MORI-ZWANZIG REDUCED MODELS FOR UNCERTAINTY QUANTIFICATION

JING LI AND PANOS STINIS*  
Pacific Northwest National Laboratory  
Richland, WA 99354, USA

(Communicated by Kevin Lin)

Abstract. In many time-dependent problems of practical interest the parameters and/or initial conditions entering the equations describing the evolution of the various quantities exhibit uncertainty. One way to address the problem of how this uncertainty impacts the solution is to expand the solution using polynomial chaos expansions and obtain a system of differential equations for the evolution of the expansion coefficients. We present an application of the Mori-Zwanzig (MZ) formalism to the problem of constructing reduced models of such systems of differential equations. In particular, we construct reduced models for a subset of the polynomial chaos expansion coefficients that are needed for a full description of the uncertainty caused by uncertain parameters or initial conditions.

Even though the MZ formalism is exact, its straightforward application to the problem of constructing reduced models for estimating uncertainty involves the computation of memory terms whose cost can become prohibitively expensive. For those cases, we present a Markovian reformulation of the MZ formalism which is better suited for reduced models with long memory. The reformulation can be used as a starting point for approximations that can alleviate some of the computational expense while retaining an accuracy advantage over reduced models that discard the memory altogether. Our results support the conclusion that successful reduced models need to include memory effects.

1. Introduction. The problem of quantifying the uncertainty of the solution of systems of partial or ordinary differential equations has become in recent years a rather active area of research. The realization that more often than not, for problems of practical interest, one is not able to determine the parameters, initial conditions, boundary conditions etc. to within high enough accuracy, has led to a flourishing literature of methods for quantifying the impact that this uncertainty imposes on the solution of the problems under investigation (see e.g. [8, 15, 17, 20, 24, 26, 2, 25]).

2010 Mathematics Subject Classification. Primary: 65C20, 65M99; Secondary: 41A10.

Key words and phrases. Uncertainty quantification, model reduction, Mori-Zwanzig formalism, renormalization, long memory, Markovian reformulation.

This research at Pacific Northwest National Laboratory (PNNL) was partially supported by the U.S. Department of Energy (DOE) Office of Advanced Scientific Computing Research (ASCR) Collaboratory on Mathematics for Mesoscopic Modeling of Materials (CM4), under Award Number DE-SC0009280 and partially by the U.S. DOE ASCR project “Uncertainty Quantification For Complex Systems Described by Stochastic Partial Differential Equations”. PNNL is operated by Battelle for the DOE under Contract DE-AC05-76RL01830.

* Corresponding author: Panos Stinis.
In this work we focus on techniques based on polynomial chaos expansions of the uncertain solution [8, 28, 26]. In principle, if the orthogonal polynomial type and the corresponding random variables are determined, both intrusive and non intrusive methods can be used to evaluate the coefficients of the expansion. For example, stochastic collocation, based on tensor products of one-dimensional quadrature rules, is often employed when dimensionality is small [18, 27, 1]. However, as the dimension increases, the number of quadrature points needed for the tensor product rule increases exponentially. To mitigate this, sparse grid and adaptive collocation methods have been proposed to deal with moderate dimensionality [19, 7, 13, 16, 6]. When the dimension of the random inputs is large, none of the above collocation methods is feasible. One way to address this problem is to look for reduced models for a subset of the variables needed for a complete description of the uncertainty. The effect of all types of uncertainty is intimately connected with the inherent instabilities that may be present in the underlying system which we subject to the uncertainty. These considerations remain equally, if not more, important when we attempt to construct reduced models for uncertainty quantification.

In the current work, we are concerned with the construction of reduced models for systems of differential equations that arise from polynomial chaos expansions of solutions of a PDE or ODE system. In particular, we focus on the case that the given PDE or ODE system contains uncertain parameters or initial conditions and we want to construct a reduced model for the evolution of a subset of the polynomial chaos expansions that are needed for a complete description of the uncertainty caused by the uncertain parameters. There are different methods to construct reduced models for PDE or ODE systems (see e.g. [9, 5] and references therein). We choose to use the Mori-Zwanzig formalism in order to construct the reduced model [3, 4].

The main issue with all model reduction approaches is the computation of the memory caused by the process of eliminating variables from the given system (referred to as the full system from this point on) [5]. The memory terms are, in general, integral terms which account for the history of the variables that are not resolved. In principle the integrands appearing in the memory terms can be computed through the solution of the orthogonal dynamics equation [4]. We present some examples where this procedure can be implemented and the resulting reduced model can be estimated. Those examples highlight the definite improvement in accuracy of a reduced model when it includes a memory term. However, it is also easy to come up with examples where the solution of the orthogonal dynamics equation becomes prohibitively expensive.

For such cases we present a Markovian reformulation of the MZ formalism which allows the calculation of the memory terms through the solution of ordinary differential equations instead of the computation of convolution integrals as they appear in the original formulation. The construction of our Markovian reformulation is better suited for situations where the reduced model is expected to have a long memory (see also the related discussion in Section 3.1.5). We also present an algorithm which allows the estimation of the necessary parameters on the fly. This means that one starts evolving the full system and use it to estimate the reduced model parameters. Once this is achieved, the simulation continues by evolving only the reduced model with the necessary parameters set equal to their estimated values from the first part of the algorithm. Of course, such an approximation of the memory term cannot work under all circumstances. We present results for a nontrivial
problem where it does yield a reduced model with improved behavior compared to
a model that ignores the memory terms altogether.

We should note that this alternative approach to computing the memory term
fits in the renormalization framework advocated recently by one of the authors [22]
in order to construct reduced models for singular PDEs. In particular, the idea is
that one embeds the MZ reduced model in a larger family of reduced models which
share the same functional form but may have additional parameters for enhanced
flexibility. These extra parameters are determined so that the reduced model repro-
duces some dynamic features of the full system. After this is done, one can switch
to the reduced model for the rest of the simulation. In the current work, the extra
parameters are the lengths of the memory appearing in the MZ reduced model (see
the related discussion in Section 3.2).

Section 2 presents a brief introduction to the MZ formalism for the construction
of reduced models of systems of ODEs. In Section 3 we develop the Markovian
reformulation of the MZ formalism and show how one can estimate adaptively the
parameters appearing in the reduced model. Section 4 presents numerical results
both for the original MZ formalism (Sections 4.1.1-4.1.2) and its Markovian re-
formulation (Section 4.2.1). Finally, in Section 5 we discuss directions for future
work.

2. The Mori-Zwanzig formalism. We begin in Section 2.1 with a brief presen-
tation of the Mori-Zwanzig formalism for constructing reduced models of systems
of differential equations (see [3, 4, 5] for more details). The MZ formalism belongs
in the class of projection methods. As such, the choice of projection operator can
affect significantly the final form of the reduced models. Section 2.2 contains a dis-
cussion of various projection operators used in the literature as well as the choice
we have made in the current work.

2.1. The Mori-Zwanzig equation. Suppose we are given the system

\[ \frac{du(t)}{dt} = R(t, u(t)), \tag{1} \]

where \( u = \{u_k\} \), \( k \in H \cup G \) with initial condition \( u(0) = u_0 \). The unknown
variables (modes) are divided into two groups, one group is indexed in \( H \) and the
order indexed in \( G \). Our goal is to construct a reduced model for the modes in the
set \( H \). The system of ordinary differential equations we are given can be transformed
into a system of linear partial differential equations

\[ \frac{\partial \varphi_k}{\partial t} = L \varphi_k, \quad \varphi_k(u_0, 0) = u_{0k}, \quad k \in H \cup G \tag{2} \]

where \( L = \sum_{k \in H \cup G} R_i(u_0) \frac{\partial}{\partial u_{0i}} \). The solution of (2) is given by \( u_k(u_0, t) = \varphi_k(u_0, t) \).
Using semigroup notation we can rewrite (2) as

\[ \frac{\partial}{\partial t} e^{Lt} u_{0k} = L e^{Lt} u_{0k} \]

Suppose that the vector of initial conditions can be divided as \( u_0 = (\hat{u}_0, \tilde{u}_0) \), where
\( \hat{u}_0 \) is the vector of the resolved variables (those in \( H \)) and \( \tilde{u}_0 \) is the vector of the
unresolved variables (those in \( G \)). Let \( P \) be an orthogonal projection on the space
of functions of \( \hat{u}_0 \) and \( Q = I - P \) (see also Section 2.2 for a more detailed discussion
of projection operators).
Equation (2) can be rewritten as
\[ \frac{\partial}{\partial t} e^{tL} u_{0k} = e^{tL} P L u_{0k} + e^{tQL} Q L u_{0k} + \int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds, \quad k \in H, \quad (3) \]
where we have used Dyson’s formula
\[ e^{tL} = e^{tQL} + \int_0^t e^{(t-s)L} P L e^{sQL} ds. \quad (4) \]
Equation (3) is the Mori-Zwanzig identity. Note that this relation is exact and is an alternative way of writing the original PDE. It is the starting point of our approximations. Of course, we have one such equation for each of the resolved variables \( u_k, k \in H \). The first term in (3) is usually called Markovian since it depends only on the values of the variables at the current instant, the second is called “noise” and the third “memory” (see [5] for a discussion of the significance of each term).

If we write
\[ e^{tQL} Q L u_{0k} = w_k, \]
\[ w_k(u_0, t) \]
satisfies the equation
\[ \begin{cases} \frac{\partial}{\partial t} w_k(u_0, t) = Q L w_k(u_0, t) \\ w_k(u_0, 0) = Q L u_{0k} = R_k(u_0) - (P R_k)(u_0). \end{cases} \quad (5) \]
If we project (5) we get
\[ P \frac{\partial}{\partial t} w_k(u_0, t) = P Q L w_k(u_0, t) = 0, \]
since \( PQ = 0 \). Also for the initial condition
\[ P w_k(u_0, 0) = P Q L u_{0k} = 0 \]
by the same argument. Thus, the solution of (5) is at all times orthogonal to the range of \( P \). We call (5) the orthogonal dynamics equation (see more details in [3]). Since the solutions of the orthogonal dynamics equation remain orthogonal to the range of \( P \), we can project the Mori-Zwanzig equation (3) and find
\[ \frac{\partial}{\partial t} P e^{tL} u_{0k} = P e^{tL} P L u_{0k} + P \int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds. \quad (6) \]
We will not present here more details about how to start from Eq. (6) and construct reduced models of different orders for a general system of ODEs. Such constructions have been documented thoroughly elsewhere (see e.g. [4]). However, we will provide such details for the specific numerical examples in Sections 4.1.1-4.1.2.

