Survey of Methods for Model Order Reduction

For the Seminar High Dimensional Methods in Scientific Computing

Evgeni Todorov
Department of Informatics
Technical University of Munich
Email: evgeni.todorov@tum.de

***Abstract*— Numerical simulations in the fields of science and engineering have grown in importance with increased computational capacity. Efficiency of modelling is now key to multiple applications. It can be enabled by reducing the number of degrees of freedom of a model while continuing to satisfy the requirements on the output - a process known as model order reduction. In this paper some of the milestone methods underlying future developments or undergoing active research are shortly outlined. A worked example from the well known component mode synthesis method illustrates typical usage. Several recent applications from the fields of aerospace engineering, robotics and biomechanics are presented. While much of the field is mature, extending model reduction methods to apply to parametric models and taking on board modern developments in machine learning for non-linear reduction remain active fields of research.**

***Keywords*— Model Order Reduction**

I. Introduction

Advances in computational technology have made it possible to more accurately than ever before represent physical, biological, psychological, social or economic phenomena in detail. This is accomplished by describing them using Partial Differential Equations (PDEs). Then a computer algorithm is implemented to find values for the parameters of interest which satisfy this idealized view of the system under investigation. Simulation continues to be increasingly successful in many fields, and the scope of application continues to grow. Systems with a large number of values of interest (degrees of freedom) like weather and climate prediction have long been the focus on computational science and high-performance computing. Improving the efficiency of the numerical solution is crucial, both to be able to better utilize finite amounts of resource but also to open new avenues which are currently restricted. The latter can include applications as diverse as *real-time* computing (for control systems or Internet of things), *many-query* context (design optimization, parameter estimation) or *outer-loop* applications (uncertainty quantification, advanced measurement) [1].

One of the common measures quantifying the cost of solving a model is the number of degrees of freedom the system has. Consequently, one of the best avenues to improve the efficiency is to decrease this number, which is the job of model order reduction (MOR) methods. Consider the general dynamical system (1):

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} &= \mathbf{f}(\mathbf{x}, \mathbf{u}), \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}, \mathbf{u}), \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the input and $\mathbf{y}$ is the output (all of them a function of time t).

When solving a numerical PDE, the state vector $\mathbf{x}$ will contain all the degrees of freedom of the system. A model order reduction method has the task to reduce (1) to (2) such that $\hat{\mathbf{y}} \approx \mathbf{y}$, where $\hat{}$ denotes reduced dimension:

$$\begin{aligned} \frac{\mathrm{d}\hat{\mathbf{x}}}{\mathrm{d}t} &= \hat{\mathbf{f}}(\hat{\mathbf{x}}, \mathbf{u}), \\ \hat{\mathbf{y}} &= \hat{\mathbf{g}}(\hat{\mathbf{x}}, \mathbf{u}). \end{aligned} \tag{2}$$

The aim is to provide a quicker way to get to a solution which is sufficiently refined to provide the knowledge needed but not any further. One way can be achieved is by eliminating degrees of freedom which are not of interest in their own right but are required to correctly compute others which are, Another way is creating a *surrogate model* which can provide actionable information but very often cannot be constructed from first principles without a prior more detailed representation. Many recent application focus on parametric models, where the state $\mathbf{x}$ and transfer functions $\mathbf{f}$ and $\mathbf{g}$ also depend on a parameter vector $\mathbf{p}$ in addition to time. Parametric models often find application in real-time and many-query settings [2].

The drive for a simpler way to get to sophisticated conclusions has a long history, from nomograms used in calculations, through ubiquitous lookup-tables to reach reduced models critically important to contemporary technology such as the Earth geopotential model [3] based on spherical harmonics, which is crucial to the functioning of GPS, and to cutting-edge aritifical intelligence developments like the recent BERT language model [4]. In computational science, model order reduction is most often used in the context of analyzing dynamical systems by solving PDEs. Some important developments include the Modal truncation method based on Arnoldi [5], Proper Orthogonal Decomposition [6], Krylov subspaces methods [7] and many others [8].

To provide a brief contemporary introduction to model order reduction, first highlights among methods for PDEs are discussed in section II. An example implementation illustrates the methods in section III and is followed by some exciting recent applications in section IV. Finally a summary of the current status and future opportunities for the field are discussed in section V.

