

Fast simulation of Gaussian-mode scattering for precision interferometry

Freise, Andreas; Brown, Daniel; Smith, Rory

DOI:

[10.1088/2040-8978/18/2/025604](https://doi.org/10.1088/2040-8978/18/2/025604)

License:

Creative Commons: Attribution (CC BY)

Document Version

Publisher's PDF, also known as Version of record

Citation for published version (Harvard):

Freise, A, Brown, D & Smith, R 2016, 'Fast simulation of Gaussian-mode scattering for precision interferometry', *Journal of Optics*, vol. 18, no. 2, 025604. <https://doi.org/10.1088/2040-8978/18/2/025604>

[Link to publication on Research at Birmingham portal](#)

Publisher Rights Statement:

Checked Feb 2016

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Fast simulation of Gaussian-mode scattering for precision interferometry

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2016 J. Opt. 18 025604

(<http://iopscience.iop.org/2040-8986/18/2/025604>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 147.188.224.215

This content was downloaded on 12/02/2016 at 12:43

Please note that [terms and conditions apply](#).

Fast simulation of Gaussian-mode scattering for precision interferometry

D Brown¹, R J E Smith² and A Freise¹

¹University of Birmingham, Edgbaston, B15 2TT, UK

²LIGO, California Institute of Technology, Pasadena, CA 91125, USA

E-mail: ddb@star.sr.bham.ac.uk

Received 2 September 2015, revised 5 November 2015

Accepted for publication 16 November 2015

Published 5 January 2016



CrossMark

Abstract

Understanding how laser light scatters from realistic mirror surfaces is crucial for the design, commissioning and operation of precision interferometers, such as the current and next generation of gravitational-wave detectors. Numerical simulations are indispensable tools for this task but their utility can in practice be limited by the computational cost of describing the scattering process. In this paper we present an efficient method to significantly reduce the computational cost of optical simulations that incorporate scattering. This is accomplished by constructing a near optimal representation of the complex, multi-parameter 2D overlap integrals that describe the scattering process (*referred to as a reduced order quadrature*). We demonstrate our technique by simulating a near-unstable Fabry–Perot cavity and its control signals using similar optics to those installed in one of the LIGO gravitational-wave detectors. We show that using reduced order quadrature reduces the computational time of the numerical simulation from days to minutes (a speed-up of $\approx 2750\times$) while incurring negligible errors. This significantly increases the feasibility of modelling interferometers with realistic imperfections to overcome current limits in state-of-the-art optical systems. While we focus on the Hermite–Gaussian basis for describing the scattering of the optical fields, our method is generic and could be applied with any suitable basis. An implementation of this reduced order quadrature method is provided in the open source interferometer simulation software FINESSE.

Keywords: simulation, scattering, gaussian beams, reduced order quadrature

(Some figures may appear in colour only in the online journal)

1. Introduction

Laser interferometers have long been an exceptional tool for enabling high-precision measurements. With ever increasing demands on their performance, new techniques and tools have been developed to design and build the next-generation of instruments. This has especially been true in the development of gravitational-wave detectors over the last several decades [1–4]. Such ground-based gravitational-wave detectors are based on a Michelson interferometer and are enhanced with Fabry–Perot cavities. Detecting gravitational waves is still

one of the major challenges in experimental physics and the interferometers used include numerous new optical technologies to reach unprecedented displacement sensitivities beyond 10^{-19} m/ $\sqrt{\text{Hz}}$. At the same time these instruments are exploring a new regime of coupling between light and macroscopic objects at the quantum level [5].

Some of these detectors are currently being upgraded to have a ten-fold increase in sensitivity using a much higher circulating power [6, 7]. To achieve their target performance the detectors undergo several years of commissioning, during which the interferometers are carefully tested and improved towards their designed operational state. Numerical simulations are important tools to diagnose causes of any unexpected behavior seen during commissioning; to suggest solutions to potential problems and for advising the design of detector upgrades. Hence, there is a long history of



Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

developing and using dedicated optical simulation tools for the commissioning and design of gravitational-wave detectors [8, 9, 11].

One of the key aspects for the current instruments is the high circulating laser power, up to hundreds of kiloWatts, required for a broadband reduction of shot-noise. It has been recognized for some time that the thermal deformations of the optics due to spurious absorption can degrade the performance of the interferometers [12]. Numerical models have been used extensively in the investigation of such problems and in the development of mitigating solutions (for example [13, 14]). Thermally induced distortions and other effects related to the laser beam shape are still limiting factors of the instruments today and are concerns for the design of future detectors [15]. Furthermore, similar effects can limit the performance of other optical precision measurements such as optical clocks [16] or the optical readout of atomic systems [17]. Mitigation strategies for beam shape distortions in complex interferometers are actively being developed and require accurate numerical models for their design and development.

