UK Atomic
Energy
Authority

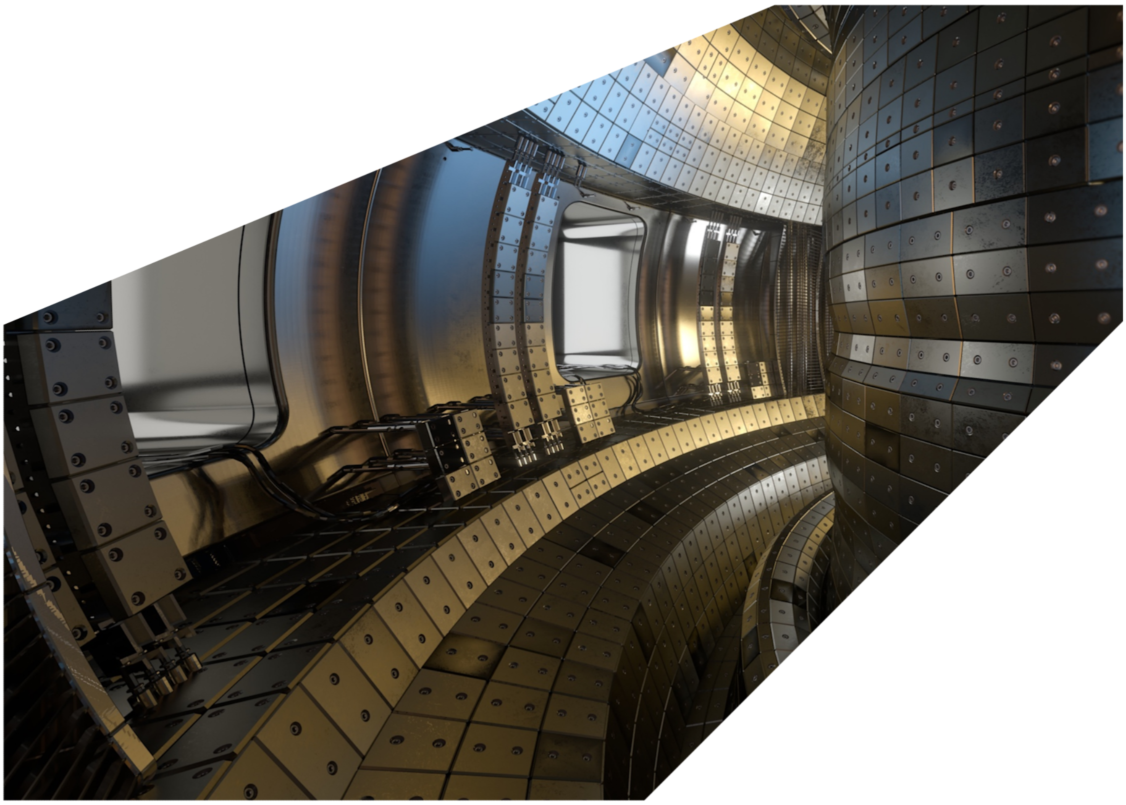
ExCALIBUR

Select techniques for MOR (Model Order Reduction)

M2.5.1

Abstract
The report describes work for ExCALIBUR project NEPTUNE at Milestone 2.5.1. This report includes an overview of three important classes of Model Order Reduction (MOR) approach: reduced basis methods, proper orthogonal decomposition, and proper generalized decomposition. Also included is a discussion of techniques for data assimilation, taken from the context of numerical weather prediction, and how these techniques overlap with MOR. Of particular interest for NEPTUNE are methods capable of addressing nonlinear problems and so some attention is drawn to the treatment of nonlinear effects in numerical weather prediction.

The report contains also some initial suggestion of how the techniques described might be applied in the initial proxyapp stage of NEPTUNE; however, a fuller discussion and more detailed conclusions have to be deferred until the outputs of the defrayed MOR grant have been received.



UKAEA REFERENCE AND APPROVAL SHEET

	Client Reference:	
	UKAEA Reference:	CD/EXCALIBUR-FMS/0031
	Issue:	1.00
	Date:	15 March 2021
Project Name: ExCALIBUR Fusion Modelling System		

	Name and Department	Signature	Date
Prepared By:	Ed Threlfall Wayne Arter BD	N/A N/A	15 March 2021 15 March 2021
Reviewed By:	Rob Akers Advanced Computing Dept. Manager		15 March 2021

1 Introduction

Modern computational models in physics and engineering achieve a high level of complexity, with numbers of degrees of freedom (alternately ‘state-space dimension’) frequently of order a billion, or more. Despite this, many problems of interest are still out of reach for direct methods, either due to the sheer number of degrees of freedom, or because the problem involves widely-varying time or space lengthscales, as occurs not infrequently in multi-physics scenarios where there are unresolved scales of variation. Problems of both these types are associated with the NEPTUNE use case of magnetically-confined fusion, in which a typical degree of freedom count is 10^{21}.

In order to surmount the problem of an unmanageably-large state-space, it might be asked whether some of the degrees of freedom, or certain *combinations* thereof, can be taken to be in some sense more important than others, with the ‘others’ being ignorable. Such an approach of retaining only the dominant behaviour of a system is termed *model order reduction* (MOR), with the resulting model being referred to as a *reduced-order model* (ROM) (synonyms are a *surrogate model* or an *emulator*). These may be hundreds to thousands of times faster in execution than the high-dimensional model they approximate. The utility of a much faster proxy is obvious for uncertainty quantification (UQ), calibration of models to large amounts of data, real-time control systems, and for allowing the solution of coupled, multiscale problems.