II. Highlights among MOR methods for PDEs

Model order reduction is a broad field with a large variety in methods and application, and multiple different classification strategies can be found in literature, e.g. *equation-based* vs *data-based* [9] (also often called equation-free [10]), *operational* vs *projection* methods [8], *reduce-then-model* vs *discretize-then-reduce* [11], *transformation* vs *projection* [12], *hierarchical* vs *data-fit* [13], etc. Here a similar but less rigorous split based on the target driving the method is used to organize a short overview of some of the key techniques.

A. Algorithm-driven

Algorithm-driven methods are generally hierarchical or multiscale methods which are targeted towards reducing a very large problem in the first place. This is done by leveraging techniques to never explore degrees of freedom which might not be of interest unless needed, or to jump to the solution much faster. The field is rich but mostly mature, with various classes of methods a standard feature of both commercial and academic software packages and training courses.

1) Multigrid methods (recent overview in e.g. [14]) are a fitting and ubiquitous example of a class of algorithms for which the main idea is to improve the performance of *smoother* iterative solver methods by solving the problems first at a coarse scale, and then interpolating and refining the solution iteratively. In the classical multigrid approach this is accomplished by adjusting the grid size but algebraic methods also exist which use directly the system matrix representation of the discretized problem, and can be combined with adaptive relaxation to achieve optimal performance [15].

2) Adaptive mesh contol is another similarly successful and popular method, which uses a similar idea of controlling the discretized domain to decrease the necessary degrees of freedom, however based on a different set of criteria and refinement techniques. One important aspect of adaptive mesh refinement is the ability to locally improve mesh size and time-step, identifying regions of quicker variation where better resolution is needed. The idea has also been extended into adaptive mesh and algorithm refinement [16], which not only adjusts the discretization parameters but also the underlying physical representation based on the user-defined resolution criteria.

3) Multiscale modeling is a broad field which deals with systems where the micro-scale and macro-scale behaviour cannot be isolated but are mutually dependent. Both of the above techniques, as well as most of the other *algorithm driven* methods find their application in the field of multi-scale modelling. It utilizes the techniques discussed previously, as well as multi-scale representation of the solution (via e.g. wavelets or Fourier series) to only concern itself with relevant degrees of freedom but perhaps even more crucial for model order reduction are the hierarchy of physical representations. Replacing for example a particle model with a continuum one can reduce the modelling scale and therefore the required number of degrees of freedom much farther any domain discretization strategy would [17].

4) Proper Generalized Decomposition (PGD) is a general strategy to model order reduction which is related to other methods such as Proper Orthogonal Decomposition and the Empricial Interpolation Method (see below) and is often used in conjunction with a snapshot based method [18]. PGD is conceptually similar to the method of separation of variables, and has historically been used to enable space-time separation. However, the advancing need for parametric models has brought increasing importance to the method since it enables a clever and efficient way to reduce the intrinsic dimensionality of a problem.

B. Data-driven

In contrast, the *data-driven* methods more often target reducing a model which can be solved in realistic time scales but where a reduced form will be beneficial. Typical applications include *many-query* context, where a single model is solved multiple times with some variation - e.g. a design optimization problem, or *real-time* computing, as required when a model needs to be solved for complex control system such as employed in robotics. While no less developed than the *algorithm driven* methods, this category is continuing to be under very active development [19], including due to synergies with recent advances in machine learning. Several key methods are outlined hereafter.

1) Proper Orthogonal Decomposition (POD) is one of the most common model order reduction techniques and a typical example from the family of projection methods. The core idea is common to dimensionality reduction techniques used in many disciplines starting with the famous Prinicpal Component Analysis (PCA) method [20]. In general, it is a method to project the system of interest into a smaller system, defined on a smaller space spanned by an orthogonal basis. It is commonly applied as the method of snapshots in which this orthogonal basis is determined based on a set of snapshots computed by a physical or numerical experiment. Consider (3), a form of (1) where we disregard the input $\mathbf{u}$ without loss of generality, and separate the linear part of the function $\mathbf{f}$ into the matrix $\mathbf{A}$ and a non-linear $\mathbf{f}$:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{A}\mathbf{x} + \mathbf{f}(\mathbf{x}). \tag{3}$$

