Two-step greedy algorithm for reduced order quadratures

**Harbir Antil, Scott E. Field, Frank Herrmann, Ricardo H. Nochetto,
Manuel Tiglio**

Abstract We present an algorithm to generate application-specific, global *reduced order quadratures* (ROQ) for multiple fast evaluations of weighted inner products between parameterized functions. If a reduced basis (RB) or any other projection-based model reduction technique is applied, the dimensionality of integrands is reduced dramatically; however, the cost of approximating the integrands by projection still scales as the size of the original problem. In contrast, using discrete empirical interpolation (DEIM) points as ROQ nodes leads to a computational cost which depends linearly on the dimension of the reduced space. Generation of a reduced basis via a greedy procedure requires a training set, which for products of functions can be very large. Since this direct approach can be impractical in many applications, we propose instead a two-step greedy targeted towards approximation of such products. We present numerical experiments demonstrating the accuracy and the efficiency of the two-step approach. The presented ROQ are expected to display very fast convergence whenever there is regularity with respect to parameter variation. We find that for the particular application here considered, one driven by gravitational wave physics, the two-step approach speeds up the offline computations to build the ROQ by more than two orders of magnitude. Furthermore, the resulting ROQ rule is found to converge exponentially with the number of nodes, and a factor of ~ 50 savings, without loss of accuracy, is observed in evaluations of inner products when ROQ are used as a downsampling strategy for equidistant samples using the trapezoidal rule. While the primary focus of this paper is on quadrature rules for inner products of parameterized functions, our method can be easily adapted to integrations of single parameterized functions, and some examples of this type are considered.

1 Introduction

Many application areas deal with parameterized problems. Throughout this paper, we consider these to be a set of functions

$$\mathcal{F}_W := \{h_\mu : \Omega \to \mathbb{C} \mid \mu \in \mathcal{P}, h_\mu \in \mathcal{C}(\Omega)\},$$

Harbir Antil
Department of Mathematical Sciences. George Mason University, Fairfax, VA 22030, USA. Tel.: +703-993-2390
Fax: +703-993-1491
E-mail: hantil@gmu.edu

Scott E. Field
Department of Physics, Joint Space Science Institute, Maryland Center for Fundamental Physics. University of Maryland, College Park, MD 20742, USA.

Frank Herrmann
Center for Scientific Computation and Mathematical Modeling, Department of Physics, Maryland Center for Fundamental Physics. University of Maryland, College Park, MD 20742, USA.

Ricardo H. Nochetto
Department of Mathematics, and Institute of Physical Science and Technology, University of Maryland, College Park, MD 20742, USA.

Manuel Tiglio
Center for Scientific Computation and Mathematical Modeling, Department of Physics, Joint Space Science Institute, Maryland Center for Fundamental Physics. University of Maryland, College Park, MD 20742, USA.

where Ω, $\mathcal{P}$ denote the physical and parameter domains, respectively, $\mathbb{C}$ is the set of complex numbers, $\mathcal{C}(\Omega)$ is the set of continuous functions on a compact domain $\Omega \subset \mathbb{R}$ and $\mathcal{F}_W \subset \mathcal{H}_W$ denotes a compact subset of the Hilbert space $\mathcal{H}_W := L^2_W(\Omega)$. In general, both Ω and $\mathcal{P}$ are multi-dimensional and can be irregular domains. Here we assume $\mathcal{P}$ to be compact in $\mathbb{R}^N$ and take the scalar product and norm between two functions $f, g \in \mathcal{H}_W$ to be

$$\langle f, g\rangle_{L^2_W} := \int_\Omega f^*(x)g(x)W(x)dx, \qquad \|f\|^2_{L^2_W} := \langle f, f\rangle_{L^2_W}, \qquad (1)$$

with $0 < W \in \mathcal{C}(\Omega)$ some weight function and f^* denoting complex conjugation. We absorb the weight above into each integrand function through[1]

$$f(x) \to W^{\frac{1}{2}}(x)f(x),$$

with $x \in \Omega$, and define the set of functions

$$\mathcal{F} = \{h_\mu \sqrt{W} \in \mathcal{C}(\Omega) \mid h_\mu \in \mathcal{F}_W\},$$

which is now a compact subset of the Hilbert space $\mathcal{H} := L^2(\Omega)$. This allows us, without loss of generality, to use the standard L^2 inner product and the corresponding norm for elements of $\mathcal{H}$, i.e.,

$$\langle f, g\rangle = \int_\Omega f^*(x)g(x)\,dx, \qquad \|f\|^2 := \langle f, f\rangle. \qquad (2)$$

Furthermore, the discrete L^2 inner product and the corresponding norm are here denoted by

$$\langle f, g\rangle_{\mathtt{d}} = \sum_{k=1}^M \omega_k f^*(x_k)g(x_k), \qquad \|f\|^2_{\mathtt{d}} = \langle f, f\rangle_{\mathtt{d}}, \qquad (3)$$

where $\{x_k, \omega_k\}_{k=1}^M$ are arbitrary quadrature points and weights. Additionally, we often denote discrete objects and vectors through bold notation, for example, $\mathbf{f} = \left(f(x_1), \ldots, f(x_M)\right)^T$ and $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_M)^T$. More details about the notation used throughout this paper are given in Table 1.

There are several challenges associated with parameterized problems and Reduced Order Modeling (ROM). One of them, when the set of functions $\mathcal{F}$ is not known a priori but requires expensive numerical simulations, is how to select "on the fly" which parameters to solve for in a nearly optimal way, and a compact –application-specific spectral– representation for every element in $\mathcal{F}$. Reduced Basis (RB) [1] is a leading candidate for such problems. We refer to Section 3.1 for details about it; for the purposes of this paper it suffices to describe in that section the algorithm and some of its key features. Another often desired aspect of any approach to parameterized problems is, once a reduced basis has been built, the ability to accelerate the computation of a particular quantity of interest. This can be fast online prediction of new solutions by solving a reduced problem [1] or, as in the case of this paper, fast online evaluations of inner products between elements of $\mathcal{F}$. Additional benefits may be gained from an interpolation-based approximation. The leading candidates for this are the Empirical Interpolation Method (EIM) [2,3] and its discrete counterpart, the Discrete Empirical Interpolation Method (DEIM) [4,5]. Again, for the purposes of this paper it suffices to summarize the algorithm in Section 3.2 and some of its properties.

The specific goal of this paper is to build application-specific global quadratures for **inner products of functions** (such as those appearing in convolutions, matched filtering, Markov Chain Monte Carlo simulations, etc.) combining both ROM and the DEIM, which for briefness we refer to as *reduced order quadratures* (ROQ). Here we explicitly consider RB for ROM, but the approach remains the same for other choices of bases generated using, for example, Proper Orthogonal or Singular Value Decompositions (POD, SVD) [6].

The construction of ROQ is somewhat similar in spirit to that one of Gaussian quadratures: the function to be integrated is approximated by a truncated expansion in a reduced basis, this expansion is replaced by interpolation (here DEIM) at special nodal points and, finally, the interpolant integrated to compute effective quadrature weights for any scalar product of functions in the set of interest $\mathcal{F}$. For many applications of interest, there are several advantages of ROQ with respect to Gaussian quadratures, though. First, regularity with respect to parameter variation, not with respect to the integration variable, is exploited to provide very fast (in many cases of interest, exponentially fast) convergence. For example, for the gravitational wave test cases considered in this paper, ROQ prove to be more efficient than Gaussian

---

[1] For generic target applications we envision these functions not to be polynomials, and the weights not to be standard ones. In particular, our proposed reduced order quadrature construction does not rely on classical orthogonal polynomial theory for which the precise form of the weight W is essential.

quadratures (see Fig. 9). In addition, the method is adaptive by nature, and the quadrature nodes are hierarchical. Our approach also gives, when acquiring data from experiments or observations, a nearly optimal way of downsampling it for matched filtering [7, 8] or other purposes, while at the same time preserving the accuracy in computing integrals with the full data set. All these benefits of ROQ come at the cost of a potentially expensive offline procedure to pre-assemble both the basis and the ROQ rule. For many applications where multiple fast evaluations are required, especially if they are needed online or in real time, this tradeoff is indeed desirable.

To the best of our knowledge, leveraging the advantages of the Empirical Interpolation Method for fast numerical integration was first suggested by Maday et al [2] (see also Refs. [9, 10]) and further investigations were carried out by Aanonsen [10]. However, our approach differs from previous ones in several respects. First, the empirical interpolant coefficients are usually found by carrying out a potentially costly (compared to a competitive quadrature rule) matrix-vector multiplication. Here we absorb this cost into an *offline* computation of the reduced order quadrature weights and explicitly write the ROQ rule as a vector-vector product (see Eq. (10) in Section 2). Additionally, the RB-DEIM model reduction used here allows for a natural factorization of the parameter and physical dependences. Finally and most important, our main objective here is fast computation of **inner products** of the form $\langle h_{\mu_i}, h_{\mu_j} \rangle$. While this is closely related to the integration of single functions in $\mathcal{F}$, for large problems there are practical and serious obstacles towards a reduced basis representation of products of functions, denoted here as

$$\widetilde{\mathcal{F}} = \{ h_{\mu_i}^* h_{\mu_j} \in \mathcal{C}\left(\Omega\right) \mid h_{\mu_i}, h_{\mu_j} \in \mathcal{F} \}, \tag{4}$$

in terms of very large training spaces. To address this we suggest a simple two-step approach targeted towards such products which reuses the algorithms necessary for reducing the underlying lower order space $\mathcal{F}$, with dramatic savings in the offline stage. Without this two-step procedure, building ROQ would be simply not possible in many realistic application problems driving our work, even if carried out offline and using large supercomputers.

This paper is organized as follows. Section 2 provides an overview of the main ideas to be developed as well as brief preliminary discussions on quadrature rules, RB-DEIM, and reduced order quadratures. Section 3 collects necessary results and algorithms required for developing ROQ. In particular, in Section 3.1 we describe a RB-greedy model reduction algorithm to generate a set of basis vectors which are subsequently interpolated at a set of points chosen by the DEIM algorithm as summarized in Section 3.2. The main difficulties in constructing a reduced basis for $\widetilde{\mathcal{F}}$, as well as our approach for overcoming them, are considered in Section 3.3. Fast integrations using the reduced model and ROQ are constructed in Section 4.1, along with some error estimates in Section 4.2. Straightforward extensions of the ROQ algorithm for resampled functions are discussed in Section 4.3. Two numerical experiments are documented in Section 5. To compare with well known results in one and two dimensions, our first experiment considers polynomials on the standard interval $[-1, 1]$ as basis, and integration of single functions. The second example draws from a non-trivial problem in gravitational wave physics and showcases the potential savings of ROQ to compute scalar products in large problems. In Section 6 we summarize the results of this paper, and comment on their possible extensions.

2 Overview and main ideas

The main goal of this paper is to efficiently compute approximations to integrals such as those in Eq. (2). We refer to the exact, continuum one as I_c,

$$I_c(i, j) := \langle h_{\mu_i}, h_{\mu_j} \rangle, \tag{5}$$

where $\{\mu_i, \mu_j\}$ are any two values in the parameter space $\mathcal{P}$. Next we discuss different approximations to I_c.

Integration by quadrature: Let us recall a typical quadrature rule, since the proposed ROQ of this paper follow a somewhat similar pattern. With $\{x_k, \omega_k\}_{k=1}^M$ denoting an arbitrary set of quadrature points and weights, respectively, we can approximate the integral (5) as

$$I_c(i, j) \approx I_{\mathtt{d}}(i, j) := \langle h_{\mu_i}, h_{\mu_j} \rangle_{\mathtt{d}} = \sum_{k=1}^{M} \omega_k h_{\mu_i}^*(x_k) \, h_{\mu_j}(x_k). \tag{6}$$

Many integration rules can be written in the form of Eq. (6), including the extended trapezoidal rule, Gaussian quadratures, and integration using domain decomposition. Among these many choices an optimal strategy should allow for a given target integration accuracy with the smallest number of operations. However, if M is large (perhaps the functions are sharply peaked, have different length scales and/or many cycles) and/or many evaluations of I_c are needed (perhaps in real-time), computing the approximations $I_{\mathtt{d}}$ can become very costly. Problems which depend on a high dimensional spaces constitute such cases. In addition, if the target functions $\{h_{\mu_j}\}$ come from acquired data, each function h_{μ_j} is

usually sampled at equally spaced or scattered points and the integrations are usually carried out using an extended trapezoidal rule, with slow convergence in general.

Integration in a reduced space: A preliminary reduced order approximation to $I_\mathtt{d}(i, j)$ would involve using an expansion in basis vectors $\{e_\ell\}_{\ell=1}^n$. Without loss of generality we can assume that the basis is orthonormal with respect to the discrete [2] inner product $\langle \cdot, \cdot \rangle_\mathtt{d}$. Then any function $h_\mu \in \mathcal{F}$ can be approximated by

$$h_\mu \approx \mathcal{P}_n h_\mu := \sum_{\ell=1}^n \langle e_\ell, h_\mu \rangle_\mathtt{d} e_\ell \,,$$

where we have introduced the orthogonal projection operator $\mathcal{P}_n$, and an approximation to (6) would be

$$I_\mathtt{d}(i, j) \approx \langle \mathcal{P}_n h_{\mu_i}, \mathcal{P}_n h_{\mu_j} \rangle_\mathtt{d} = \sum_{k=1}^n \langle h_{\mu_i}, e_k \rangle_\mathtt{d} \langle e_k, h_{\mu_j} \rangle_\mathtt{d} \,. \tag{7}$$

Now, computing the approximation (7) requires knowledge of the projection coefficients $\left\{ \langle h_{\mu_i}, e_k \rangle_\mathtt{d} \right\}$. Modulo the latter, performing an integration via Eq. (7) is of improved computational cost (n multiplications and $n-1$ additions) whenever the number of basis elements is smaller than the number of quadrature points, $n < M$. A greedy construction of a *reduced basis* (see Sec. 3.1), as opposed to a standard basis choice such as Jacobi polynomials, involves the use of a problem-dependent *training space*. If the functions to be integrated, namely h_{μ_i} and h_{μ_j}, are members of this training space, the projection coefficients have been precomputed offline while building the basis (see Section 3.1). The more interesting case in practice is that one in which h_{μ_i} and h_{μ_j} were not members of the training space, or one is unable to store the set of projection coefficients.

Integration with reduced order quadratures (ROQ): We advocate an application-specific quadrature scheme for efficiently evaluating the discrete integral in (6), with essentially no loss of accuracy, with a computational cost that depends only on the reduced basis space dimension, and without requiring projection coefficients. This new approach combines the flexibility of quadrature rules with powerful dimensionality reduction. We seek to approximate integrands of the form $g_{ij} = h_{\mu_i}^* h_{\mu_j}$,

$$g_{ij} \approx \widetilde{\mathcal{P}}_m g_{ij} := \sum_{\ell=1}^m \langle \widetilde{e}_\ell, g_{ij} \rangle_\mathtt{d} \widetilde{e}_\ell \,, \tag{8}$$

with $g_{ij} \in \widetilde{\mathcal{F}}$, and a truncated expansion is carried with a set of basis vectors $\{\widetilde{e}_\ell\}_{\ell=1}^m$ approximating elements in $\widetilde{\mathcal{F}}$. As before, we are faced with computing $\langle \widetilde{e}_\ell, g_{ij} \rangle_\mathtt{d}$, the cost of which can in principle be expensive. The empirical interpolation method (EIM) offers an attractive alternative: consider these m basis $\{\widetilde{e}_\ell\}_{\ell=1}^m$, let $\{\widetilde{p}_\ell\}_{\ell=1}^m \subset \{x_k\}_{k=1}^M$ be a set of points (the generation of which will be described later), and let the EIM interpolant of g_{ij} be

$$\widetilde{\mathcal{I}}_m[g_{ij}] := \sum_{\ell=1}^m \widetilde{c}_\ell \widetilde{e}_\ell \quad \text{s.t.} \quad \sum_{\ell=1}^m \widetilde{c}_\ell \widetilde{e}_\ell(\widetilde{p}_k) = g_{ij}(\widetilde{p}_k), \quad k = 1, \dots, m \,,$$

where the coefficients $\{\widetilde{c}_\ell\}_{\ell=1}^m$ solve the interpolation problem. If the interpolant is accurate then $g_{ij} \approx \widetilde{\mathcal{I}}_m[g_{ij}]$ and, substituting this into Eq.(6), our approximation to the quadrature rule $\{x_k, \omega_k\}_{k=1}^M$ becomes

$$I_d(i, j) \approx \sum_{k=1}^M \omega_k \widetilde{\mathcal{I}}_m[g_{ij}](x_k) = \sum_{k=1}^M \omega_k \sum_{\ell=1}^m \widetilde{c}_\ell \widetilde{e}_\ell(x_k) = \sum_{\ell=1}^m \left[\sum_{k=1}^M \omega_k \widetilde{e}_\ell(x_k) \right] \widetilde{c}_\ell \,. \tag{9}$$

Written in this way, the bracketed expression in the last term of Eq. (9) is seen to be the integration of the basis. Later on (cf. Eq.(23)) we show that this expression can be rewritten in a more recognizable form

$$I_\mathtt{d}(i, j) \approx I_{\text{ROQ}}(i, j) := \sum_{\ell=1}^m \omega_\ell^{\text{ROQ}} h_{\mu_i}^*(\widetilde{p}_\ell) h_{\mu_j}(\widetilde{p}_\ell) = \sum_{\ell=1}^m \omega_\ell^{\text{ROQ}} g_{ij}(\widetilde{p}_\ell) \,. \tag{10}$$

Eq. (10) is our proposed Reduced Order Quadrature, which has an online evaluation cost of $\mathcal{O}(m)$. The interpolation points $\{\widetilde{p}_\ell\}_{\ell=1}^m \subset \{x_k\}_{k=1}^M$ are outputs of the EIM algorithm. Both these points and the weights ω_ℓ^{ROQ} only depend on

---

[2] In carrying out the RB-greedy algorithm we resolve all the integrals involved within machine precision, so for all practical purposes within that context the discrete and continuum scalar products agree with each other.