2.2. The choice of the projection operator \( P \). Before we proceed we would like to comment on choices of the projection operator \( P \) that have appeared in the literature as well as the choice that we have opted for in the current work. As we have seen in Eq. (3), what the MZ formalism offers is a way to decompose the RHS of the equations for the resolved variables. This decomposition involves three terms. The choice of the projection operator affects how the information content of the RHS of the equation is distributed among these three terms. The popular choices for the projection operator (see e.g. [4]) include the conditional expectation and the finite-rank projection with respect to an *invariant* measure for the system. The projection operator based on the conditional expectation also comes with the added property of being optimal in an \( L_2 \) sense. One uses these projection operators
when the objective is to produce trajectories for the resolved variables starting from initial conditions sampled from the invariant measure or when one wants to study the relaxation to the invariant measure when starting from a non-typical initial condition.

For the examples that we have studied, there are two factors that motivate us to use a different projection operator: i) we do not have access to an invariant measure and ii) our study has a different objective than the ones stated in the previous paragraph. In particular, we want to study the cascade of activity from large scales to small ones (in random space). We can use initial conditions that have no activity in small scales, represented as 0 for the unresolved variables (recall that our variables are the coefficients of an expansion of the randomness in Legendre polynomials similar to a Fourier series in physical space).

As a result, the projection $P$ we have chosen is defined as $(Pf)(\hat{u}_0) = P(f(u_0)) = f(\hat{u}_0, \tilde{u}_0) = f(\hat{u}_0, 0)$. What this definition means is that our projection operator, when applied to a function of the initial conditions, assigns the value 0 to the unresolved variables. One way to think about this that makes contact with other choices for the projection operator that have appeared in the literature, is that we have used a projection operator which is defined with respect to a measure on the initial conditions that has a delta measure for the unresolved variables centered at 0. Thus, it does not allow fluctuations for the initial condition of the unresolved variables.

Our choice of operator comes with a few distinctive features. First, unlike the more popular choices of the projection operator presented above, our chosen operator commutes with nonlinear functions. This property turns out to be convenient for analytical calculations of expressions that appear in the MZ formalism. Second, for our choice of projection operator, the Markovian term is identical to the term that would result from the Galerkin projection that sets the unresolved variables to zero for all time. However, note that our reduced models do not rely only on the Markovian term, but they incorporate the memory terms too which are there to account for the interaction of the resolved and unresolved variables. Third, since our choice of projection operator does not allow fluctuations in the initial conditions of the unresolved variables, then when we apply the projection operator to the MZ equation to cancel the noise term, we get an equation which is valid pathwise. In other words, the projected, noiseless MZ equation (6) is valid for the prediction of each trajectory that starts with whatever initial conditions we have chosen for the resolved variables and 0 for the unresolved variables. Finally, we would like to add that this particular choice of projection operator has been used successfully by the current authors before to tackle a variety of problems from detection and tracking of singularities to locating bifurcations (see e.g. [11, 16, 21, 23]).

3. Markovian reformulation of the MZ formalism. While the MZ model given by Eq. (6) is exact, its construction can be involved and most importantly, very costly. The main source of computational expense is the memory term. Technically, the cost associated with the memory term comes from two sources: i) the presence of the orthogonal dynamics equation solution operator $e^{sQL}$ and ii) the need to find an expression in terms of the resolved variables and time for $PLe^{sQL}QLu_{0k}$ which appears in the memory integrand. The presence of $e^{sQL}$ is problematic because the orthogonal dynamics equation is, for the general case, a PDE in as many dimensions as the original system of ODEs. Also, finding an expression for $PLe^{sQL}QLu_{0k}$ is problematic because, in general, it is not possible to separate
the dependence of the expression on time and on the resolved variables. Both are formidable tasks and we will show with several examples how they can increase the cost of constructing the reduced model. For some cases (see e.g. Section 4.1.1 and 4.1.2) both tasks can be tackled through the use of a finite-rank projection for the operator \( P \). However, we will show with a simple example (see Section 4.2.1) that the use of a finite-rank projection may be too costly itself. For such cases, we need an alternative approach to the construction of the memory term. In this section we describe a reformulation of the problem of computing the memory term which can alleviate some of these issues. Also, we present numerical results from the application of this approach in Section 4.2.1.

There are two cases that the reformulation can be applied to: i) the finite memory case and ii) the infinite memory case. The finite memory case is much more involved and we explain the main steps of the reformulation in Section 3.1 (see Sections 3.1.1-3.1.4). A discussion on the applicability of the reformulation can be found in Section 3.1.5. We also provide a summary of the main steps in Section 3.1.6. The infinite memory case is simpler and is presented in Section 3.2. In the same section we offer an interpretation of the finite memory case construction through the perspective of renormalization.

3.1. Finite memory. We focus on the case when the memory has a finite extent only. The case of infinite memory is simpler and is a special case of the formulation presented below (see Section 3.2).

3.1.1. Differential equation hierarchy for the memory estimation. Let \( w_{0k}(t) = P \int_0^t e^{(t-s)L} PLe^{QL}QLu_{0k}ds = P \int_0^t e^{sL} PLe^{(t-s)QL}QLu_{0k}ds \), by the change of variables \( t' = t - s \). Note, that \( w_{0k} \) depends both on \( t \) and the resolved part of the initial conditions \( \hat{u}_0 \). We have suppressed the \( \hat{u}_0 \) dependence for simplicity of notation. If the memory extends only for \( t_0 \) units in the past (with \( t_0 \leq t \)) then

\[
w_{0k}(t) = P \int_{t-t_0}^t e^{sL} PLe^{(t-s)QL}QLu_{0k}ds.
\]

The evolution of \( w_{0k} \) is given by

\[
\frac{dw_{0k}}{dt} = Pe^{tL} PQLQLu_{0k} - Pe^{(t-t_0)L} PLe^{t_0QL}QLu_{0k} + w_{1k}(t),
\]

where \( w_{1k}(t) = P \int_{t-t_0}^t e^{sL} PLe^{(t-s)QL}QLQLu_{0k}ds \).

To allow for more flexibility, let us assume that the integrand in the formula for \( w_{1k}(t) \) contributes only for \( t_1 \) units with \( t_1 \leq t_0 \). Then

\[
w_{1k}(t) = P \int_{t-t_1}^t e^{sL} PLe^{(t-s)QL}QLQLu_{0k}ds.
\]

We can proceed and write an equation for the evolution of \( w_{1k}(t) \) which reads

\[
\frac{dw_{1k}}{dt} = Pe^{tL} PQLQLu_{0k} - Pe^{(t-t_1)L} PLe^{t_1QL}QLQLu_{0k} + w_{2k}(t),
\]

where \( w_{2k}(t) = P \int_{t-t_1}^t e^{sL} PLe^{(t-s)QL}QLQLQLu_{0k}ds \).
Similarly, if this integral extends only for $t_2$ units in the past with $t_2 \leq t_1$, then

$$w_{2k}(t) = P \int_{t-t_2}^{t} e^{sL} P L e^{(t-s)QL} Q L Q L Q L u_{0k} ds.$$  

This hierarchy of equations continues indefinitely. Also, we can assume for more flexibility that at every level of the hierarchy we allow the interval of integration for the integral term to extend to fewer or the same units of time than the integral in the previous level. If we keep, say, $n$ terms in this hierarchy, the equation for $w_{(n-1)k}(t)$ will read

$$\frac{dw_{(n-1)k}}{dt} = P e^{tL} P L (QL)^{n-1} Q L u_{ok} - P e^{(t-t_{n-1})L} P L e^{t_{n-1}QL} (QL)^{n-1} Q L Q L u_{0k} + w_{nk}(t)$$  

(9)

where

$$w_{nk}(t) = P \int_{t-t_n}^{t} e^{sL} P L e^{(t-s)QL} (QL)^{n} Q L Q L u_{0k} ds$$

Note that the last term in (9) involves the unknown evolution operator for the orthogonal dynamics equation. This situation is the well-known closure problem. We can stop the hierarchy at the $n$th term by assuming that $w_{nk}(t) = 0$.

In addition to the closure problem, the unknown evolution operator for the orthogonal dynamics equation appears in the equations for the evolution of the quantities $w_{0k}(t), \ldots, w_{(n-1)k}(t)$ through the various terms $P e^{(t-t_{n})L} P L e^{t_{n}QL} Q L Q L u_{0k}, \ldots, P e^{(t-t_{0})L} P L e^{t_{0}QL} (QL)^{n-1} Q L Q L u_{0k}$ respectively.

3.1.2. Discretization of the memory integrals to estimate the orthogonal dynamics.

We describe now a way to express the terms involving the unknown orthogonal dynamics operator through known quantities so that we obtain a closed system for the evolution of $w_{0k}(t), \ldots, w_{(n-1)k}(t)$.

For $w_{0k}(t)$ we have

$$w_{0k}(t) = P \int_{t-t_0}^{t} e^{sL} P L e^{(t-s)QL} Q L Q L u_{0k} ds$$

$$= \left[ P e^{tL} P L Q L u_{0k} + P e^{(t-t_0)L} P L e^{t_0QL} Q L Q L u_{0k} \right] \frac{t_0}{2} + O(t_0^2)$$

from which we find

$$P e^{(t-t_0)L} P L e^{t_0QL} Q L Q L u_{0k} = \left( \frac{2}{t_0} \right) w_{0k}(t) - P e^{tL} P L Q L u_{0k} + O(t_0^2)$$

and from (7)

$$\frac{dw_{0k}}{dt} = - \left( \frac{2}{t_0} \right) w_{0k}(t) + 2 P e^{tL} P L Q L u_{0k} + w_{1k}(t) + O(t_0^2).$$

Note that having a memory of finite length results in the appearance of the dissipative term $-\left( \frac{2}{t_0} \right) w_{0k}(t)$ whose magnitude depends on the actual length of the memory.

Similarly, for $w_{1k}(t)$ we have

$$w_{1k}(t) = P \int_{t-t_1}^{t} e^{sL} P L e^{(t-s)QL} Q L Q L u_{0k} ds$$
If we drop the error terms \(w\) integrals up to and including the dynamics operator on the RHS of the equations for the evolution of all the memory variables to determine the evolution of the resolved variables. We can continue in this way and estimate the expressions involving the orthogonal complement of the memory integrals to ensure numerical consistency.

### 3.1.3. Decomposition of the memory integrals to ensure numerical consistency

Since we want to treat the case where \(t_0, t_1, \ldots, t_{n-1}\) are not necessarily small, we should modify the construction above to ensure numerical consistency between the integral discretizations and the ODE solver used to solve the resulting auxiliary system. In particular, if the memory lengths \(t_0, t_1, \ldots, t_{n-1}\) are longer than the stepsize \(\Delta t_{res}\) used for the ODE solver, we need to decompose the integrals defining \(w_{0k}(t), w_{1k}(t), \ldots, w_{(n-1)k}(t)\) into sums of integrals of shorter length. We explain now how this affects the formulas for the computation of the memory in our Markovian reformulation.