POD requires that a snapshot matrix $\mathbf{S} = [\mathbf{x}_1 \ldots \mathbf{x}_s]$ is defined for s snapshots of the system. $\mathbf{S}$ is then decomposed via singular value decomposition. The largest r singular values $\sigma_1 \ldots \sigma_r$ are determined, based on some criterion about the precision of the reduction. A common criterion based on least-squares fit is

$$\frac{\sum_{i=0}^{r} \sigma_i^2}{\sum_{i=0}^{s} \sigma_i^2} > \mathrm{threshold}. \tag{4}$$

The corresponding r singular vectors $\mathbf{v}$ form the orthogonal reduction matrix $\mathbf{V} = [\mathbf{v}_1 \ldots \mathbf{v}_r]$ such that $\mathbf{x} \approx \mathbf{V}\hat{\mathbf{x}}$ and define $\hat{\mathbf{A}} := \mathbf{V}^\mathbf{T}\mathbf{A}\mathbf{V}$ and $\hat{\mathbf{f}} := \mathbf{V}^\mathbf{T}\mathbf{f}(\mathbf{V}\hat{\mathbf{x}})$. The reduced model is then given by:

$$\frac{\mathrm{d}\hat{\mathbf{x}}}{\mathrm{d}t} = \hat{\mathbf{A}}\mathbf{x} + \hat{\mathbf{f}}(\mathbf{x}). \tag{5}$$

For linear models, the reduced model is very cheap to evaluate since $\hat{\mathbf{A}}$ is cheap to compute but the $\mathbf{V}\hat{\mathbf{x}}$ term in the non-linear part can be expensive [1].

2) Empirical Interpolation method (EIM) introduced by [21] is an example of a class of techniques to reduce non-linear systems, sometimes referred to as hyper-reduction [22]. EIM deals better with the nonlinear term by conducting interpolation on a low-dimensional basis [1]. A specification of EIM to finite dimensional systems of differential equations (the Discrete Empirical Interpolation Method, DEIM [23]) is presented here. In summary, DEIM interpolates the non-linear term from (3) on a suitable reduced basis $\mathbf{U} = [\mathbf{u}_1 \ ... \ \mathbf{u}_r]$ following a greedy algorithm to determine a suitable interpolation matrix $\mathbf{P}$ such that $\hat{\mathbf{f}}(\mathbf{x}) = \mathbf{U}(\mathbf{P^T U})^{-1}\mathbf{P^T f}(\mathbf{x})$. Since the term in front of $\mathbf{f}$ can be precomputed, this significantly improves the performance of the algorithm. The basis $\mathbf{U}$ is not specificed by DEIM but is generally constructed similarly to the POD basis considering only the non-linear term. The interpolation matrix $\mathbf{P} = [\mathbf{e}_{\iota_1} \ ... \ \mathbf{e}_{\iota_r}]$ is a permutation matrix consisting of the columns of the identity matrix $\mathbf{I} = [\mathbf{e}_1 \ ... \ \mathbf{e}_s]$, $\mathbf{e_1} = [1, 0, ..., 0]$ for the $\iota_1 ... \iota_r$ indices where the absolute value of the components of the residual $\boldsymbol{\rho} = [\rho_{\iota_1} \ ... \ \rho_{\iota_r}]$ defined in (6) for consecutive values of $l = 1 \ ... \ r$ are maximized:

$$\begin{aligned} \mathbf{P^T U c} &= \mathbf{P^T u}_l, \\ \boldsymbol{\rho} &= \mathbf{u}_l - \mathbf{Uc}. \end{aligned} \tag{6}$$

EIM aims to get at the non-linear term by interpolation, while other hyper-reduction methods attempt to use regression or reformulate the problem in its weak form to use an estimate of the integral [13].

3) Non-linear machine learning approaches can also be used for model order reduction, since it is closely related to the problem of dimensionality reduction. A variety of dimensionality reduction methods have been proposed by the machine learning community [24] and new ones are continuing to be actively investigated. Those approaches are typically targeted to the non-linear behaviours in the model, approximating it as e.g. a manifold to be extracted. Diffusion maps [25] are an example of a manifold learning technique which has been applied to construct surrogate models [9]. Neural networks have of course also been utilized for model order reduction applications, although they are often best used in conjunction with another reduction method to extend a reduced model across parameter space [26].