MOR is a fairly new field, though a rapidly-developing one since it stands on the shoulders of a large body of techniques from linear algebra and systems theory. It is an attempt to tackle (if not ‘solve’ in the exhaustive sense) today’s ‘intractable’ problems using advances in algorithm technology rather than hardware; working ‘smarter’, rather than necessarily ‘harder’. In an era where FLOPS per watt is a marketable figure of merit, the energy efficiency of such approaches might be considered a bonus, in addition to the elegance of avoiding doing unnecessary work. There is also the sheer convenience of being able to make progress using only light computing platforms, given that a powerful computer means a physically large computer, currently and for the foreseeable future.

The focus here is on algorithmic approaches to MOR, as opposed to the more traditional application of physicists’ or engineers’ domain knowledge. The goal is a systematic procedure to identify the dominant emergent degrees of freedom from potentially highly-nonlinear coupled systems. Such an algorithm works by reducing the genericity of an existing computational method e.g. a finite element framework, incorporating problem-specific constraints and boundary conditions; in this way it constrains the physics and determines approximate ‘modes’ dominating the system response. Thus, MOR works in tandem with generic discretization routines (the spectral / *hp* methods chosen for exploration in NEPTUNE are an example) rather than supplanting them: most ROM approaches tend to deal with the post-discretization system. Some analyses seek to reduce the size of the set of input parameters, and hence the burden associated with data assimilation - this is not specifically addressed here, though clearly UQ techniques can be used to identify the most ‘important’ input parameters in terms of their effect on the mean or variance of the output (and this UQ might itself make use of a ROM for Monte Carlo sampling).

Section 2 of this report is written with the intent of surveying the currently popular techniques for MOR: *reduced basis methods*, *proper orthogonal decomposition*, and *proper generalized decomposition*, though note that it is intended as an initial exposition in anticipation of further investigative work. The near-term future goal is to focus on computational models of the edge region of a magnetically-confined fusion plasma, which tends to be described using fluid dynamics equations (an example of state-space basis reduction in a fusion context, using Krylov subspace methods, can be found in [1]), though ultimately it may be useful to apply ROMs also to particle kinetics [2] and gyrokinetics [3]. It is worth noting that there are currently several open-source MOR toolkits and modules; a list can be found in [4]. The use of Gaussian processes in model order reduction is not included here since this work has been assigned to a third party, though the value of the Gaussian process approach in facilitating non-intrusive UQ is to be noted.

Another important area of model validation is the need to incorporate experimental measurements (e.g. in order to fit parameters in a model or in order to define an inverse problem for an initial condition). The process of including data into models is called *data assimilation* (DA). There are various analytic techniques for data assimilation, most of which start with the assumption of Gaussian errors. Here, ROMs find relevance in that there is frequently a need to integrate a complicated model (at least once) in order to propagate the information derived from experimental measurements and their corresponding uncertainties either forward or backward in time; an effective ROM can make this process a lot less cumbersome. There is also the fact that DA techniques can be applied to the Bayesian calibration of ROMs, for example, the ensemble Kalman filter can be used as a parameter estimation framework.

Because of this overlap, an overview of the basic DA methods is included in Section 3 of this report. This section is really structured as an exposition of analytic methods derived for the special case of linear models and Gaussian errors, generalized to treatments of real models, and some mention of the methods for incorporating known physics into the process is made. The canonical example of DA, and to a large extent the context of the discussion presented here, is *numerical weather prediction* (NWP); here, a large amount of observational data is available (in the form of wind velocity, pressure, humidity) and the system to be modeled, though based on well-understood microscopic physics, comprises vastly many degrees of freedom interacting nonlinearly over a vast range of lengthscales. It goes without saying that the field of meteorology has a great deal of experience with DA (and to a lesser extent, MOR, though the entire workings of NWP could be viewed as a MOR), with forecasts having been made on a daily basis using methods improved incrementally over many decades. The techniques here for extending the analytic Gaussian results to a real nonlinear system are of great interest because one aim of NEPTUNE is to use knowledge of plasma physics to improve in a similar fashion our understanding of the 'weather' inside a tokamak. Tokamak physics also contains noteworthy similarities to certain phenomena encountered in numerical weather prediction, for example multiple scales and turbulent flow.

An appendix gathers some useful facts concerning Gaussian statistics.

2 Techniques for Model Order Reduction

For illustrative purposes, a concrete definition of a typical target problem is given. In the further interest of clarity, a linear time-invariant system is considered (note that nonlinear models are typically time-updated using a linearization, though in that case the property of time-invariance is lost).

Consider an input $u(t) \in \mathbb{R}^m$ to a model with state defined by $x(t) \in \mathbb{R}^n$ and giving an output $y(t) \in \mathbb{R}^p$:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) && (1) \\ y(t) &= Cx(t). && (2) \end{aligned}$$

Here the number of model states, that is the dimension of the vector $x(t)$, given by n above, may be very large (and generally is much larger than the number of inputs m and the number of outputs p); in a modern application n might be taken to be of the order of a billion (with numbers such as this, it is feasible to store the vector of model states but not to store the $n \times n$ matrix A). This full model is a high-dimensional state space linear model (sometimes called the high-dimensional model in the context of MOR). Now MOR aproaches aim to replace the full internal state by a smaller state vector, of size $r \ll n$, defined by $x(t) = V\tilde{x}(t)$; then choosing another matrix defined by $W^T V = I$ the system is

$$\begin{aligned} \dot{\tilde{x}}(t) &= W^T AV\tilde{x}(t) + W^T Bu(t) && (3) \\ y(t) &= CV\tilde{x}(t). && (4) \end{aligned}$$

Here W and V are $n \times r$ matrices (of tall, thin shape). The task of MOR approaches is to find the projection matrices W, V such that the response of this reduced system mimics as closely as possible that of the full system.