Initially, the simulation tools for investigating distorted beams used a grid-based field description. Beam distortions can also be modeled effectively using an expansion into spatial cavity eigenmodes [18], such as Hermite–Gauss modes. The interaction of the beam shape with a distorted optical surface often requires the computation of a scattering matrix based on measured or simulated profiles of the distorted surface. This is always true for mode-based simulation programs but is also required for grid-based codes when specific shapes of the beam are important, for example, for the investigations of parametric instabilities [19, 20]. If this matrix has to be re-generated, for example when the effects of a change of a surface shape is being investigated, this element of the computation can dominate the total time required for the entire simulation. A prominent example is that when the circulating laser power within the LIGO interferometers thermally warps the mirror surfaces changing the shape of the laser beams and requiring a re-calculation of many scattering matrices. Including this effect can increase the computation time from minutes to days.

Some of us are providing numerical simulation support for the commissioning of the LIGO interferometers [21]. We use our own simulation tool FINESSE [22] and are maintaining parameter files for the detectors [23]. Commissioning tackles the unexpected behavior of the interferometers and must take into account the sometimes rapid progress of the experimental setup. Therefore support provided with numerical models must fulfil two criteria: (a) we must be able to accurately model the current experimental setup in the presence of distortions and deviations from the design and (b) we must be able to provide a quick response to new questions to inform the management of the activities on site in real time. FINESSE is a frequency-domain tool, using Hermite–Gauss modes to describe beam distortions and is thus ideally suited as a rapid and accurate tool.

Our investigations with numerical tools typically consist of a sequence of different subtasks, sometimes using different

tools, alternating with an expert review of intermediate or preliminary results. This is a very different pattern of tasks to those that benefit from a computer cluster or super computer. Instead, our work requires lightweight and flexible tools with computing times up to minutes or hours. Because of this, strategies to ameliorate the run time of simulations are of high importance to provide fast diagnosis of unexpected behavior; to allow the parameter space of the simulations to be probed exhaustively; to improve the resolution of simulations at a fixed run-time, and to allow simulations to be run on less powerful and cheaper hardware.

In this paper we present a new approach that reduces the computational time of simulations based on modal models by several orders of magnitude. We specifically target the computational cost of computing scattering matrices for optical simulations. Our approach is based on a near-optimal formulation of the integrals required to compute the scattering matrices, known as a *reduced order quadrature* (ROQ) [24]. The ROQ has already been applied in the context of astronomical data analysis with LIGO [25] where the repeated computation of quantities similar to the scattering matrix dominate the run time of the analysis codes. Crucially, the ROQ is designed to provide huge improvements to computational efficiency while maintaining computational precision.

The ROQ can be regarded as a type of near-optimal, application specific, downsampling of the integrands needed to compute the integrals for the scattering matrices [24]. It is analogous to Gaussian quadrature, but whereas Gaussian quadrature is designed to provide exact results for polynomials of a certain degree, the ROQ produces nearly-exact results for arbitrary parametric functions. Importantly, we are able to place tight error bounds on the accuracy of the ROQ for a particular application [24] making it an ideal technique to speed up costly integrals. It exploits an offline/online methodology in which we recast the expensive integrals used to compute scattering matrices into a more computationally efficient form in the ‘offline’ stage. This is then used for the rapid ‘online’ evaluation of the scattering matrices. The offline stage can itself be computationally expensive, however it need only be performed once and is easily parallelized. The data computed in the offline stage—that is needed by the ROQ—can be stored and shared for particular use cases in the online stage so that the offline cost does not need to be factored in at run time.

We describe the algorithm in a general form and report on the implementation and performance of this method in an example task for the LIGO interferometers. The implementation of the method described in this article is available as open source as part of the FINESSE source code and the Python based package PYKAT [26], which will also contain the offline computed data to enable others to model Advanced LIGO like arm cavities. Our particular implementation here is used to provide a simple, real-world example. However, the algorithm can be easily implemented in other types of simulation tools, for example, time domain simulations or grid based tools (also known as FFT simulations) that compare beam shapes. In all cases our algorithm can significantly