$\mathcal{F}$:	set of target functions.	
μ:	parameter in set $\mathcal{P}$.	
h_μ:	a sample target function in $\mathcal{F}$.	
$\widetilde{\mathcal{F}}$:	set of product of target functions.	
g_μ:	a sample target function in $\widetilde{\mathcal{F}}$.	
$\widetilde{\mu}$:	parameter in Cartesian product set $\mathcal{P} \times \mathcal{P}$	
$\{e_\ell\}_{\ell=1}^n$:	basis whose span F_n approximates $\mathcal{F}$.	
$\mathcal{P}_n$:	orthogonal projection to F_n.	
$\mathcal{I}_n[]$:	empirical interpolation associated with F_n.	
$\{\widetilde{e}_\ell\}_{\ell=1}^m$:	basis whose span $\widetilde{F}_m$ approximates $\widetilde{\mathcal{F}}$.	
$\widetilde{\mathcal{P}}_m$:	orthogonal projection to $\widetilde{F}_m$.	
$\widetilde{\mathcal{I}}_m[]$:	empirical interpolation associated with $\widetilde{F}_m$.	
M:	number of samples to evaluate discrete inner products.	
n:	number of EIM points (and reduced basis) to approximate elements in $\mathcal{F}$.	
m:	number of EIM points (and reduced basis) to approximate elements in $\widetilde{\mathcal{F}}$. Number of ROQ points.	
$\mathcal{F}_{\text{train}}$:	training space sampling $\mathcal{F}$.	
K:	number of elements in $\mathcal{F}_{\text{train}}$.	
K^2:	number of training space elements in principle needed to *directly* sample $\widetilde{\mathcal{F}}$.	
n^2:	number of training space elements used in our two-step greedy approach to sample $\widetilde{\mathcal{F}}$. In general $n^2 \ll K^2$.	

Table 1 NOTATION. Notice that tildes refer to approximations or quantities related to $\widetilde{\mathcal{F}}$, defined in Eq. (4).

the underlying quadrature rule given by $\{x_k, \omega_k\}_{k=1}^M$ and on the choice of reduced basis vectors $\widetilde{e}_\ell$; they are precomputed off-line and do not depend on the integrand g_{ij}. Note that the ROQ rule (10) achieves our efficiency requirements whenever $m < M$. In many cases one can expect exponential convergence of $I_{\text{ROQ}} \to I_{\text{d}}$ with respect to m, and for $I_{\text{d}} \approx I_c$ we expect this convergence rate to be inherited by the limit to the continuum $I_{\text{ROQ}} \to I_c$. Furthermore, even though this might be problem-dependent, in our numerical experiments (see Section 5.2) we have found that $m \sim 2n \ll M$ (the equality $m = 2n$ exactly holds for polynomial bases). For the case $m = M$, when all degrees of freedom have been exhausted, consistency requires both quadrature rules $\{\widetilde{p}_\ell, \omega_\ell^{\text{ROQ}}\}_{\ell=1}^m$ and $\{x_k, \omega_k\}_{k=1}^M$ to be identical. Indeed, in this case the quadratures points trivially coincide and one can show $\omega^{\text{ROQ}} = \omega$ (see Corollary 1).

A straightforward and largely self contained blueprint to generate the quadrature rule $\{\widetilde{p}_\ell, \omega_\ell^{\text{ROQ}}\}_{\ell=1}^m$ is given in Algorithm 2.

Approximation of products with a two-step greedy: The ROQ rule requires an approximation of integrands, which is where the main challenge lies; namely, in integrating products of target functions and not individual ones. A greedy construction of a reduced basis space approximating $\mathcal{F}$ involves the use of a *training set* $\mathcal{T}_K := \{\mu_i\}_{i=1}^K \subset \mathcal{P}$ of size K which densely samples the continuum $\mathcal{P}$. If a faithful approximation of $\mathcal{F}$ requires a training set $\mathcal{T}_K$ of size $\mathcal{O}(K)$ then a training set for $\widetilde{\mathcal{F}}$ would in principle be of size $\mathcal{O}(K^2)$. Since for large problems $\mathcal{O}(K)$ can already be computationally challenging, this direct approach might be unfeasible in some applications, even when building the basis is an offline and parallelizable calculation.

A greedy algorithm for $\mathcal{F}$ returns a set of n *greedy points* $\{\mu_\ell\}_{\ell=1}^n$ and orthonormal reduced basis $\{e_\ell\}_{\ell=1}^n$. The basic idea of the two-step greedy approach is to take advantage of the observation that all integrands can be accurately approximated by

$$h_{\mu_i}^* h_{\mu_j} \approx (\mathcal{P}_n h_{\mu_i})^* \mathcal{P}_n h_{\mu_j} = \left(\sum_{k=1}^n \langle e_k, h_{\mu_i}\rangle_{\text{d}} e_k\right)^* \left(\sum_{l=1}^n \langle e_l, h_{\mu_j}\rangle_{\text{d}} e_l\right),$$

which involves a sum of n^2 terms of the form $e_k^* e_l$. We then carry out a second layer of dimensional reduction – a second greedy for the products $\{e_k^* e_l\}_{k,l=1}^n$. The training set for this second greedy is of size $\mathcal{O}(n^2)$, as opposed to $\mathcal{O}(K^2)$ (with, usually, $n^2 \ll K^2$). One may also consider a training set for $\widetilde{\mathcal{F}}$ given by the Cartesian product of greedy points $\{\mu_i\}_{i=1}^n$ thereby sampling $\widetilde{\mathcal{F}}$ is a smarter way. Both ideas are more thoroughly explored in Sec. 3.3.

3 Reduced basis and the empirical interpolation method

The ROQ rule proposed here requires an accurate interpolation representation for all integrands in the space of interest. In the next two subsections we describe the necessary RB-greedy and EIM algorithms to achieve this. Here we favor an RB-greedy approach, since it is able to handle very large problems, such as those that we are interested in. Our proposal is also applicable to other basis choices such as those found through a Proper Orthogonal/Singular Value

Decomposition [6]. In Sec. 3.3 we propose a two-step RB-greedy approach for large sets of products of functions. Without loss of generality we assume all functions to be normalized with respect to the discrete inner product $\langle \cdot, \cdot \rangle_{\mathsf{d}}$.

3.1 Greedy construction of a reduced basis

The RB approach, combined with a greedy algorithm to generate the basis, provides a way to construct an application-specific expansion, where the basis elements are members of the space under consideration itself. Here the sets of interest are $\mathcal{F}$ and $\widetilde{\mathcal{F}}$. For definiteness in notation we will describe the approach for $\mathcal{F}$, since an approach for $\widetilde{\mathcal{F}}$ or any other space is identical; we will refer to this approach applied to $\widetilde{\mathcal{F}}$ as a *direct* or one-step RB-greedy.

A greedy approach is highly efficient in practice. It yields a nested basis set that is hierarchically constructed. As is customary for spectral expansions, having an orthonormal basis $\{e_\ell\}_{\ell=1}^n$ simplifies computing the orthogonal projection $\mathcal{P}_n h_\mu$ with respect to $\langle \cdot, \cdot \rangle_{\mathsf{d}}$ of a function $h_\mu \in \mathcal{F}$ onto the span F_n of the basis,

$$F_n := \mathrm{span}\{e_1, \ldots, e_n\}\,.$$

We recall (see, for example, [11]) that the Kolmogorov n-width of $\mathcal{F}$ in $\mathcal{H}$

$$d_n(\mathcal{F}; \mathcal{H}) := \inf_{\dim X_n \le n} \sup_{h_\mu \in \mathcal{F}} \inf_{f \in X_n} \left\| h_\mu - f \right\| = \inf_{\dim X_n \le n} \sup_{h_\mu \in \mathcal{F}} \left\| h_\mu - \mathcal{P}_n h_\mu \right\|\,, \tag{11}$$

measures the error of the best n-dimensional subspace $X_n \subset \mathcal{H}$ approximating $\mathcal{F}$, and the last equality follows from $\mathcal{H}$ being a Hilbert space. Computation of the n-width d_n, or any basis achieving it, is in most practical applications not possible (however, see [11] for some cases where it can be done). Nevertheless, obtaining a convergence rate for the n-width provides valuable information towards understanding the approximability of a space by greedy algorithms.

We summarize a greedy strategy to build a reduced basis $\{e_\ell\}_{\ell=1}^n$, with an approximation error

$$\sigma_n(\mathcal{F}; \mathcal{H}) := \sup_{h_\mu \in \mathcal{F}} \left\| h_\mu - \mathcal{P}_n h_\mu \right\|\,, \tag{12}$$

which is nearly optimal with respect to the Kolmogorov n-width defined in Eq. (11). Then we will state some of the available convergence rates for the *greedy error* $\sigma_n(\mathcal{F}; \mathcal{H})$ based on [12,13], which make precise the quoted near optimality, under the assumption that up to machine precision $\left\| h_\mu - \mathcal{P}_n h_\mu \right\| \simeq \left\| h_\mu - \mathcal{P}_n h_\mu \right\|_{\mathsf{d}}$. While the greedy error (12) has been defined over $\mathcal{F}$, in practice one samples the continuum using a training set $\mathcal{T}_K := \{\mu_i\}_{i=1}^K \subset \mathcal{P}$ of size K and a *training space* $\mathcal{F}_{\mathrm{train}}$ of associated normalized functions $\{h_{\mu_i}\}_{i=1}^K$. Typically, if there is redundancy, the number of reduced basis needed to represent the training space is much smaller than the number of samples: $n \ll K$.

Next let $\epsilon > 0$ be a user-specified error tolerance. Its role is to guarantee that the approximation error ensured by the RB-Greedy algorithm is strictly bounded as

$$\sigma_n(\mathcal{F}_{\mathrm{train}}; \mathcal{H}) := \sup_{h_\mu \in \mathcal{F}_{\mathrm{train}}} \left\| h_\mu - \mathcal{P}_n h_\mu \right\|_{\mathsf{d}} \le \epsilon\,. \tag{13}$$

To a-priori ensure that the training space $\mathcal{F}_{\mathrm{train}}$ is a faithful approximation of the continuum $\mathcal{F}$ is, in general, difficult. Construction and adaptive management of the training space $\mathcal{F}_{\mathrm{train}}$ is an area of active research; see, for example, [14, 9,15,16]. A good choice is dictated by the problem, further details for our application are given in the numerical results Section 5.2. To check that the reduced basis space F_n is a faithful approximation of $\mathcal{F}$ we typically do convergence tests with respect to the number of samples K in $\mathcal{T}_K$, and Monte Carlo reconstruction studies of the continuum; see for example [17,18,19] and Section 5.2. For any given tolerance $\epsilon > 0$, the Algorithm 4 of Appendix A ensures the strict bound (13) over the entire training space. However, a-posteriori validation has allowed us to establish a more impressive bound in all of our applications in gravitational wave physics so far [17,18,19],

$$\| h_\mu - \mathcal{P}_n h_\mu \|_{\mathsf{d}} \le \sigma_n(\mathcal{F}; \mathcal{H}) \lesssim \epsilon \quad \forall \mu \in \mathcal{P}\,. \tag{14}$$

Such observation is obviously problem-dependent, and in general it should read $\sigma_n(\mathcal{F}; \mathcal{H}) \le \widetilde{\epsilon}$ where in principle $\widetilde{\epsilon} \ge \epsilon$.

For any $\epsilon > 0$ and training set $\mathcal{T}_K$ the greedy algorithm to build a reduced basis is given in App. A. This algorithm returns a nested, hierarchical set of n greedy points $\{\mu_\ell\}_{\ell=1}^n$ and orthonormal reduced basis $\{e_\ell\}_{\ell=1}^n$ which are nearly optimal with respect to the n-width (11), in the following sense. Recall from [13, Corollary 3.3] that if the n-width defined in (11) decays exponentially, then

$$d_n(\mathcal{F}; \mathcal{H}) \le C e^{-c_0 n^\alpha} \quad \rightarrow \quad \sigma_n(\mathcal{F}; \mathcal{H}) \le \sqrt{2C} e^{-c_1 n^\alpha}\,, \tag{15}$$

whence the greedy error in (12) decays exponentially, where C, c_0, α, and $c_1 := 2^{-1-2\alpha} c_0$ are positive constants. Similar statements can be made if the Kolmogorov n-width decays with a polynomial order,

$$ d_n(\mathcal{F}; \mathcal{H}) \leq C_a n^{-\beta} \quad \rightarrow \quad \sigma_n(\mathcal{F}; \mathcal{H}) \leq 2^{5\beta+1} C_a n^{-\beta} , \tag{16} $$

where C_a, β are positive constants.

In Section 5.2 we summarize the observed exponential decay of the greedy error for a particular family of gravitational wave solutions. We have also found that for a fixed representation error and parameter domain $\mathcal{P}$ we can reconstruct any member function in $\mathcal{F}$ with a *finite number of basis* elements. That is, as the size of $\mathcal{T}_K$ increases, a finite asymptotic number of basis is needed to represent it; see [20, 21, 18, 17, 19] for further studies.

Remark 1 While it is obvious that $d_n(\mathcal{F}; \mathcal{H}) \leq \sigma_n(\mathcal{F}; \mathcal{H})$, a direct comparison obtained in [13, Corollary 3.3] reads $\sigma_n(\mathcal{F}; \mathcal{H}) \leq \sqrt{2\, d_{n/2}(\mathcal{F}; \mathcal{H})}$ for any n-width decay rate. If $d_{n/2}(\mathcal{F}; \mathcal{H})$ decays exponentially, then one gets (15). Similarly, if $d_{n/2}(\mathcal{F}; \mathcal{H})$ decays with a polynomial order, then one has (16).

We now turn to some immediate applications of the reduced space expressed with matrix-vector notation, which will prove convenient throughout the remainder of this paper. Let

$$ \mathbf{V} := [\mathbf{e}_1, \ldots, \mathbf{e}_n] \in \mathbb{C}^{M \times n} , $$

where the ℓth column of the matrix $\mathbf{V}$ is $\mathbf{e}_\ell = \left(e_\ell(x_1), \ldots, e_\ell(x_M) \right)^T \in \mathbb{C}^M$; i.e., it corresponds to the ℓth reduced basis e_ℓ sampled at the set of quadrature points $\{x_i\}_{i=1}^M$ used in the greedy algorithm[3]. In matrix-vector notation $\mathcal{P}_n \mathbf{h}_\mu = \mathbf{V} \left[\mathbf{V}^\dagger (\boldsymbol{\omega} \circ \mathbf{h}_\mu) \right]$ and the reduced discrete integral (7) becomes

$$ I_{\mathtt{d}}(i,j) \approx \langle \mathcal{P}_n h_{\mu_i}, \mathcal{P}_n h_{\mu_j} \rangle_{\mathtt{d}} = \sum_{k=1}^n \langle h_{\mu_i}, e_k \rangle_{\mathtt{d}} \langle e_k, h_{\mu_j} \rangle_{\mathtt{d}} = \left(\mathbf{V}^\dagger (\boldsymbol{\omega} \circ \mathbf{h}_{\mu_i}) \right)^\dagger \left(\mathbf{V}^\dagger (\boldsymbol{\omega} \circ \mathbf{h}_{\mu_j}) \right) , \tag{17} $$

where $\mathbf{V}^\dagger$ denotes the conjugate transpose of $\mathbf{V}$ and $\circ$ denotes the Hadamard (pointwise) product between the vector components. Here we have reduced the overall *dimensionality* of the problem to n, but two issues remain. First, the number of $\{h_\mu(x_k)\}_{k=1}^M$ functional evaluations depends on M (Sec. 3.3 of Ref. [4] discusses the issue for a variety of nonlinear parametrized functions). Second, the dominant operation count for the approximation is seen to be of order $\mathcal{O}(Mn)$. We will return to these issues shortly.

3.2 The empirical interpolation method

The main idea behind the EIM is to replace an expensive approximation by projection with a relatively inexpensive interpolation, without sacrificing accuracy. The algorithm identifies a set of basis-specific interpolation points through a greedy selection criteria. When applied to several physical dimensions, irregular shaped domains, and rather generic parameterized spaces, the EIM [2, 10, 9] and its discrete counterpart, the DEIM [4, 5], and points have shown remarkable robustness and efficiency.

Suppose we have a set of basis vectors. In particular, the algorithm applies to the reduced basis for the space F_n or the space $\widetilde{F}_m$ to be constructed in Sec. 3.3. The algorithm selects the DEIM points in physical space Ω and builds an associated interpolation matrix $\mathbf{P}$. This matrix $\mathbf{P}$ interpolates the columns of the matrix $\mathbf{V}$ introduced in Sec. 3.1 (see Appendix A.2). The columns of $\mathbf{P}$ are unit vectors with a single unit entry at the location of the empirical interpolation points and zero elsewhere. For example, the first column of $\mathbf{P}$ has a unit entry at the location $\arg\max |\mathbf{e}_1|$ (see step 2 of Algorithm 5). Furthermore, $\mathbf{P}^T \mathbf{h}_\mu$ extracts an n-subvector which is exactly equivalent to evaluating h_μ at n empirical interpolation points $\{p_i\}_{i=1}^n$. The algorithm to generate $\mathbf{P}$ and $\{p_i\}_{i=1}^n$ is given in App. A.

With $\{p_i\}_{i=1}^n$ the DEIM approximation is [4]

$$ \mathbf{h}_\mu \approx \mathcal{I}_n[\mathbf{h}_\mu] := \mathbf{V}(\mathbf{P}^T \mathbf{V})^{-1} \mathbf{P}^T \mathbf{h}_\mu , \tag{18} $$

which is indeed an interpolant $\mathbf{P}^T \mathcal{I}_n[\mathbf{h}_\mu] = \mathbf{P}^T (\mathbf{V}(\mathbf{P}^T \mathbf{V})^{-1} \mathbf{P}^T \mathbf{h}_\mu) = \mathbf{P}^T \mathbf{h}_\mu$. Note that $\mathbf{P}^T \mathbf{V}$ is invertible when the columns of $\mathbf{V}$ are linearly independent. Substituting $\mathcal{I}_n[\mathbf{h}_\mu]$ from Eq. (18) into Eq. (6) gives

$$ I_{\mathtt{d}}(i,j) \approx \langle \mathcal{I}_n[\mathbf{h}_{\mu_i}], \mathcal{I}_n[\mathbf{h}_{\mu_j}] \rangle_{\mathtt{d}} \approx \left((\mathbf{P}^T \mathbf{V})^{-1} \mathbf{P}^T \mathbf{h}_{\mu_i} \right)^\dagger \left((\mathbf{P}^T \mathbf{V})^{-1} \mathbf{P}^T \mathbf{h}_{\mu_j} \right) . \tag{19} $$

---

[3] For certain applications it may be desirable to evaluate the functions at a different set of points $\{y_i\}_{i=1}^{M'}$, see Section 4.3 for details.

