-, see Eq. (32), or two-, see Eq. (33), dimensional subspace which is reproduced poorly by a reduction to a four-dimensional space, which has to consider the time evolution also.

C. Nonlinear diffusion

This system is defined by Eq. (11). As initial conditions we have chosen a front with uniformly distributed position X and normally distributed height H,

$$\Phi(t=0,x_i) = \frac{H}{2}\tanh[(x_i - X)10]. \qquad (33)$$

Under these conditions all methods were able to reproduce the dynamics well, see Fig. 9. Surprisingly the full POD method gave poorer results than even the Fourier mode-based reduction. This is to a lower extent also true for the initialization run of the DMRG POD. The iteration led to an improvement although the second iteration gave similar results as the initialization. Further iterations again increase the accuracy, so no general statement can be made. After a fast saturation by applying the iteration procedure we observed repeatedly a decay in the quality of the results, which we attribute to the accumulation of numerical errors.

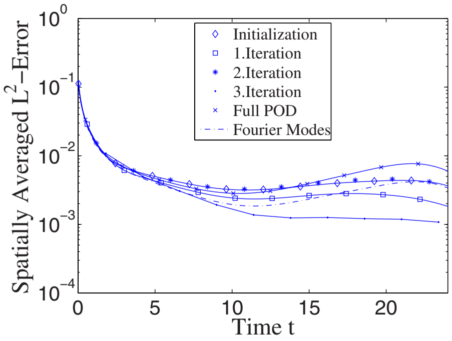

FIG. 9. (Color online) Reduced nonlinear diffusion equation L^2-error $E(t)$, statistical initial conditions, $N=30$, $h_t=0.03$, $d=0.01$, and $a=0.5$. The error is expressed in units of Φ; for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

TABLE I. Naive estimation of the computational load with full system size N, superblock size M, and numbers of iterations N_i. The computational load for the simulation is denoted with $S(\cdot)$

Full system POD	DMRG POD
$\mathcal{O}(N^3)+S(N)$	$NN_iO(M)+S(M)$

VI. DISCUSSION

A. Computational load

For all calculation steps, e.g., diagonalization, Gram-Schmidt orthonormalization, etc., standard algorithms were applied [6,21]. The focus was more on a concise assessment of the new algorithm instead of an optimal solution of the toy problems. For the diagonalization of the covariance matrix, e.g., first a Householder tridiagonalization was performed [21], which is an $\mathcal{O}(N^3)$ algorithm. The calculation of the POD, either for the complete system or for the superblock system, was performed with the same routine. This comprised the simulation as well as the diagonalization.

For a POD the simulation of the system in the time span of interest is additionally necessary. The required computational load for this simulation is implementation dependent and is denoted with $S(\cdot)$. Within our approach the simulation and diagonalization is performed only on the superblock system. Comparing the results from Figs. 6 and 7 suggests that the necessary number of iterations (sweeps) does not depend on the full system size N. If we denote the superblock size with M and the number of iterations with N_i, a naive estimation of the computational load is given in Table I.

For a more quantitative analysis we have measured the time necessary to perform a full POD comprised of simulation and diagonalization. Then we did the same for the initialization of the DMRG POD algorithm including all simulation and diagonalization steps until the superblock system described the full system of dimensionality N, compare Fig. 2, and a first reduced basis had been calculated. We also measured the computing time for one further iteration step in the same way as for the initialization. The computing time is constant for all iteration steps so further data were extrapolated. The underlying equation was the deterministic initialized Burgers equation although the choice for an equation affects the computational load only marginally. As parameters we have chosen $h_t=0.005$, $d=0.01$, and $\nu=0.1$. Figure 10 shows a logarithmic plot of the results. The DMRG POD approach shows a lower amount of computer time for the initialization step. For higher system size this holds also for the iterations. Generally the scaling with N is favorable. Note that here only the DMRG method should be assessed. For this purpose public assessable standard algorithms are sufficient, although much more effective methods could be possible. All calculations were performed on an Intel Dual Core machine, using a single CPU.