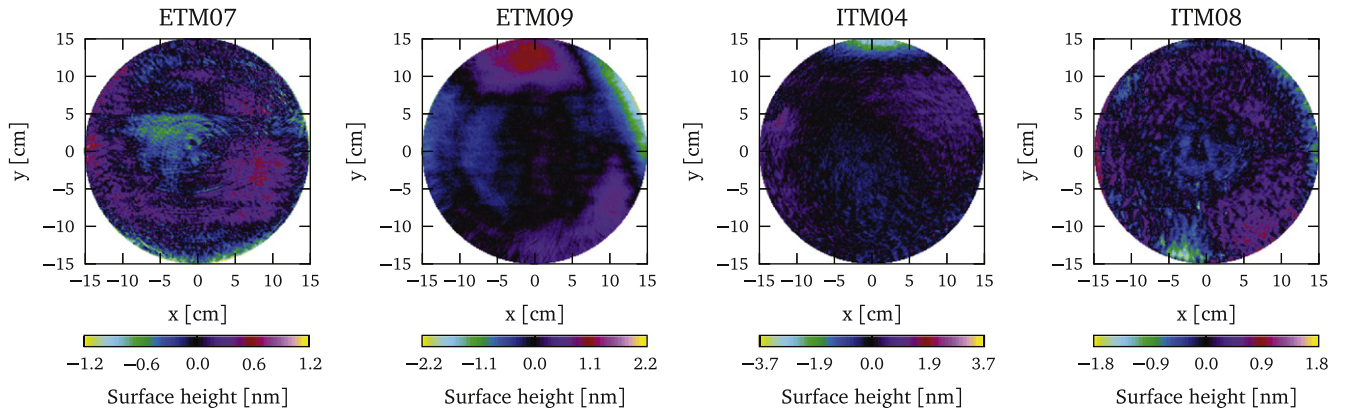


Figure 1. Measured surface distortions for the mirrors currently installed in the Livingston LIGO site (shown here are the distortions of the test masses before they were coated). ETM09 and ITM08 are installed in the x -arm and ETM07 and ITM04 in the y -arm [29]. These measurements have been processed to remove the overall mirror curvatures, offset and tilts [30].

reduce the computation time for evaluating overlap integrals of Gaussian modes with numerical data.

The paper is outlined as follows: in section 2 we give an overview of the paraxial description of the optical eigenmodes and scattering into higher order modes. In section 3 we provide the mathematical background and algorithm for producing the ROQ. Section 3 heavily relies on an additional mathematical technique known as the ‘empirical interpolation (EI) method’ [27]. We assume no prior knowledge of this and provide the main details and results necessary for the ROQ. Section 4 then highlights an exemplary case to demonstrate our method for modeling near-unstable optical cavities. Finally in section 5 the computational performance of our method is analysed.

2. Higher-order optical modes

Gravitational wave detectors are constructed of multiple optical cavities based on a Michelson interferometer. The circulating laser beams in such an optical setup is well described by the the paraxial Gaussian eigenmodes of a spherical cavity in the Hermite–Gaussian (HG) basis; an efficient basis for describing the spatial properties of a laser beam in the transverse plane to the propagation axis and any perturbations to it [28]. The complex transverse spatial amplitude of these HG modes is given by:

$$u_{nm}(x, y, q_x, q_y) = u_n(x, q_x)u_m(y, q_y), \quad (1)$$

$$u_n(x, q) = \left(\frac{2}{\pi}\right)^{1/4} \left(\frac{1}{2^n n! w_0}\right)^{1/2} \left(\frac{q_0}{q}\right)^{1/2} \times \left(\frac{q_0 q^*}{q_0^* q}\right)^{n/2} H_n\left(\frac{\sqrt{2}x}{w(z)}\right) \exp\left(-i\frac{kx^2}{2q}\right). \quad (2)$$

Here n defines the order of the Hermite polynomials H_n in the x axis and m for the y , w_x and w_y are the beam spot sizes in the x and y directions, k is the wavenumber of the laser light, $\mathbf{q} = \{q_x, q_y\}$ are the complex *beam parameters* in the x and y directions, and $q_0 = q(z = 0)$ the beam parameter at the

beam waist. The shape of the Gaussian mode is fully defined by the wavelength of the light λ and the beam parameter

$$q_x = z_x + iz_{R,x} = z_x + i\frac{\pi w_{0,x}^2}{\lambda}, \quad (3)$$

where z_x is the distance from the waist, $z_{R,x}$ is the Rayleigh range, $w_{0,x}$ is the size of the waist and $\lambda = 1064$ nm is the wavelength of the Nd:YAG laser used in current GW detectors. The same set of parameters exists for q_y . The order of the optical mode is $\mathcal{O} = n + m$ and individual modes are typically referred to as TEM_{nm} . Up to an order \mathcal{O} there are