The following discussions are necessarily brief and many important considerations, such as the stability of the ROMs, are not included. More detailed presentations of the material can be found in e.g. the review [5] and also in the documentation for the open-source Python MOR toolkit *pyMOR* [6].

2.1 Reduced basis methods

To focus on the case of reduction for parameterized PDEs, consider a family of PDEs defined by a single equation with a dependence on parameters μ belonging to a set $\mathcal{P}$ (this would represent the inputs to the model). To give a parabolic example, let the solution be labelled by $u_\mu(t)$; the semi-discrete problem is then, in weak form

$$\begin{aligned} u_\mu(0) &= u_0 && (5)\\ \langle v, \dot{u}_\mu(t)\rangle + b_\mu\left(v, u_\mu(t)\right) &= f(v) \;\; \forall v \in V_h && (6)\\ y_\mu(t) &= g(u_\mu(t)). && (7) \end{aligned}$$

This is the full-order model (note the first equation is simply the initial condition). States u_μ belong to a large-dimensional vector space V_h which spans the states available to the system post-discretization. Here the discretized spatial part of the PDE is contained in the matrix operator b_μ, which also contains dependence on the set of parameters (an example would be a spatially-varying diffusion constant).

To seek an approximate solution on a subspace $V_N \subset V_h$, a solution the same problem is sought for vectors in the reduced set and using an initial condition Galerkin projected onto this space. The general idea here is to construct iteratively a reduced space for the internal degrees of freedom, given a predefined set of parameters restricted to the region of interest in parameter space and referred to as the *training* parameter set. The choice of vector to add at a given stage in the process is the main problem, and this choice is motivated by the definition of the Kolmogorov N-width d_N:

$$d_N = \inf_{V_N \subseteq V, dim(V_N) \leq N} \sup_{\mu \in \mathcal{P}} \inf_{v \in V_N} \|u(\mu) - v\|. \qquad (8)$$

This quantity, which gives an upper bound for the possible quality of a reduced space i.e. the worst-case error, needs some explanation. For a particular solution $u(\mu)$ of the full-order model (known as a *snapshot*), the best approximation in the current reduced space is first computed (this is the inner infimum, which simply means identifying the reduced-space vector v that best approximates the snapshot), then the error in this is maximized (the supremum) over the training set of μ, giving the worst-case best approximation. Taking the infimum of this over all possible reduced spaces V_N gives the Kolmogorov N-width.

In practice, the inner infimum is dealt with by orthogonal projection of the snapshot $u(\mu)$ into V_N. Denoting the reduced basis vectors by v_i for $i = 1, ..., N$ the relevant coefficents (projections) λ_i can be found by solving

$$\sum_{j=1}^{N} \lambda_j \langle v_j, v_i\rangle = \langle v, v_i\rangle, \qquad (9)$$

which can also be expressed as the matrix equation $G\Lambda = R$; the matrix G is referred to as the Gramian. Note that a naive implementation would lead to the condition number of the Gramian getting increasingly large as new vectors are added into the reduced basis (because of the increasing amount of linear dependence); in order to avoid this issue, the reduced basis tends to be orthonormalized using the standard Gram-Schmidt procedure (reducing the Gramian to an identity matrix). Note the basis construction is an example of the *method of snapshots* due to the use of particular solutions of the full-order model.

Each new vector added to the reduced basis ought to give in some sense the maximum possible improvement in the approximation and this can be achieved by so-called ‘greedy’ methods: the aim is to find the parameter value μ^* that maximizes the error between the reduced model and the full-order model, then add the solution vector corresponding to this μ^* to the reduced basis (having first subtracted the part projected onto the existing reduced state space). This search over the (remaining) parameters in the training set forms the inner loop of the algorithm; clearly, using the full-order model here (giving the *strong greedy algorithm*) incurs a significant speed cost in constructing the ROM. The process can be streamlined by using the current ROM as a proxy for the full-order model in this inner loop (giving the *weak greedy algorithm*) and effectively increasing the size of the training sets (and thereby the dimension of the parameter spaces) that can be used.

A further simplification, in order to avoid needing to recompute the system matrices for each choice of parameters, involving repeated computation of the relevant bilinear forms using the full-order model, is to assume that the parameter dependence of the operators can be put into separable form (also known as an *affine decomposition*), viz.

$$b_\mu(v,u) = \sum_{q=1}^{Q} \theta_q(\mu)\; b_q(v,u). \tag{10}$$

This gives what is termed *online efficiency* - new parameter values can be tested without expensive recomputation. Clearly this will not work in the case of a nonlinear operator as the linearized operators are then state-dependent. One solution to this problem is *empirical interpolation*: the behaviour of the nonlinear operator is represented by a small number of pointwise evaluations with an (affine) interpolation to general parameter values. The technique can also be applied to problems where there is no affine decomposition of the operators.

2.2 Proper orthogonal decomposition (POD)

POD is another method for creating a reduced basis out of snapshot data and it is related to the familiar *Principal Components Analysis* (PCA) used in statistical analysis [7] and also to the *Karhunen-Loéve expansion* used in stochastic process modelling. In PCA, a vector of random variables $\underline{x}$ is subjected to a linear transformation to a new variable $\underline{y}$ such that the covariance matrix $\langle(\underline{y}-\bar{\underline{y}})(\underline{y}-\bar{\underline{y}})^T\rangle$ of the new variable is diagonal (this can always be done, since the covariances form a real symmetric matrix). The Karhunen-Loéve expansion is basically the continuous analogue of this. PCA-type analyses rely on the basic fact that a matrix acting on a general vector will give greater weight to components of the vector that lie in the subspace spanned by the eigenvectors of the matrix with the largest eigenvalues.