B. Stability

Many numerical schemes and the explicit Euler method in particular show instabilities for certain parameter ranges. For

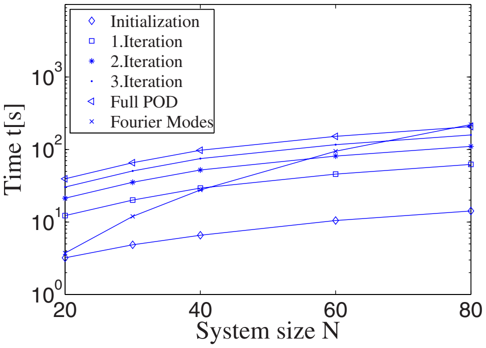

FIG. 10. (Color online) Computing time for various system sizes and approaches, obtained by the reduced Burgers equation, statistical initial conditions, $N=40$, $h_t=0.005$, $d=0.01$, and $\nu=0.1$.

the explicit Euler method the stability condition is

$$|1+\lambda h_t| \leqslant 1, \tag{34}$$

where λ is the largest eigenvalue of the generator of evolution and h_t the size of the time step. For the Laplace operator the highest frequency component is the first to become unstable while increasing h_t. The eigenvalue is $\frac{-4}{\delta x^2}\sin^2\left(\frac{\pi(N-1)}{N}\right) \approx \frac{-4}{(\delta x)^2}$. Consequently, we should have $h_t \leqslant \frac{(\delta x)^2}{2d}$, d being the diffusion constant. We have performed calculation directly at this limit, see Fig. 11, and have seen no signs of instability. Although the nonlinear Burgers equation was considered, the previous point holds, since the symmetric derivative operator has no nonzero real part eigenvalues. By increasing h_t the instability appears for all methods. These calculations were only first tests and further work is required. However, it can be assumed that in the DMRG POD calculation the instability originates only from the newly inserted nodes which correspond to the highest spatial resolution. Combined implicit-explicit methods [22] could be used to solve this problem.

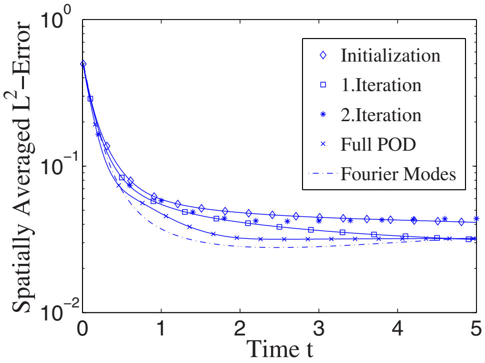

FIG. 11. (Color online) Reduced Burgers equation L^2-error $E(t)$, statistical initial conditions, $N=40$, $h_t=0.00625$, $d=0.05$, and $\nu=0.1$. The error is expressed in units of Φ; for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

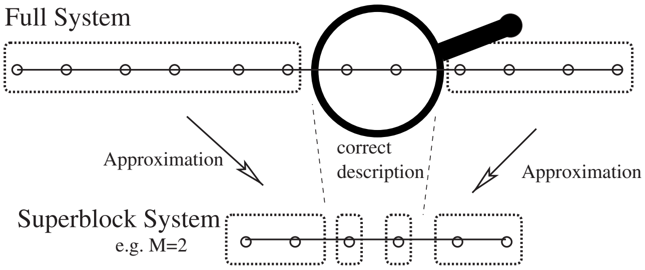

FIG. 12. Pictorial representation of the approximation.

C. Interpretation of the algorithm

The various steps, necessary for a DMRG version of the proper orthogonal decomposition, seem to be complex on the first glance. It is also not clear why this approach should be effective. We now shortly depict the basic idea behind this algorithm.

The DMRG algorithm decomposes the spatial domain but considers an interaction of the domains due to the superblock concept. The single blocks, together with information about interaction with neighboring blocks and the reduced operators, describe spatial regions with a higher number of nodes than the number of DOFs actually retained in the block. Inserting nodes described by the full (yet already discretized) dynamics corresponds to increasing the spatial resolution locally. The surrounding blocks simulate the environment for a small subsystem with correct dynamics. In the DMRG POD algorithm now the area with high resolution is moved through the system. Thereby the parameters of the superblock system, i.e., the block basis and operator matrix elements, are adapted to approximate the full system. For more graphical illustration, see Fig. 12. These arguments are a bit heuristic, but until now no rigorous proof for the algorithm has been given.