We divide the interval \([t - t_0, t]\) in \(n_0\) subintervals. Define

\[
\begin{align*}
    w^{(1)}_{0k}(t) &= P \int_{t - t_0}^{t} e^{sL} PLe^{(t-s)QL}QLu_{0k} ds \\
    w^{(2)}_{0k}(t) &= P \int_{t - 2t_0}^{t - t_0} e^{sL} PLe^{(t-s)QL}QLu_{0k} ds \\
    &\vdots \\
    w^{(n_0)}_{0k}(t) &= P \int_{t - t_0}^{t - (n_0-1)t_0} e^{sL} PLe^{(t-s)QL}QLu_{0k} ds,
\end{align*}
\]

from which we find

\[
P e^{(t-t_1)LPLQL}QLu_{0k} = \left(\frac{2}{t_1}\right) w_{1k}(t) - P e^{tL} PQLQLu_{0k} + O(t_1^2)
\]

and from (8)

\[
\frac{dw_{1k}}{dt} = -\left(\frac{2}{t_1}\right) w_{1k}(t) + 2P e^{tL} PQLQLu_{0k} + w_{2k}(t) + O(t_1^2).
\]

We can continue in this way and estimate the expressions involving the orthogonal dynamics operator on the RHS of the equations for the evolution of all the memory integrals up to and including \(w_{n-1k}(t)\). We find

\[
\frac{dw_{n-1k}}{dt} = -\left(\frac{2}{t_{n-1}}\right) w_{n-1k}(t) + 2P e^{tL} PQLQLu_{0k} + w_{nk}(t) + O(t_{n-1}^2).
\]

If we drop the error terms \(O(t_0^2), O(t_1^2), \ldots, O(t_{n-1}^2)\) (and assume that \(w_{nk}(t)\) is negligible as before), we obtain the required auxiliary system of ODEs

\[
\begin{align*}
    \frac{dw_{0k}}{dt} &= -\left(\frac{2}{t_0}\right) w_{0k}(t) + 2P e^{tL} PQLQLu_{0k} + w_{1k}(t) \\
    \frac{dw_{1k}}{dt} &= -\left(\frac{2}{t_1}\right) w_{1k}(t) + 2P e^{tL} PQLQLu_{0k} + w_{2k}(t) \\
    \frac{dw_{n-1k}}{dt} &= -\left(\frac{2}{t_{n-1}}\right) w_{n-1k}(t) + 2P e^{tL} PQLQLu_{0k} + w_{nk}(t).
\end{align*}
\]

If we know the values of \(t_0, t_1, \ldots, t_{n-1}\) (see Section 3.1.4 for an approach to their estimation), we can solve this system in conjunction with the MZ equations for the resolved variables to determine the evolution of the resolved variables.
where $n_0 \Delta t_0 = t_0$ and $w_{0k}(t) = \sum_{i=1}^{n_0} w_{0k}^{(i)}(t)$. Similarly, we can define the quantities $w_{1k}^{(1)}(t), \ldots, w_{1k}^{(n_1)}(t)$

$$w_{1k}^{(1)}(t) = P \int_{t_{k-1}}^{t} e^{sL} P Le^{(t-s)Q} L Q L u_{0k} ds$$

$$w_{1k}^{(2)}(t) = P \int_{t_{k-2}}^{t_{k-1}} e^{sL} P Le^{(t-s)Q} L Q L u_{0k} ds$$

$$\cdots$$

$$w_{1k}^{(n_1)}(t) = P \int_{t_{n_1-1}}^{t_{n_1}} e^{sL} P Le^{(t-s)Q} L Q L u_{0k} ds,$$

where $n_1 \Delta t_1 = t_1$ and $w_{1k}(t) = \sum_{i=1}^{n_1} w_{1k}^{(i)}(t)$. In a similar fashion we can define corresponding quantities for all the memory terms up to $w_{(n-1)k}(t) = \sum_{i=1}^{n-1} w_{(n-1)k}^{(i)}(t)$.

As before, we want to construct differential equations for the evolution of the quantities $w_{0k}^{(1)}(t), w_{0k}^{(2)}(t), \ldots, w_{(n-1)k}(t)$. In order to proceed we need to make an approximation for the integrals over the subintervals. This will allow us to estimate the expressions involving the orthogonal dynamics evolution operator. We will use again the trapezoidal rule.

For $w_{0k}^{(1)}(t)$ we have

$$w_{0k}^{(1)}(t) = P \int_{t_{k-1}}^{t} e^{sL} P Le^{(t-s)Q} L Q L u_{0k} ds$$

$$= \left[ Pe^{L} P L Q L u_{0k} + Pe^{(t-s)L} P L e^{L} v_{0k} Q L Q L u_{0k} \right] \frac{\Delta t_0}{2} + O((\Delta t_0)^2)$$

from which we find

$$Pe^{(t-s)L} P L e^{L} v_{0k} Q L Q L u_{0k} = \left( \frac{2}{\Delta t_0} \right) w_{0k}^{(1)}(t) - Pe^{sL} P L Q L u_{0k} + O((\Delta t_0)^2)$$

and from (7)

$$\frac{dw_{0k}^{(1)}}{dt} = - \left( \frac{2}{\Delta t_0} \right) w_{0k}^{(1)}(t) + 2Pe^{sL} P L Q L u_{0k} + w_{1k}^{(1)}(t) + O((\Delta t_0)^2).$$

Similarly, for $w_{0k}^{(2)}(t)$ we find

$$\frac{dw_{0k}^{(2)}}{dt} = \left( \frac{4}{\Delta t_0^2} \right) w_{0k}^{(1)}(t)$$

$$- \left( \frac{2}{\Delta t_0} \right) w_{0k}^{(2)}(t) - 2Pe^{sL} P L Q L u_{0k} + w_{1k}^{(2)}(t) + O((\Delta t_0)^2).$$

In general,

$$\frac{dw_{0k}^{(i)}}{dt} = - \left( \frac{2}{\Delta t_0} \right) w_{0k}^{(i)}(t) + (-1)^{i+1} 2Pe^{sL} P L Q L u_{0k}$$

$$+ \sum_{j=1}^{i-1} \left( \frac{4}{\Delta t_0^2} \right) (-1)^{i+j+1} w_{0k}^{(j)}(t) + w_{1k}^{(i)}(t) + O((\Delta t_0)^2) \text{ for } i = 1, \ldots, n_0. \quad (10)$$
Similarly,

\[
\frac{dw_{(i)}^{(i)}_{k}}{dt} = -\left(\frac{2}{\Delta t_{i}}\right)w_{(i)}^{(i)}_{1k}(t) + (-1)^{i+1}2Pe^{tL}PLQLQLu_{ok} + \left[\sum_{j=1}^{i-1} \left(\frac{4}{\Delta t_{j}}\right)(-1)^{i+j+1}w_{(i)}^{(j)}_{1k}(t)\right] + w_{2k}^{(i)}(t) + O((\Delta t_{i})^{2}) \quad \text{for } i = 1, \ldots, n_{1}
\]

\[
\ldots
\]

\[
\frac{dw_{(n-1)}^{(i)}}{dt} = -\left(\frac{2}{\Delta t_{n-1}}\right)w_{(n-1)}^{(i)}_{1k}(t) + (-1)^{i+1}2Pe^{tL}PLQLQLu_{ok} + \left[\sum_{j=1}^{i-1} \left(\frac{4}{\Delta t_{j}}\right)(-1)^{i+j+1}w_{(n-1)}^{(j)}_{1k}(t)\right] + O((\Delta t_{n-1})^{2}) \quad \text{for } i = 1, \ldots, n_{n-1}.
\]

By dropping the \(O((\Delta t_{0})^{2}), \ldots, O((\Delta t_{n-1})^{2})\) terms we obtain a system of \(n_{0} + n_{1} + \ldots + n_{n-1}\) differential equations for the evolution of the quantities \(w_{0}^{(i)}_{1}(t), \ldots, w_{(n-1)}^{(n-1)}_{1k}(t)\). This system allows us to determine the memory term \(w_{0}^{(i)}(t)\).

We want to comment on the required numerical consistency between the memory integral approximations and the ODE solver used. Since the approximation we have used for the integrals leads to an error \(O(\Delta t)^{2}\), where \(\Delta t = \max(\Delta t_{0}, \Delta t_{1}, \ldots, \Delta t_{n-1})\), the ODE solver for the auxiliary system should also be \(O(\Delta t)^{2}\). We have used the modified Euler method to solve numerically both the auxiliary system and the equations for the resolved variables. Also, we must have \(\Delta t \approx \Delta t_{res}\) where \(\Delta t_{res}\) is the stepsize used for the ODE solver. Finally, note that if we decide to use the same length \(t_{0} = t_{1}, \ldots, \) then the number of subintervals will be the same for the different layers of the approximation, thus simplifying somewhat the expressions.

The implementation of the above scheme requires the knowledge of the expressions for \(Pe^{tL}PLQLQLu_{ok}, \ldots, Pe^{tL}PLQLQLu_{ok}\). Since the computation of these expressions for large \(n\) can be rather involved for nonlinear systems (see Section 4.2.1), we expect that the above scheme will be used with a small to moderate value of \(n\). Finally, we mention that the above construction can be carried out for integration rules of higher order e.g. Simpson’s rule.

3.1.4. Estimation of the memory length. The construction presented above relies on an accurate determination of the memory lengths \(t_{0}, t_{1}, \ldots, t_{n-1}\). We present in this section a way to estimate these quantities on the fly. This means that we start evolving the full system, use it to estimate \(t_{0}, t_{1}, \ldots, t_{n-1}\) and then switch to the reduced model with the estimated values for \(t_{0}, t_{1}, \ldots, t_{n-1}\).

For simplicity of presentation we assume that we evolve only \(w_{0}^{(i)}(t)\). If we use the trapezoidal rule to discretize \(w_{0}^{(i)}(t)\) and eliminate the term \(Pe^{(t-t_{0})L}PLe^{t_{0}QL}QLu_{ok}\) from (7), the reduced model reads

\[
\frac{dP_{uk}}{dt} = Pe^{tL}PLu_{ok} + w_{ok}(t) \quad (12)
\]

\[
\frac{dw_{ok}}{dt} = 2Pe^{tL}QLQLu_{ok} - \frac{2}{t_{0}}w_{ok}(t) \quad (13)
\]

for \(k \in H\). We can solve (13) formally and substitute in (12) to get

\[
\frac{dP_{uk}}{dt} = Pe^{tL}PLu_{ok} + \int_{0}^{t} e^{-\lambda_{0}(t-s)}2Pe^{sL}QLQLu_{ok}ds \quad (14)
\]

Similarly,
where \( \lambda_0 = 2/t_0 \). Recall that, for the resolved variables, we have from the full system

\[
\frac{dP_{u_k}}{dt} = Pe^{tL}PLu_{0k} + Pe^{tL}QLu_{0k}.
\]  

(15)

We would like to estimate the memory decay parameter \( t_0 \) so that the reduced equation (14) for \( u_k \) reproduces the behavior of \( u_k \) as predicted by the full system (15). We can do that by requiring that the evolution of some integral quantity of the solution is the same when predicted by the reduced and full systems.