In POD the snapshot vectors are assembled into the *snapshot matrix* (this is a tall, thin matrix: the dimension of the full state is n and the number of samples n_s, with $n_s \ll n$)

$$A = \begin{pmatrix} \vdots & \vdots & \cdots & \vdots \\ u(\mu_1) & u(\mu_2) & \cdots & u(\mu_{n_s}) \\ \vdots & \vdots & \cdots & \vdots \end{pmatrix} \tag{11}$$

Then the *singular value decomposition* (SVD) of A is computed

$$A = U\Sigma V^T. \tag{12}$$

For clarity, here U is $n \times n_s$ and the columns of this matrix are the *left-singular vectors* of A; V is $n_s \times n_s$ and the columns are the *right-singular vectors* of A. The diagonal entries of the $n_s \times n_s$ matrix Σ are the *POD singular values* of A and are arranged in monotonically decreasing order (denoting the components of Σ as $\sigma_1 \geq \sigma_2 \, ... \geq \sigma_{n_s}$). The magnitude of each of these singular values (which can be clearly seen to generalize the eigenvalues for a square matrix) tells how important is each vector in the columns of U for the approximation of the snapshot data. The procedude is then to retain a given number of these vectors, and in the sense of minimizing the $L2$-norm, an optimal approximation space is obtained. It is worth noting that individual errors may be large, despite the least-squares optimality (this is different to the greedy methods applied in the preceding subsection).

The POD is an efficient lower-dimensional representation of the snapshot data in the sense that if the first r singular vectors are retained $r \leq n_s$ then the POD basis minimizes the last-squares error relative to the n_s-dimensional reduced space; and this least-squares error is represented by the sum of the squares of the remaining eigenvalues, a fact that can be used to choose an appropriate size of POD basis: one approach, referred to as the fractional 'energy' in the POD modes, is to choose r such that

$$\frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2} > \kappa \tag{13}$$

with κ being something approaching unity, e.g. 99.9%.

Note that this error assessment applies only to the error in the reconstruction of the snapshots and *not* the error relative to the full model. Error analysis with this wider scope is not possible in all cases - see for example [5], p.9.

There is also the issue of how best to choose the snapshot set (optimal snapshot selection). Uniform sampling is likely to be effective only for problems with a handful of parameters; for larger-dimensional parameter spaces, sparse grid sampling or greedy sampling may be used. Again, UQ analysis to exclude less important parameters and restriction to physically reasonable location in parameter space are useful.

Note that because the POD basis is constructed from sampled solutions, this method can be applied straightforwardly to linear and nonlinear systems.

2.3 Proper generalized decomposition (PGD)

Many MOR techniques (including the POD method outlined above) operate in a sequential fashion, constructing a reduced basis and then projecting the problem solution onto it. A potential issue here is that if the initial reduced basis is in some way unsuitable, the second step will yield a poor reduced model, and the only way to proceed is to repeat step one and construct a new basis. One way to address this problem is to seek a method that works incrementally i.e. the reduced basis is capable of being enriched with new states during the construction of the ROM.

In PGD the simplification is made that the solution to the problem is separable in time and in each (or some) of the space coordinates. Thus the solution to a parametric problem may be sought in the form

$$u^N(\underline{x}, t, \mu_1, ...\mu_M) \approx \sum_{i=1}^{N} X_i(\underline{x}) T_i(t) \prod_{m=1}^{M} P_i^m(p_m). \tag{14}$$

The important question now is the algorithm for constructing the separated functions. An enrichment scheme is used, with e.g.

$$u^n(\underline{x}, t) = u^{n-1}(\underline{x}, t) + X_n(\underline{x}) T_n(t) \tag{15}$$

being substituted into the equation and the nonlinear problem for the unknown functions $X_n(\underline{x})$ and $T_n(t)$ solved. The enrichment scheme itself is an iterative process that involves solving for the sequences $X_n^0, X_n^1, ..., X_n^p$ and $T_n^0, T_n^1, ..., T_n^p$, where e.g. X_n is taken to be X_n^p once a tolerance

$$\|X_n^p(\underline{x}) T_n^p(t) - X_n^{p-1}(\underline{x}) T_n^{p-1}(t)\| < \epsilon \tag{16}$$

is reached (here ||...|| is a suitable norm). The procedure involves an initial guess for all but one of the unknown functions in the product and then solving the equation for the remaining unknown (this is called an *alternating direction strategy*). The actual solution mechanism is typically a greedy algorithm applied to the weak formulation of the problem.

It is well worth noting that the extent to which the spatial dependence can be separated stems from the choice of spatial computational mesh: meshes with symmetry directions result in a reduction of the complexity of the problem needed to solve each iteration of the alternating direction problem. As an example, a mesh of rectilinearly-aligned hexahedra would represent the easiest case (three spatial symmetry directions), with an extruded (unstructured) triangular mesh of prisms being easier than the fully three-dimensional tetrahedral-meshed case. This is a clear example of the relevance of the spatial discretization to the efficiency of subsequent algorithms.

Note that the modes obtained using PGD are not necessarily orthogonal, unlike POD modes.