VII. CONCLUSION AND OUTLOOK

To summarize, we have given a demonstration of applicability for an algorithm to calculate an approximate POD without ever simulating the full system. Our approach also makes practically no assumption on the equations that define the dynamics. The approach has been tested for linear systems where its performance was even higher than the full system POD results but considerably worse than the optimal reduction. Several nonlinear systems have been considered. For the Burgers equation the results of the full POD and our algorithm were comparable and significantly better than a Fourier mode based reduction.

Further work on this topic will include extensions to higher dimensional systems. A method for two and three dimensions is currently in progress. Further, driven systems and systems with noise shall be implemented. A closer analysis of the quality of the approximations as well as the limitations of the method has to be performed. We have access to the amount of discarded information from the discarded eigenvalues in the truncations as well as in the POD steps. This suggests an adaptive approach for the reduction.

ACKNOWLEDGMENTS

I would like to thank Professor F. Schmid, Professor Ph. Blanchard, and Javier Rodriguez-Laguna for discussion and the German Science Foundation (DFG) for support [Contract No. DE 896/1-(1,2) 2004–2007].

APPENDIX

For completeness, we assess in the following the error of the reduced evolution for the linear case. For the optimal reduction we require a minimal L^2-error for the reduced field with respect to the unreduced evolution. The full time evolution in the N-dimensional phase space is generated by L as

$$\Phi(t) = e^{(t-t_0)L}\Phi(t_0). \tag{A1}$$

The explicit Euler algorithm approximates this by

$$\Phi(t) \approx (\mathbb{1} + h_t L)\Phi(t_0). \tag{A2}$$

We assume that all eigenvalues of L are negative or zero. A positive eigenvalue would lead to an unbounded exponential growth in Eq. (A1) which is unphysical. Considering only linear projections the reduction is defined by the operator P which is the orthogonal projection to the relevant subspace Range(P). P can be constructed from an ONB of this space. Equivalently, it can be defined via the ONB (namely C) of Kern(P) so that $P=\mathbb{1}-CC^\dagger$.

The reduced time evolution becomes

$$\hat{\Phi}(t) = e^{(t-t_0)PLP}P\Phi(t_0) = e^{(t-t_0)\hat{L}}\hat{\Phi}(t_0), \tag{A3}$$

since after each (infinitesimal) time step the components within the irrelevant subspace, i.e., Kern(P), are projected out. For a general P the eigenvectors of $\hat{L}$ are not the same as for L, but known eigenvectors of $\hat{L}$ are always the column vectors of C.

1. Long time optimized projection

If we assume that the eigenvalues of L are $\leqslant 0$, for long times $t \gg 1$ the time evolution operators $e^{t\hat{L}}$, e^{tL} become the projectors onto the kernels of L or $\hat{L}$, respectively. In the eigenbasis ψ_i^{eig} it is simply

$$\psi_i^{eig} e^{tL} \psi_j^{eig} = \delta_{ij} e^{t\lambda_i}. \tag{A4}$$

The product of the reduced evolution operator e^{tPLP} and P converges for long times to the projector onto Kern(PLP)$\cap$Range(P). More explicitly this is

$$\lim_{t\to\infty} e^{tL} = \mathbb{1}|_{\mathrm{Kern}(L)}, \tag{A5}$$

$$\lim_{t\to\infty} e^{tPLP} = \mathbb{1}|_{\mathrm{Kern}(PLP)}. \tag{A6}$$

This gives for the error

$$E_\infty = \lim_{t\to\infty} E\Phi(t) = (\mathbb{1}|_{\mathrm{Kern}(L)} - \mathbb{1}|_{\mathrm{Kern}(PLP)}P)\Phi(t_0). \tag{A7}$$

In the long time limit we can obtain a zero error for all initial conditions if we have

$$\mathrm{Kern}(PLP) \cap \mathrm{Range}(P) \equiv \mathrm{Kern}(L). \tag{A8}$$

This is achieved by requiring

$$\mathrm{Kern}(L) \subset \mathrm{Range}(P), \tag{A9}$$

and

$$\mathrm{Range}(P) \quad L\ \text{invariant}, \tag{A10}$$

as we show now.