4) The Koopman operator can be used to derive another type of data-driven model order reduction. The Koopman operator theory originates in the analysis of dynamical systems, where one of its main advantages has been allowing to do spectral analysis in a linear setting to represent the full non-linear behaviour of the system. [27] This advantage comes at the cost of a disconnection from intuitive interpretation of dynamical system, which is perhaps a reason why despite the growing body of academic work, industry uptake is much slower.

While a complete treatment of the mathematics behind the Koopman operator is beyond the current scope, it can be defined (passing over some of the formalism) for a discrete dynamical system

$$\mathbf{x_{n+1}} = \mathbf{T}(\mathbf{x_n}), \tag{7}$$

where $\mathbf{T}$ is the *shift operator*, and for observables (functions, which when evaluated provide meaningful information about the system) $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ ... \ f_k(\mathbf{x})]^T$ as the functional operator K in (8) which projects the observables further in time:

$$(\mathrm{K}\mathbf{f})(\mathbf{x}) = \mathbf{f}(\mathbf{T}(\mathbf{x})) = \mathbf{f}(\mathbf{x_{n+1}}). \tag{8}$$

While this definition holds for arbitrary functions in the domain of K, numerical approximations of the operator typically use finite truncations and thus (8) only holds exactly in finite subspaces [27].

Spectral decomposition of the Koopman modes is not guaranteed to yield a useful lower dimensional representation of a dynamical system, which means it does not fit with classical model reduction techniques but several data-driven approaches have been defined which work well for PDEs. Dynamic Mode Decomposition (DMD) originated in [28] and was extended in [29] provides a way to approximate the spectral properties of the operator based on high-fidelity measurement data from the system. It has recently been combined with a neural network used to construct a trainable dictionary which defines the observables used by the EDMD without a priori knowledge [30].

III. Example method: Component Mode Synthesis

A simplified example from the field of structural mechanics is presented to illustrate the creation and use of model order reduction. It uses the Component mode synthesis (CMS) method (particularly the Craig-Bampton implementation) which is considered industry standard for structural simulation and is implemented in major commercial packages such as ANSYS and NASTRAN. In addition to the widespread usage, CMS serves as a good example since it is a hybrid method employing generalized coordinate projection including physical and modal coordinate basis vectors. While very application specific, CMS exhibits all key aspects of modal reduction methods. Moreover, it allows introduction to the concept of *substructuring* which while not directly relevant to the mathematical formalism is integral to many practical implementations.

A. Setup

The dynamics of a general mechanical system are governed by (9) for a generalized coordinate q, displacement function u, mass m, stiffness k and applied force f:

$$m(q)\ddot{u}(q) - k(q)u(q) = f(q), \tag{9}$$

This can be discretized to (10) by finite elements, finite differences or another method:

$$\mathbf{M}\ddot{\mathbf{u}} - \mathbf{K}\mathbf{u} = \mathbf{f}. \tag{10}$$

To relate this to the general format defined in (1), we can define the state space $\mathbf{x} = [\ddot{\mathbf{u}}^{\mathbf{T}}, \dot{\mathbf{u}}^{\mathbf{T}}, \mathbf{u}^{\mathbf{T}}]^{\mathbf{T}}$. In this section we consider an already discrete 1D example inspired by [31] as shown in fig. 1. The matrices for (10) are then given by:

$$\mathbf{u} = \begin{pmatrix} u_A \\ u_B \\ u_C \\ u_D \\ u_E \end{pmatrix} \mathbf{M} = \begin{bmatrix} 0 & & & & \\ & m_B & & & \\ & & m_C & & \\ & & & m_D & \\ & & & & m_E \end{bmatrix} \mathbf{K} =$$

$$\begin{bmatrix} k_{AB} & -k_{AB} & & & \\ -k_{AB} & k_{AB}+k_{BC} & -k_{BC} & & \\ & -k_{BC} & k_{BC}+k_{CD} & -k_{CD} & \\ & & -k_{CD} & k_{CD}+k_{DE} & -k_{DE} \\ & & & -k_{DE} & k_{DE} \end{bmatrix}.$$