As we only have to evaluate $\mathbf{P}^T \mathbf{h}_\mu \in \mathbb{C}^n$, with $n < M$ in many applications, we have reduced the complexity (that is, the number of required functional evaluations) to $\mathcal{O}(n)$ (compare with (17)). However, due to the two matrix-vector multiplications an evaluation cost of $\mathcal{O}(n^2)$ has appeared. Such unacceptable scaling arises when approximating the individual functions h_μ instead of the products (integrands) $h_{\mu_i}^* h_{\mu_j}$ directly. We consider the generation of a reduced basis for products of functions next.

3.3 Two-step greedy approach for the approximation of product of target functions

$$\widetilde{\mathcal{F}} \xrightarrow[\underset{Path\ \#2\ (i)}{}]{\overset{\nearrow \quad \overset{Path\ \#1}{\longrightarrow} \quad \searrow}{}} F_{n^2} \xrightarrow[Path\ \#2\ (ii)]{} \widetilde{F}_m \tag{20}$$

As discussed at the end of Secs. 3.1 and 3.2, we seek to approximate all possible integrands $g_{ij} = h_{\mu_i}^* h_{\mu_j} \in \widetilde{\mathcal{F}}$, where $\widetilde{\mathcal{F}}$ is a subset of the Hilbert space $\mathcal{H} = L^2(\Omega)$. A natural approach, which we refer to as a *direct* one, is to build a reduced basis for $\widetilde{\mathcal{F}}$ through a greedy algorithm as described in Algorithm 4 and is marked as *Path #1* in (20). In that case the results of Secs. 3.1 and 3.2 are directly applicable, with $\mathcal{F}$ and F_n replaced by $\widetilde{\mathcal{F}}$ and $\widetilde{F}_m$ respectively. In Sections 2 and 5.2 we argue that this direct approach can be impractical for some applications. Advanced algorithms for sampling or building the training space, perhaps adaptively [14, 9, 15, 16], might overcome this issue, though, and allow for a direct approach to such large problems.

In Sec. 2 we motivated a *two-step greedy approach* via *Path #2* in (20). In the first step (*Path #2 (i)* in (20)) we generate, through a first greedy as in Algorithm 4, a reduced basis whose span F_n approximates $\mathcal{F}$. Next one might construct a set F_{n^2}, consisting of all n^2 products $e_k^* e_l / \|e_k^* e_l\|_{\mathrm{d}}$. Another option, and indeed the one used in our numerical experiments, is to use a training set $\mathcal{T}_n^2 := \{(\mu_i, \mu_j)\}_{i,j=1}^n$ and associated normalized products $h_{\mu_i}^* h_{\mu_j} / \|h_{\mu_i}^* h_{\mu_j}\|_{\mathrm{d}}$ to define F_{n^2}, where $\{\mu_i\}_{i=1}^n$ are the greedy points identified by the first greedy algorithm when approximating $\mathcal{F}$ by F_n. In other words, we do not necessarily use an orthonormal reduced basis for F_n (see Remark 2, below). In either case, we approximate F_{n^2} through a second RB-greedy (*Path 2 (ii)* in (20)), carried out again as in Algorithm (4) but with F_{n^2} as the training space. The result is an orthonormal set of m reduced basis $\{\widetilde{e}_i\}_{i=1}^m$ such that $\widetilde{F}_m := \mathrm{span}\{\widetilde{e}_1, \ldots, \widetilde{e}_m\}$ accurately approximates F_{n^2}, and in turn the full set $\widetilde{\mathcal{F}}$.

To summarize, the second step of our proposed two-step greedy algorithm to build $\widetilde{F}_m$ is given by Algorithm 1, and for simplicity we choose the tolerance for the second greedy step to be equal to the first one (ϵ in Eq.(13)). To better present the algorithm it will be useful to introduce new notation for elements of $\mathcal{T}_n^2$. Define $\widetilde{\mu}_{k(i,j)} := (\mu_i, \mu_j)$ to be a re-indexing of the array $\{(\mu_i, \mu_j)\}_{i,j=1}^n$ into $\{\widetilde{\mu}_k\}_{k=1}^{n^2}$, where k is a one-to-one function which takes two integers i and j and returns a unique integer $k(i,j)$.

Remark 2 The first greedy returns both the greedy points $\{\mu_i\}_{i=1}^n$, an orthonormal basis $\{e_i\}_{i=1}^n$, and the *greedy basis functions* $\{h_{\mu_i}\}_{i=1}^n$, serving as a non-orthonormal basis set for the same space F_n. As discussed above, from either $\{h_{\mu_i}\}_{i=1}^n$ or $\{e_i\}_{i=1}^n$ one can build the set F_{n^2}. In our numerical experiments we choose $\{h_{\mu_i}\}_{i=1}^n$ as basis, partly because we need not store the orthonormal basis vectors $\{e_i\}_{i=1}^n$ but only the greedy points $\{\mu_i\}_{i=1}^n$. Clearly the implementation and interpretation of Algorithm 1 will depend on this choice. For example, when using $\{h_{\mu_i}\}_{i=1}^n$ it makes sense to discuss the selected greedy parameter points $\widetilde{\mu}_k$ while for $\{e_i\}_{i=1}^n$ no such interpretation holds; instead one should view Step 5d as returning a column index when F_{n^2} is thought of as an $M \times n^2$ matrix.

Algorithm 1 (Two-step RB-Greedy approximation for $\widetilde{F}_m$)

$\left[\{\widetilde{e}_\ell\}_{\ell=1}^m, \{\widetilde{\mu}_\ell\}_{\ell=1}^m \right]$ = Two-step RB-Greedy $(\epsilon, \{\mu_i\}_{i=1}^n, \{e_i\}_{i=1}^n)$
Comment: $\{e_i\}_{i=1}^n$ are either orthonormal or greedy basis functions (cf. remark 2)

(1) Let $\mathcal{T}_n^2 = \{(\mu_i, \mu_j)\}_{i,j=1}^n$
(2) Let $F_{n^2} = \{e_i^* e_j / \|e_i^* e_j\|_{\mathrm{d}}\}_{i,j=1}^n$
(3) Set $m = 0$ and define $\sigma_0(F_{n^2}; \mathcal{H}) := 1$
(4) Choose an arbitrary $g \in F_{n^2}$ and set $\widetilde{e}_1 := g$ **Comment:** $\|g\|_{\mathrm{d}} = 1$
(5) do, while $\sigma_m(F_{n^2}; \mathcal{H}) \geq \epsilon$
    (a) $m = m + 1$
    (b) $\sigma_m(g) := \left\| g - \widetilde{\mathcal{P}}_m g \right\|_{\mathrm{d}}$ for all $g \in F_{n^2}$

(c) $\sigma_m(F_{n^2}; \mathcal{H}) = \sup_{g \in F_{n^2}} \{\sigma_m(g)\}$

(d) $\widetilde{\mu}_{m+1} := \operatorname{argsup}_{g \in F_{n^2}} \{\sigma_m(g)\}$ (greedy sweep)

(e) $\widetilde{e}_{m+1} := g_{\widetilde{\mu}_{m+1}} - \widetilde{\mathcal{P}}_m g_{\widetilde{\mu}_{m+1}}$ (Gram-Schmidt)

(f) $\widetilde{e}_{m+1} := \widetilde{e}_{m+1} / \|\widetilde{e}_{m+1}\|_{\mathtt{d}}$ (normalization)

A DEIM approximation for the integrand leads to an efficient and simple reduced order quadrature rule (10) and is the subject of the next Section.

4 Reduced order quadratures (ROQ)

In classical theory of Gaussian quadratures one seeks to maximize the exactness for the integration of polynomials. Here the goal is to empirically maximize the accuracy of inner products between elements of $\mathcal{F}$. Notice that an approximation of (discrete) inner products may be carried out in either a reduced space $F_n \approx \mathcal{F}$ or $\widetilde{F}_m \approx \widetilde{\mathcal{F}}$. In Sec. 3.2 we noted that working in F_n leads to a pessimistic $\mathcal{O}(n^2)$ cost for inner products (see the discussion below Eq. (19)). Thus, for applications where one needs fast evaluations of many scalar products the approximation space $\widetilde{F}_m$ is preferable. For the construction of $\widetilde{F}_m$ we refer to (20).

Section 4.1 focuses on a ROQ stemming from $\widetilde{F}_m$, although only very minimal changes are needed to adapt Algorithm 2 for obvious variations. Indeed, one such variation is considered in our first numerical experiment. In Section 4.3 we consider extensions of the ROQ construction to situations where one might only be able to evaluate integrands at a set of points $\{y_i\}_{i=1}^{M'}$ which do not correspond to the nodes used in the quadrature rule $\{x_k, \omega_k\}_{k=1}^M$. In Section 4.2.1 we present the DEIM interpolation error estimates, followed by some ROQ error estimates in Section 4.2.2.

4.1 ROQ Algorithm

Suppose we are given a set of functions $\mathcal{F}$ and an arbitrary quadrature rule $\{x_k, \omega_k\}_{k=1}^M$ for the discrete inner product (6). The following algorithm generates the ROQ nodes and weights.

Algorithm 2 (Construction of Reduced Order Quadratures)
$$\left[\{\widetilde{p}_\ell, \omega_\ell^{\mathrm{ROQ}}\}_{\ell=1}^m\right] = \mathrm{ROQ}\left(\epsilon, \mathcal{T}_K, \{x_k, \omega_k\}_{k=1}^M\right)$$

(1) **Approximation of $\widetilde{\mathcal{F}}$:** We have two options to approximate $\widetilde{\mathcal{F}}$ by $\widetilde{F}_m$, (cf. (20)).

Path #1 Consider the training set $\mathcal{T}_K^2 = \{(\mu_i, \mu_j)\}_{i,j=1}^K$ and apply the Greedy Algorithm 4, with obvious modifications, to generate $\widetilde{F}_m = \operatorname{span}\{\widetilde{e}_\ell\}_{\ell=1}^m$ and greedy points $\{\widetilde{\mu}_\ell\}_{\ell=1}^m \subset \mathcal{T}_K^2$ such that any product $h_{\mu_i}^* h_{\mu_j} \in \widetilde{\mathcal{F}}$ can be approximated by its projection $\widetilde{\mathcal{P}}_m$ as:

$$\left\| h_{\mu_i}^* h_{\mu_j} - \widetilde{\mathcal{P}}_m \left(h_{\mu_i}^* h_{\mu_j} \right) \right\|_{\mathtt{d}} \le \epsilon, \qquad \forall \left(\mu_i, \mu_j \right) \in \mathcal{T}_K^2.$$

Note: This direct approach can be expensive; we therefore advocate the following two-step procedure.

Path #2   (i) **Approximation of $\mathcal{F}$:** Consider the training set $\mathcal{T}_K = \{\mu_i\}_{i=1}^K$ and apply the Greedy Algorithm 4 to generate $F_n = \operatorname{span}\{e_\ell\}_{\ell=1}^n$ and greedy points $\mathcal{T}_n = \{\mu_\ell\}_{\ell=1}^n \subset \mathcal{T}_K$ such that any $h_\mu \in \mathcal{F}$ can be approximated by its projection $\mathcal{P}_n$ as:

$$\left\| h_\mu - \mathcal{P}_n h_\mu \right\|_{\mathtt{d}} \le \epsilon, \qquad \forall \mu \in \mathcal{T}_K.$$

    (ii) **Construction of F_{n^2}:** Define the training set $\mathcal{T}_n^2 = \mathcal{T}_n \times \mathcal{T}_n = \{(\mu_i, \mu_j)\}_{i,j=1}^n$ and training space $F_{n^2} = \left\{ e_i^* e_j / \|e_i^* e_j\|_{\mathtt{d}} \right\}_{i,j=1}^n$. Another choice for F_{n^2} is $\left\{ h_{\mu_i}^* h_{\mu_j} / \|h_{\mu_i}^* h_{\mu_j}\|_{\mathtt{d}} \right\}_{i,j=1}^n$, we refer to Remark 2 for justification.

    (iii) **Approximation of $\widetilde{\mathcal{F}}$:** Apply the Greedy Algorithm 1 to generate $\widetilde{F}_m = \operatorname{span}\{\widetilde{e}_\ell\}_{\ell=1}^m$ and $\{\widetilde{\mu}_\ell\}_{\ell=1}^m \subset \mathcal{T}_n^2 \subset \mathcal{T}_K^2$ such that any product $e_i^* e_j \in F_{n^2}$ can be approximated by its projection $\widetilde{\mathcal{P}}_m$ as:

$$\left\| e_i^* e_j - \widetilde{\mathcal{P}}_m \left(e_i^* e_j \right) \right\|_{\mathtt{d}} \le \epsilon.$$

(2) Define the matrix $\widetilde{\mathbf{V}} = [\widetilde{\mathbf{e}}_1, \ldots, \widetilde{\mathbf{e}}_m] \in \mathbb{C}^{M \times m}$ where, for example, the ℓth column is $\widetilde{\mathbf{e}}_\ell = \left(\widetilde{e}_\ell(x_1), \ldots, \widetilde{e}_\ell(x_M)\right)^T$.

(3) **Generation of empirical interpolation points**: Apply the DEIM Algorithm 5 with $\widetilde{\mathbf{V}}$ as an input to compute the DEIM points $\{\widetilde{p}_\ell\}_{\ell=1}^m \subset \{x_i\}_{i=1}^M$ and the interpolation matrix $\widetilde{\mathbf{P}}$.

(4) Compute the ROQ weights by

$$\left(\boldsymbol{\omega}^{\text{ROQ}}\right)^T := \boldsymbol{\omega}^T \widetilde{\mathbf{V}}(\widetilde{\mathbf{P}}^T \widetilde{\mathbf{V}})^{-1}\,. \tag{21}$$

Step 4 deserves a bit more explanation. With the interpolation matrix $\widetilde{\mathbf{P}}$ from Step 3 we first build an expression for the DEIM interpolant of some product $g \in \widetilde{\mathcal{F}}$

$$\mathbf{g} \approx \widetilde{\mathcal{I}}_m[\mathbf{g}] := \widetilde{\mathbf{V}}(\widetilde{\mathbf{P}}^T \widetilde{\mathbf{V}})^{-1} \widetilde{\mathbf{P}}^T \mathbf{g}\,. \tag{22}$$

The reduced order weights are then found by substituting this expression into Eq. (6) to produce the desired approximation with $\mathbf{g} = \mathbf{h}_{\mu_i}^* \circ \mathbf{h}_{\mu_j}$,

$$I_{\mathsf{d}}(i,j) = \boldsymbol{\omega}^T \mathbf{g} \approx \boldsymbol{\omega}^T \widetilde{\mathcal{I}}_m[\mathbf{g}] = \left[\boldsymbol{\omega}^T \widetilde{\mathbf{V}}(\widetilde{\mathbf{P}}^T \widetilde{\mathbf{V}})^{-1}\right] \widetilde{\mathbf{P}}^T \mathbf{g} = \sum_{i=1}^m \omega_i^{\text{ROQ}} g(\widetilde{p}_i)\,. \tag{23}$$

Notice that the term $\widetilde{\mathbf{P}}^T \mathbf{g}$ is just evaluation of g at the DEIM points $\{\widetilde{p}_i\}_{i=1}^m$ (which need not be ordered, but can be if desired). Furthermore, the weights are explicitly parameter independent and for less accuracy we might consider using the first $m' \leq m$ functional evaluations of g

$$\boldsymbol{\omega}^T \widetilde{\mathcal{I}}_{m'}[\mathbf{g}] \approx \sum_{i=1}^{m'} \alpha_i^{\text{ROQ}} g(\widetilde{p}_i)\,,$$

which, as this is equivalent to carrying out the computation in an m'-dimensional subspace of $\widetilde{F}_m$, is also a valid integration rule. In general the weights α_i^{ROQ} associated with this m' point quadrature rule will not be a subset of the weights ω_i^{ROQ} computed for the m point rule. Notice that the hierarchical nature of the empirical interpolation set allows one to build application-specific nested quadratures of arbitrary order and depth (i.e. more than one embedded method). For example, suppose $m = 20$; one might combine the 10-point, 15-point, and 20-point ROQ rules to yield a nested quadrature scheme.

Remark 3 (FLOP-Count) Let K be the number of parameters in the training space to approximate $\mathcal{F}$ by F_n. If $\widehat{m}$ and m denote the number of reduced basis for direct (*Path #1*) and two-step greedy (*Path #2*) respectively, then numerically we observe that $m \approx \widehat{m}$. The cost of ROQ Algorithm 2 using a direct and a two-step greedy is $\mathcal{O}(K^2 M \widehat{m})$ and $\mathcal{O}(n^2 M m + K M n)$, respectively. This implies that for $K \gg n$ (as is observed in our numerical experiments), the two-step greedy is highly efficient. For the details of FLOP counts we refer to Appendix B.

Next we state some results which highlight the importance of using an accurate quadrature rule $\{x_k, \omega_k\}_{k=1}^M$; these will be verified in Section 5.1.

Theorem 1 *Integration of the basis functions $\{\widetilde{e}_\ell\}_{\ell=1}^m$ with either of the quadrature rules $\{x_i, \omega_i\}_{i=1}^M$ or $\{\widetilde{p}_\ell, \omega_\ell^{\text{ROQ}}\}_{\ell=1}^m$ yield identical results.*

Proof The integration of basis $\{\widetilde{e}_i\}_{i=1}^m$ using ROQ (cf. (21), (23) in vector notation) is

$$I_{\text{ROQ}} = \left(\boldsymbol{\omega}^{\text{ROQ}}\right)^T \widetilde{\mathbf{P}}^T \widetilde{\mathbf{e}}_i = \boldsymbol{\omega}^T \widetilde{\mathbf{V}}(\widetilde{\mathbf{P}}^T \widetilde{\mathbf{V}})^{-1} \widetilde{\mathbf{P}}^T \widetilde{\mathbf{e}}_i\,, \quad i = 1, \ldots, m\,.$$

Recalling that $\widetilde{\mathbf{V}} = [\widetilde{\mathbf{e}}_1, \ldots, \widetilde{\mathbf{e}}_m]$, integration of all the basis $\{\widetilde{e}_i\}_{i=1}^m$ can be written as $\left(\boldsymbol{\omega}^{\text{ROQ}}\right)^T \widetilde{\mathbf{P}}^T \widetilde{\mathbf{V}} = \boldsymbol{\omega}^T \widetilde{\mathbf{V}}$, where $\boldsymbol{\omega}^T \widetilde{\mathbf{V}}$ is precisely the integration of all the basis vectors $\{\widetilde{e}_i\}_{i=1}^m$ with the quadrature rule $\{x_k, \omega_k\}_{k=1}^M$.