2.4 System-theoretic approaches

The idea behind *balanced truncation* is to intelligently reduce the state vector by removing states that are either hard to reach or hard to observe. This is done by computing the *reachability Gramian* $\mathcal{P}$ and the *observability Gramian* $\mathcal{Q}$ defined by

$$\begin{aligned} A\mathcal{P} + \mathcal{P}A^T + BB^T &= 0 \qquad (17) \\ A^T\mathcal{Q} + \mathcal{Q}A + C^TC &= 0; \qquad (18) \end{aligned}$$

note the matrices A, B, C are as defined in 2 above. To find the states to remove, a *balancing transformation* that simultaneously diagonalizes $\mathcal{P}$ and $\mathcal{Q}$ is sought; and matrices V and W (again, defined in 2) are computed.

There are also frequency-domain approaches, for example, *rational interpolation* involves approximating the transfer function (the Laplace transform of the impulse response) of the system by a rational function.

More information can be found at [6].

3 Data Assimilation

The two main classes of data assimilation method are *sequential methods*, used to propagate the state of the model and the information from observations forward in time one step at a time (embodied in the *Kalman filter*); and *variational methods*, in which the assimilation is performed for all timesteps at once (hence, the information contained in an observation at a given time propagates backward and forward in time). Both classes of method are based on analytic results derived for the combination of Gaussian-distributed errors and using linear dynamical models.

3.1 Kalman filtering

The *filtering problem* is how to maintain an optimally-predictive stochastic model in the presence of two sources of error: i) model error (also known as system noise); and ii) observation error. This means calculating the PDF for the state at a given point in time, conditioned on observations up to that time.

First, the model state is propagated for one time step into the future and this propagation incurs an error known as the *system noise*. The propagation dynamics may be linear or not, though linearity is required in order to keep the Gaussian statistics for which the Kalman filter is strictly optimal. Secondly, new experimental observations are assumed to become known at the new time, in the form of measurement data with its own errors, called *observation error*. There are therefore two fundamental steps: combining the error in the model at the previous timestep with the model error incurred by the time-advance, then conditioning the resulting system on the probability distribution associated to the new observation.

The problem of propagating sequentially the probability density function (PDF) $p(x)$ of the state from time index i to $i+1$ is easily understood, at least in the case of a linear system (and it might be noted here that many models are propagated in time using a linearization about their true dynamics). A simple pedagogical example is that of a random walk where the individual (spatial) 'step' is itself a random variable, ϵ_i, with PDF $q_i(x)$ for each individual step i; thus, the state of the walk updates as

$$x_{i+1} = x_i + \epsilon_i \tag{19}$$

and the evolution of the PDF is given by the convolution

$$p_{i+1}(x) = \int_{-\infty}^{+\infty} p_i(x-u)q_i(u)du. \tag{20}$$

This formula, which can be interpreted as a probabilistic sum over all possible update steps, is most easily solved by transforming to the Fourier domain, in which the convolution becomes a simple product of the Fourier-transformed PDFs (known as *characteristic functions*): viz., indicating a Fourier-transformed quantity with a tilde,

$$\tilde{p}_{i+1}(k) = \tilde{p}_i(k)\tilde{q}_i(k). \tag{21}$$

This can be used to show - basically by completing the square - that, if the initial $p_0(x)$ is Gaussian and the distribution of the step lengths is also Gaussian, then the error in the sum is always Gaussian and that the variance is incremented by the variance of the individual steps (the familiar square-root dependence of the expected displacement via a random walk can be seen to emerge from this rule). Here it is assumed that the errors in different steps labelled by i are uncorrelated. Note this works because of the basis change to the eigenspace of the space translation operator (this sort of diagonalization - a transform to an 'uncorrelated' basis - comes up again and again e.g. in Principal Components Analysis, mentioned in Subsection 2.2). These simple conclusions survive for any choice of model time-update that is linear.

The other thing to consider is the fact that, as well as the system noise (represented by the distribution $q_i(x)$ in the above), there are also observational data that take the form of measurements, these with their own errors (observation error). Given the PDF of the state at time index $i+1$ (i.e. $p_{i+1}(x)$ in the above) and the new observation, the question of the correct new PDF for the system is answered by *Bayes' theorem*, which enables us to make use of the information about the state from the previous time step (in fact, recursively, all past time steps and prior observations) and the new information in the current observation.

Bayes' theorem is usually stated as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{22}$$

where $P(A|B)$ is to be read as the probability distribution of A given prior knowledge B. The term $P(A)$ represents prior knowledge and is simply referred to as the *prior*; the left-hand distribution, incorporating this knowledge, is called the *posterior distribution*.

To continue in the vein of treating the simplest possible example, consider the following Bayesian derivation of the posterior, given a prior $p(x) \propto \exp\left(-(x-x_0)^2/2\sigma_0^2\right)$ and an observation x_e taken from a distribution $\propto \exp\left(-(x-x_e)^2/2\sigma_e^2\right)$. Up to normalization, the posterior distribution is simply the product of these two PDFs, giving a new Gaussian with a variance σ that is clearly given by

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_0^2} + \frac{1}{\sigma_e^2}, \tag{23}$$

the new mean given by $\mu = \frac{\sigma_e^2 x_0 + \sigma_0^2 x_e}{\sigma_0^2 + \sigma_e^2}$; both of these formulae can be obtained by minimizing the variance when combining two uncorrelated random variables - a simple example of least squares. The benefit of all this is that observational 'evidence' is weighted according to its degree of uncertainty.

It is somewhat amusing that this update involves a square completion in real space while the update for the model error had a square completion in k-space (i.e. adding Gaussian random variables involves adding real-space variances, while conditioning involves adding k-space variances - hence the reciprocals in 23). This example demonstrates the property that a Gaussian prior and a Gaussian-distributed observation error lead to a Gaussian posterior (and note that the variance in the posterior does not depend on the actual measured value). It is also clear that the model expectation value undergoes a discontinuous jump on incorporation of the new observation. Again, the basic conclusions here survive for any observation $y(x)$ that is a linear function of the underlying model variable x.