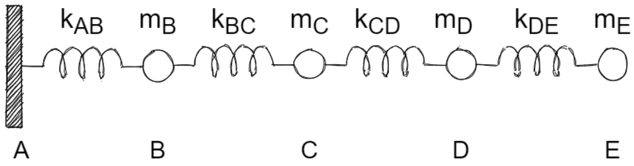

Fig. 1. Example discrete dynamical system

B. Substructuring

Substructuring is the subdivision of a domain into a number of subdomains, usually with the aim of obtaining reduced order models for each subdomain (component), and assemblying them into a reduced order model for the whole area of interest. In structural engineerig, the term *superelement* is commonly used for a substructure component. As mentioned previously, the concept of substructuring is one of the main strategies of utilizing reduced order models. This is especially relevant in the context of design applications, be it of structures, circuits or control systems.

Application of this philosophy to the example system from fig. 1 is shown in fig. 2, where two superelements are defined. The underlying physical degrees of freedom are distributed within several sets of a superelement. Most relevant are the interior degrees of freedom $\mathcal{I}$: $\mathcal{I}_1 = \{B\}$, $\mathcal{I}_2 = \{D, E\}$ and the boundary degrees of freedom $\mathcal{B}$: $\mathcal{B}_1 = \mathcal{B}_2 = \{C\}$

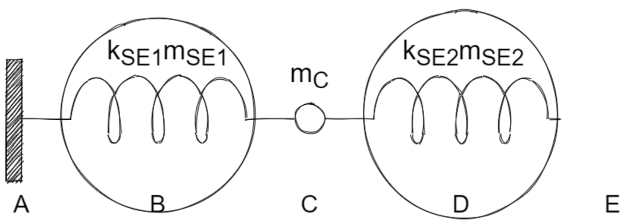

Fig. 2. Substructuring of the example system with 2 superelements

C. Craig-Bampton Transformation

Component mode synthesis works, simply put, by substructuring the domain, identifying component modes for each domain and removing redundant (or low impact) component modes to achieve model reduction. Here we continue with the illustrative example and the Craig-Bampton method, where only fixed-interface and constraint component modes are considered. An overview of other types of CMS methods and modes, as well as further detail about CMS, can be found in [32, Chapter 17].

It is useful to partition (10) for the internal and boundary points of a single superelement:

$$\begin{bmatrix} \mathbf{M}_{\mathcal{II}} & \mathbf{M}_{\mathcal{IB}} \\ \mathbf{M}_{\mathcal{BI}} & \mathbf{M}_{\mathcal{BB}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathcal{I}} \\ \mathbf{u}_{\mathcal{B}} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{\mathcal{II}} & \mathbf{K}_{\mathcal{IB}} \\ \mathbf{K}_{\mathcal{BI}} & \mathbf{K}_{\mathcal{BB}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_{\mathcal{I}} \\ \ddot{\mathbf{u}}_{\mathcal{B}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_{\mathcal{B}} \end{bmatrix}. \quad (11)$$

Further for this example, we consider SE2 as defined in fig. 2, with $\mathbf{u}_{\mathcal{I}_2} = [u_D, u_E]$ and $\mathbf{u}_{\mathcal{B}_2} = [u_C]$. This means:

$$\mathbf{K}_{\mathcal{II}_2} = \begin{bmatrix} k_{CD}+k_{DE} & -k_{DE} \\ -k_{DE} & k_{DE} \end{bmatrix} \mathbf{K}_{\mathcal{IB}_2} = \begin{bmatrix} -k_{CD} \\ 0 \end{bmatrix}. \quad (12)$$

For the constraint modes $\boldsymbol{\phi}_{\mathcal{IB}}$ which apply the boundary condition to the internal nodes, we have:

$$\boldsymbol{\phi}_{\mathcal{IB}} = -\mathbf{K}_{\mathcal{II}}^{-1} \mathbf{K}_{\mathcal{IB}} \mathbf{I}_{\mathcal{BB}}, \quad (13)$$

with $\mathbf{I}_{\mathcal{BB}}$ a suitably sized identity matrix.