Corollary 1 (Consistency of ROQ rule) *When $m = M$ the quadrature rules $\{x_i, \omega_i\}_{i=1}^M$ and $\{\widetilde{p}_\ell, \omega_\ell^{\text{ROQ}}\}_{\ell=1}^m$ are identical up-to ordering of the quadrature points.*

Proof Using Theorem 1, we have $\left(\boldsymbol{\omega}^{\text{ROQ}}\right)^T \widetilde{\mathbf{P}}^T \widetilde{\mathbf{V}} = \boldsymbol{\omega}^T \widetilde{\mathbf{V}}$. Apply $\widetilde{\mathbf{V}}^{-1}$ on the right to deduce $\left(\boldsymbol{\omega}^{\text{ROQ}}\right)^T \widetilde{\mathbf{P}}^T = \boldsymbol{\omega}^T$.

Remark 4 If the quadrature rule $\{x_i, \omega_i\}_{i=1}^{M}$ integrates the basis functions $\{\widetilde{e}_\ell\}_{\ell=1}^{m}$ exactly then ROQ built using Algorithm 2 also integrates those basis functions exactly.

Remark 5 As a special case consider $\widetilde{\mathcal{F}}$ to be a space of polynomials of degree $m-1$ with weight $W = 1$ and suppose the basis is specified by an ordered set of Legendre polynomials. Then the m-point ROQ rule will exactly integrate any polynomial of at most degree $m-1$ provided the quadrature rule $\{x_k, \omega_k\}_{k=1}^{M}$ is at least $m-1$ exact.

4.2 Convergence Estimates

4.2.1 DEIM Convergence

We shall first state a DEIM interpolation error estimate with respect to the discrete L^∞-norm. Then we will recall the DEIM error estimate with respect to the discrete L^2-norm from [4] and, finally, combine this estimate with those for the RB-greedy basis construction from [13].

Theorem 2 (discrete empirical interpolation method) *Let the set of reduced basis $\{e_\ell\}_{\ell=1}^{n}$ be orthonormal with respect to the discrete inner product defined in (3) and $h_\mu^{opt} \in F_n$ be the optimal approximation of h_μ with respect to the discrete L^∞-norm. Then for every $\mu \in \mathcal{P}$ and x_k, with $k = 1, \ldots, M$,*

$$\max_{1 \leq k \leq M} \left| h_\mu(x_k) - \mathcal{I}_n[h_\mu](x_k) \right| \leq (1 + \Lambda_{n,\infty}) \max_{1 \leq k \leq M} \left| h_\mu(x_k) - h_\mu^{opt}(x_k) \right|, \tag{24}$$

where $\Lambda_{n,\infty} = \|\mathcal{I}_n\|_\infty = \|\mathbf{V}(\mathbf{P}^T\mathbf{V})^{-1}\mathbf{P}^T\|_\infty$ denotes the Lebesgue constant.
In the case of the discrete L^2-norm, $h_\mu^{opt} = \mathcal{P}_n h_\mu$ and we have for all $\mu \in \mathcal{P}$

$$\left\| h_\mu - \mathcal{I}_n[h_\mu] \right\|_{\mathsf{d}} \leq \Lambda_{n,2} \left\| h_\mu - \mathcal{P}_n h_\mu \right\|_{\mathsf{d}}, \tag{25}$$

where $\Lambda_{n,2} = \|\mathcal{I}_n\|_2 = \|\mathbf{V}(\mathbf{P}^T\mathbf{V})^{-1}\mathbf{P}^T\|_2$ [4]

Proof The proof of (24) and (25) are straightforward and are omitted. Furthermore, (25) follows from [4, Lemma 3.2], but we state it for completeness. Since $\mathcal{I}_n[h_\mu^{opt}](x_k) = h_\mu^{opt}(x_k)$ for $k = 1, \ldots, M$, we get

$$\left\| h_\mu - \mathcal{I}_n[h_\mu] \right\|_{\mathsf{d}} = \left\| (\mathbb{I} - \mathcal{I}_n)[h_\mu - h_\mu^{opt}] \right\|_{\mathsf{d}}$$
$$\leq \|\mathcal{I}_n\|_2 \left\| h_\mu - h_\mu^{opt} \right\|_{\mathsf{d}},$$

where the last equality follows from the fact that $\|\mathbb{I} - \mathcal{I}_n\|_2 = \|\mathcal{I}_n\|_2$ (see, for example [22,23]).

Remark 6 Notice that the DEIM Algorithm 5 seeks to minimize the interpolation error as measured by a discrete L^∞-norm (cf. Step 6, Algorithm 5), for which an error bound of the type (24) is closely related. The RB-greedy Algorithm 4, however, exactly minimizes the term $\left\| h_\mu - \mathcal{P}_n h_\mu \right\|_{\mathsf{d}}$ on the right hand side of the error bound (25) and is therefore computable. Observing that $\Lambda_{n,2} \leq \|\mathbf{V}\|_2 \|(\mathbf{P}^T\mathbf{V})^{-1}\|_2$ (because $\|\mathbf{P}^T\|_2 = 1$), one might consider directly minimizing the error $\left\| h_\mu - \mathcal{I}_n[h_\mu] \right\|_{\mathsf{d}}$ by minimizing the norm $\|(\mathbf{P}^T\mathbf{V})^{-1}\|_2$. Indeed, whenever $\mathbf{V}^\dagger\mathbf{V} = \mathbf{1}_{n \times n}$ we have exactly $\Lambda_{n,2} = \|(\mathbf{P}^T\mathbf{V})^{-1}\|_2$ which is more efficient to compute than $\|\mathbf{V}(\mathbf{P}^T\mathbf{V})^{-1}\mathbf{P}^T\|_2$. Throughout the numerical experiments section 5, in addition to the error bound (25) we monitor

$$\|h_\mu - \mathcal{I}_n[h_\mu]\|_{\mathsf{d}} \leq \|\mathbf{V}\|_2 \|(\mathbf{P}^T\mathbf{V})^{-1}\|_2 \left\| h_\mu - \mathcal{P}_n h_\mu \right\|_{\mathsf{d}}, \tag{26}$$

which we have written here for future reference.

---

[4] The 2-norm and ∞-norm of a matrix $\mathbf{Q} \in \mathbb{C}^{M \times n}$ is defined by

$$\|\mathbf{Q}\|_2 := \max_{\mathbf{u} \neq 0} \frac{\|\mathbf{Q}\mathbf{u}\|_{\mathsf{d}}}{\|\mathbf{u}\|_{\mathsf{d}}} = \left(\lambda_{\max}(\mathbf{Q}^\dagger\mathbf{Q}) \right)^{1/2}, \qquad \|\mathbf{Q}\|_\infty := \max_{1 \leq k \leq M} \sum_{\ell=1}^{n} |q_{k\ell}|,$$

where $q_{k\ell}$ denotes the (k, ℓ) entry of $\mathbf{Q}$.

Corollary 2 (RB-Greedy-DEIM error bounds) *The following practical RB-greedy-DEIM estimate holds*

$$\|h_\mu - \mathcal{I}_n[h_\mu]\|_\mathsf{d} \lesssim \Lambda_{n,2}\epsilon, \quad \forall h_\mu \in \mathcal{F}.$$

Furthermore, if the Kolmogorov n-width decays exponentially (with order α) as in (15), then the error (25) decays with the same order,

$$\|h_\mu - \mathcal{I}_n[h_\mu]\|_\mathsf{d} \lesssim \sqrt{2C}\Lambda_{n,2}e^{-c_1 n^\alpha},$$

whereas if the Kolmogorov n-width decays algebraically with order β as in (16), then (25) again decays with the same order β

$$\|h_\mu - \mathcal{I}_n[h_\mu]\|_\mathsf{d} \lesssim 2^{5\beta+1}C_a\Lambda_{n,2}n^{-\beta},$$

where $C_a > 0$ and $\beta > 0$ are some constants.

Proof Consider (14) and (25) and apply (15) and (16). $\square$

4.2.2 ROQ Convergence

Next we present the error estimate for the case in which the reduced basis $\{\widetilde{e}_\ell\}_{\ell=1}^m$ for $\widetilde{\mathcal{F}}$ are generated through a direct approach via *Path #1* in (20).

Theorem 3 (approximation error ROQ (direct approach)) *Let $\epsilon > 0$ be an error tolerance and $\widetilde{\mathcal{P}}_m : \widetilde{\mathcal{F}} \to \widetilde{F}_m$ be the projection operator defined in Algorithm 2 (Path #1), with accuracy $\epsilon > 0$ in the discrete L^2-norm $\|\cdot\|_\mathsf{d}$. Furthermore, let $\widetilde{\mathcal{I}}_m : \widetilde{\mathcal{F}} \to \widetilde{F}_m$ be a DEIM operator built using Step (3) (RB-greedy-DEIM) of Algorithm 2. Given two arbitrary functions $h_{\mu_i}, h_{\mu_j} \in \mathcal{F}$, the following estimate holds*

$$\left|I_\mathsf{d}(i,j) - I_{\mathrm{ROQ}}(i,j)\right| \lesssim \epsilon \left(|\Omega|_\mathsf{d}\,\|\!|\widetilde{\mathcal{I}}_m\|\!|_2\right)\left\|h_{\mu_i}^* h_{\mu_j}\right\|_\mathsf{d}.$$

Proof We recall that given $h_{\mu_i}, h_{\mu_j} \in \mathcal{F}$, with $\mu_i, \mu_j \in \mathcal{P}$, we have a discrete quadrature rule $I_\mathsf{d}(i,j)$ given in Eq. (6). On the other hand, we have the following expression for the ROQ integral

$$I_{\mathrm{ROQ}}(i,j) = \sum_{k=1}^M \omega_k \widetilde{\mathcal{I}}_m[h_{\mu_i}^* h_{\mu_j}](x_k).$$

An application of Cauchy-Schwarz implies

$$\left|I_\mathsf{d}(i,j) - I_{\mathrm{ROQ}}(i,j)\right| \leq \sum_{k=1}^M \left|\omega_k\left(h_{\mu_i}^*(x_k)h_{\mu_j}(x_k) - \widetilde{\mathcal{I}}_m[h_{\mu_i}^* h_{\mu_j}](x_k)\right)\right|$$

$$\leq |\Omega|_d\left\|h_{\mu_i}^* h_{\mu_j} - \widetilde{\mathcal{I}}_m[h_{\mu_i}^* h_{\mu_j}]\right\|_\mathsf{d}.$$

Next the fact that $\widetilde{\mathcal{I}}_m\left[\widetilde{\mathcal{P}}_m[h_{\mu_i}^* h_{\mu_j}]\right] = \widetilde{\mathcal{P}}_m[h_{\mu_i}^* h_{\mu_j}]$, implies

$$h_{\mu_i}^* h_{\mu_j} - \widetilde{\mathcal{I}}_m[h_{\mu_i}^* h_{\mu_j}] = \left(\mathbb{I} - \widetilde{\mathcal{I}}_m\right)\left[h_{\mu_i}^* h_{\mu_j} - \widetilde{\mathcal{P}}_m[h_{\mu_i}^* h_{\mu_j}]\right]$$

Finally using $\|\!|\mathbb{I} - \widetilde{\mathcal{I}}_m\|\!|_2 = \|\!|\widetilde{\mathcal{I}}_m\|\!|_2$ (cf. [22, 23]), we get

$$\left|I_\mathsf{d}(i,j) - I_{\mathrm{ROQ}}(i,j)\right| \leq |\Omega|_d\,\|\!|\widetilde{\mathcal{I}}_m\|\!|_2\,\sigma_m\left(\widetilde{\mathcal{F}}; \mathcal{H}\right)\left\|h_{\mu_i}^* h_{\mu_j}\right\|_\mathsf{d}.$$

Then the greedy estimate *Path #1* in Algorithm 2 gives the required estimate. $\square$

Corollary 3 (ROQ error bound) *The following practical ROQ error estimate holds for every $\mu_i, \mu_j \in \mathcal{P}$*

$$\left|I_\mathsf{c}(i,j) - I_{\mathrm{ROQ}}(i,j)\right| \lesssim \left|I_\mathsf{c}(i,j) - I_\mathsf{d}(i,j)\right| + \epsilon\left(|\Omega|_d\,\|\!|\widetilde{\mathcal{I}}_m\|\!|_2\right)\left\|h_{\mu_i}^* h_{\mu_j}\right\|_\mathsf{d}. \tag{27}$$

Our numerical examples indicate the ROQ built from a two-step greedy (using *Path #2*) has an error bound as in Theorem 3, but proving such bound is beyond the scope of this paper.

4.3 ROQ at new quadrature points

Consider the scenario in which the integrands are available at a set of points which is different from the ones used to build the reduced basis. A typical example would be the case in which Gaussian quadratures are used in the RB-greedy algorithm but data is given at equally spaced samples. We continue denoting the set of points and quadrature rule used to build the basis as $\{x_k, \omega_k\}_{k=1}^{M}$, and the second rule $\{y_i, \tau_i\}_{i=1}^{M'}$. Therefore, it is not unreasonable to assume that $M' > M$. Our goal here is to generate an ROQ with respect to $\{y_i, \tau_i\}_{i=1}^{M'}$ under this scenario.

Algorithm 3 (Construction of ROQ for $\{y_i, \tau_i\}_{i=1}^{M'}$)
$$\left[\{\widetilde{p}_\ell, \tau_\ell^{\mathrm{ROQ}}\}_{\ell=1}^{m}\right] = \mathrm{ROQ\text{-}NEW}\left(\{\widetilde{\mu}_\ell\}_{\ell=1}^{m}, \{y_i, \tau_i\}_{i=1}^{M'}\right)$$

(1) Orthogonalize the (ordered) product of functions $\{g_{\widetilde{\mu}_\ell}\}_{\ell=1}^{m}$ using the quadrature rule $\{y_i, \tau_i\}_{i=1}^{M'}$ and let the resulting vectors form the columns of a matrix $\widetilde{\mathbf{V}} = [\widetilde{\mathbf{e}}_1, \ldots, \widetilde{\mathbf{e}}_m] \in \mathbb{C}^{M' \times m}$ where, for example, the ℓth column is $\widetilde{\mathbf{e}}_\ell = \left(\widetilde{e}_\ell(y_1), \ldots, \widetilde{e}_\ell(y_{M'})\right)^T$.

(2) Apply Steps 3 and 4 from Algorithm 2 to generate the ROQ points $\{\widetilde{p}_\ell\}_{\ell=1}^{m} \subset \{y_i\}_{i=1}^{M'}$, interpolation matrix $\widetilde{\mathbf{P}}$ and ROQ weights $(\boldsymbol{\tau}^{\mathrm{ROQ}})^T := \boldsymbol{\tau}^T \widetilde{\mathbf{V}}(\widetilde{\mathbf{P}}^T \widetilde{\mathbf{V}})^{-1}$.

Accuracy and conditioning of the resulting ROQ rule are guaranteed provided (i) we are able to carry out the orthogonalization in Step 1 up to a certain tolerance and (ii) the new quadrature rule is accurate enough such that $\left\| h_\mu - \mathcal{P}_n h_\mu \right\|_\tau \lesssim \epsilon$.

5 Numerical examples

Here we discuss two sets of numerical experiments. As a first example we compare ROQ using orthogonal polynomials on the interval $[-1, 1]$ as basis (that is, the basis is not built through a greedy approach) with the well known case of Gaussian quadratures and, in particular, integration of Runge's function. This study reveals several important features: i) ROQ are well conditioned, ii) ROQ and Gaussian quadrature rules have a similar point and weight distribution, iii) the importance of an accurate quadrature rule $\{x_i, \omega_i\}_{i=1}^{M}$ (cf. Theorem 1), iv) the ROQ rule's efficiency, compared to Gaussian quadratures, is expected to increase with the number of spatial dimensions.

Next we turn to a challenging example drawn from gravitational wave physics, namely the calculation of overlaps between "chirp" gravitational waves for compact binary coalescences. Details on implementation, error bounds, and a discussion of computational cost are provided, as well as the physical motivation of the problem. In particular, we find that ROQs exhibit exponential asymptotic convergence in the calculation of overlaps, with a profile of the error function in terms of quadrature points qualitatively similar to the greedy error curve of the training space in terms of the number of reduced basis elements. *We emphasize that this exponential convergence in computing overlaps is, as expected, present even when the function is sampled at equally spaced quadrature points. In particular, this means that reduced order quadratures can be used as an efficient down-sampling strategy when function evaluations from experimental observations are given at (for example) fixed intervals. We also show the staggering offline savings of our two-step greedy approach compared to a direct one.*

5.1 Comparison with Gauss-Legendre quadratures

In order to highlight the essential features of our approach, in this subsection we use Legendre polynomials $\{P_\ell(x)\}_{\ell=1}^{m}$ defined on $x \in [-1, 1]$ as an orthonormal basis (with respect to $W(x) = 1$) instead of a reduced basis as generated in Step 1 Algorithm 2. Steps 3 and 4 of Algorithm 2 require specifying an underlying quadrature rule, which in this subsection will be either an M point Gauss-Legendre or extended trapezoidal rule. The resulting ROQ point and weight distribution (see Subsection 5.1.1), conditioning (see Subsection 5.1.2) and application to Runge's function (see Subsection 5.1.3) are considered next.

5.1.1 Point and weight distribution of ROQ

To build the ROQ points and weights an extended trapezoidal rule is used and all $m = 24$ Legendre polynomials are sampled at $M = 1,000$ equidistant points. Fig. 1(a) compares the distribution of points and weights for the Gauss-Legendre quadrature (red dots) and ROQ (blue crosses). Both weight and point distributions are similar, and in particular

the points are not equally spaced but instead cluster towards the end points at $x = \pm 1$. One of the ROQ weights is equal to -0.00496089441576999 at 0.775775775775776. Negative weights can be problematic in quadrature rules due to roundoff errors leading to poorly conditioned formula [24]. We turn to this issue next.