Taken together, and extended to a general linear system of random variables (the scalar quantities above become vectors; the variances become convariance matrices), the two key ideas above (adding Gaussian random variables and Gaussian-on-Gaussian conditioning) combine to give an analytic filtering algorithm known as the Kalman filter.

Taking the system dynamics to be

$$x_{k+1} = Mx_k + \eta_k \tag{24}$$

and the observation definition to be

$$y_k = Hx_k + \epsilon_k \tag{25}$$

and the system noise and observation errors respectively $\eta_k \sim N(0, Q)$ and $\epsilon_k \sim N(0, R)$, there follow formulae for the expectation vector and covariance matrix of the model at time step k, called the *analysis* values and given subscript a (and because of the Gaussian statistics, this exhausts the properties of the model):

$$x_k^a = P_k^a \left(H^T R^{-1} y_k^o + \left(P_k^f \right)^{-1} x_k^f \right), \tag{26}$$

$$P_k^a = \left(\left(P_k^f \right)^{-1} + H^T R^{-1} H \right)^{-1}, \tag{27}$$

where the *forecast* values for the mean and covariance are

$$x_f^k = M x_{k-1}^a \tag{28}$$

and

$$P_k^f = M P_{k-1}^a M^T + Q. \tag{29}$$

These forms make manifest the similarity to the scalar Bayesian formulae above; there are alternate forms which may or may not be easier to evaluate in practice due to the different dimensions of the matrices to be inverted. Note that the analysis covariance is updated using a procedure similar to 23 above; the covariance associated to the measurement error is added in reciprocal.

One alternative form for the update of the mean is

$$x_k^a = x_k^f + P_k^a H^T R^{-1} \left(y_k^o - H x_k^f \right); \tag{30}$$

the matrix $K \equiv P_k^a H^T R^{-1} = P_k^f H^T \left(H P_k^f H^T + R \right)^{-1}$ is known as the *Kalman gain* matrix and the observation-forecast difference the *innovation*. (The second form of K can be derived from the first by postmultiplying by the identity in the form $\left(H P_k^f H^T + R \right) \left(H P_k^f H^T + R \right)^{-1}$ and using the formula connecting the forecast and analysis variances.)

The Kalman filter is optimal (indeed analytic) in the case of strictly Gaussian errors, a property that is destroyed if either the system dynamics or the measurements involve nonlinearity. In most practical applications the dynamics are nonlinear, and observation errors are often manifestly non-Gaussian (for examples such as humidity, the values are strictly positive and bounded from above by physics e.g. saturation). The absence of Gaussianity means that the conditional mean and conditional modes differ; also, properties such as the new variance being independent of the measured values and the fact that the incorporation of the observation decreases the variance may not hold.

Nonlinear systems are treated using the *extended Kalman filter*, in which the mean propagates according to the nonlinear dynamics and the variance is evolved according to a linearization about this mean (see the next subsection for more on this).

In practice, it is difficult to apply the Kalman filter directly to a high-dimensional system because of the need to store the covariance matrices of large numbers of variables (typically 10^6 and up). One possible approach is to replace the continuous PDF and the full covariance matrix with a set of random samples from the appropriate distributions, i.e. resort to Monte Carlo. For such a sample, the Kalman filter is approximated by the *ensemble Kalman filter* (EnKF) and, again, the easiest case to understand is the linear Gaussian scenario, under which there is a simple procedure for generating random variables taken from the correct analysis distribution. Starting with x^i drawn from the forecast distribution (noting that here i denotes one sample in the ensemble) and observation error ϵ_i taken from $N(0, R)$, form the combination

$$\xi^i = x^i + K \left(y^o - (H x^i + \epsilon^i) \right). \tag{31}$$

It is also necessary to approximate the Kalman gain matrix K and the appropriate procedure is to form the approximate gain

$$\hat{K} = X Y^T (Y Y^T)^{-1}, \tag{32}$$

where X is a matrix whose columns are $(N_e - 1)^{-1}\left(x^i - \hat{x}^f\right)$ and Y is a matrix with columns $(N_e - 1)^{-1}\left(Hx^i + \epsilon_i - H\hat{x}^f - \hat{\epsilon}\right)$ (here N_e is the size of the ensemble and in the limit $N_e \to \infty$ the mean and covariance converge to those given by the Kalman filter update).

There are many other details but the key point is that the Monte Carlo approach avoids the curse of dimensionality for very large systems of correlated variables. It is clear that ensemble-based schemes involve potentially a large number of model evaluations, implying that an efficient ROM could be of great utility here.

Note that the EnKF can also be used for parameter estimation; in *joint estimation*, the parameters are included as additional degrees of freedom in the state (there are also other applications of extending the state beyond the degrees of freedom of the numerical model). Such an approach can be used to calibrate a reduced order model; the advantage of doing this, as opposed to an elementary least-squares fit, is that the calibration is done 'correctly' in a Bayesian sense. For a ROM with r internal states and p model parameters and assuming N_y measurements, the approximate gain (32) is a matrix of dimensions $(r + p) \times N_y$, which is relatively inexpensive to compute and use; also note that there are also techniques to avoid needing to compute the $N_y \times N_y$ matrix inverse $\left(YY^T\right)^{-1}$.

A more detailed introduction to the Kalman filter can be found in [8].