Consider a $\phi \in \mathrm{Range}(P)$. Then $P\phi=\phi$ and due to the L invariance of Range(P) it is $L\phi \in \mathrm{Range}(P)$ resulting in $PLP\phi=PL\phi=L\phi$. This gives for P with Range(P) being L invariant,

$$\mathrm{Kern}(PLP) \cap \mathrm{Range}(P) = \mathrm{Kern}(L) \cap \mathrm{Range}(P). \tag{A11}$$

Equation (A8) can be retrieved from Eq. (A11) just by requiring condition (A9). Thus, in the long time limit Eq. (A7) becomes identically zero.

2. Short time optimized projection

For short times we consider here the reduction from an N-dimensional to an $(N-1)$-dimensional system. For further reductions the results can be applied by iteration. The projector P becomes then $P_{ij}=\mathbb{1}_{ij}-c_i c_j$, where $\mathbf{c}$ is the removed state. In order to minimize the error for the short time evolution measured by the L^2 norm we have to minimize

$$E_s(t) = \|e^{tL}\phi - e^{PLP}P\phi\|_2 \approx \|(\mathbb{1} + tL - P - PLP)\phi\|_2 = \|E\phi\|_2. \tag{A12}$$

Here we have already used an expansion in powers of t and truncated after the first order terms.

Since we have no information on ϕ, we minimize Eq. (A12) by using the Frobenius norm $|\cdot|_F$ of the error operator E. The Frobenius norm is consistent with the L^2 norm [21], i.e.,

$$\|Ax\|_2 \leqslant |A|_F \|x\|_2 \quad \forall\ A \in R^{n\times n}, \quad x \in R^n. \tag{A13}$$

By inserting $P=\mathbb{1}-C$ we obtain for the error operator

$$\begin{aligned} E &= \mathbb{1} - P + t[L - (\mathbb{1} - C)L(\mathbb{1} - C)] \\ &= C + t(L - L + LC + CL - CLC) \\ &= C + t(LC + CL - CLC). \end{aligned} \tag{A14}$$

We assume L to be symmetric, i.e., $L_{ij}=L_{ji}$. Thus L has an orthonormal eigenbasis $\{\varphi_{i\alpha}\}_{\alpha=1\ldots N}$ where the columns are the eigenvectors of L. The eigenvalues are λ_α and the matrix elements of the error operator E are decomposed in this basis as

$$\begin{aligned} E_{ij} &= \sum_{\alpha\beta} \varphi_{\alpha i} E_{\alpha\beta} \varphi_{\beta i} \\ &= C_{ij} + t\sum_n \left(L_{in}C_{nj} + C_{in}L_{nj} - \sum_m C_{in}L_{nm}C_{mj} \right) \end{aligned} \tag{A15}$$

with

$$C_{ij}=\sum_{\alpha\beta}\varphi_{\alpha i}c_{\alpha}c_{\beta}\varphi_{\beta j}, \tag{A16}$$

$$\sum_{mn}C_{in}L_{nm}C_{mj}=\sum_{\alpha\beta nm}\varphi_{\alpha i}c_{\alpha}c_{n}L_{nm}c_{m}c_{\delta}\varphi_{\beta j}, \tag{A17}$$

$$\sum_{n}L_{in}C_{nj}=\sum_{\alpha\beta n}\varphi_{\alpha n}L_{in}c_{\alpha}c_{\beta}\varphi_{\beta j}, \tag{A18}$$

$$\sum_{n}C_{in}L_{nj}=\sum_{\alpha\beta n}\varphi_{\alpha i}c_{\alpha}c_{\beta}L_{nj}\varphi_{\beta n}. \tag{A19}$$

We use the orthogonality of the φ_{α}, i.e.,

$$\sum_{i}\varphi_{\alpha i}\varphi_{\beta i}=\delta_{\alpha\beta}=\sum_{i}\varphi_{i\alpha}\varphi_{i\beta}. \tag{A20}$$

In the eigenbasis the removed degree of freedom $\mathbf{c}$ can be written as $\tilde{\mathbf{c}}$ with components

$$\tilde{c}_i=\sum_{\alpha}\varphi_{\alpha i}c_{\alpha},\quad c_{\beta}=\sum_{i\alpha}\varphi_{\alpha i}\varphi_{\beta i}c_{\alpha}=\sum_{i}\varphi_{\beta i}\tilde{c}_i. \tag{A21}$$