For the fixed boundary modes $\boldsymbol{\Phi}_{\mathcal{II}}$ we need to solve the eigenvalue problem (14). The index k denotes index of eigenvalues and modes calculated. The total number of modes used later on determines the degree of reduction:

$$\begin{gathered} -\omega^{(k)} \mathbf{M}_{\mathcal{II}} \boldsymbol{\phi}_{\mathcal{II}}^{(k)} + \mathbf{K}_{\mathcal{II}} \boldsymbol{\phi}_{\mathcal{II}}^{(k)} = \mathbf{0} \\ \boldsymbol{\Phi}_{\mathcal{II}} = [\boldsymbol{\phi}_{\mathcal{II}}^{(1)} \; \boldsymbol{\phi}_{\mathcal{II}}^{(2)} \; ...]. \end{gathered} \quad (14)$$

The Craig-Bampton transformation matrix is given by:

$$\boldsymbol{\Phi}_{CB} = \begin{bmatrix} \boldsymbol{\phi}_{\mathcal{IB}} & \boldsymbol{\Phi}_{\mathcal{II}} \\ \mathbf{I}_{\mathcal{BB}} & \mathbf{0} \end{bmatrix}. \quad (15)$$

It is important to note that the transformation matrix is applied to each individual superelement's stiffness and mass matrices to produce the reduced counterparts, for example for the right-hand side superelement (SE2):

$$\widehat{\mathbf{K}}_2 = \boldsymbol{\Phi}_{CB_2}^T \mathbf{K}_2 \boldsymbol{\Phi}_{CB_2} \qquad \widehat{\mathbf{M}}_2 = \boldsymbol{\Phi}_{CB_2}^T \mathbf{M}_2 \boldsymbol{\Phi}_{CB_2}. \quad (16)$$

Finally the superelements need to be assembled back with the residual structure (the part of the domain outside the open sets of all the subdomains) to construct the reduced version of the model (17). Care should be taken to properly combine the remaining physical degrees of freedom from $\mathbf{u}$ of (10) with the modal degrees of freedom derived by (16) into the generalized reduced coordinates $\hat{\mathbf{q}}$:

$$\widehat{\mathbf{M}} \ddot{\hat{\mathbf{q}}} + \widehat{\mathbf{K}} \hat{\mathbf{q}} = \hat{\mathbf{f}}. \quad (17)$$

IV. Recent applications of model order reduction

The maturing of the field of model order reduction has resulted in a lot of research dedicated to applications and case studies. Some recent interesting applications in the field of aerospace engineering, robotics and bioengineering are discussed below. A recent collection of examples is [33].

A. Comparison for Aeroelastic Analysis

Coupling of aerodynamic effects to the structural response (known as aeroelastics) is an important component of aircraft design, and an area still under development due to the complexity of interactions and the wide spectrum of conditions which need to be evaluated. In [34] a benchmarking study for a variety of typical model order reduction methods used in structural engineering is presented. This includes condensation methods such as Guyan reduction [35] and Dynamic condensation [36], some of the classical projection methods discussed earlier, and several types of CMS (including Craig-Bampton as presented in section III).

While some incremental improvements of the methods (e.g. [37]) are more recent, most of the tools in use in structural engineering as discussed in [34] were established in the 1960s. Recent developments see less wide application, partially due to competition with established methods which sufficiently satisfy the existing requirements. For example, some of the methods in [34] (e.g. POD) could not be benchmarked at a comparable level due to absence of mature enough implementation compatible with the complex model. The development of more successful parametric techniques can be a turning point, changing the paradigm in the structural engineering field by enabling conceptually new applications [12]. As an example, instead of a surrogated model of the structure of the wing, a robust parametric model would allow reducing the aeroelastic model to terms of flight parameters and design variables, and be integrated in a possible design loop.

B. Control of Soft Robots

Another example of reducing a structural simulation but for a very different purpose is the simulation and control of soft robots [38]. Among the challenges of modelling soft robots is the complex non-linear material properties. These, combined with the multi-body simulation nature of robotics simulation, are beyond the capability of the industry standard modelling tools in real-time required for the actuator and control system design of the robots. In [38] a couple of applications of reduced order models in the design stage are very successful with real-time online simulation after a one-off expensive offline creation. An even more interesting use, integrating a reduced model within the control system of the robot, has some remaining limitation related to contact modelling but nevertheless highlights the potential of using reduced models in sophisticated control applications.