$$N_{\text{HOM}}(\mathcal{O}) = \frac{1}{2}(\mathcal{O} + 1)(\mathcal{O} + 2) \quad (4)$$

higher order optical modes. A laser field with a single optical frequency component ω can be expanded into a beam basis whose shape is described by \mathbf{q} as:

$$E(x, y, t; \mathbf{q}) = \sum_{n=0, m=0}^{n+m \leq \mathcal{O}_{\max}} a_{nm} u_{nm}(x, y; \mathbf{q}) e^{i\omega t} \quad (5)$$

where a_{nm} is a complex value describing the amplitude and phase of a mode TEM_{nm} and \mathcal{O}_{\max} is the maximum order of modes included in the expansion.

2.1. Scattering into higher-order modes

When a field interacts with an optical component its mode content is typically changed. Here we define scattering as the relationship between the mode content of the outgoing beams \bar{a} with a beam shape \mathbf{q} , and the mode content of the incoming beam \bar{a}' described with \mathbf{q}' . Mathematically this is simply $\bar{a} = \hat{k}\bar{a}'$ where \hat{k} is known as the *scattering matrix*. Now consider the spatial profile of a beam reflected from an imperfect optic $E'(x, y; \mathbf{q}') = A(x, y)E_{\text{in}}(x, y; \mathbf{q}')$, where E_{in} is the incident beam and $A(x, y)$ is complex function describing the perturbation it has undergone. For example, on reflection a beam will be clipped by the finite size of the mirror $\alpha(x, y)$ and reflected from a surface with height variations $z(x, y)$. Thus, both the amplitude and phase of the beam will be affected and $A(x, y) = \alpha(x, y)e^{i2kz(x,y)}$. An example of the measured surface height variations present on

LIGO test mass mirrors can be seen in figure 1 [29]. The mode content of the outgoing beam $E(x, y; \mathbf{q})$ is computed by projecting E' into the outgoing beam basis \mathbf{q} . For any incoming HOM $u_{n'm'}$ the amount of outgoing u_{nm} can be computed via an overlap integral, this complex value is known as a *coupling coefficient*:

$$k_{nm,n'm'}(q_x, q'_x, q_y, q'_y; A) = \iint_{-\infty}^{\infty} \mathcal{K}(\boldsymbol{\lambda}_x; x)A(x, y)\mathcal{K}(\boldsymbol{\lambda}_y; y)dx dy, \quad (6)$$

where the integral kernels $\mathcal{K}(\boldsymbol{\lambda}_x; x)$ and $\mathcal{K}(\boldsymbol{\lambda}_y; y)$ are given by:

$$\mathcal{K}(\boldsymbol{\lambda}_x; x) = u_n^*(x, q_x)u_{n'}(x, q'_x), \quad (7)$$

$$\mathcal{K}(\boldsymbol{\lambda}_y; y) = u_m^*(y, q_y)u_{m'}(y, q'_y), \quad (8)$$

and the parameter vectors are given by $\boldsymbol{\lambda}_x = (n, n', q_x, q'_x)$ and $\boldsymbol{\lambda}_y = (m, m', q_y, q'_y)$. There are two general cases when computing (6): $q \neq q'$ which we refer to as *mode-mismatched* and $q = q'$ as *mode-matched*.

Computing the scattering matrix \hat{k} requires evaluating the integral (6) for each of its elements. This matrix is of size $N_{\text{HOM}} \times N_{\text{HOM}}$. If the couplings between the modes up to and including order \mathcal{O} are considered then the number of elements in the scattering matrix \hat{k} is:

$$N_k(\mathcal{O}) = N_{\text{HOM}}(\mathcal{O})^2, \\ = \frac{\mathcal{O}^4 + 6\mathcal{O}^3 + 13\mathcal{O}^2 + 12\mathcal{O} + 4}{4}, \quad (9)$$

and the computational cost of evaluating this many integrals can be very expensive. In our experience [21, 30] a typical LIGO simulation task involving HOMs can be performed with $\mathcal{O} = 6$ to 10. While some cases, such as those that include strong thermal distortions or clipping, require a higher maximum order to converge on a result