(a) ROQ and Gauss-Legendre point and weight distributions.   (b) ROQ (solid) and Gauss-Legendre (dashed) condition number.
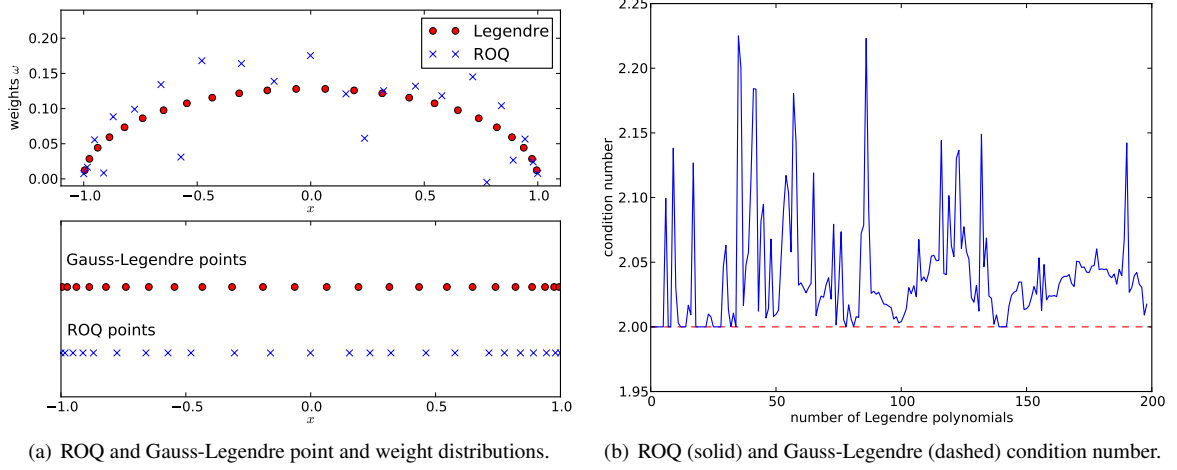
Fig. 1 The top left scatter plot shows the weight ω_k and point distributions for Gauss-Legendre and ROQ using $m = 24$ Legendre polynomials as a basis (see Subsection 5.1.1 for more details), whereas the bottom left one shows just the point locations. The right plot gives the condition number $\sum_{k=1}^{m} |\omega_k|$ for an m-point Gauss-Legendre and ROQ rule, with $m \in [2, 200]$ (see Subsection 5.1.2 for more details).

5.1.2 Conditioning of ROQ

Consider the absolute condition number $\sum_k |\omega_k|$ of a quadrature formula which, on the integration domain $x \in [-1, 1]$, is exactly 2 if all the weights ω_k are positive as can be seen by integration of unity. If some weights are negative then the condition number is necessarily larger than 2. A classic example of ill-conditioning is the Newton-Cotes rule, where $\sum_k |\omega_k| > 2$ and grows without bound when using more than 8 nodes [24]. Fig. 1(b) shows the absolute condition number $\sum_k |\omega_k|$ for a Gauss-Legendre (dashed red line) and reduced order (solid blue line) quadrature. The dashed red line shows the optimal value of 2, which is achieved by the Gauss-Legendre quadrature case. The condition number for ROQ remains below 2.25 for the first 200 Legendre polynomials (see Subsection 5.1.1) and has no noticeable growth trend. This plot was generated using $M = 1,000$ equally spaced points on $x \in [-1, 1]$: this resolution is sufficient for demonstrating well conditioning, but the resulting ROQ integration rule would be of low accuracy. We have considered up to $100,000$ points, always obtaining behavior similar to that one of Fig. 1(b).

5.1.3 Integration of Runge's function

Many general features of ROQ can be ascertained by integration of Runge's function

$$\int_{-1}^{1} \frac{1}{(1 + x^2)} \, dx = 2 \tan^{-1}(1),$$

which is a classic example of oscillatory errors due to high-order interpolation. In the following experiment, integration of Runge's function is carried out with three different ROQ rules built from a basis consisting of m Legendre polynomials. In each case the Legendre polynomials are evaluated at $M_1 = 10,000$ equally spaced and $M_2 = 400$ Gauss-Legendre points, and the integration errors are computed as $|2 \tan^{-1}(1) - I_{\text{ROQ}}|$.

First, integration of Runge's function is carried out using an m-point ROQ-trapezoidal rule which is built from an M_1-point extended trapezoidal rule and the m DEIM point selections. The resulting integration error is given by the dash-dot black line in Fig. 2. Notice that the ROQ-trapezoidal error curve flattens out around 10^{-9}. Beyond this value the $10,000$ point trapezoidal rule is unable to accurately integrate the Legendre polynomials of degree higher than ~ 20, thereby limiting the ROQ accuracy (see Theorem 1).
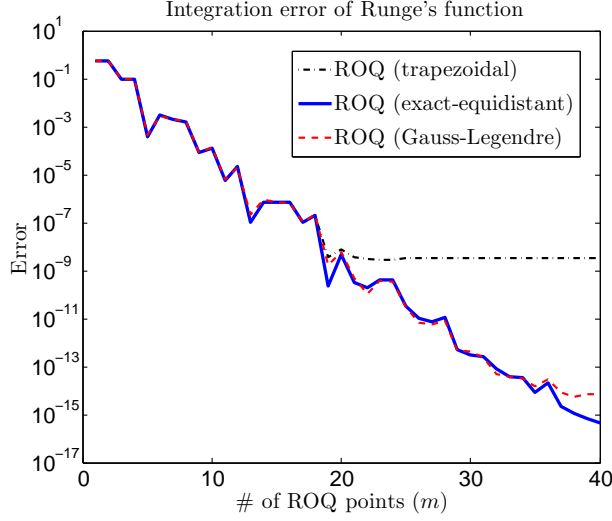
Fig. 2 Convergence of the integral $\int_{-1}^{1} \left(1 + x^2\right)^{-1} dx$ to $2\tan^{-1}(1)$ using three different ROQ rules whose generation (using Legendre polynomials as basis) is described in Sec. 5.1.3. Integration error from a ROQ-trapezoidal constructed with a $M_1 = 10,000$ point extended trapezoidal rule is given by the dash-dot black line. Due to the extended trapezoidal rule's inability to accurately integrate higher order Legendre polynomials this curve flattens out around 10^{-9}. If we *exactly* integrate these basis functions by setting $\boldsymbol{\omega}^T \widetilde{\mathbf{V}} = [\sqrt{2}, 0, \ldots, 0]$ in Eq. (21) while continuing to use the ROQ-trapezoidal's quadrature points the accuracy is restored as seen from the ROQ-equidistant error curve (solid blue). Integration error from an ROQ constructed with a $M_2 = 400$ point Gauss-Legendre is given by the dashed red line.

Next we show how accuracy may be restored in this particular case. Notice that the factor $\boldsymbol{\omega}^T \widetilde{\mathbf{V}}$ in Eq. (21) for the ROQ weights is precisely the numerical integration of m Legendre polynomials. Furthermore, we know that the first Legendre polynomial integrates to $\sqrt{2}$ while all others integrate to zero. Hence, in this particular case we may integrate them "by hand" by setting $\boldsymbol{\omega}^T \widetilde{\mathbf{V}} = [\sqrt{2}, 0, \ldots, 0]$, while continuing to generate interpolation points by way of the DEIM algorithm. The resulting ROQ rule, denoted in the figure as ROQ-equidistant, is then used to integrate Runge's function with an error given by the solid blue line in Fig. 2. This highlights the importance of accurate integration of the basis functions which, in a general setting where the basis vectors are selected by a greedy algorithm, should be carried out with a good underlying quadrature rule.

Lastly, an ROQ rule is constructed from a $M_2 = 400$ point Gauss-Legendre rule (dashed red line of Fig. 2) and shows excellent agreement with the ROQ-equidistant case. Of particular noteworthiness, all three ROQ rules clearly show exponential convergence with the number of quadrature points and no apparent issues stemming from Runge's phenomena. Notice that the dashed red and solid blue lines are nearly indistinguishable, confirming the expectation that if the basis functions are sampled densely enough (a statement which certainly depends on m) and integrated accurately enough by the underlying quadrature rule, the resulting rules perform similarly.

5.1.4 Higher dimensional integrals

Let $d \geq 1$ be the space dimension and consider the integrals

$$I_1 = \int_{\Omega} \left[(x - \mu_1)^2 + 0.1^2 \right]^{-1/2} dx, \quad I_2 = \int_{\Omega} \left[(x - \mu_1)^2 + (y - \mu_2)^2 + 0.1^2 \right]^{-1/2} dx\, dy, \qquad (28)$$

for the following two cases: i) $d = 1$, $\Omega = [-1, 1]$ and $\mu_1 \in [-.1, .1]$ and ii) $d = 2$, $\Omega = [-1, 1]^2$ and $\mu = (\mu_1, \mu_2) \in [-.1, .1]^2$. In the first case , numerical integration is carried out using an $M_1 \leq 150$ point Gauss-Legendre rule and an ROQ rule built from the 150-point GQ rule. For the second case, numerical integration is carried out using an $M_2 \leq 150^2$ point tensor product Gauss-Legendre rule and an ROQ rule built from the 150^2-point GQ rule. Basis vectors are selected by the greedy algorithm in both cases. From Fig. 3 one can see that ROQ provide a factor of ~ 4 savings in the 1-dimensional case for a maximum error below 10^{-4}, while the savings in the 2-dimensional case is greater than ~ 12. For many problems the expected ROQ savings will continue to increase when compared to quadrature rules arising from tensor product grids whose computational cost (also the cost incurred while building the basis) scales like M^d for d spatial dimensions. On the other hand, an ROQ nodal set is formed by scattered point distributions tailored to the problem.
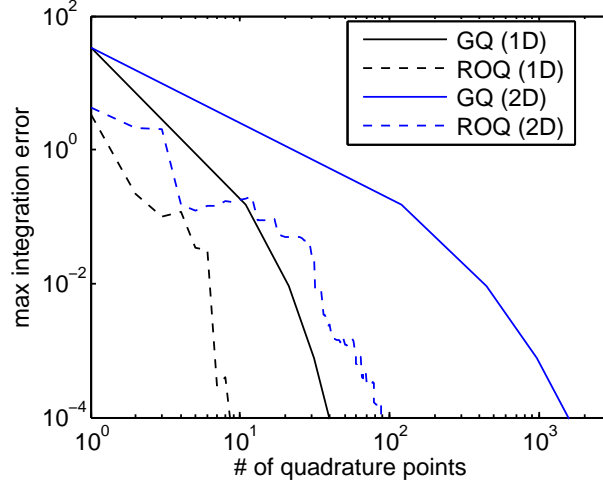
Fig. 3 Error curves for the 1-dimensional (case i) and 2-dimensional (case ii) numerical integrals of Eq. (28) using Gauss-Legendre and ROQ rules. Errors are computed by taking a maximum over the entire training set. The ROQ savings increase from ~ 4 to ~ 12 as the number of spatial dimensions is increased from one to two. As discussed in Sec. 5.1.4, further savings are expected as the number of spatial dimensions increase.

5.2 Gravitational waves

The inspiral and merger of neutron stars (NSs), black holes (BHs) or mixed pairs, known generically as compact binary coalescences, is believed to be one of the main sources of gravitational waves to be detected by the upcoming generation of earth-based observatories [25, 26, 27, 28, 29, 30]. A passing wave distorts the length between any two points separated by a non-zero spatial distance. These distortions are measured by gravitational wave detectors as an effective strain $h_\mu(x)$, where x denotes either time or frequency. In this problem the parameter μ depends on quantities such as the mass and spin of the compact objects, or detector orientation. In principle h_μ is obtained by numerically solving Einstein's equations, which are a set of parameterized quasilinear hyperbolic-elliptic equations [31, 32, 33, 34]. However, due to the cost of these simulations and the properties of the target system, simplified models are often used.

Here we focus on testing our proposed ROQ on the post-Newtonian (PN) approximation [35], under which the gravitational waves in the leading order Stationary Phase Approximation (SPA) are given by [36, 37, 38],

$$h_{\mathcal{M}_c}(f) = \mathcal{A} f^{-7/6} \cdot \exp\left(i \left\{ -\frac{\pi}{4} + \frac{3}{128} \left(\pi \cdot \frac{G}{c^3} \cdot f \cdot \mathcal{M}_c \right)^{-5/3} \right\} \right), \tag{29}$$

where f denotes the frequency, G Newton's gravitational constant [39], c the speed of light, and $\mathcal{A}$ an overall amplitude that depends on quantities such as the distance to the compact binary coalescence. We will refer to Eq. (29) as a gravitational *waveform*, two representative examples are shown in Fig. 4. The single parameter μ in this model is the *chirp mass* $\mathcal{M}_c := (m_1 m_2)^{3/5} (m_1 + m_2)^{-1/5}$ [38], with m_i the individual mass of each object. All quantities here use MKS units.

There are various reasons one might need to repeatedly compute inner products between waveforms. One setting where this occurs is in gravitational wave searches using matched filtering [40, 41, 42, 38]. Given data s recorded by a detector, a matched filtering search is carried out by computing all possible inner products of the data s against waveforms drawn from a (usually large) catalog. It can be shown [43, 44] that the optimal matched filtering strategy requires one to compute inner products with a weight given by the reciprocal of the detector's power spectral density $S(f) = W^{-1}$. As a representative example we use $S(f)$ from the initial Laser Interferometer Gravitational Wave Observatory (LIGO) [45], which can be modeled by the curve [46]

$$S(y) = 9 \times 10^{-46} \left[(4.49y)^{-56} + 0.16y^{-4.52} + 0.52 + 0.32 \cdot y^2 \right], \quad y = \frac{f}{150 Hz}. \tag{30}$$

To ensure that no signals are missed and that the parameters are correctly estimated many inner products are needed for each segment of data [18, 47, 48, 49]. These significant computational costs motivate a need for faster ways of performing these integrals, often in real-time [50, 51, 52].
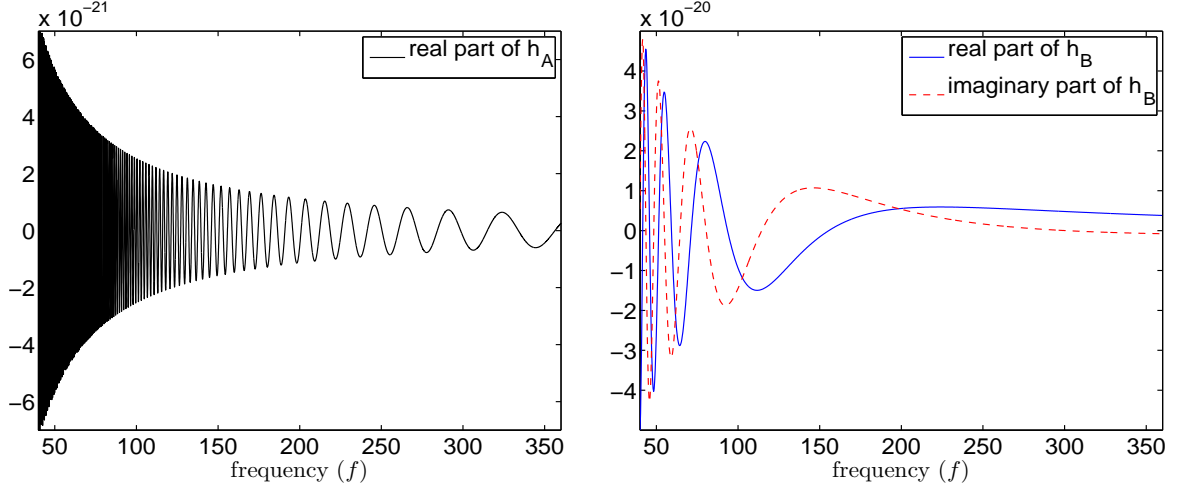
Fig. 4 Gravitational waveforms h_A (left) and h_B (right) correspond to the smallest ($A = 2.611651689888372M_\odot$) and largest ($B = 26.11651689888372M_\odot$) chirp masses of the training set $\mathcal{T}_K$ defined in (31). Waveforms are complex valued, but for clarity we do not show the imaginary part of h_A.

In the following series of experiments we consider gravitational waves of the form given by Eq. (29), generated by inspiralling binary black hole (BBH) systems with mass components in the interval $m_i \in [3, 30]M_\odot$, where $1M_\odot = 1.98892 \times 10^{30}Kg$ is the astrophysical unit of mass known as a *solar mass*. Therefore the parameter domain under consideration is $\mathcal{P} = [A, B]$, where $A = 2.611651689888372M_\odot$ and $B = 26.11651689888372M_\odot$. The physical domain frequency interval is given by $\Omega = [40, 366.3383434841933]$ Hz. Here the lower value is set by the Earth's seismic noise which decreases the detector's sensitivity. The upper bound is set by the limitations of the PN approximation which breaks down at frequencies corresponding to the *Innermost Stable Circular Orbit*. While this is a mass dependent frequency, for simplicity we choose the upper limit as the maximum over the mass range here considered.

Our primary goal in the forthcoming sections is to describe the construction and performance of an ROQ rule for computing inner products between gravitational waveforms given by Eq. (29). These steps are conveniently summarized in Algorithm 2. The first step, an approximation for $\widetilde{\mathcal{F}} \approx \widetilde{F}_m$, can be carried out either by means of a direct greedy (Path #1) or two-step greedy (Path #2). Due to a *significantly* reduced offline cost we advocate the two-step greedy approach, which in turn requires an intermediate approximation $\mathcal{F} \approx F_n$. Results for this intermediate approximation are considered in Section 5.2.1. We contemporaneously remark on the performance of the DEIM algorithm when applied to single functions to facilitate a better understanding of this algorithm in a simpler setting. In Sections 5.2.2 and 5.2.3 we complete the second half of the two-step greedy approximation, identify the interpolation points and finally generate an ROQ rule. In this section we also answer the fundamental question: *Do the direct and two-step greedy approaches give comparable results and is the former much more expensive than the latter?* As we will see, we have strong numerical evidence that, as predicted by our estimates, the two-step greedy gives enormous computational savings without sacrificing the accuracy or compactness of the reduced basis.

5.2.1 RB-greedy and DEIM for single functions

Here we consider experiments focused on approximations of the set of functions $\mathcal{F} = \{h_{\mathcal{M}_c}(f) : \mathcal{M}_c \in \mathcal{P}\}$; with $h_{\mathcal{M}_c}$ as in Eq. (29) and the range $\mathcal{P} = [A, B]$ for the chirp mass $\mathcal{M}_c$ –serving here as the parameter μ– as described above. We find the DEIM points and numerically confirm the DEIM error bounds from Sec. 4.2.2. This subsection is also precursor to Sections 5.2.2 and 5.2.3, where ROQs are constructed for waveform inner products.