We begin by discretizing the integral term in (14). Suppose that we are evolving the full system with a step size \( \delta t \), where \( t = n_t \delta t \) (note that \( n_t \) increases as \( t \) increases). If we discretize the integral with the trapezoidal rule we find

\[
\frac{dP_{u_k}}{dt} = Pe^{tL}PLu_{0k}
\]  

(16)

\[
+ [f_k(t, \hat{u}_0) + 2 \sum_{j=1}^{n_t-1} e^{-\lambda_0(t-j\delta t)} f_k(j\delta t, \hat{u}_0) + e^{-\lambda_0t} f_k(0, \hat{u}_0)] \frac{\delta t}{2}
\]

where \( f_k(j\delta t, \hat{u}_0) = 2Pe^{j\delta tL}QLu_{0k} \) for \( j = 0, \ldots, n_t \). The quantities \( f_k(j\delta t, \hat{u}_0) \) can be computed from the full system.

There is freedom in the choice of the integral quantity whose evolution the reduced model should be able to reproduce. For example, we can use \( \sum_{k \in H} |P_{u_k}(t)|^2 \) the squared \( l_2 \) norm of the resolved variables. If we use this integral quantity, then from (16) and (15) we find that the unknown parameter \( t_0 \) must satisfy

\[
\sum_{k \in H} 2Re\{I_k(t, t_0)(P_{u_k})^*(t)\} = \sum_{k \in H} 2Re\{Pe^{tL}QLu_{0k}(P_{u_k})^*(t)\},
\]  

(17)

where

\[
I_k(t, t_0) = [f_k(t, \hat{u}_0) + 2 \sum_{j=1}^{n_t-1} e^{-\lambda_0(t-j\delta t)} f_k(j\delta t, \hat{u}_0) + e^{-\lambda_0t} f_k(0, \hat{u}_0)] \frac{\delta t}{2}
\]

and \( Re\{\cdot\} \) denotes the real part.

Let \( y = \exp[-\lambda_0 \delta t] \). Then,

\[
I_k(t, t_0) = [f_k(t, \hat{u}_0) + 2 \sum_{j=1}^{n_t-1} y^{n_t-j} f_k(j\delta t, \hat{u}_0) + y^{n_t} f_k(0, \hat{u}_0)] \frac{\delta t}{2}.
\]  

(18)

With this identification, equation (17) becomes a polynomial equation for \( y \) with \( y \in [0, 1] \). It is not difficult to solve equation (17) with an iterative method, for example Newton’s method. For the numerical results we present in Section 4.2.1, Newton’s method converged to double precision accuracy within 4-5 iterations. After an estimate \( \hat{y} \) has been obtained, we can find the estimate \( \hat{t}_0 \) of \( t_0 \) (recall \( \lambda_0 = 2/t_0 \)) from

\[
\hat{t}_0 = -\frac{2\delta t}{ln \hat{y}}.
\]  

(19)

**Determination of optimal estimate \( \hat{t}_0 \).** For each time instant \( t \) we can obtain through equations (17) and (19), an estimate \( \hat{t}_0(t) \) for \( t_0 \). Thus, the most important issue that we have to address is that of deciding which is the best estimate of \( t_0 \). In other words, at what time \( t_f \) should we stop estimating the value of \( t_0 \) so that we can use the estimated value \( \hat{t}_0(t_f) \) to evolve the reduced model from then on.
We define $\epsilon(t) = \max_{t \in [1, n_t]} |\hat{g}'(t + \delta t) - \hat{g}'(t)|$. The quantity $\epsilon(t)$ monitors the convergence of not only the value of the estimate $\hat{g}$ as a function of the time $t$, but of the whole function $e^{-\lambda_0(t-s)}$. Ideally, $\epsilon(t)$ converges to zero with increasing $t$. That will be the case if the approximation of the memory term only through Eq. (13) is enough. In other words, if the evolution of $w_{1k}(t), w_{2k}(t), \ldots$ is not needed. However, this will not always be the case. If keeping $Pe^{tL}PLQLu_{0kr}$ is not enough, then $\epsilon(t)$ will decrease with increasing $t$ up to some time $t_{min}$ when it will reach a nonzero minimum. After that time, it starts increasing. This signals that keeping only $Pe^{tL}PLQLu_{0kr}$ is not enough to describe accurately the memory.

In order to proceed we have two options: (i) construct a higher order model and (ii) identify $t_f = t_{min}$ and thus $t_0(t_f) = t_0(t_{min})$. Results for higher order models will be presented elsewhere (see also discussion in Section 5). In the numerical experiments we present in Section 4.2 we have chosen $t_0(t_f) = t_0(t_{min})$. Note that the procedure just outlined allows the automation of the algorithm. This means that there is no adjustable reduced model parameter that needs to be specified at the onset of the algorithm.

We are now in a position to state the adaptive Mori-Zwanzig algorithm which constructs a reduced model with the necessary memory term parameter $t_0$ estimated on the fly.

**Adaptive Mori-Zwanzig Algorithm**

1. Evolve the full system and compute, at every step, the estimate $\hat{t}_0(t)$. Use estimates of $t_0$ from successive steps to calculate $\epsilon(t) = \max_{t \in [1, n_t]} |\hat{g}'(t + \delta t) - \hat{g}'(t)|$.
2. When $\epsilon(t)$ reaches a minimum (possibly non zero) value at some instant $t_{min}$, pick $t_0(t_{min})$ as the final estimate of $t_0$.
3. For the remaining simulation time ($t > t_{min}$), switch from the full system to the reduced model. The reduced model is evolved with the necessary parameter $t_0$ set to its estimated value $\hat{t}_0(t_{min})$.

This procedure can be extended to the computation of optimal estimates for $t_1, t_2, \ldots$, i.e. when we evolve, in addition to $w_{0k}(t)$, the quantities $w_{1k}(t), w_{2k}(t), \ldots$. Also, as we have discussed before, we can assume that $t_0 = t_1 = t_2, \ldots$ thus simplifying the construction by having to determine only one parameter. Results for such higher order models will be presented elsewhere.

3.1.5. **Discussion on the applicability of the finite memory construction.** By inspecting the equations for the evolution of the various memory integrals (either in the form of Section 3.1.2 or the form of Section 3.1.3), we see that the construction requires in essence the knowledge of the value of the orthogonal dynamics operator $e^{tQL}$ and its derivatives at $t = 0$ i.e., its Taylor expansion around 0. There is a reason why these values are needed. The way we have developed the construction assumes that the memory is long. What this means is that the orthogonal dynamics operator evolves on a timescale that is at least comparable (if not longer) to the timescale of the resolved variables. Thus, the hope is that if $e^{tQL}$ evolves slowly then we can approximate it using only relatively few terms of its Taylor expansion. This case is relevant for realistic applications where more often than not the variables we wish to eliminate are not significantly faster than the variables we want to keep. For example, if the system we want to reduce is linear and thus the orthogonal dynamics operator is in the form $e^{tQL} = \exp(\alpha t)$, our construction would require that $|\alpha| < 1$ so that successively higher derivatives of $e^{tQL}$ evaluated at 0
have shrinking values. That would make the assumption that the hierarchy can be closed after relatively few terms more plausible.

The construction we have presented is more suitable for the case when the memory is long. In unpublished work, we have also attempted a variant of the construction for the case when the memory is short. In this case one can also develop a hierarchical system of ODEs to estimate the memory term. However, the construction does not involve the Taylor expansion of the orthogonal dynamics operator $e^{tQL}$. While on paper the resulting construction is similar to the one for the case of long memory presented above, there are mathematical and numerical issues that we have not been able to address successfully so far. We hope to report on the short memory construction in a future publication.

3.1.6. Summary of steps for the finite memory Markovian reformulation. We present here a summary of the steps needed to set up the Markovian reformulation of the memory term. This is only meant to be an outline and the details for each step are relegated to the appropriate section.

1. Choose the order $n$ where the hierarchy will be terminated. This means that for each of the resolved variables, we will use $n$ additional quantities $(w_{0k}(t), \ldots, w_{n-1k}(t))$ for the evaluation of the hierarchical approximation of the memory (see Section 3.1.1).

2. Choose the numerical scheme to discretize the integrals appearing in the definition of $w_{0k}(t), \ldots, w_{n-1k}(t)$ in order to ensure consistency with the ODE solver for the differential equations for the resolved variables. For our numerical example, we used the trapezoidal rule whose order is consistent with the modified Euler scheme used for the ODE solver (see Section 3.1.2 and discussion in Section 3.1.3). The extension to higher order schemes both for the integral discretization and the ODE solver is possible.

3. Let $\Delta t_{res}$ be the stepsize for the ODE solver and $t_0, t_1, \ldots, t_{n-1}$ the memory lengths. As discussed in Section 3.1.5, we expect the memory lengths to be longer than $\Delta t_{res}$. Let $\Delta t = \max(\Delta t_0, \ldots, \Delta t_{n-1})$ where $\Delta t_0, \ldots, \Delta t_{n-1}$ are the subinterval lengths for the decomposition of the memory integrals appearing in $w_{0k}(t), \ldots, w_{n-1k}(t)$ respectively. Enforce numerical consistency by using $\Delta t \approx \Delta t_{res}$ and then apply the discretization construction outlined in Step 2 (see also Section 3.1.3).

4. Estimate the optimal memory lengths $\hat{t}_0, \hat{t}_1, \ldots, \hat{t}_{n-1}$ by requiring that the reduced model reproduces the evolution of certain quantities e.g. various norms of the resolved variables, as computed from the full system (see Section 3.1.4).

3.2. Infinite memory and renormalization. By infinite memory we mean that the left end-point of the integral defining $w_{0k}(t)$ extends all the way to 0 i.e., $w_{0k}(t) = P \int_0^t e^{sL}PLE^{(t-s)QL}QLu_{0k}ds$. We want to present briefly this case because it sheds more light on the finite memory construction presented above.