3.2 Four-dimensional variational data assimilation and methods from weather prediction

Four-dimensional variational data assimilation or *4D-Var* is a procedure that minimizes the mismatch between a temporal sequence of model states and a set of observational data. Importantly, in doing so it propagates information forward and backward in time (the Kalman filter propagates information forward only). Note that the problem of fitting a model to a set of measurements all at a particular point in time is termed *3D-Var*.

What is referred to as *weak-constraint 4D-Var* allows for uncertainty in the dynamical propagation (i.e. system noise) and works by minimizing a scalar functional such as

$$J(x_0, x_1, ..., x_K) = \frac{1}{2}(x_0 - x_0^b)^T \left(P_0^b\right)^{-1} (x_0 - x_0^b) + \frac{1}{2}\sum_{k=0}^{K}(H_k x_k - y_k)^T \left(R_k\right)^{-1} (H_k x_k - y_k) + \frac{1}{2}\sum_{k=0}^{K-1}(x_{k+1} - M_k x_k)^T \,(\tag{33}$$

The terms here represent, respectively, the deviation of the fitted model from the initial background data, the error in the observed quantities, and the model error. The assimilation is done over a sequence of $K + 1$ points in time labelled by the index k. It is clear, at least in the linear, Gaussian-error case, that $J(x_0, x_1, ..., x_K)$ represents the negative logarithm of the PDF and hence its minimization gives a maximum-likelihood estimate for the system state conditioned on the observed data.

In the case of a realistic scenario, the problem with this is that a direct minimization would require the evaluation of a large number of derivatives, each of which would involve a specific integration of the dynamical model. Making the method feasible is the *adjoint method*, a technique that allows the numerical computation of the gradient of a scalar function at a cost of at most a few times the cost of the direct computation of that function. The adjoint method is basically a recursive trick for calculating the gradients of a function of the form

$$f(x) = \frac{1}{2}\sum_{k=0}^{K} \left(x_k - y_k\right)^T C^{-1} \left(x_k - y_k\right) \tag{34}$$

in the particular case that $x_{k+1} = M_k x_k$. The price of using this technique is that an *adjoint model* must also be coded, in addition to the actual model.

In the case of a linear model and linear measurements, the problem is amenable to the analytic, Gaussian techniques outlined in the previous subsection. However, even if the model error is ignored (giving what is called *deterministic 4D-Var*), it turns out that there are severe problems fitting these models to observations for the case of real weather models, which are not linear. As so far defined, a vector x is sought that minimizes the 4D-Var penalty function

$$J(x) = \frac{1}{2}\sum_{k=0}^{K} \left(x - x^b\right)^T B^{-1} \left(x - x^b\right) + \frac{1}{2}\sum k = 0^K \left(y - y^o\right)^T R^{-1} \left(y - y^o\right), \tag{35}$$

where the first quadratic term is a Gaussian background prior. This extremization is done using local gradients evaluated using the adjoint method. The problem is that this gradient typically fails to lead to a minimum of the penalty function, which is nonlinear due to the nonlinearity of $y = H(M(x))$. Examples of atmospheric physics which lead to this failure are given in [9]; these examples split into fast processes not captured by scales simulated in the model and 'macroscopic' phenomena (e.g. cloud formation) not captured by local gradients. An overall issue seems to be attempts to predict large-scale nonlinear phenomena based on local gradients (equivalently, the forecasting model that has been written down is chaotic). Note that this problem relates to the evolution of the prior under the model and the observations *neglecting* any additional system noise or measurement error. Another issue is that outside the linear, Gaussian case, minimizing the argument of the negative exponential does not estimate the mean of the distribution - it estimates the mode.

A strategy for dealing with this is to track the evolution of the mean using the full nonlinear model $\bar{M}$ and approximate the evolution of perturbations about this mean by a Gaussian model $\tilde{M}$ called the *perturbation forecast* (PF). In practice, this proceeds by introducing and iteratively improving a 'guess' x^g for the mean of the new state (this is termed the *incremental approach*). Also, in the absence of a true model $\bar{M}$ for the mean of the PDF, the normal NWP model M is used. Now the evaluation of the statistics from the perturbation forecast model form the inner loop and this needs to be evaluable without recomputing the full nonlinear model $\bar{M}$ i.e. it uses $\tilde{M}$. This can be viewed as a sort of reduced-order model.

Another critical issue in NWP is the so-called *butterfly effect* in which errors or fluctuations at small scales quickly grow and affect larger scales. This means that current NWP algorithms cannot simply be expected to continue to work at a higher resolution on a more powerful computer without some degree of modification. A solution mentioned in [9] is to filter the PF model to prevent the rapid growth of the affected scales. Another issue is the potentially chaotic nature of DA systems (and note that deterministic chaos concerns extreme sensitivity to initial conditions, rather than the coupling between scales that characterises the butterfly effect). This chaotic dynamics becomes worse for larger systems where there are basically not enough observations to control the growth of dynamics governed by the increasing number of positive Lyapunov exponents.

More detailed expositions of 4D-Var can be found in [9], [10].

4 Summary

This report has provided a brief overview of each of three important approaches to model order reduction: reduced basis methods, proper orthogonal decomposition and proper generalized decomposition. Specific recommendations for methods expected to work well in the magnetically-confined fusion (MCF) use case are not yet possible, though it is clear that methods capable of handling nonlinearity are indicated, inviting a further study of POD methods and also indicating the need for techniques such as empirical interpolation to preserve the online efficiency of reduced basis methods. Another, though perhaps longer-term, consideration is the requirement for an MCF control system to be able to quantify and mitigate extreme events in order to prevent damage to a fusion machine and in this sense, algorithms of the greedy type are useful in that they seek to identify worst-case events rather than focussing on a global least-squares minimization.