C. Structural Health Monitoring

A different field where efficient evaluation of the model is important are outer-loop applications to measurement. In [39], the authors use a parametric reduced order model of the experimental apparatus to generate the large quantities of training data needed to train a machine learning classifier to identify damage. Unlike the application in robotics discussed above, the reduced model is still used in the offline stage for the measurement to generate the extremely large number of simulations with varying parameters required to train the classifier. For a full-size damage detection system (which can be applied to a civil or mechanical structure) the cost for generating the data would be prohibitive for a full order model.

D. Cardiovascular system modelling

Biomechanical simulations are another growing research field where model order reduction has opened new pathways. Two uses of reduced order models in cardiovascular simulation are discussed in [40].

When modelling wall shear stress at the carotid artery bifurcation, a parametric reduced order model of a finite element model is used to replace the computationally expensive Navier-Stokes equations simulating blood flow in response to two parameters, one representing stenosis (obstruction) and the other the flow rate in the common carotid artery. While currently no application which requires real-time simulation has been identified, the availability of the technique will hopefully pave the way for future development.

In the other application discussed in [40], a parametric reduced order model is used in the evaluation of activation maps resulting from cardiac electrophysiology measurements. The real-time simulation allows for replicating virtual counterparts of the measurements, which can improve quantifying the uncertainty as well as facilitate optimization of intervention procedures, aiding clinical practice.

V. Summary and Outlook

State-of-the-art

Model order reduction is a maturing field with a wide spectrum of well-developed methodologies proven in a variety of applications. Some examples where the technology finds ready industrial application is circuit design or linear mechanics simulation, where robust algorithms are adopted within the industry standard toolset. Nevertheless, the drive for extending the domain of simulation to more performance critical domains including real-time and many-query contexts means that the existing methods still fall short of the technological potential. While a wide selection of academic-grade research, including software packages, is available, the commercial uptake for recent advanced methods remains limited. [41]

Open Avenue for Future Work

The rich world of classical model order reduction algorithms has left little space for recent advances, except in some limited areas where growth has been driven by high interest in machine learning e.g. neural networks. However, developments of data-driven and parametric MOR methods and very promising application to industrially hot topics like multi-disciplinary optimization, digital twins, autonomous control systems, augmented reality, data rich measurements, etc. are bound to continue. In addition, the existing gap between scientific advancements and practical utilization needs to be filled. Since this gap is related both to a high barrier to entry due to the nuances of a large field for the users and a diminishing return for the software vendors, a combination of well-communicated and efficient solutions to industry relevant problems is the most promising way forward.

REFERENCES

[1] F. Chinesta, A. Huerta, G. Rozza, and K. Willcox, "Model reduction methods," *Encyclopedia of Computational Mechanics Second Edition*, pp. 1–36, 2017.

[2] A. Quarteroni, G. Rozza *et al.*, *Reduced order methods for modeling and computational reduction*. Springer, 2014, vol. 9.

[3] D. Barnes, D. Barnes, J. Beale, H. Small, and S. Ingalls, "Introducing egm2020," in *EGU General Assembly Conference Abstracts*, 2020, p. 9884.

[4] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[5] W. E. Arnoldi, "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Quarterly of applied mathematics*, vol. 9, no. 1, pp. 17–29, 1951.

[6] L. Sirovich, "Turbulence and the dynamics of coherent structures i-iii," *Quarterly of applied mathematics*, 1987.

[7] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 4, pp. 352–366, 1990.

[8] W. H. Schilders, H. A. Van der Vorst, and J. Rommes, *Model order reduction: theory, research aspects and applications*. Springer, 2008, vol. 13.

[9] F. Dietrich, "Data-driven surrogate models for dynamical systems," Ph.D. dissertation, Technische Universität München, 2017.

[10] I. G. Kevrekidis, C. W. Gear, and G. Hummer, "Equation-free: The computer-aided analysis of complex multiscale systems," *AIChE Journal*, vol. 50, no. 7, pp. 1346–1355, 7 2004.