We start by differentiating the expression for $w_{0k}(t)$ w.r.t. time and we obtain

$$\frac{dw_{0k}}{dt} = Pe^{tL}PLQLu_{0k} + w_{1k}(t)$$

where $w_{1k}(t) = P \int_0^t e^{sL}PLE^{(t-s)QL}QLu_{0k}ds$. Note the difference between the above equation and Eq. (7). In the equation above there is no longer the term
Recall that it was due to the need to evaluate this particular term that we had to resort to the discretization of the integral e.g. through the trapezoidal rule in order to find an expression for $P e^{(t-t_0)L} P e^{t_0 Q L} Q L u_{0k}$ in terms of known quantities. And due to that discretization we may then had to resort to breaking up the integral into a sum of integrals of shorter length. All these complications disappear when the memory extends from the current time $t$ all the way back to time $0$. In this respect, if the memory does not have finite length $t_0$, the formulas become simpler. In fact, in this case the system of ODEs for the determination of $w_{0k}(t)$ becomes

$$\frac{dw_{0k}}{dt} = P e^{tL} P L Q L u_{0k} + w_{1k}(t)$$
$$\frac{dw_{1k}}{dt} = P e^{tL} P L Q L Q L u_{0k} + w_{2k}(t)$$
$$\ldots$$
$$\frac{dw_{(n-1)k}}{dt} = P e^{tL} P L (Q L)^{n-1} Q L u_{0k} + w_{nk}(t).$$

One still needs to compute the expressions for the first terms on the RHS of the equations but these do not involve the orthogonal dynamics operator. As a result, the rest of the construction involving the discretization of the integrals is not needed. The system of ODEs above (after assuming $w_{nk}(t)$ is negligible) can be solved to provide the memory term $w_{0k}(t)$ in the equation for the resolved variable $u_{0k}$. For the numerical example of Burgers where we do use the reformulation we only treat the case of finite memory. We have tried using the infinite memory construction of this section but the resulting reduced model was unstable.

The infinite memory construction clarifies conceptually the finite memory construction. As we have described already in Section 3.1.4, the length of the memory can be computed so that the reduced model reproduces certain features of the dynamics of the resolved variables. In this respect, the meaning of the memory length, say $t_0$, is akin to that of a renormalization parameter [22]. Thus, the finite memory Markovian reformulation can be viewed as a hierarchical way to renormalize the MZ formalism and incorporate dynamic information from the evolution of the full system. We insist on using the adjective hierarchical and not perturbative since we have not identified a small parameter. According to our discussion in Section 3.1.5 we expect our construction to be better suited for problems where the orthogonal dynamics operator evolves on a timescale that is comparable (if not longer) to the timescale of the resolved variables. In this case, the rate of decay of the memory kernel should be small and one can think of this as an effective small parameter.

4. Numerical examples. We present 3 numerical examples of the application of the MZ formalism in the construction of reduced order models for subsets of polynomial chaos coefficients for time-dependent systems. The first two examples (a linear ODE with an uncertain coefficient and a nonlinearly damped and randomly forced particle) involve the direct application of the MZ formalism while the third example (the 1D viscous Burgers equation with uncertain initial condition) contains an application of the Markovian reformulation of MZ. Because the first two examples are for the direct application we have collected in Section 4.1 the main construction needed to evaluate the memory kernels.
4.1. Direct application of the Mori-Zwanzig formalism. Before we present the numerical examples for the direct application of MZ we provide some details about the calculation of the memory term (see [4] for more details).

We define the inner product

\[(f, g) = \int fg d\omega.\]  

where \(\omega\) is the joint probability measure with respect to the initial conditions for all the variables in the full system. This can be any measure we choose and for our numerical examples (Sections 4.1.1 and 4.1.2) we have chosen it to be a Gaussian for a very specific purpose which we now explain. The projection operator \(P\) we have used sets the initial conditions for the unresolved variables equal to 0. In order to compute the memory kernel we utilize a finite-rank projection operator \(P\) which approximates \(P\). Since the original operator \(P\) sets the initial conditions for the unresolved variables equal to 0, we have chosen \(P\) to be defined through a measure \(\omega\) which assigns a Gaussian distribution of small variance for the initial condition of all the variables in the system. The mean of this Gaussian distribution for each variable is equal to the initial value of this variable (see e.g. Eq. (31)). In other words, the choice of the measure \(\omega\) does not have to respect any invariance for the system dynamics. If one has access to such an invariant measure, it can be used for the definition of a finite-rank projection operator (see e.g. [4]).

For a function \(\varphi_j(u_0, t)\) of the initial conditions and time, the finite-rank projection reads (see e.g. [4])

\[(P\varphi_j)(\hat{u}_0, t) = \sum_{\nu \in I} (\varphi_j(u_0, t), h^\nu(\hat{u}_0)) h^\nu(\hat{u}_0),\]  

where \(h^\nu(\hat{u}_0)\) are tensor product Hermite polynomials up to some order \(p\), \(\nu\) is the multi-index \(\nu = (\nu_0, \ldots, \nu_\Lambda)\) with \(|\nu| = \sum_{i=0}^\Lambda \nu_i\) and \(I\) is the index set up to order \(p\), i.e., \(I = \{\mu | |\mu| \leq p\}\). For the numerical example in Section 4.1.1 we set the order \(p\) for the basis functions to 5 for a total of 21 basis functions. For the numerical example in Section 4.1.2 we set the order \(p\) for the basis functions to 3 for a total of 10 basis functions.

For each \(j \leq \Lambda\), the component \(F_j(u_0, t)\) denotes the solution of the orthogonal dynamics

\[\frac{\partial}{\partial t} F_j(u_0, t) = Q L F_j(u_0, t) = L F_j(u_0, t) - P L F_j(u_0, t),\]  

\[F_j(u_0, 0) = Q L u_0j = R_j(u_0) - P L u_0j.\]  

(23) is equivalent to the Dyson formula:

\[F_j(u_0, t) = e^{tL} F_j(u_0, 0) - \int_0^t e^{(t-s)L} P L F_j(u_0, s) ds.\]  

(24)

Eq. (24) is a Volterra integral equation for \(F_j(u_0, t)\). To proceed, we replace the projection operator \(P\) with the finite-rank projection operator \(P\) and find

\[K_j(\hat{u}_0, s) = P L F_j(u_0, s) \approx P L F_j(u_0, s) = \sum_{\nu \in I} a^\nu_j h^\nu(\hat{u}_0),\]  

where

\[a^\nu_j(s) = (L F_j(u_0, s), h^\nu(\hat{u}_0)).\]
Consequently,
\[ e^{(t-s)\mathcal{L}}P\mathcal{L}F_j(u_0, s) = \sum_{\nu \in I} a^\nu_j(s)h^\nu(\varphi(u, t - s)). \]

We substitute \( e^{(t-s)\mathcal{L}}P\mathcal{L}F_j(u_0, s) \) for \( e^{(t-s)\mathcal{L}}P\mathcal{L}F_j(u_0, s) \) in Eq. (24), multiply both sides by \( \mathcal{L} \) and take the inner product with \( h^\mu(\hat{u}_0) \); the result is (dropping the approximation sign)
\[
(\mathcal{L}F_j(u_0, t), h^\mu(\hat{u}_0)) = (\mathcal{L}e^{t\mathcal{L}}F_j(u_0, 0), h^\mu(\hat{u}_0)) - \int_0^t \sum_{\nu \in I} a^\nu_j(s)(\mathcal{L}e^{(t-s)\mathcal{L}}h^\nu(\hat{u}_0), h^\mu(\hat{u}_0))ds. \tag{26}
\]

Eq. (26) is a Volterra integral equation for the function \( a^\nu_j(t) \), which can be rewritten as follows:
\[
a^\nu_j(t) = f^\nu_j(t) - \int_0^t \sum_{\nu \in I} a^\nu_j(s)g^{\nu\mu}(t - s)ds, \tag{27}
\]
where
\[
f^\nu_j(t) = (Le^{t\mathcal{L}}F_j(u_0, 0), h^\mu(\hat{u}_0)), \quad g^{\nu\mu}(t) = (Le^{t\mathcal{L}}h^\nu(\hat{u}_0), h^\mu(\hat{u}_0)).
\]

The functions \( f^\nu_j(t) \), \( g^{\nu\mu}(t) \) can be found by averaging over a collection of experiments or simulations, with initial conditions drawn from the initial distribution. In this example, we use a sparse grid quadrature rule for the multi-dimensional integrals [27]. However, we need to clarify how the expressions \( Le^{t\mathcal{L}}F_j(u_0, 0) \) and \( Le^{t\mathcal{L}}h^\nu(\hat{u}_0) \) that appear in them can be estimated. Note that both of these expressions involve an application of the operator \( \mathcal{L} \) which differentiates w.r.t. to the initial conditions to expressions, \( e^{t\mathcal{L}}F_j(u_0, 0) \) and \( e^{t\mathcal{L}}h^\nu(\hat{u}_0) \) which depend on the solution at time \( t \). Thus, they are unknown functions of the initial conditions. To proceed with the differentiation we use the formula (see [3])
\[
LB(u(u_0, t)) = \sum_{r=0}^M R_r(u_0) \frac{\partial}{\partial u_0} B(u(u_0, t)) = \sum_{r=0}^M R_r(u(u_0, t)) (\frac{\partial B}{\partial u_0}(u(u_0, t)), \tag{28}
\]
which holds for any function \( B(u(u_0, t)) \) of the solution at time \( t \) and where \( u(u_0, t) = e^{t\mathcal{L}}u_0 \). For example, for \( Le^{t\mathcal{L}}F_j(u_0, 0) \) we find
\[
Le^{t\mathcal{L}}F_j(u_0, 0) = \sum_{r=0}^M R_r(u(u_0, t)) \left( \frac{\partial F_j}{\partial u_0} \right)(u(u_0, t)).
\]

Finally, we perform one more projection to eliminate the noise term (see Section 2) and the memory term becomes
\[
\int_0^t P e^{(t-s)\mathcal{L}}K_j(\hat{u}_0, s)ds.
\]

This can be approximated by
\[
\int_0^t \sum_{\nu, \mu \in I} a^\nu_j(s)\gamma^{\nu\mu}(t - s)h^\mu(\hat{u}_0)ds,
\]
where
\[
\gamma^{\nu\mu}(t) = (e^{t\mathcal{L}}h^\nu(\hat{u}_0), h^\mu(\hat{u}_0)).
\]
After calculating $a_i^\mu$ and $\gamma^{\mu\nu}$ we obtain the following reduced system,
\begin{equation}
\frac{d}{dt}\hat{u}(t) = R(\hat{u}(t)) + \int_0^t A(s)\Gamma(t-s)h(\hat{u}_0)ds. \quad \hat{u}(0) = \hat{u}_0,
\end{equation}
here $A$ and $\Gamma$ are the matrix form of $a_i^\mu$ and $\gamma^{\mu\nu}$, and $\hat{u}_0$ is the initial condition of the resolved variables.