The average of L in the removed state $\mathbf{c}$, is

$$\begin{aligned}\langle L\rangle_{\mathbf{c}}&:=\sum_{nm}c_nL_{nm}c_m\\&=\sum_{nmij}\varphi_{ni}\tilde{c}_iL_{nm}\varphi_{mj}\tilde{c}_j\\&=\sum_{nij}\varphi_{ni}\tilde{c}_i\lambda_j\varphi_{nj}\tilde{c}_j\\&=\sum_{i}\tilde{c}_i^2\lambda_i.\end{aligned} \tag{A22}$$

The matrix elements from Eqs. (A16)–(A19) become

$$C_{ij}=\tilde{c}_i\tilde{c}_j, \tag{A23}$$

$$\sum_{mn}C_{in}L_{nm}C_{mj}=\tilde{c}_i\langle L\rangle_{\mathbf{c}}\tilde{c}_j, \tag{A24}$$

$$\sum_{n}L_{in}C_{nj}=\sum_{\alpha n}\varphi_{\alpha i}L_{\alpha n}c_n\tilde{c}_j=\sum_{n}\lambda_i\varphi_{ni}c_n\tilde{c}_j=\lambda_i\tilde{c}_i\tilde{c}_j, \tag{A25}$$

$$\sum_{n}C_{in}L_{nj}=\sum_{n}\tilde{c}_ic_n\lambda_j\varphi_{nj}=\tilde{c}_i\tilde{c}_j\lambda_j. \tag{A26}$$

Thus for the matrix elements of the error operator we obtain

$$E_{ij}=\tilde{c}_i\tilde{c}_j[1+t(\lambda_i+\lambda_j-\langle L\rangle_{\mathbf{c}})]. \tag{A27}$$

We minimize the Frobenius norm of E given by

$$|E|_F=\sum_{ij}|E_{ij}|^2=\sum_{ij}\tilde{c}_i^2\tilde{c}_j^2[1+t(\lambda_i+\lambda_j-\langle L\rangle_{\mathbf{c}})]^2 \tag{A28}$$

for a normalized $\mathbf{c}$, i.e.,

$$1=\|c\|_2^2=\sum_i c_i^2=\sum_i\tilde{c}_i^2. \tag{A29}$$

Since E is a linear operator it follows that $\|E\mathbf{x}\|_2=\|\mathbf{x}\|_2\|E\hat{\mathbf{x}}\|_2$ with $\mathbf{x}=\hat{\mathbf{x}}\|\mathbf{x}\|_2$. Without no restriction to $\|\mathbf{x}\|_2$ the zero vector would always minimize $\|E\mathbf{x}\|_2$. Furthermore, each lower bound K for $\|\mathbf{x}\|_2$ will lead to the same $\hat{\mathbf{x}}$ with $\|\hat{\mathbf{x}}\|_2=K$. This is not true for the general nonlinear case as in [23].

Incorporating this condition $|E|_F$ reduces to

$$\begin{aligned}|E|_F^2&=1+2t\langle L\rangle_{\mathbf{c}}+2t\langle L\rangle_{\mathbf{c}}-2t\langle L\rangle_{\mathbf{c}}+t^2\langle L\rangle_{\mathbf{c}}^2\\&\quad+2t^2\langle L\rangle_{\mathbf{c}}^2-2t^2\langle L\rangle_{\mathbf{c}}^2+t^2\langle L\rangle_{\mathbf{c}}^2-2t^2\langle L\rangle_{\mathbf{c}}^2+t^2\langle L\rangle_{\mathbf{c}}^2\\&=1+2t\langle L\rangle_{\mathbf{c}}+t^2\langle L\rangle_{\mathbf{c}}^2\\&=(1+t\langle L\rangle_{\mathbf{c}})^2\Rightarrow|E|_F\\&=|1+t\langle L\rangle_{\mathbf{c}}|.\end{aligned} \tag{A30}$$

Consequently, in order to minimize $|E|_F$ we have to minimize $\langle L\rangle_{\mathbf{c}}$.