[11] K. S. Mohamed, *Machine learning for model order reduction*. Springer, 2018, vol. 664.

[12] E. J. Yoo, "Parametric model order reduction for structural analysis and control," Ph.D. dissertation, Technische Universität München, 2010.

[13] Q. He, "Model order reduction and data-driven computational modeling for linear and nonlinear solids," Ph.D. dissertation, University of California, San Diego, 2018.

[14] J. H. Bramble, *Multigrid methods*. Chapman and Hall/CRC, 2019.

[15] A. Brandt, J. Brannick, K. Kahl, and I. Livshits, "Bootstrap AMG," *SIAM Journal on Scientific Computing*, vol. 33, no. 2, pp. 612–632, 2011.

[16] A. L. Garcia, J. B. Bell, W. Y. Crutchfield, and B. J. Alder, "Adaptive mesh and algorithm refinement using direct simulation Monte Carlo," *Journal of computational Physics*, vol. 154, no. 1, pp. 134–155, 1999.

[17] E. Weinan, *Principles of Multiscale Modeling*. Cambridge University Press, 2011.

[18] F. Chinesta and P. Ladevèze, "3 proper generalized decomposition," in *Snapshot-Based Methods and Algorithms*. De Gruyter, 2020, pp. 97–138.

[19] J. V. Aguado, D. Borzacchiello, E. Lopez, E. Abisset-Chavanne, D. González, E. Cueto, and F. Chinesta, "New trends in computational mechanics: model order reduction, manifold learning and data-driven," in *9th Annual US-France symposium of the International Center for Applied Computational Mechanics*, 2016.

[20] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[21] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations," *Comptes Rendus Mathematique*, vol. 339, no. 9, pp. 667–672, 2004.

[22] D. Ryckelynck, "Hyper-reduction of mechanical models involving internal variables," *International Journal for Numerical Methods in Engineering*, vol. 77, no. 1, pp. 75–89, 2009.

[23] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010.

[24] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, no. 66-71, p. 13, 2009.

[25] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[26] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, "Model reduction and neural networks for parametric PDEs," *arXiv preprint arXiv:2005.031806*, 2020.

[27] M. Budišić, R. Mohr, and I. Mezić, "Applied koopmanism," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, p. 047510, 2012.

[28] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[29] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[30] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 10, p. 103111, 2017.

[31] MSC.Software, *MSC.Nastran Superelement User's Guide*, MSC.Software Corporation, CA, USA, 2004.

[32] R. R. Craig Jr and A. J. Kurdila, *Fundamentals of structural dynamics*. John Wiley & Sons, 2006.

[33] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, *Model Order Reduction: Volume 3: Applications*. De Gruyter, 2020.

[34] P. V. Thomas, M. S. ElSayed, and D. Walch, "Review of model order reduction methods and their applications in aeroelasticity loads analysis for design optimization of complex airframes," *Journal of Aerospace Engineering*, vol. 32, no. 2, 2019.

[35] R. J. Guyan, "Reduction of stiffness and mass matrices," *AIAA journal*, vol. 3, no. 2, pp. 380–380, 1965.

[36] A. Y.-T. Leung, "An accurate method of dynamic condensation in structural analysis," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 11, pp. 1705–1715, 1978.

[37] D. J. Rixen, "A dual Craig–Bampton method for dynamic substructuring," *Journal of Computational and applied mathematics*, vol. 168, no. 1-2, pp. 383–391, 2004.

[38] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1576, 2018.

[39] T. Taddei, J. Penn, M. Yano, and A. T. Patera, "Simulation-based classification; a model-order-reduction approach for structural health monitoring," *Archives of Computational Methods in Engineering*, vol. 25, no. 1, pp. 23–45, 2018.

[40] N. Dal Santo, A. Manzoni, S. Pagani, and A. Quarteroni, "Reduced-order modeling for applications to the cardiovascular system," in *Model Order Reduction Volume III: Applications*. De Gruyter, 2020, pp. 251–278.

[41] B. Haasdonk, "MOR software," in *Model Order Reduction Volume III: Applications*. De Gruyter, 2020, pp. 431–460.