4.1. A linear ODE with uncertain coefficient. Consider the following linear ordinary equation with an uncertain coefficient
\begin{equation}
\frac{du}{dt} = -\kappa u, \quad u(0, \cdot) = u^o,
\end{equation}
where $\kappa \sim U[0,1]$. This equation has the solution $u = u^o\exp(-\kappa t)$. To represent the dependence of the solution of (30) on $\kappa$, we can expand it in a general polynomial chaos (gPC) expansion [27], say using Legendre polynomials. Let
\begin{equation}
u(t, \cdot) \approx \sum_{i=0}^M u_i(t)\phi_i(\xi),
\end{equation}
where $\phi_i(\xi)\sim U[-1, 1]$ and $\{\phi_i\}$ are normalized Legendre polynomials which are orthonormal with respect to the uniform distribution of $\xi$, i.e.,
\begin{equation}
\int_{-1}^1 \phi_i(\xi)\phi_j(\xi)\frac{1}{2}d\xi = \delta_{ij}.
\end{equation}
We can write $\kappa$ as $\kappa = \frac{1}{2}\xi + \frac{1}{2} = \sum_{i=0}^1 k_i\phi_i(\xi)$. We substitute this expansion in (30) and obtain (through Galerkin projection) the (truncated) system up to order $M$
\begin{equation}
\frac{du_r}{dt} = -\sum_{i=0}^M \sum_{j=0}^M k_iu_j e_{ijr}, \quad u_r(0) = u_{0r}, \quad r = 0, \ldots, M,
\end{equation}
where $e_{ijk} = \int_{-1}^1 \phi_i(\xi)\phi_j(\xi)\phi_k(\xi)\frac{1}{2}d\xi$ and $u_{00} = u^o$, $u_{0r} = 0$ for $r = 1, \ldots, M$ (for details, see e.g [28]). For the numerical results presented here we have chosen $u^o = 1$.

Because of the spectral decay in the gPC coefficients, it is natural to choose the coefficients $u_i$, $i = 0, \ldots, \Lambda$ of the lower degree Legendre polynomials to be the resolved variables $\hat{u}$, and $u_i$, $i = \Lambda + 1, \ldots, M$ to be the unresolved variables $\tilde{u}$ respectively. To conform with the notation in Section 2, we have $H = \{0, \ldots, \Lambda\}$ and $G = \{\Lambda + 1, \ldots, M\}$. We have chosen $M = 6$ for the full system and $\Lambda = 1$ for the reduced system. The solution of the full system is converged for $M = 6$ and thus, we do not need to keep further terms in the expansion.

As we have explained at the end of Section 2, the projection $P$ we have chosen is defined as $(Pf)(\hat{u}_0) = P(f(u_0)) = P(f(\hat{u}_0, \tilde{u}_0)) = f(\hat{u}_0, 0)$. Also, we define $Q = I - P$. To be consistent with the notation in Section 2, we have
\begin{equation}
R_r(u_0) = -\sum_{i=0}^M \sum_{j=0}^M k_iu_j e_{ijr}.
\end{equation}
\begin{equation}
PLu_{0r} = -\sum_{i=0}^\Lambda \sum_{j=0}^\Lambda k_iu_{0j} e_{ijr}.
\end{equation}
In order to be able to compute the expressions for the memory terms we use a finite-rank projection $\mathbb{P}$ to approximate the projection $P$. As explained in Section 4.1, to define the finite-rank projection we need to introduce a measure for the distribution of the initial conditions of the coefficients $u_{0r}$. We consider the coefficients $u_{0r}$ to be i.i.d Gaussian random variables with mean at the values given initially (see (31)) and a prescribed variance for $i = 0, \ldots, M$. In the case of the linear ODE, the variance was set to 0.01 for all the coefficients $u_{0r}$ in the full system. As explained in Section 4.1, we use the finite-rank projection onto the function space expanded by Hermite polynomials (up to order 5) to represent the orthogonal dynamics (total of 21 functions) and solve the Volterra equation for the memory kernels.

Fig. 1 shows the evolution of the memory kernel $(Le^{tQ}QLu_1, h_{01})$ which is indicative of the behavior of the memory kernels. The basis function $h_{01}$ is the product of the zero order Hermite polynomial in the variable $u_0$ and the first order Hermite polynomial in the variable $u_1$. We see that the memory kernel is rather slowly decaying which means that the resulting reduced order model will have a long memory. Fig. 2 shows the solution for the resolved variables as predicted by the full system and two different reduced order models, the Markovian model which results from dropping the memory term in (29) and the non-Markovian reduced model given by (29). It is obvious from Fig. 2 that the Markovian model loses accuracy quickly. On the other hand, the non-Markovian model retains its accuracy for the length of the simulation interval. This difference in behavior is quantified in Fig. 3 where we see that for both resolved variables the relative error of the Markovian model becomes greater than 50% by the end of the simulation interval. On the other hand, the error of the non-Markovian model remains less than 1% for the whole simulation interval.

4.1.2. Nonlinearly damped and randomly forced particle. Consider the following equation describing a particle moving in a double well potential and driven by a force term (see [27])

\[
\frac{du}{dt} = u - u^3 + f(t, \xi),
\]

\[
u(0) = u^0,
\]

where $f = \sin(t + t_0)\xi$ and $\xi \sim U[-1, 1]$. We use order $M = 6$ polynomials in $\xi$ to approximate the full system solution up to time 10. We want to construct a reduced model for the first 2 coefficients of the polynomial expansion ($\Lambda = 1$). As before, we let $u(t, \xi) \approx \sum_{i=0}^{M} u_i(t)\phi_i(\xi)$ and we obtain through Galerkin projection the system

\[
\frac{du_i}{dt} = u_i - \sum_{j,k,m=0}^{M} u_j u_k u_m e_{jkm} + f_i,
\]

\[
u_i(0) = u_{0i}, \quad \text{for } i = 0, \ldots, M.
\]

where $e_{jkm} = \int \phi_j(\xi)\phi_k(\xi)\phi_m(\xi)\phi_i(\xi)\frac{1}{2}d\xi$ and $u_{0i} = \{ u^0, \quad i = 0; 0, \quad \text{otherwise} \}$. For the numerical results presented here we choose $u^0 = 1$. 
Let $R$ be the vector with $R_i = u_i - \sum_{j,k,m=0}^M u_j u_k u_m c_{jkm} + f_i$. In order to apply the MZ formalism we need an autonomous system of equations to begin with. For this purpose, we introduce an auxiliary time-variable $\tau$, such that $\tau = t$ and $\frac{d\tau}{dt} = 1$. The projection operator $P$ projects onto the function space of the first two coefficients and $\tau$. Again, we use the finite-rank projection onto the function space expanded by Hermite polynomials up to order 3 to represent the orthogonal dynamics (total of 10 functions) and solve the Volterra equation for the memory.
Figure 3. Linear ODE: Relative error with respect to the true solution for the (Markovian) reduced model without memory (blue line) and the (non-Markovian) reduced model with memory (red line).

kernels as we did for the linear ODE example. The variance for the Gaussian variables used to define the inner product for the finite-rank projection was set to $10^{-2i-2}$ for the coefficient $u_i$ with $i = 0, 1, \ldots, 6$. The reason we used a decreasing sequence of variances as we go up in the order of Legendre polynomials is to stabilize the behavior of the reduced model.

As can be seen from Fig. 5, the difference between the (memoryless) Markovian and non-Markovian reduced models is even more pronounced than in the case of the linear ODE. The inclusion of the memory term is indeed crucial for maintaining the accuracy of the reduced model for long times. For the case of the resolved variable $u_1$, the relative error spikes at a couple of points even for the otherwise very accurate non-Markovian reduced model. As can be seen from Fig. 4, this is because the exact value of $u_1$ becomes zero at these points so that the relative error becomes very large even for an accurate approximation. However, the significant improvement in accuracy with the inclusion of the memory term is evident in Fig. 5 which plots the error in a logarithmic scale.

4.2. Markovian reformulation of the Mori-Zwanzig formalism. We present in this subsection an example where the direct application of the MZ formalism is too expensive. In particular, the number of basis functions needed to obtain an accurate representation of the memory integrand is very large. We choose to apply the Markovian reformulation which allows us to construct an approximation to the memory that is computationally manageable. We only use the first term in the Markovian hierarchy presented earlier in Section 3. As a result we do not obtain an exact representation of the memory. However, the approximation of the memory results in more accurate predictions for the statistics than a reduced model that completely ignores the memory.

4.2.1. Viscous 1D Burgers with uncertain initial conditions. In this section we show how the above MZ formulation can be used for uncertainty quantification for the one-dimensional Burgers equation with uncertain initial conditions. As is explained later, the calculation of the MZ memory term cannot proceed as for the last two examples. The reason is that it is prohibitively expensive due to the number of
basis functions needed. Thus, we will apply the alternative construction that was presented in Section 3.

The equation is given by

\[ u_t + uu_x = \nu u_{xx}, \quad (34) \]

where \( \nu > 0 \). Equation (34) should be supplemented with an initial condition \( u(x,0) = u_0(x) \) and boundary conditions. We solve (34) in the interval \([0, 2\pi]\) with periodic boundary conditions. This allows us to expand the solution in Fourier series

\[ u_N(x,t) = \sum_{k \in \mathcal{F}} u_k(t)e^{ikx}, \]
where $F = [-N, N - 1]$. The equation of motion for the Fourier mode $u_k$ becomes

$$\frac{du_k}{dt} = -\frac{ik}{2} \sum_{p+q=k} u_p u_q - \nu k^2 u_k. \quad (35)$$

We assume that the initial condition $u_0(x)$ is uncertain (random) and can be expanded as $u_0(x, \xi) = (\alpha_0 + \alpha_1 \xi)v_0(x)$ where $\xi$ is uniformly distributed in $[-1, 1]$ and $v_0(x)$ a given function. In the numerical experiments we have taken $\alpha_0 = \alpha_1 = 1$ and $v_0(x) = \sin x$. Thus, the initial condition varies “uniformly” between the functions 0 and $2\sin x$.