Reduced Basis: We begin by building a reduced basis space F_n approximating $\mathcal{F}$ by using the RB-greedy approach. Since the number of cycles is proportional to an inverse power of $\mathcal{M}_c$, we have found it advantageous to populate the training space with a logarithmic spacing between the samples, thereby clustering more points at low values of $\mathcal{M}_c$. More precisely, a training set of size K is here given by

$$\mathcal{T}_K = \left\{ A \left(\frac{B}{A} \right)^{\frac{i}{K-1}} \mid i = 0, \ldots, K - 1 \right\}, \tag{31}$$

and an associated training space of normalized waveforms

$$\mathcal{F}_{\text{train}} = \{h_{\mathcal{M}_c}(f) \mid \mathcal{M}_c \in \mathcal{T}_K\}.$$

Through numerical experiments we have found that for this problem the number of basis for any given greedy error saturates with at most $K = 3,000$ training space elements, with – for example – 178 RB elements $\{e_i\}_{i=1}^{178}$ needed to achieve a tolerance[5] of $\epsilon^2 = 10^{-12}$. Therefore, the results shown below use $3,000$ training space samples. Fig. 5(a) shows the distribution of points selected by the greedy algorithm. They cluster at low $\mathcal{M}_c$, which corresponds to more lower-frequency oscillations in Eq. (29). Such clustering is expected, since the number of cycles in a frequency range $f \in [f_{min}, f_{max}]$ is given to lowest post-Newtonian approximation order by (graphically seen from Fig 4)

$$\mathcal{N}_{\text{cycles}}(\mathcal{M}_c) = \mathcal{N}(f_{min}; \mathcal{M}_c) - \mathcal{N}(f_{max}; \mathcal{M}_c),$$

where (see Eq. (4.23), and Eq. (5.247) from Ref. [44])

$$\mathcal{N}(f; \mathcal{M}_c) = 1/(32\pi^{8/3}) \left(G\mathcal{M}_c/c^3\right)^{-5/3} f^{-5/3}.$$

The truly interesting aspect is that the distribution of greedy points closely matches the functional form of $\mathcal{N}_{\text{cycles}}(\mathcal{M}_c)$.

The solid black lines labeled "projection error" in both plots of Figure 6 show the square of the greedy error $\sigma_n(\mathcal{F}_{\text{train}}; \mathcal{H})$, defined in Eq. (12), over the training space $\mathcal{F}_{\text{train}}$, as a function of the number of RB elements. After a slow decay, the error decays with a very fast exponential falloff, a feature that we have found in this family of inspiral gravitational waveforms and generalizations thereof [18, 19].

(a) Chirp Mass $\mathcal{M}_c$ (b) Frequency f
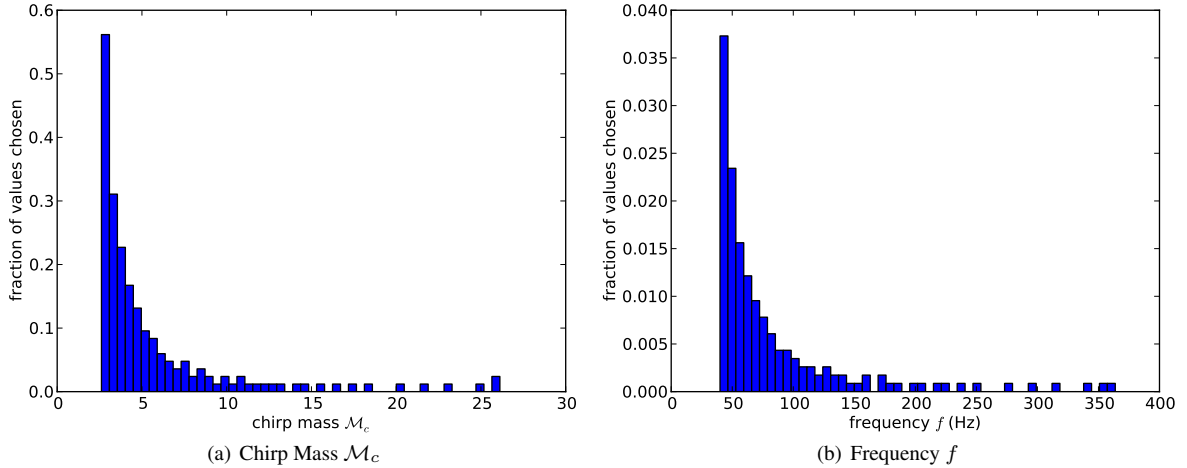
Fig. 5 The left figure (a) shows the parameters $\mathcal{M}_c$ selected by greedy Algorithm 4. The right figure (b) shows the frequency points selected by DEIM Algorithm 5 out of $20,000$ equidistant frequency sample points. We see from the histograms that both the chirp masses selected and the frequency interpolating points cluster at small values. This is intuitively expected, because smaller chirp masses correspond to a larger number of waveform cycles in Eq. (29).

Empirical Interpolation: Since the reduced basis set $\{e_i\}_{i=1}^{178}$ generated above is application-based (as opposed to, say, Jacobi polynomials), an appropriate set of interpolation points is *in principle* not known, and that is where the DEIM algorithm of Section 3.2 enters the game. We now generate a hierarchical set of DEIM points for the reduced basis built (these results are not used in the construction of ROQ for inner products), choosing the maximum number of interpolating points which equals the dimension of the RB space. That is, using $n = 1, 2, 3, \ldots, 178$ reduced basis elements, we sequentially computed the equivalent number of DEIM points. We recall that the RB-greedy approach selects points in parameter space, in this case the chirp mass, while the DEIM generates interpolating points in physical

---

[5] Our numerical experiments are carried out with double precision arithmetic. This translates into a double precision computation of the quantity $\left\|h_\mu - \mathcal{P}_n h_\mu\right\|_{\text{d}}^2$ found in step 3b of Algorithm 4 (RB-Greedy Algorithm), and hence an accuracy of about 10^{-7} in the computation of its square root. These observations motivate a choice for the greedy error tolerance to be $\epsilon = 10^{-6}$. Refs. [53,54] address this issue in greater detail and propose alternatives for improving error computations.

space (here frequency), and that both methods are hierarchical. That is, a seed choice for the first parameter value defines the first RB element, and from it the first interpolating point can be computed. Next, the greedy method chooses a second parameter value and the second RB element. By applying the DEIM to this new set (of so far two basis elements), the second interpolating point can be computed. And so on. Equivalently (the output is exactly the same), one can generate the whole RB first, up to the desired representation error tolerance, and then generate all of the interpolation points. Put differently, if there are n RB elements, up to n DEIM points can be generated, and generating some $n_1 < n$ DEIM only requires the first n_1 RB elements. This discussion, though perhaps trivial, might be helpful to keep in mind when later discussing the results from Fig. 6.

As input to the DEIM algorithm we must provide the RB functions sampled on some set of frequency points. Two cases are here considered: i) the basis vectors are sampled at $1,701$ Gauss-Legendre points, which was the integration rule used to build the reduced basis in Subsection 5.2.1, ii) the basis vectors are sampled at $20,000$ equidistant points [6]. Figure 5(b) depicts the frequency distribution of selected points for the equidistant sampling case, which is of particular practical importance when one seeks to downsample experimental data. The overall structure for the Gauss-Legendre points case was found to be essentially identical, and is therefore not shown.

Next, we randomly pick $10,000$ waveforms, not necessarily in the training space, and represent each of them as a DEIM interpolant (that is, using Eq. (22)). Both sampling at Gauss-Legendre and equidistant points are considered. For each waveform the DEIM error is calculated as $\|h_\mu - \mathcal{I}_n[h_\mu]\|_\mathsf{d}^2$. Fig. 6 compares the largest DEIM error over all $10,000$ waveforms (solid blue line) with the greedy error (solid black line). Notice that the black line lies strictly below the blue line; for waveforms in the training space this is guaranteed by the optimality of the L^2 orthogonal projection $\mathcal{P}_n h_\mu$. In turn, the DEIM interpolation error is bounded, again as expected, from above by the a-priori theoretical error estimates given in by Eq. (25) (red line) and Eq. (26) (purple line). In principle these two error estimates strictly hold only for waveforms in the training space, but our numerical results show that they evidently continue to hold for waveforms between elements of the training space when the latter is sufficiently dense. The red line provides a sharper bound on the error but it is also more expensive to calculate (cf. Remark 6). Finally, notice that, at least for elements in the training space, the asymptotic convergence rate of the DEIM is predicted to be bounded by the RB-greedy representation error (with the Lebesgue constant as a proportionality constant), see Corollary 2. From the figure we can see that the similar asymptotic convergence rates for the DEIM and reduced basis representation continue to hold for waveforms not necessarily in the training space.
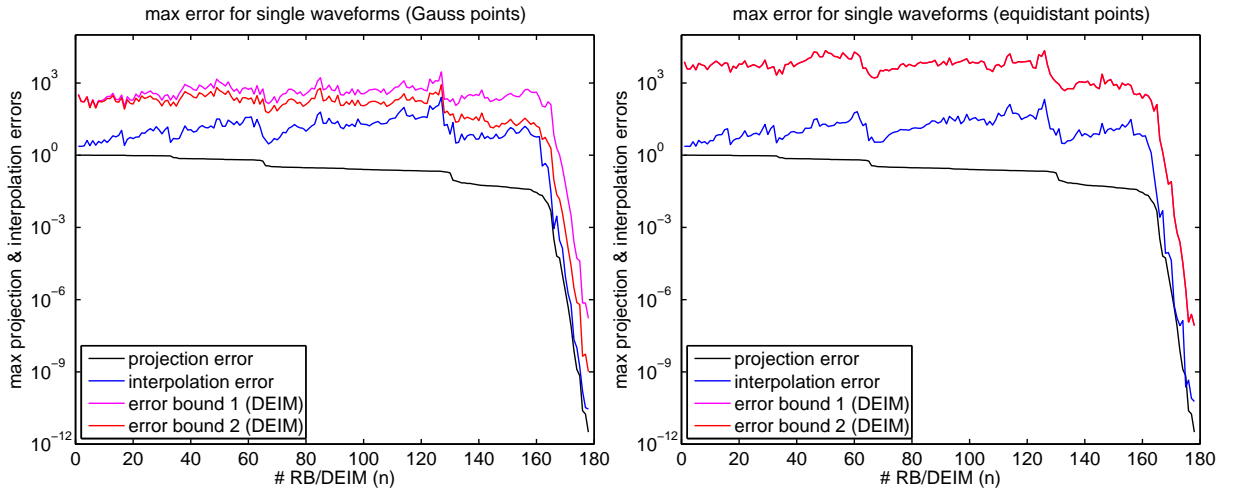
Fig. 6 Maximum projection and interpolation errors, as well as the predicted error bounds, for single waveforms as a function of the number of reduced basis (equal to the number of DEIM points used). In both figures "projection error" plots the square of the greedy error $\sigma_n(F_{\text{train}}; \mathcal{H})$ and "interpolation error" plots a maximum error $\|h_{\mathcal{M}_c} - \mathcal{I}_n[h_{\mathcal{M}_c}]\|_\mathsf{d}^2$ taken over $10,000$ randomly drawn values of $\mathcal{M}_c$ not necessarily in the training space. The DEIM interpolant error curves found in the left (right) figures correspond to the Gauss-Legendre (equidistant) cases described in Sec. 5.2.1. Error bounds given by Eq. (26) (error bound 1, magenta) and Eq. (25) (error bound 2, red) are identical up to numerical accuracy when equidistant samples are used (cf. remark 6). For both cases the maximum interpolation error is nearly identical, as expected.

---

[6] See the discussion in Section 4.3 related to the subtleties involved when sampling the RB functions at a set different from those used to compute the underlying quadratures to build the RB itself.

5.2.2 **Direct and two-step greedy: results and comparison**
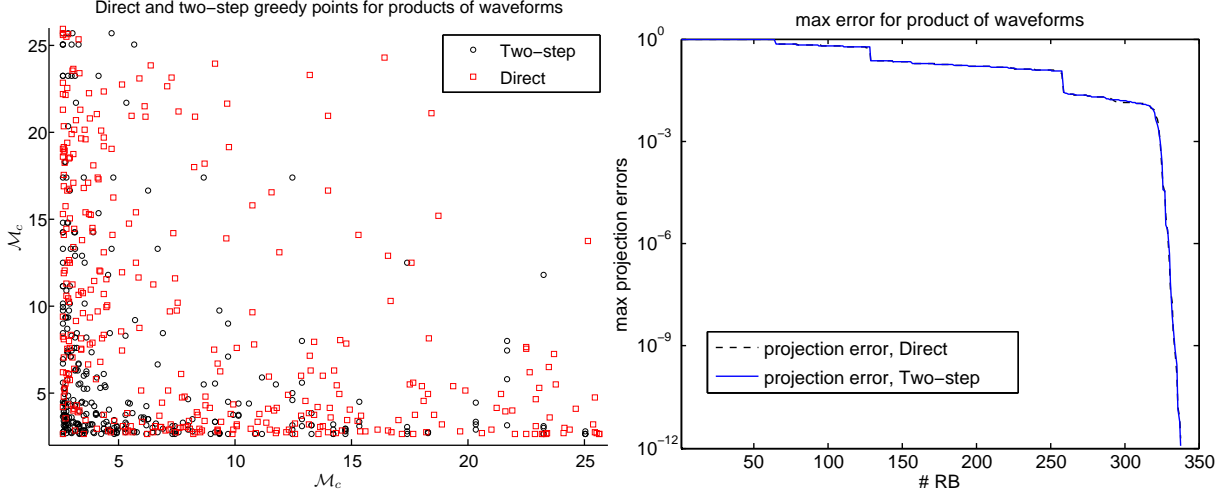
Fig. 7 Selected chirp mass $\mathcal{M}_c$ values (left) and square of the greedy error (right) for both the direct and two-step greedy approaches. Each approach seeks to compress a training space which consists of normalized products of waveforms. However, these training spaces are constructed in very different ways. The two-step training space is built from a Cartesian product of greedy points identified while building an RB space for waveforms (as opposed to their products) and thus has 178^2 members. A direct greedy uses a training space from the Cartesian product of the $\mathcal{T}_K$ defined in Eq. (31) and has 3000^2 members. Remarkably, despite these differences, we find that *exactly* 339 basis vectors are needed to achieve an error tolerance of $\epsilon^2 = 10^{-12}$ for both the two-step and direct greedy. The left figure shows the point distribution is slightly different with more clustering at lower masses for the two-step process (black circles) as compared with the direct one (red squares).

In the previous subsection we constructed an approximation space $F_n \approx \mathcal{F}$. It was found that 178 basis elements are needed to represent any member of a 3000 member training set given by (31) with an accuracy better than $\epsilon^2 = 10^{-12}$. We continue following Algorithm (2) with an aim towards approximating $\widetilde{\mathcal{F}}$. Results for both the two-step and direct greedy algorithms are given, followed by a short discussion.

Two-step greedy results: As discussed in Sec. 3.3 a second RB-greedy is used to build a space $\widetilde{F}_m$ which approximates $\widetilde{\mathcal{F}}$. Following the prescription outlined there, we begin by building a training set $\mathcal{T}_n^2 = \{(\mu_i, \mu_j)\}_{i,j=1}^n$ and a training space F_{n^2} of associated normalized products $\{h_{\mu_i}^* h_{\mu_j} / \|h_{\mu_i}^* h_{\mu_j}\|_{\mathtt{d}}\}_{i,j=1}^n$, where $\{\mu_i\}_{i=1}^n$ are the greedy points identified by the first greedy approximation $\mathcal{F} \approx F_n$ carried out in Sec. 5.2.1. Our definition of F_{n^2} is not the only choice and, in particular, one may build a training space from normalized products of the orthogonalized basis $\{e_i\}_{i=1}^{178}$ (cf. Remark 2).

Through numerical experiments we have found that 339 reduced basis elements are needed to achieve a tolerance of $\epsilon^2 = 10^{-12}$ for approximating F_{n^2}. Fig. 7 (left) shows the distribution of points selected by the greedy algorithm (the two-step greedy results are denoted by black circles). As expected they cluster towards lower values of $\mathcal{M}_c$ (see Sec. 5.2.1). The solid blue line labeled "projection error, Two-step" in Fig. 7 (right) plots the square of the greedy error $\sigma_m(F_{n^2}; \mathcal{H})$, defined in Eq. (12), over the training space of F_{n^2}, as a function of the number of reduced basis elements. Furthermore, through Monte Carlo sampling of the continuum we find any waveform to be accurately represented (see "interpolation error" of Fig. 8, right) by these same 339 basis elements.

Direct greedy results: One may consider direct approximation of $\widetilde{\mathcal{F}}$ through a single RB-greedy (Path # 1 in Algorithm 2). For most problems this will be prohibitively expensive even as an offline computation. Nevertheless, we provide details on it here mainly for comparison with the two-step approach (Path # 2 in Algorithm 2). We take our training set to be a Cartesian product $\mathcal{T}_K^2 = \{(\mu_i, \mu_i)\}_{i,j=1}^K$ of the $K = 3000$ element training set given by (31) and a training space F_{K^2} to be the associated normalized products $h_{\mu_i}^* h_{\mu_j} / \|h_{\mu_i}^* h_{\mu_j}\|_{\mathtt{d}}$. To achieve an approximation tolerance of $\epsilon^2 = 10^{-12}$ we have found that 339 RB elements were needed. Fig. 7 (left) shows the distribution of points selected by the greedy algorithm (the direct greedy results are denoted by red squares). The dashed black line labeled "projection error, Direct" in Figure 7 (right) shows the square of the greedy error $\sigma_m(\mathcal{F}_{K^2}; \mathcal{H})$, defined in Eq. (12), over the training space $\mathcal{F}_{K^2}$, as a function of the number of RB elements.

Discussion and comparison of two-step and direct greedy: We first notice that when building F_{n^2} from products of basis waveforms (as opposed to products of orthonormal basis vectors (cf. Remark 2)) one has $F_{n^2} \subset F_{K^2}$. In light of this observation one may view the two-step greedy as a smarter choice of sampling $\widetilde{\mathcal{F}}$. Furthermore, to better assist with

our comparison, in the numerical experiments carried out above we have initialized both the two-step and direct greedy algorithms with the same seed.