The minimization itself is performed using Lagrangian multipliers for the constraint Eq. (A29). The necessary condition for a minimum is

$$\begin{aligned}0&=\frac{\partial}{\partial\tilde{c}_k}[\langle L\rangle_{\mathbf{c}}+\eta(1-\|\mathbf{c}\|_2^2)]\\&=\frac{\partial}{\partial\tilde{c}_k}\sum_i[\tilde{c}_i^2\lambda_i+\eta(1-\tilde{c}_i^2)]\\&=2\tilde{c}_k(\lambda_k-\eta).\end{aligned} \tag{A31}$$

This is true if either $\tilde{c}_k=0$ or $\eta=\lambda_k$. The last equation can only be true for a single value of λ_k. We denote the nonzero component as $\tilde{c}_{k'}\neq0$ and $\tilde{c}_k=\delta_{kk'}\tilde{c}_{k'}$. From Eq. (A29) it follows further that $\tilde{c}_k=\delta_{kk'}$.

Inserting this in Eq. (A30) we obtain

$$|E|_F=\left|1+t\sum_k\tilde{c}_k^2\lambda_k\right|=|1+t\lambda_{k'}|. \tag{A32}$$

For small t, i.e., $t<|\lambda_i|^{-1}$ $\forall$ i this is clearly minimal if we choose $\lambda_{k'}$ to be the smallest eigenvalue.

Further iterations, e.g., n times, of selecting the irrelevant states remove successively the eigenstates corresponding to the n lowest eigenvalues. This is due to the fact that the spaces Kern(C)$\equiv$Range(P) and Range(C)$\equiv$Kern(P) are by construction L invariant. This also makes the iteration unambiguous, a feature that is in general not present for nonlinear problems.

Note also that since $\lambda_i\leqslant0$ the reduced states always belong to Range(L) as long as any remaining eigenvalue, i.e., an eigenvalue of $P_{n-1}LP_{n-1}$, is nonzero. Here, P_{n-1} results from the previous reduction step. In this case the error always vanishes for long times.

Summarizing, the optimal short time projection leads to results that are not only consistent with the long time accuracy requirements, but even include them.

[1] L. Sirovich, Q. Appl. Math. **XLV**, 561 (1987).

[2] J. D. Murray, *Mathematical Biology*, 3rd ed. (Springer, Berlin, 2002).

[3] D. Lucia, P. S. Beran, and W. A. Silva, Prog. Aerosp. Sci. **40**, 51 (2004).

[4] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).

[5] J. M. Burgers, *The Nonlinear Diffusion Equation* (Riedel, Boston, 1974).

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes* (Cambridge Univerity Press, Cambridge, 2007).

[7] R. FitzHugh, Biophys. J. **1**, 445 (1961).

[8] T. Yanagita, Y. Nishiura, and R. Kobayashi, Phys. Rev. E **71**, 036226 (2005).

[9] A. N. Gorban, I. V. Karlin, and A. Y. Zinovyev, Physica A **333**, 106 (2004).

[10] A. Steindl and H. Troger, Int. J. Solids Struct. **38**, 2131 (2001).

[11] E. N. Lorenz, Statistical Forecasting Project MIT Scientific Report No. 1, 1956 (unpublished).

[12] G. Berkooz, P. Holmes, and J. L. Lumley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Monographs on Mechanics (Cambridge University Press, Cambridge, 1998).

[13] C. W. Rowley, T. Colonius, and R. M. Murray, Physica D **189**, 115 (2003).

[14] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmore, and F. Thiele, J. Fluid Mech. **497**, 335 (2003).

[15] T. Bui-Thanh, M. Damodaran, and K. Willcox, AIAA Pap. **42** 1505 (2004).

[16] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems* (Cambridge University Press, Cambridge, 2005).

[17] C. W. Rowley, Int. J. Bifurcation Chaos Appl. Sci. Eng. **15**, 997 (2001).

[18] D. J. Lucia, P. I. King, and P. S. Beran, Comput. Fluids **32**, 917 (2003).

[19] S. R. White and R. M. Noack, Phys. Rev. Lett. **68**, 3487 (1992).

[20] S. R. White, Phys. Rev. B **48**, 10345 (1993).

[21] G. H. Golub and C. F. VanLoan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, 2006).

[22] B. A. Fryxell, J. Comput. Phys. **63**, 283 (1986).

[23] A. Degenhard and J. Rodríquez Laguna, J. Stat. Phys. **106**, 1093 (2002).