To proceed we expand the solution $u_k(t, \xi)$ for $k \in F$ in a polynomial chaos expansion using the standard Legendre polynomials which are orthogonal in the interval $[-1, 1]$. In particular, we have that

$$\int_{-1}^{1} \phi_i(\xi) \phi_j(\xi) \frac{1}{2} d\xi = \frac{1}{2i + 1} \delta_{ij},$$

where $\phi_i(\xi)$ is the standard Legendre polynomial of order $i$. For each wavenumber $k$ we expand the solution $u_k(t, \xi)$ of (38) in Legendre polynomials and keep the first $M$ polynomials

$$u_k(t, \xi) \approx \sum_{i=0}^{M-1} u_{ki}(t) \phi_i(\xi), \quad \text{where} \quad \xi \sim U[-1, 1]. \quad (36)$$

Similarly, the initial condition can be written as $u_0(x, \xi) = \sin x \sum_{i=0}^{1} \alpha_i \phi_i(\xi)$ since $\phi_0(\xi) = 1$ and $\phi_1(\xi) = \xi$. Substitution of (36) in (35) and use of the orthogonality property of the Legendre polynomials gives

$$\frac{du_{kr}(t)}{dt} = -\frac{ik}{2} \sum_{l=0}^{M-1} \sum_{m=0}^{M-1} \sum_{p+q=k}^{F} u_{pl} u_{qm} c_{lmr} - \nu k^2 u_{kr} \quad (37)$$

for $k \in F$ and $r = 0, \ldots, M - 1$. Also

$$c_{lmr} = \frac{E[\phi_l(\xi) \phi_m(\xi) \phi_r(\xi) \delta_{ij}]}{E[\phi_i^2(\xi)]},$$

where the expectation $E[\cdot]$ is taken with respect to the uniform density on $[-1, 1]$. The Legendre polynomial triple product integral defines a tensor which has the following sparsity pattern: $E[\phi_l(\xi) \phi_m(\xi) \phi_r(\xi)] = 0$, if $l + m < r$ or $l + r < m$ or $m + r < l$ or $l + m + r = \text{odd} [10]$. Due to this sparsity pattern, for a given value of $M$ only about 1/4 of the $M^3$ tensor entries are different from zero.

The cost of direct application of Mori-Zwanzig. Before we proceed we have to comment on the cost of applying the MZ formalism to construct a reduced model. We have set the viscosity coefficient to $\nu = 0.03$. The solution of the full system was computed with $N = 196$ Fourier modes ($F = [-98, 97]$) and the first 7 Legendre polynomials ($M = 7$). The first 7 Legendre polynomials were enough to obtain converged statistics for the full system. We want to construct reduced models for the evolution of the coefficients of the first 2 Legendre polynomials i.e., $u_{k0}$, $u_{k1}$ for $k \in F$. If we want to apply the MZ formalism in the way we did for the previous two examples (employing a finite-rank projection etc.) we would need to construct a basis in $2 \times 98$ dimensions (exploiting the fact that the solution of the Burgers equation is real-valued). Any attempt to use basis functions up to a high order is
infeasible for such a high-dimensional situation. We have attempted to use only low
order basis functions but they are not enough to guarantee accuracy of the reduced
model. Thus, we turn to the reformulated reduced model that was presented in
Section 3.

Reformulated MZ reduced model. To conform with the Mori-Zwanzig formalism we
set
\[
R_{kr}(u) = -\frac{ik}{2} \sum_{l=0}^{M-1} \sum_{m=0}^{M-1} \sum_{p+q=k, p, q \in F} u_pl u_qm c_{l mr} - \nu k^2 u_{kr},
\]
where \( u = \{u_{kr}\} \) for \( k \in F \) and \( r = 0, \ldots, M - 1 \). Thus, we have
\[
\frac{du_{kr}}{dt} = R_{kr}(u)
\]  
(38)
for \( k \in F \) and \( r = 0, \ldots, M - 1 \). We proceed by dividing the variables in resolved
and unresolved. In particular, we consider as resolved the variables \( \hat{u} = \{u_{kr}\} \) for
\( k \in F \) and \( r = 0, \ldots, \Lambda - 1 \), where \( \Lambda < M \). Similarly, the unresolved variables are
\( \tilde{u} = \{u_{kr}\} \) for \( k \in F \) and \( r = \Lambda, \ldots, M - 1 \). In the notation of Section 2 we have
\( H = F \cup (0, \ldots, \Lambda - 1) \) and \( G = F \cup (\Lambda, \ldots, M - 1) \). In other words, we resolve,
for all the Fourier modes, only the first \( \Lambda \) of the Legendre expansion coefficients and
we shall construct a reduced model for them.

The system (38) is supplemented by the initial condition \( u_0 = (\hat{u}_0, \tilde{u}_0) \). We
focus on initial conditions where the unresolved Fourier modes are set to zero, i.e.
\( u_0 = (\hat{u}_0, 0) \). We also define \( L \) by
\[
L = \sum_{k \in F} \sum_{r = 0}^{M - 1} R_{kr}(u_0) \frac{\partial}{\partial u_{kr}}.
\]
To construct a MZ reduced model we need to define a projection operator \( P \). For
a function \( h(u_0) \) of all the variables, the projection operator we will use is defined
by \( P(h(u)) = P(h(\hat{u}_0, \tilde{u}_0)) = h(\hat{u}_0, 0) \), i.e. it replaces the value of the unresolved
variables \( \tilde{u}_0 \) in any function \( h(u_0) \) by zero. Note that this choice of projection is
consistent with the initial conditions we have chosen. Also, we define the Markovian
term
\[
PLu_{0k} = PR_k(u_0) = -\frac{ik}{2} \sum_{l=0}^{\Lambda-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k, p, q \in F} u_{0 pl} u_{0qm} c_{l mr} - \nu k^2 u_{0kr}.
\]
The Markovian term has the same functional form as the RHS of the full system
but is restricted to a sum over only the first \( \Lambda \) Legendre expansion coefficients for
each Fourier mode.

For the the term \( PLQLu_{0kr} \) we find
\[
PLQLu_{0kr} = 2 \times \left[ -\frac{ik}{2} \sum_{l=\Lambda}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k, p, q \in F} PLu_{0 pl} u_{0qm} c_{l mr} \right].
\]  
(39)
To see this, first we compute the expression for \( QLu_{0kr} \).
\[
QLu_{0kr} = Lu_{0kr} - PLu_{0kr}
\]
\[ L_{u0kr} = -\frac{ik}{2} \sum_{\ell=0}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k} \sum_{p,q \in F} u_{0pl} u_{0qm} c_{lmr} - \nu k^2 u_{0kr} \]

We have

\[ PL_{u0kr} = -\frac{ik}{2} \sum_{\ell=0}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k} \sum_{p,q \in F} u_{0pl} u_{0qm} c_{lmr} - \nu k^2 u_{0kr} \]

Also, we have

\[ PL_{u0kr} = -\frac{ik}{2} \sum_{\ell=0}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k} \sum_{p,q \in F} u_{0pl} u_{0qm} c_{lmr} - \nu k^2 u_{0kr} \]

We combine and find

\[ QL_{u0kr} = L_{u0kr} - PL_{u0kr} \]

\[ = 2(\frac{-ik}{2}) \sum_{\ell=0}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k} \sum_{p,q \in F} u_{0pl} u_{0qm} c_{lmr} \]

\[ - \frac{ik}{2} \sum_{\ell=0}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k} \sum_{p,q \in F} u_{0pl} u_{0qm} c_{lmr} \]

Then we compute the expression for \( LQL_{u0kr} \). We find

\[ LQL_{u0kr} = \]

\[ 2(\frac{-ik}{2}) \sum_{\ell=0}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k} \sum_{p,q \in F} L_{u0pl} u_{0qm} c_{lmr} + 2(\frac{-ik}{2}) \sum_{\ell=0}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k} \sum_{p,q \in F} u_{0pl} L_{u0qm} c_{lmr} \]

\[ - \frac{ik}{2} \sum_{\ell=0}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k} \sum_{p,q \in F} L_{u0pl} u_{0qm} c_{lmr} - \frac{ik}{2} \sum_{\ell=0}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p+q=k} \sum_{p,q \in F} u_{0pl} L_{u0qm} c_{lmr} \]

To complete the derivation of \( PLQL_{u0kr} \), we have to apply \( P \) to \( LQL_{u0kr} \). Because the projection we have chosen sets to 0 the initial condition for the unresolved variables, the last 3 terms in the expression for \( LQL_{u0kr} \) are zero and we obtain Eq. (39).

Finally, to implement any method to solve equation (17) for the estimation of \( t_0 \) we need to specify the RHS of the equation (17). This requires the evaluation of
the expression $Pe^{tL}QLu_{0kr}$. For the case of the viscous Burgers equation, we find

$$Pe^{tL}QLu_{0kr} = 2\left(-\frac{ik}{2}\right) \sum_{l=\Lambda}^{M-1} \sum_{m=0}^{\Lambda-1} \sum_{p,q=k}^{M-1} u_{pl} u_{qm} c_{lmr}$$

(40)

$$-\frac{ik}{2} \sum_{l=\Lambda}^{M-1} \sum_{m=\Lambda}^{M-1} \sum_{p,q=k}^{M-1} u_{pl} u_{qm} c_{lmr}.$$  

Note that since we restrict attention to initial conditions for which the unresolved variables are zero and the projection sets the unresolved variables to zero, the quantity $Pe^{tL}QLu_{0kr}$ can be computed through the evolution of the full system (38).

The full system was solved with the modified Euler method with $\delta t = 0.001$. The reduced model uses $N = 196$ Fourier modes but only the first two Legendre polynomials, so $\Lambda = 2$. It was solved using the modified Euler method with $\delta t = 0.001$. The parameter $t_0$ needed for the evolution of the memory term was found to be 0.3783 through the procedure described in Section 3.1.4.

Figure 6 shows the evolution of the mean energy of the solution

$$E[\hat{E}(t)] = \frac{1}{2} \sum_{k \in F} \sum_{r=0}^{1} 2\pi |u_{kr}(t)|^2 \frac{1}{2^r+1}$$

as computed from the full system (with $M = 7$ Legendre polynomials), the MZ reduced model with $\Lambda = 2$ without memory (keeping only the Markovian term) and the MZ reduced model with $\Lambda = 2$ with memory. Figure 7 shows the evolution of
the standard deviation of the energy of the solution. The variance of the energy is given by

$$Var[E(t)] = \frac{1}{4} \sum_{k_1, k_2 \in F} \sum_{r_1, \ldots, r_4 = 0}^{1} (2\pi)^2 u_{k_1 r_1} u_{k_2 r_2}^* u_{k_2 r_3} u_{k_2 r_4}^* d_{r_1 r_2 r_3 r_4} - \{E[E(t)]\}^2,$$

where

$$d_{r_1 r_2 r_3 r_4} = \int_{-1}^{1} \phi_{r_1}(\xi) \phi_{r_2}(\xi) \phi_{r_3}(\xi) \phi_{r_4}(\xi) \frac{1}{2} d\xi.$$

The reduced model performs equally well with or without memory. Of course, the reduced model with memory is slower than the reduced model without memory. However, the reduced model with memory is still about 4 times faster than the full system.