Remarkably, despite the differences in the two approaches in terms of computational cost, we find that the greedy error curves are nearly identical and in both cases *exactly* 339 basis vectors are needed to achieve an error tolerance of $\epsilon^2 = 10^{-12}$ (the greedy errors are defined with respect to different training spaces, however). In both cases after a slowly decaying region we observe very fast exponential convergence of the form $Ce^{-c_0 n^\alpha}$. For the two-step greedy, $\sigma_m(F_{n^2}; \mathcal{H})$ can be fitted by

$$C = 4.19 \times 10^{-3}, c_0 = 0.981, \text{ and } \alpha = 0.923,$$

while for the direct one $\sigma_{\widehat{m}}(\mathcal{F}_{K^2}; \mathcal{H})$ can be fitted by

$$\widehat{C} = 3.98 \times 10^{-3}, \widehat{c}_0 = 1.07 \text{ and } \widehat{\alpha} = 0.875.$$

The number m of basis found to be needed so that $\widetilde{F}_m$ represents $\widetilde{\mathcal{F}}$ within machine precision, $m = 339$, is remarkably close to twice that one needed so that F_n represents $\mathcal{F}$ with the same accuracy, $n = 178$. If we were dealing with polynomials, it would be exactly $m = 2n$. From a practical perspective, with the two-step greedy approach we remove significant redundancy amongst elements of F_{n^2} where the dimensionality of the space to represent products of waveforms is compressed from n^2 to $\sim 2n$, with a compression ratio of about 90 for this problem.

Notice the savings in the offline stage when building the reduced basis space $\widetilde{F}_m$ using our two-step greedy approach, compared to a direct one. For the problem here considered, which is not particularly large in terms of the number of physical parameters (typically an 8 dimensional parameter space for a faithful description of compact binary coalescences [19]), we needed 3,000 training space points to build F_n. Using a direct approach to build $\widetilde{F}_m$ one needs 9×10^6 training space points, compared to the 178^2 needed in the two-step greedy, *with offline savings when building $\widetilde{F}_m$ of* ~ 284. At the end of the day, both basis are able to accurately represent any waveform product and hence perfectly well suited for an ROQ construction. Clearly the two-step is preferable when considering these costs; we refer to Remark 3.

Remark 7 The savings for larger problems, for example when using a lower cutoff frequency of 10Hz, as estimated for the upcoming generation of earth-based detectors, or including spins in the modeling of each compact objects would be considerably larger (even in the absence of precession). For such cases we have typically used $\sim 10^6$ elements in the training space in order to build a high accuracy RB, with less than $\sim 2,000$ RB elements needed to represent F_n [17,18, 19]. For those cases a direct RB-greedy construction of $\widetilde{F}_m$ would in principle require $\sim 10^{12}$ training space elements – *our two-step greedy would then save the offline cost by a factor of $\sim 10^5$*.

### 5.2.3 DEIM and ROQ for overlap/inner-product integration

In the previous subsection we constructed the approximation space $\widetilde{F}_m \approx \widetilde{\mathcal{F}}$ using a direct and two-step greedy algorithms. While both approaches identify an accurate and compact reduced basis set $\{\widetilde{e}_i\}_{i=1}^{339}$ approximating $\widetilde{\mathcal{F}}$, the two-step one has a much smaller computational cost (see Remark 3). We now continue with Algorithm (2). The results shown are for a reduced basis generated from the two-step greedy.

Discrete Empirical Interpolation: We now generate the corresponding set of interpolation points using the DEIM algorithm of Section 3.2. Results for Gauss-Legendre sampling (case 1) are shown in Figure 8, equidistant sampling is qualitatively similar. A distribution of selected DEIM points is depicted in the leftmost plot. Notice that the overall structure is very similar to the single waveform case shown in Figure 5(b).

For 20,000 randomly sampled normalized waveform products $g \in \widetilde{\mathcal{F}}$, not necessarily in the training space, the DEIM interpolant is evaluated using Eq. (22). The DEIM interpolation error is calculated as $\|g - \widetilde{\mathcal{I}}_m[g]\|_{\mathtt{d}}^2$. Fig. 8 (right) compares the largest DEIM interpolation error over all 20,000 products (solid blue line) with the square of the greedy error $\sigma_m(F_{n^2}; \mathcal{H})$ (solid black line). Notice that the DEIM interpolation error is bounded from above by the a-priori error estimates Eq. (25) (red line) and Eq. (26) (purple line). These two error estimates strictly hold for waveforms in the training space, and evidently continue to hold for waveforms outside of the training space when the latter is sufficiently dense. The red line provides a sharper bound on the error, but its also more expensive to compute (cf. Remark 6).

Reduced Order Quadratures: Having assembled a set of basis vectors and empirical interpolation points for products, we compute the reduced order quadrature weights ω_i^{ROQ} with Eq. (21) to complete the ROQ rule. Like the previous experiments of this section, we randomly sample 20,000 inner products (with normalized integrands) to compute, and monitor the maximum error from this computation.

Figure 9 compares accuracy versus computational degrees of freedom (number of quadrature nodes) for a variety of integration schemes. The black and blue lines denote trapezoidal and Gauss-Legendre quadratures, respectively. Two
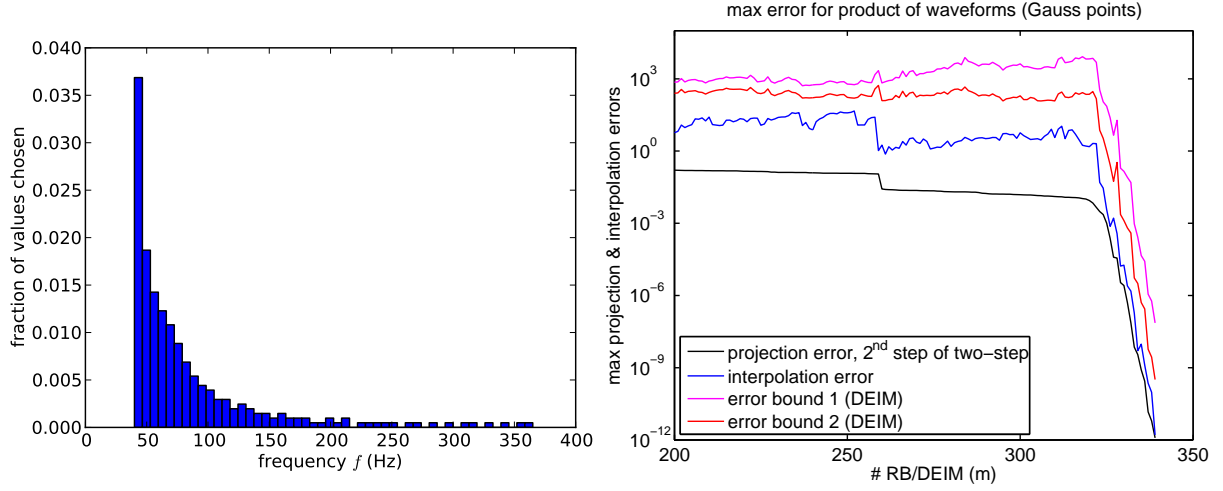
Fig. 8 The left figure shows the distribution of empirical interpolation points selected by the DEIM algorithm with the basis vectors sampled at $1,701$ Gauss-Legendre points. The right figure shows maximum projection and interpolation error (and error bounds) for products of waveforms as a function of the number of reduced basis used; notice that the profile is similar to that one of Fig. 6. On the right figure "projection error" plots the square of the greedy error $\sigma_m(F_{n^2}; \mathcal{H})$ and "interpolation error" plots a maximum error $\|g - \widetilde{\mathcal{I}}_m[g]\|_d^2$ taken over $20,000$ randomly drawn normalized products $g \in \widetilde{\mathcal{F}}$. A-priori interpolation error bounds given by Eq. (25) (error bound 2) and Eq. (26) (error bound 1) are also shown. Results for equidistant sampling are qualitatively similar to the Gauss-Legendre case.

cases are considered within reduced order quadratures. The ROQ leading to the red line stems from a reduced basis space $\{\widetilde{e_i}\}_{i=1}^{339}$ and a $1,701$ point Gauss-Legendre quadrature rule. The magenta one, in turn, uses a $20,000$ point trapezoidal rule. One can see that ROQ have a factor of ~ 2 of savings when compared to Gauss-Legendre points for a maximum error below $10^{-2} - 10^{-1}$, while the savings compared to the extended trapezoidal rule at high resolutions are greater than 50. The Gauss-Legendre comparison provides a benchmark test against the best quadrature for smooth functions and already provides benefits for applications that may take days or weeks to run (e.g. parameter estimation studies with Markov chain Monte Carlo). The more relevant comparison, however, for data driven applications is with the trapezoidal rule. Real data (e.g. measurements taken at GW observatories or, more broadly, signal detection) will not be given at Gauss-Legendre points but rather equally spaced ones and in this setting ROQ significantly outperforms its counterpart.

6 Final remarks

Numerical integration is a well studied topic (see Refs. [55, 24] for excellent introductions). One may wish to identify when Reduced Order Quadratures (ROQ) are likely to be a competitive option over alternative, more standard integration methods. First and foremost, the set of functions to be integrated should be well approximated by a relatively compact basis. Although we have here focused on bases obtained through the Reduced Basis-greedy approach, ROQ apply unchanged to any other basis sets; for example, those obtained through a Proper Orthogonal/Singular Value decomposition. Second, since the cost of building the reduced basis either by a RB-greedy or POD/SVD is potentially large, it should be done *offline*. For larger problems, however, a direct reduced order modeling approach might be unfeasible, even when building the basis is an offline and parallelizable calculation. To overcome this we proposed a two-step greedy targeted towards approximation of products of functions (such as those appearing in weighted inner products) which, for the problem considered here, accelerated offline ROQ computations by two orders of magnitude. Finally, and crucially, one should determine which *online* costs to reduce. Benefits of ROQ are greatest when multiple fast evaluations of weighted inner products (or perhaps simply weighted integrals) are required between parametrized functions. Are the functional evaluations themselves costly? Quadrature rules which are designed for high-order integration of polynomials might lead to significantly more functional evaluations compared to an ROQ rule. Can one sample the function at arbitrary points or is the data sampling specified? In our numerical experiments with gravitational waves the savings were moderate (a factor of ~ 2) when we were able to dictate the sampling location whereas they were much greater (a factor of ~ 50) when the function was sampled at equally spaced points. This suggests the possibility of using ROQ in data analysis applications such as matched filtering and Bayesian parameter estimation.

There are many potential applications of ROQ which deserve further consideration. As pointed out in Ref. [2], empirical interpolation is remarkably versatile in its handling of multi-dimensional problems on irregular domains, and a
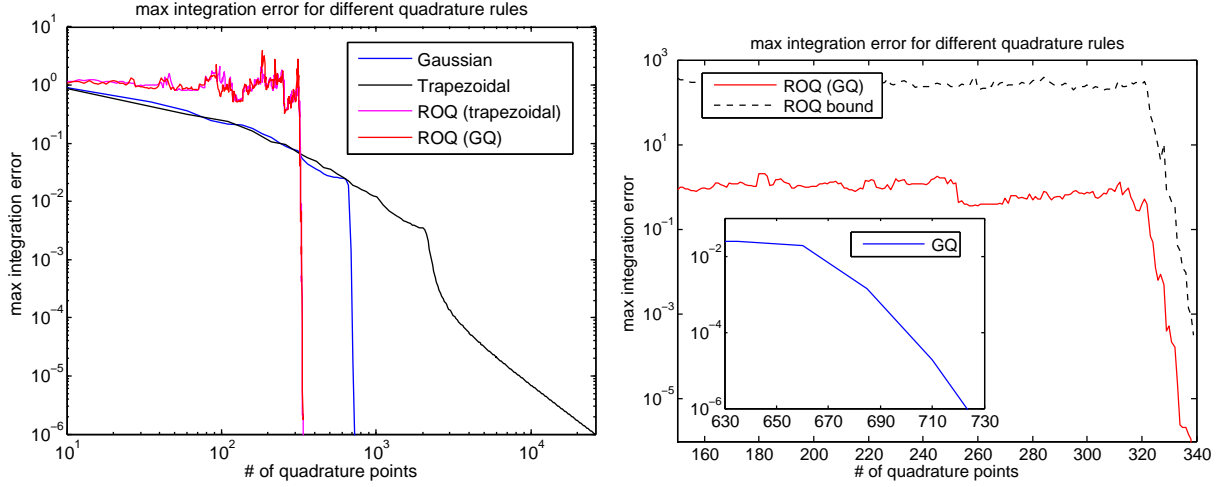
Fig. 9 We randomly draw $20,000$ normalized pair products $g \in \widetilde{\mathcal{F}}$ and for each compute an accurate inner product (whose error is smaller than 10^{-6}) which we take to be its exact value I_c. Next, for each product we compute an inner product using a i) Gauss-Legendre quadrature (blue line), ii) trapezoidal rule (black line), iii) ROQ built from Gauss-Legendre quadratures (red line), and iv) ROQ built from the trapezoidal rule (pink line). In each of the four cases we monitor the maximum errors $|I_c - I_d|$ (for the Gauss-Legendre and trapezoidal rules) and $|I_c - I_{\mathrm{ROQ}}|$ (for the ROQ accelerated Gauss-Legendre and trapezoidal rules). Once the underlying product is well resolved by the empirical interpolant we see very fast exponential convergence of $I_{\mathrm{ROQ}} \to I_c$ (compare with "interpolation error" plotted in the right panel of Figure 8). Notice that both ROQ rules perform similarly, which is to be expected whenever the underlying integration scheme is able to accurately integrate the reduced basis (cf. the discussion following Algorithm 2 and Fig. 2). After an error of about 10^{-1} both ROQ rules outperform their discrete counterparts and in some cases significantly so. The right figure is a semi-log "zoom-in" plot to clearly show the exponential convergence along with error bounds given by Eq. 27. Results for ROQ (trapezoidal) are qualitatively similar and omitted for clarity on the right figure.

natural application of ROQ would be to such problems. Discontinuous functions (in physical space) with smooth variation with respect to parameters might admit a reduced basis approximating to the full space with fast convergence, and if so it might be possible to generate fast converging ROQ for families of discontinuous or noisy functions. Furthermore, as pointed out in Sec. 4, the hierarchical construction of ROQ node locations allows for very natural application-specific nested quadrature rules of arbitrary order and depth.

Acknowledgments

A Greedy and EIM algorithms

A.1 Greedy algorithm

For any $\epsilon > 0$ and training set $\mathcal{T}_K$ the greedy algorithm to build a reduced basis is as follows:

Algorithm 4 (**RB Greedy Algorithm**)
$\left[\{e_\ell\}_{\ell=1}^n, \{\mu_\ell\}_{\ell=1}^n \right] = \text{RB-Greedy } (\epsilon, \mathcal{T}_K)$

(1) Set $n = 0$ and define $\sigma_0(\mathcal{F}_{\mathrm{train}}; \mathcal{H}) := 1$

(2) Choose an arbitrary $\mu_1 \in \mathcal{T}_K$ and set $e_1 := h_{\mu_1}$ **Comment:** $\|h_{\mu_1}\|_{\mathtt{d}} = 1$

(3) do, while $\sigma_n(\mathcal{F}_{\mathrm{train}}; \mathcal{H}) \geq \epsilon$
    (a) $n = n + 1$
    (b) $\sigma_n(h_\mu) := \left\| h_\mu - \mathcal{P}_n h_\mu \right\|_{\mathtt{d}}$ for all $\mu \in \mathcal{T}_K$
    (c) $\sigma_n(\mathcal{F}_{\mathrm{train}}; \mathcal{H}) = \sup_{\mu \in \mathcal{T}_K} \left\{ \sigma_n(h_\mu) \right\}$

(d) $\mu_{n+1} := \mathrm{argsup}_{\mu \in \mathcal{T}_K} \left\{ \sigma_n(h_\mu) \right\}$ (greedy sweep)

(e) $e_{n+1} := h_{\mu_{n+1}} - \mathcal{P}_n h_{\mu_{n+1}}$ (Gram-Schmidt Orthogonalization)

(f) $e_{n+1} := e_{n+1}/\|e_{n+1}\|_{\mathtt{d}}$ (normalization)

Remark 8 In step 3b the error is computed exactly as $\left\| h_\mu - \mathcal{P}_n h_\mu \right\|_{\mathtt{d}}$. In the setting of PDEs, RB methods avoid exact error computations by using parametric error estimators and without computing full solutions. In the context of the empirical interpolation method Ref. [2] suggests using $\hat{\sigma}(h_\mu) = \|h_\mu - I_n[h_\mu]\|_{\mathtt{d}}$, where $I_n[h_\mu]$ is the empirical interpolant. Indeed, for applications limited by computational resources using $\hat{\sigma}(h_\mu)$ could be desirable. Within the Hilbert space setting described in this paper, however, an exact error $\sigma(h_\mu)$ computation results in a reduced basis space which will more accurately approximate the full space (either $\mathcal{F}$ or $\widetilde{\mathcal{F}}$) and should be used whenever possible.

A.2 EIM algorithm

Algorithm 5 (Selection of DEIM Points)

$\left[\mathbf{P}, \{p_i\}_{i=1}^n \right] = \mathrm{DEIM} \left(\mathbf{V}, \{x_k\}_{k=1}^M \right)$ **Comment:** the column vectors of $\mathbf{V}$ must be linearly independent

(1) $j = \mathrm{argmax} \, |\mathbf{e}_1|$ **Comment:** here argmax takes a vector and returns the *index* of its largest entry

(2) Set $\mathbf{U} = [\mathbf{e}_1]$, $\mathbf{P} = [\hat{\mathbf{e}}_j]$, $p_1 = x_j$ **Comment:** $\hat{\mathbf{e}}_j$ is a unit column vector with a single unit entry at index j

(3) for $i = 2, \ldots, n$ do

   (4) Solve $(\mathbf{P}^T \mathbf{U})\mathbf{c} = \mathbf{P}^T \mathbf{e}_i$ for $\mathbf{c}$

   (5) $\mathbf{r} = \mathbf{e}_i - \mathbf{U}\mathbf{c}$

   (6) $j = \mathrm{argmax} \, |\mathbf{r}|$

   (7) Set $\mathbf{U} = [\mathbf{U} \quad \mathbf{r}]$, $\mathbf{P} = [\mathbf{P} \quad \hat{\mathbf{e}}_j]$, $p_i = x_j$

The DEIM algorithm described above is nearly identical to the one given in Ref. [4] with the exception of Step 7 which is $\mathbf{U} = [\mathbf{U} \quad \mathbf{e}_i]$ in that reference. In Appendix C we show these algorithms to be equivalent and, furthermore, we show a relationship between DEIM and Gauss-elimination with partial pivoting. Owing to the lower triangular form of $\mathbf{P}^T \mathbf{U}$, in Appendix B we show that Algorithm 5 has a reduced computational cost.