Existing MOR implementations encapsulate much of the theory and computational machinery outlined in this report and are able to interface with external PDE solvers. One simple initial proposal is therefore to apply tools such as *pyMOR* to simulations of fluid turbulence (a simple model with a handful of inputs and a single main physically-relevant output in the time-averaged quasi-steady-state heat flux across the domain but a large number of internal degrees of freedom); such models are a proxy for heat transport near the outer boundary of a tokamak.

A subsequent section presented an overview of data assimilation and sketched the two main approaches of Kalman filtering and variational data assimilation. Some particular issues in NWP were highlighted, given that similar problems are expected in the case of tokamak edge physics modelling, key shared features being nonlinearity, multiple scales and turbulence; the main interest is how to extend the techniques of DA to work in cases where the errors are non-Gaussian and where the dynamical model is not linear. It also revealed that NWP models use what is spiritually a reduced-order model in the inner loop of a perturbation forecast model for estimating model variance in non-Gaussian scenarios. The ensemble Kalman filter was highlighted as a technique for fitting not only the model state but also as a tool for Bayesian parameter estimation.

Acknowledgement

The support of the UK Meteorological Office and Strategic Priorities Fund is acknowledged.

A Gaussianity

The Gaussian-distributed random variable is, in a sense, the simplest random quantity, in that the statistical properties are specified completely the the first *two* cumulants, these representing the mean and the variance of the probability distribution. Higher cumulants are zero; the Gaussian is the only distribution with this property (indeed, even a uniform probability distribution has an infinite set of non-zero cumulants).

The well-known *Central Limit Theorem* states that a linear combination of a large number of uncorrelated random variables tends to a Gaussian-distributed random variable; indeed, any linear function of a Gaussian random variable is itself a Gaussian random variable i.e. there is no need to take the limit in this case (this means that linear dynamical models preserve the Gaussianity of errors in the model state). This can be expressed in the fact that the Gaussian is *stable* under linear combination of like-distributed variables. There are other stable distributions (for example, the Cauchy distribution), but the Gaussian is the only one with finite variance. Note that it has been demonstrated in the main text that the Gaussian is also stable under the conditional probability rules for the incorporation of new observations that are themselves Gaussian distributed.

If an N-component random vector $\underline{x}$ is Gaussian, then it has probability density

$$p(\underline{x}) = \frac{1}{(2\pi)^{\frac{N}{2}}} |P|^{-\frac{1}{2}} \ \exp\left(-\frac{1}{2}(\underline{x} - \bar{\underline{x}})^T P^{-1} (\underline{x} - \bar{\underline{x}}) \right). \tag{36}$$

Here $\bar{x}$ is the expectation $E(\underline{x})$ and $P = cov(\underline{x}) \equiv E((\underline{x} - \bar{\underline{x}})(\underline{x} - \bar{\underline{x}})^T)$.

Note that the mode of the Gaussian distribution is equal to the mean (this fact means that the mean can be evaluated by minimizing the quadratic argument of the decaying exponential, or equivalently that a maximum-likelihood estimate gives the mean).

The Fourier transform of a zero-mean Gaussian is easily evaluated by completing the square, giving

$$FT\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/2\sigma^2} \right) \propto e^{-\sigma^2 k^2/2}, \tag{37}$$

showing that the Fourier transform of a Gaussian is a k-space Gaussian.

References

[1] Matteo Bonotto, Paolo Bettini, and Angelo Cenedese. Model order reduction of large-scale state-space models in fusion machines via Krylov methods. *IEEE Transactions on Magnetics*, 53, 2017.

[2] Julio L. Nicolini, Dong-Yeop Na, and Fernando L. Teixeira. Model order reduction of electomagnetic particle -in-cell kinetic plasma simulations via proper orthogonal decomposition. *IEEE Transactions on Plasma Science*, 47, 2019.

[3] C. Gillot, G. Dif-Pradalier, X. Garbet, P. Ghendrih, V. Grandgirard, and Y. Sarazin. Model order reduction approach to the one-dimensional collisionless closure problem. *Physics of Plasmas*, 28, 2021.

[4] Model order reduction wiki. `https://en.wikipedia.org/wiki/Model_order_reduction`, 2020. Accessed: October 2020.

[5] Francisco Chinesta, Antonio Huerta, Gianluigi Rozza, and Karen Willcox. *Model Order Reduction, chapter in Encyclopedia of Computational Mechanics*. John Wiley & Sons, 2017.

[6] pyMOR online documentation. `https://docs.pymor.org`, 2021. Accessed: March 2021.

[7] N. Gershenfeld. *The Nature of Mathematical Modeling*. Cambridge University Press, 1999.

[8] C. Snyder. Introduction to the Kalman filter. In É. Blayo, M. Bocquet, E. Cosme, and L. F. Cugliandolo, editors, *Advanced Data Assimilation (Les Houches 2012 Special Issue)*, chapter 3, page 75120. Oxford University Press, Oxford, 2012.

[9] A.C. Lorenc. Four-dimensional variational data assimilation. In É. Blayo, M. Bocquet, E. Cosme, and L. F. Cugliandolo, editors, *Advanced Data Assimilation (Les Houches 2012 Special Issue)*, chapter 2, pages 31–73. Oxford University Press, Oxford, 2012.

[10] O. Talagrand. 4D-VAR: four-dimensional variational assimilation. In É. Blayo, M. Bocquet, E. Cosme, and L. F. Cugliandolo, editors, *Advanced Data Assimilation (Les Houches 2012 Special Issue)*, chapter 1, pages 3–30. Oxford University Press, Oxford, 2012.