Figure 8 shows the evolution of the mean squared $l_2$ norm of the gradient of the solution

$$E[G(t)] = \sum_{k \in F} \sum_{r = 0}^{1} 2\pi k^2 |u_{kr}(t)|^2 \frac{1}{2r + 1}$$

as computed from the full system (with $M = 7$ Legendre polynomials), the MZ reduced model with $\Lambda = 1$ without memory (keeping only the Markovian term) and the MZ reduced model with $\Lambda = 1$ with memory. Figure 9 shows the evolution of the standard deviation. The variance is given by

$$Var[G(t)] = \sum_{k_1, k_2 \in F} \sum_{r_1, \ldots, r_4 = 0}^{1} (2\pi)^2 k_1^2 k_2^2 u_{k_1 r_1} u_{k_2 r_2}^* u_{k_2 r_3} u_{k_2 r_4}^* d_{r_1 r_2 r_3 r_4} - \{E[G(t)]\}^2.$$
Figure 8. Burgers equation: Evolution of the mean of the squared $l_2$ norm of the gradient of the solution calculated using only the first two Legendre polynomials.

Figure 9. Burgers equation: Evolution of the standard deviation of the squared $l_2$ norm of the gradient of the solution calculated using only the first two Legendre polynomials.
The large values of the standard deviation of the mean squared $l_2$ norm of the gradient are justified by the uncertainty in the initial condition. Recall that we have chosen an initial condition which can vary “uniformly” between the functions 0 and $2\sin x$. As a result, the standard deviation is large because it has to account for a wide range of possible initial conditions.

It is evident from the figures that the inclusion of the memory term improves the performance of the reduced model. Also, it is evident that there is room for improvement of the reduced model with memory. In particular, more terms are needed in the reformulated MZ model to approximate better the memory.

Recall that the solution of Burgers equation is a contraction \cite{14}. Eventually, the complete description of the uncertainty caused by the uncertainty in the initial condition requires only a few polynomial chaos expansion coefficients. This happens at a time scale that is dictated by the magnitude of the viscosity coefficient. That is why for long times the reduced model with and without memory have comparable behavior to that of the full system. However, for short times, the inclusion of the memory term does make a difference because information from the higher chaos expansion coefficients is needed. The higher chaos expansion coefficients will have a more prolonged contribution for systems that possess unstable modes. In such cases, the inclusion of the memory term becomes imperative for short as well as long times. Results for such cases will be presented elsewhere.

5. **Discussion and future work.** We have examined the application of the Mori-Zwanzig formalism to the problem of constructing reduced models for uncertainty quantification. In particular, we have constructed reduced models for subsets of the polynomial chaos expansion coefficients needed to describe fully the uncertainty. We have examined cases of parametric or initial condition uncertainty. The main conclusion from the current work is that while the MZ formalism can be applied for the construction of reduced models, the task of constructing an efficient (or even feasible) reduced model can be involved. For cases where the straightforward application of the MZ formalism is not possible, we have offered an alternative construction. The implementation of this alternative construction is reminiscent of renormalization constructions used to describe the evolution of complex solutions of PDEs \cite{22}.

The current work aims at a proof of concept of the applicability of the Mori-Zwanzig formalism for the construction of reduced order models for subsets of the necessary polynomial chaos coefficients. We want to show that the construction of accurate reduced order models requires the inclusion of memory effects. For the examples that we have presented we also performed the UQ computations using a collocation method for the full system (see e.g. \cite{27}). Because the source of uncertainty is one-dimensional one can get the same accuracy as the reduced order model with relatively few collocation points. So, for these examples there is no computational advantage of constructing a reduced model compared to a collocation method. Our expectation is that the construction of reduced order models will become more competitive in the case of high-dimensional sources of uncertainty where the collocation method also becomes expensive even for the calculation of lower order moments.

There are several directions for future work. First, we should investigate whether there is a more economical way of choosing the basis functions for cases when the basis functions have many arguments (as was the case for the Burgers example). This is important because the calculation of the memory kernels through the finite-rank
projection is well defined and the solution of the corresponding Volterra equations can be performed with high accuracy. A related question is whether there is sparsity in the coefficients of the basis functions. It is plausible that even though in principle the number of basis functions to reach a specific order may be very large, many of them may not contribute to the representation. A related approach would be the use of machine learning algorithms to obtain a more efficient representation of the memory term. Finally, a related issue to be investigated is how to ensure the stability of the reduced model when the finite-rank projection is employed. For example, for the nonlinearly damped and forced particle case, we had to assign smaller variances for the higher coefficients to stabilize the reduced model. This procedure needs to be investigated and, if possible, automated.

A second interesting research direction has to do with the representation of the memory term when the finite-rank projection is not possible due to a prohibitively large number of basis functions. We have explored here an expansion of the memory term that involves, in essence, a Taylor expansion of the orthogonal dynamics operator. Such an expansion seems more plausible when the timescale of the orthogonal dynamics is \textit{slower} than that of the resolved variables. However, as we have already mentioned in Section 3.1.5, there is an alternative way of performing the expansion of the memory term that is more suited to the case when the orthogonal dynamics is \textit{faster} than the resolved variables. Such an expansion leads to a Taylor expansion of the whole memory term, not just the orthogonal dynamics operator. If the memory kernel becomes insignificant after a time interval \( t_0 \), then one can use the full system up to time \( t_0 \), estimate the Taylor expansion of the whole memory term around time \( t_0 \) and then switch to the \textit{reduced} model with the memory given by the Taylor expansion. We have started investigating this alternative formulation but it presents certain mathematical and numerical issues that we have not resolved successfully so far. We will continue studying this alternative memory representation and report the results elsewhere.

\textbf{Acknowledgments.} We want to thank D. Barajas-Solano, H. Lei and A. Tartakovsky for useful discussions and comments. We also want to thank the anonymous reviewers for their insightful questions which helped to clarify and strengthen our manuscript.

\textbf{REFERENCES}

[1] I. Babuska, F. Nobile and R. Tempone, \textit{A stochastic collocation method for elliptic partial differential equations with random input data}, \textit{SIAM J. Numer. Anal.}, 45 (2007), 1005–1034.
[2] D.A. Barajas-Solano and A.M. Tartakovsky, Probabilistic density function method for nonlinear dynamical systems driven by colored noise, \textit{Phys. Rev. E} 93 (2016).
[3] A.J. Chorin, O.H. Hald and R. Kupferman, Optimal prediction and the Mori-Zwanzig representation of irreversible processes, \textit{Proc. Nat. Acad. Sci. USA} 97 (2000), 2968–2973.
[4] A.J. Chorin, O.H. Hald and R. Kupferman, Optimal prediction with memory, \textit{Physica D} 166 (2002), 239–257.
[5] A.J. Chorin and P. Stinis, Problem reduction, renormalization and memory, \textit{Comm. App. Math. Comp. Sci.} 1 (2005), 1–27.
[6] A. Doostan A and H. Owhadi, A non-adapted sparse approximation of PDEs with stochastic inputs, \textit{J. Comput. Phys.}, 230 (2011), 3015–3034.
[7] J. Foo and G.E. Karniadakis, Multi-element probabilistic collocation method in high dimensions, \textit{J. Comput. Phys.}, 229 (2010), 1536–1557.
[8] R. Ghanem and P.D. Spanos, \textit{Stochastic finite elements: a spectral approach}, Springer-Verlag, 1998.
[9] D. Given, R. Kupferman and A. Stuart, Extracting macroscopic dynamics: model problems and algorithms, *Nonlinearity* **17** (2004), R55–R127.

[10] M. Gupta and S.G. Narasimhan, Legendre polynomials Triple Product Integral and lower-degree approximation of polynomials using Chebyshev polynomials, Technical Report - CMU-RI-TR-07-22, Carnegie Mellon (2007).

[11] O.H. Hald and P. Stinis, Optimal prediction and the rate of decay for solutions of the Euler equations in two and three dimensions, *Proc. Nat. Acad. Sci. USA* **104** (2007), 6527–6532.

[12] T.Y. Hou, W. Luo, B. Rozovskii and H.M. Zhou, Wiener chaos expansions and numerical solutions of randomly forced equations of fluid mechanics, *J. Comput. Phys.* **216** (2006) 687–706.

[13] J.D. Jakeman and S.G. Roberts, Local and dimension adaptive stochastic collocation for uncertainty quantification, in *Sparse grids and applications*, Springer (2013), 181–203.

[14] P.D. Lax, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, SIAM Publications, Philadelphia, 1972.

[15] G. Leonenko and T. Phillips, On the solution of the Fokker-Planck equation using a high-order reduced basis approximation, *Comput. Methods Appl. Mech. Engrg.*, **199** (2009), 158–168.

[16] J. Li and P. Stinis, A unified framework for mesh refinement in random and physical space, *J. Comp. Phys.* **323** (2016), 243–264.

[17] X. Ma and N. Zabaras, An adaptive hierarchical sparse grid collocation method for the solution of stochastic differential equations, *J. Comput. Phys.*, **228** (2009), 3084–3113.

[18] L. Mathelin and M.Y. Hussaini, A stochastic collocation algorithm for uncertainty analysis, Technical Report NASA/CR-2003-212153, NASA Langley Research Center, (2003).

[19] F. Nobile, R. Tempone and C.G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Numer. Anal.*, **46** (2008), 2309–2345.

[20] A. Nouy and O.P. Le Maître, Generalized spectral decomposition for stochastic nonlinear problems, *J. Comp. Phys.*, **228** (2009), 202–235.

[21] P. Stinis, A phase transition approach to detecting singularities of PDEs, *Comm. App. Math. Comp. Sci.*, **4** (2009), 217–239.

[22] P. Stinis, Renormalized reduced models for singular PDEs, *Comm. App. Math. Comp. Sci.*, **8** (2013), 39–66.

[23] P. Stinis, Renormalized Mori-Zwanzig reduced models for systems without scale separation, Proc. Roy. Soc. A **471** (2015).

[24] D. Venturi, A fully symmetric nonlinear biorthogonal decomposition theory for random fields, *Physica D*, **240** (2011), 415–425.

[25] D. Venturi, H. Cho and G.E. Karniadakis, The Mori-Zwanzig approach to uncertainty quantification, in *Handbook for Uncertainty Quantification*, Springer (2016), 1–36.

[26] X. Wan and G.E. Karniadakis, Multi-element generalized polynomial chaos for arbitrary probability measures, *SIAM J. Sci. Comput.*, **28** (2006), 901–928.

[27] D. Xiu and J.S. Hesthaven, High-Order Collocation Methods for Differential Equations with Random Inputs, *SIAM J. Sci. Comput.*, **27** (2005), 1118–1139.

[28] D. Xiu and G.E. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, **24** (2002), 619–644.