B Asymptotic FLOP Count

B.1 DEIM FLOP Count

In Algorithm 5 as $\mathbf{P}^T \mathbf{U}$ is a lower triangular matrix therefore (after m iterations) Step (4) costs $\mathcal{O}(m^3)$, Step (5) costs $\mathcal{O}(Mm)$ subtractions and $\mathcal{O}(Mm^2)$ matrix vector multiplications, where latter is the dominant one. The improvement in our implementation of Algorithm 5 as compared to [4] is that the our implementation scales as $\mathcal{O}(m^3)$ as compared to $\mathcal{O}(m^4)$ in [4]. Therefore the total DEIM cost is $\mathcal{O}(Mm^2 + m^3)$.

B.2 ROQ FLOP Count

The dominant cost (assuming $M > m$) of computing reduced order quadrature weights $\left(\boldsymbol{\omega}^{\mathrm{ROQ}} \right)^T = \boldsymbol{\omega}^T \widetilde{\mathbf{V}} \left(\widetilde{\mathbf{P}}^T \widetilde{\mathbf{V}} \right)^{-1}$ in (21) is $\mathcal{O}(Mm^2)$. Here we used the fact that the operation $\widetilde{\mathbf{P}}^T \widetilde{\mathbf{V}}$ is equivalent to selecting m rows corresponding to DEIM points $\{\widetilde{p}_\ell\}_{\ell=1}^m$.

Then the overall cost to compute ROQ using one-step (Algorithm 4 [*Path #1*] with modified *Gram-Schmidt* in greedy is:

$$\mathcal{O}(K^2 M \widehat{m} + M \widehat{m}^2)$$

and two-step (Algorithm 4 [*Path #2*] is:

$$\mathcal{O}(n^2 Mm + KMn + Mm^2) \, .$$

Here m, $\widehat{m}$ denote the number of reduced basis used in approximation of $\widetilde{\mathcal{F}}$ via two-step and direct approach respectively and n is the number of reduced basis used to approximate $\mathcal{F}$.

C DEIM and LU Decomposition with Partial Pivoting

The original DEIM described in [4] has as Step 7

$$\mathbf{U} = [\mathbf{U} \quad \mathbf{e}_i], \quad i = 2, \ldots n \, .$$

Since the columns of $\mathbf{U}$ are linearly independent, $\mathbf{P}^T \mathbf{U}$ is always invertible, but in this form the matrix $\mathbf{P}^T \mathbf{U}$ can be dense. Next we show that with a slight modification to the original DEIM, we get Algorithm 5 and at every iteration $i = 2, \ldots, n$ with $\mathbf{r}_i := \mathbf{r}$,

$$\mathbf{U} := [\mathbf{U} \quad \mathbf{r}_i] \, ,$$

gives the same result. One of the advantages of looking at DEIM in this format is that the matrix $\mathbf{P}^T \mathbf{U}$ is lower triangular, therefore the system can be solved for $\mathbf{c}$ with $\mathcal{O}(n^2)$ operations, as compared to using $\mathbf{e}_i$, which gives a dense matrix requiring $\mathcal{O}(n^3)$ operations.

Next we prove that we get the same result using both formats.

Proposition 1 *In [4], Step 7 of Algorithm 5 was presented as* $\mathbf{U} = [\mathbf{U} \quad \mathbf{e}_i]$, $i = 2, \ldots, n$, *we can replace it by*

$$\mathbf{U} = [\mathbf{U} \quad \mathbf{r}_i], \qquad i = 2, \ldots, n,$$

where $\mathbf{r}_i := \mathbf{r}$, *at every iteration, and* $\mathbf{r}$ *is as given in Step 5 of Algorithm 5.*

Proof Step 4 of Algorithm 5, at $i = n$, gives the coefficient matrix

$$\mathbf{P}^T \mathbf{U} = \begin{pmatrix} \mathbf{e}_1(p_1) & \mathbf{e}_2(p_1) & \cdots & \mathbf{e}_{n-1}(p_1) \\ \mathbf{e}_1(p_2) & \mathbf{e}_2(p_2) & \cdots & \mathbf{e}_{n-1}(p_2) \\ \mathbf{e}_1(p_3) & \mathbf{e}_2(p_3) & \cdots & \mathbf{e}_{n-1}(p_3) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e}_1(p_{n-1}) & \mathbf{e}_2(p_{n-1}) & \cdots & \mathbf{e}_{n-1}(p_{n-1}) \end{pmatrix}. \tag{32}$$

Next we write the row reduced echelon form using forward Gaussian elimination for $(\mathbf{P}^T \mathbf{U})^T$, where the first step implies

$$\begin{pmatrix} \mathbf{e}_1(p_1) & 0 & \cdots & 0 \\ \mathbf{e}_1(p_2) & \mathbf{r}_2(p_2) & \cdots & \mathbf{e}_{n-1}(p_2) - \frac{\mathbf{e}_{n-1}(p_1)}{\mathbf{e}_1(p_1)}\mathbf{e}_1(p_2) \\ \mathbf{e}_1(p_3) & \mathbf{r}_2(p_3) & \cdots & \mathbf{e}_{n-1}(p_3) - \frac{\mathbf{e}_{n-1}(p_1)}{\mathbf{e}_1(p_1)}\mathbf{e}_1(p_3) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e}_1(p_{n-1}) & \mathbf{r}_2(p_{n-1}) & \cdots & \mathbf{e}_{n-1}(p_{n-1}) - \frac{\mathbf{e}_{n-1}(p_1)}{\mathbf{e}_1(p_1)}\mathbf{e}_1(p_{n-1}) \end{pmatrix}.$$

The final row reduced echelon form for $(\mathbf{P}^T \mathbf{U})^T$ is

$$\begin{pmatrix} \mathbf{e}_1(p_1) & 0 & \cdots & 0 \\ \mathbf{e}_1(p_2) & \mathbf{r}_2(p_2) & \cdots & 0 \\ \mathbf{e}_1(p_3) & \mathbf{r}_2(p_3) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e}_1(p_{n-1}) & \mathbf{r}_2(p_{n-1}) & \cdots & \mathbf{r}_n(p_{n-1}) \end{pmatrix}. \tag{33}$$

Hence the proposition.

Remark 9 Algorithm 5 has a flavor of Gauss elimination with row pivoting. We showed the elimination part in Proposition 1. Now we show that the DEIM points $\{p_1, \ldots, p_n\}$ are in fact the pivots. Our goal is to arrive at (33) row echelon form of $(\mathbf{P}^T \mathbf{U})^T$ with with partial pivoting. Consider the matrix (32). Let the location of the first pivot, i.e., the maximum of the absolute value of the first column be p_1 (this is the same as Step 1 in Algorithm 5). Apply the first step of forward elimination as in the proof of Proposition 1. Next compute the second pivot p_2, which is the maximum of the absolute value of the so-obtained second column. Following in this manner we arrive at the matrix in (33).

References

1. A. Patera, G. Rozza, Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering. http://augustine.mit.edu/methodology/methodology_bookPartI.htm.
2. Y. Maday, N. C. Nguyen, A. T. Patera, S. H. Pau, A general multipurpose interpolation procedure: the magic points, Communications on Pure and Applied Analysis 8 (2009) 383–404.
3. M. Barrault, Y. Maday, N. C. Nguyen, A. T. Patera, An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations, Comptes Rendus Mathematique 339 (2004) 667–672. doi:10.1016/j.crma.2004.08.006.
4. S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing 32 (5) (2010) 2737–2764. doi:10.1137/090766498.
   URL http://link.aip.org/link/?SCE/32/2737/1
5. S. Chaturantabut, D. Sorensen, Discrete empirical interpolation for nonlinear model reduction, in: Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, 2009, pp. 4316 –4321. doi:10.1109/CDC.2009.5400045.
6. M. Hinze, S. Volkwein, Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and sub-optimal control, in: Dimension reduction of large-scale systems, Vol. 45 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2005, pp. 261–306.
7. D. Manolakis, V. Ingle, S. Kogon, Statistical and adaptive signal processing: spectral estimation, signal modeling, adaptive filtering, and array processing, Artech House signal processing library, Artech House, 2000.
   URL http://books.google.com/books?id=3RQfAQAAIAAJ
8. L. A. Wainstein, V. D. Zubakov, Extraction of signals from noise, Prentice-Hall, London, 1962.
9. J. L. Eftang, B. Stamm, Parameter multi-domain empirical interpolation, International Journal for Numerical Methods in Engineering 90 (4) (2012) 412–428. doi:10.1002/nme.3327.
   URL http://dx.doi.org/10.1002/nme.3327

10. T. O. Aanonsen, Empirical interpolation with application to reduced basis approximations, Ph.D. thesis, Norwegian University of Science and Technology (2009).
    URL `http://ntnu.diva-portal.org/smash/record.jsf?pid=diva2:348771`
11. A. Pinkus, N-widths in approximation theory, Springer, Amsterdam, 1985.
12. P. Binev, A. Cohen, W. Dahmen, R. A. DeVore, G. Petrova, P. Wojtaszczyk, Convergence rates for greedy algorithms in reduced basis methods., SIAM J. Math. Analysis 43 (3) (2011) 1457–1472.
    URL `http://dblp.uni-trier.de/db/journals/siamma/siamma43.html#BinevCDDPW11`
13. R. DeVore, G. Petrova, P. Wojtaszczyk, Greedy algorithms for reduced bases in banach spaces, Arxiv preprint arXiv:1204.2290.
14. J. L. Eftang, A. T. Patera, E. M. Ronquist, An hp certified reduced basis method for parametrized elliptic partial differential equations, SIAM J. Sci. Comput. 32 (2010) 3170–3200.
15. T. Bui-Thanh, Model-constrained optimization methods for reduction of parameterized large-scale systems, PhD thesis, Massachusetts Institute of Technology.
16. B. Haasdonk, M. Dihlmann, M. Ohlberger, A training set and multiple bases generation approach for parametrized model reduction based on adaptive grids in parameter space, Tech. Report, SRC SimTech.
17. S. Caudill, S. E. Field, C. R. Galley, F. Herrmann, M. Tiglio, Reduced Basis representations of multi-mode black hole ringdown gravitational waves, Class. Quant. Grav. 29 (2012) 095016. `arXiv:1109.5642`.
18. S. E. Field, C. R. Galley, F. Herrmann, J. S. Hesthaven, E. Ochsner, M. Tiglio, Reduced basis catalogs for gravitational wave templates, Phys. Rev.Lett. 106 (2011) 221102. `arXiv:1101.3765, doi:10.1103/PhysRevLett.106.221102`.
19. S. E. Field, C. R. Galley, E. Ochsner, Towards beating the curse of dimensionality for gravitational waves using reduced basis, Phys. Rev. D 86 (2012) 084046. `doi:10.1103/PhysRevD.86.084046`.
    URL `http://link.aps.org/doi/10.1103/PhysRevD.86.084046`
20. K. Cannon, A. Chapman, C. Hanna, D. Keppel, A. C. Searle, et al., Singular value decomposition applied to compact binary coalescence gravitational-wave signals, Phys. Rev. D82 (2010) 044025. `arXiv:1005.0012, doi:10.1103/PhysRevD.82.044025`.
21. K. Cannon, C. Hanna, D. Keppel, Efficiently enclosing the compact binary parameter space by singular-value decomposition, Phys.Rev. D84 (2011) 084003. `arXiv:1101.4939, doi:10.1103/PhysRevD.84.084003`.
22. J. Xu, L. Zikatanov, Some observations on babuška and brezzi theories, Numer. Math. 94 (1) (2003) 195–202. `doi:10.1007/s002110100308`.
    URL `http://dx.doi.org/10.1007/s002110100308`
23. D. B. Szyld, The many proofs of an identity on the norm of oblique projections, Numer. Algorithms 42 (3-4) (2006) 309–323. `doi:10.1007/s11075-006-9046-2`.
    URL `http://dx.doi.org/10.1007/s11075-006-9046-2`
24. A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer, Berlin, 2010.
25. B. C. Barish, R. Weiss, LIGO and the detection of gravitational waves, Phys. Today 52N10 (1999) 44–50.
26. S. J. Waldmann, Status of LIGO at the start of the fifth science run, Class. Quantum Grav. 23 (2006) S653–S660.
27. S. Hild, The status of GEO 600, Class. Quantum Grav. 23 (2006) S643–S651.
28. F. Acernese, et al., The Virgo status, Class. Quantum Grav. 23 (2006) S635–S642.
29. B. Abbott, et al., LIGO: The Laser Interferometer Gravitational-Wave Observatory `arXiv:0711.3041`.
30. J. Abadie, et al., Predictions for the Rates of Compact Binary Coalescences Observable by Ground-based Gravitational-wave Detectors, Class. Quantum Grav. 27 (2010) 173001. `arXiv:1003.2480, doi:10.1088/0264-9381/27/17/173001`.
31. J. Centrella, J. G. Baker, B. J. Kelly, J. R. van Meter, Black-hole binaries, gravitational waves, and numerical relativity, Rev.Mod.Phys. 82 (2010) 3069. `arXiv:1010.5260, doi:10.1103/RevModPhys.82.3069`.
32. F. Pretorius, Binary Black Hole Coalescence `arXiv:0710.1338`.
33. T. W. Baumgarte, S. L. Shapiro, Numerical relativity and compact binaries, Physics Reports 376 (2) (2003) 41–131. `arXiv:gr-qc/0211028`.
34. O. Sarbach, M. Tiglio, Continuum and Discrete Initial-Boundary-Value Problems and Einstein's Field Equations To appear in Living Reviews in Relativity, 191 pages. `arXiv:1203.6443`.
35. L. Blanchet, Gravitational radiation from post-newtonian sources and inspiralling compact binaries, Living Reviews in Relativity 9 (4).
    URL `http://www.livingreviews.org/lrr-2006-4`
36. L. Blanchet, T. Damour, B. R. Iyer, Gravitational waves from inspiralling compact binaries: Energy loss and wave form to second post-Newtonian order, Phys. Rev. D51 (1995) 5360. `arXiv:gr-qc/9501029, doi:10.1103/PhysRevD.51.5360`.
37. C. M. Will, A. G. Wiseman, Gravitational radiation from compact binary systems: gravitational waveforms and energy loss to second post-Newtonian order, Phys. Rev. D 54 (1996) 4813–4848. `arXiv:gr-qc/9608012`.
38. B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, J. D. Creighton, FINDCHIRP: An Algorithm for detection of gravitational waves from inspiraling compact binaries `arXiv:gr-qc/0509116`.
39. L.-C. Tu, Q. Li, Q.-L. Wang, C.-G. Shao, S.-Q. Yang, et al., New determination of the gravitational constant G with time-of-swing method, Phys.Rev. D82 (2010) 022001. `doi:10.1103/PhysRevD.82.022001`.
40. B. J. Owen, Search templates for gravitational waves from inspiraling binaries: Choice of template spacing, Phys. Rev. D 53 (1996) 6749.
41. B. J. Owen, B. S. Sathyaprakash, Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement, Phys. Rev. D 60 (1999) 022002. `arXiv:gr-qc/9903108`.
42. S. Babak, R. Balasubramanian, D. Churches, T. Cokelaer, B. S. Sathyaprakash, A template bank to search for gravitational waves from inspiralling compact binaries: I. Physical models, Classical and Quantum Gravity 23 (2006) 5477–5504. `arXiv:arXiv:gr-qc/0604037, doi:10.1088/0264-9381/23/18/002`.
43. L. S. Finn, Detection, measurement and gravitational radiation, Phys.Rev. D46 (1992) 5236–5249. `arXiv:gr-qc/9209010, doi:10.1103/PhysRevD.46.5236`.
44. M. Maggiore, Gravitational Waves Volume 1: Theory and Experiments, Oxford University Press, Oxford, 2008.
45. B. Abbott, et al., LIGO: The Laser Interferometer Gravitational-Wave Observatory, Rept. Prog. Phys. 72 (2009) 076901. `arXiv:0711.3041, doi:10.1088/0034-4885/72/7/076901`.
46. P. Ajith, S. Bose, Estimating the parameters of non-spinning binary black holes using ground-based gravitational-wave detectors: Statistical errors, Phys.Rev. D79 (2009) 084032. `arXiv:0901.4936, doi:10.1103/PhysRevD.79.084032`.

47. B. Abbott, et al., Einstein@home search for periodic gravitational waves in ligo s4 data, Phys. Rev. D 79 (2009) 022001. `doi:10.1103/PhysRevD.79.022001`.
    URL `http://link.aps.org/doi/10.1103/PhysRevD.79.022001`
48. P. R. Brady, T. Creighton, Searching for periodic sources with LIGO. 2. Hierarchical searches, Phys.Rev. D61 (2000) 082001. `arXiv:gr-qc/9812014,doi:10.1103/PhysRevD.61.082001`.
49. M. Vallisneri, Beyond Fisher: exact sampling distributions of the maximum-likelihood estimator in gravitational-wave parameter estimation, Phys. Rev. Lett. 107 (2011) 191104. `arXiv:1108.1158,doi:10.1103/PhysRevLett.107.191104`.
50. J. Kanner, et al., LOOC UP: Locating and observing optical counterparts to gravitational wave bursts, Class. Quantum Grav. 25 (2008) 184034. `arXiv:0803.0312,doi:10.1088/0264-9381/25/18/184034`.
51. P. S. Shawhan, LIGO Scientific Collaboration, Virgo Collaboration, LOOC-UP: Seeking Optical Counterparts to Gravitational-Wave Signal Candidates, in: Bulletin of the American Astronomical Society, Vol. 42 of Bulletin of the American Astronomical Society, 2010, pp. 230–+.
52. K. Cannon, R. Cariou, A. Chapman, M. Crispin-Ortuzar, N. Fotopoulos, et al., Toward Early-Warning Detection of Gravitational Waves from Compact Binary Coalescence, Astrophys.J. 748 (2012) 136. `arXiv:1107.2665`.
53. F. Casenave, Accurate a posteriori error evaluation in the reduced basis method, Comptes Rendus Mathematique 350 (9âĂŞ10) (2012) 539 – 542. `doi:10.1016/j.crma.2012.05.012`.
    URL `http://www.sciencedirect.com/science/article/pii/S1631073X12001483`
54. C. Canuto, T. Tonn, K. Urban, A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear pdes, SIAM J. Numer. Anal. 47 (3) (2009) 2001–2022. `doi:10.1137/080724812`.
    URL `http://dx.doi.org/10.1137/080724812`
55. W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, Numerical Recipes, 2nd Edition, Cambridge University Press, New York, 1992.