In simple cases where $A(x, y) = 1$ or $A(x, y)$ represents a tilted surface, analytical results are available for both mode matched and mismatched cases [31, 32]. In general however $A(x, y)$ is of no particular form and the integral in (6) must be evaluated numerically. It is possible to split multiple distortions into separate scattering matrices $A(x, y) \Rightarrow A(x, y)B(x, y)$ and the coupling coefficients become a product of two separate matrices:

$$k_{nm,nm'}(\mathbf{q}, \mathbf{q}'; A B) = \sum_{\tilde{n}, \tilde{m}=0}^{\infty} k_{nm,\tilde{n}\tilde{m}}(\mathbf{q}, \tilde{\mathbf{q}}; A)k_{\tilde{n}\tilde{m},nm'}(\tilde{\mathbf{q}}, \mathbf{q}'; B), \quad (10)$$

where $\tilde{\mathbf{q}}$ is an expansion beam parameter which we are free to choose. Thus our scattering matrix becomes $\hat{k}(\mathbf{q}, \mathbf{q}') = \hat{k}_A(\mathbf{q}, \tilde{\mathbf{q}})\hat{k}_B(\tilde{\mathbf{q}}, \mathbf{q}')$. By choosing $\tilde{\mathbf{q}} = \mathbf{q}$ or \mathbf{q}' we can set the mode-mismatching to be in either one matrix or the other. This is ideal, as a mode-matched \hat{k} is a Hermitian matrix whose symmetry can be exploited to only compute one half of the matrix. By ensuring that this matrix also contains any distortions that require numerical integration the computational cost can be nearly halved. It is then possible

to benefit from the fast analytic solutions of (6) [31, 32] to account for mode-mismatching in the other matrix.

3. Efficiently computing scattering matrices: integration by interpolation

For a discretely sampled mirror map with L sample points in both the x and y directions, the coupling coefficient (6) can be approximated using a composite Newton–Cotes quadrature rule

$$k_{nm,nm'}(q_x, q'_x, q_y, q'_y; A) \approx \sum_{k=1}^L \sum_{l=1}^L W_{kl}\mathcal{K}(\boldsymbol{\lambda}_x; x_k)A(x_k, y_l)\mathcal{K}(\boldsymbol{\lambda}_y; y_l), \quad (11)$$

where W is an $L \times L$ matrix describing the 2D composite Newton–Cotes quadrature weights over the area of the map. This weight matrix is found by taking the outer product of the 1D composite Newton–Cotes quadrature weights [33] in both x and y directions.

When L^2 is large, as in the cases of interest for this paper, there are two major bottlenecks: (i) evaluation of the kernel at each discrete x_k, y_k and, (ii) evaluation of the double sum. With a set of $M \ll L$ basis elements that accurately spans the kernel space, it is possible to replace the double sum (11) with a ROQ rule (20) containing only M^2 terms, reducing the overall cost of the by a factor of $\sim L^2/M^2$, provided the kernel can be directly evaluated.

The ROQ scheme is implemented in three steps. The first two are carried out offline, while the final, mirror-map-dependent step is performed in preparation for the simulations; once per map. The steps are as follows: *Step 1*—construct a reduced basis (offline); a set of M basis elements whose span describes the kernel space. *Step 2*—Construct an interpolant using the basis (offline) by requiring it to exactly match any kernel at M carefully chosen spatial subsamples $\{X_k\}_{k=1}^M$ [34] (and similarly for y). *Step 3*—Use the interpolant to replace the inner product evaluations in (11) with the ROQ (20) (online).

3.1. The EI method

The EI method is an efficient technique performing this off-line/online procedure and has been demonstrated in the context of astronomical data analysis with LIGO [25]. Provided the kernels vary smoothly with $\boldsymbol{\lambda}_x$ over x and $\boldsymbol{\lambda}_y$ over y then there exists a set of kernels at judiciously chosen parameter values that represent any kernel—and hence any integral (6)—for an arbitrary parameter value. This set of kernels constitutes the reduced basis: given any parameter value $\boldsymbol{\lambda}_x$ or $\boldsymbol{\lambda}_y$ we can find the best approximation to the kernel at $\boldsymbol{\lambda}_x$ or $\boldsymbol{\lambda}_y$ as linear combination of the reduced basis.

The ability to exploit the reduced basis to quickly evaluate (6) depends on being able to find an affine parameterization of the integral kernels. In general, the kernels do not admit such a parameterization. However, the EI method finds a near-optimal affine approximation whose accuracy is

bounded by the accuracy of the reduced basis [24]. This affine approximation is called the *empirical interpolant*. The spatial integrals over $dx dy$ in (6) will only depend on the reduced basis (and hence only have to be computed once for a given mirror map) and the parameter variation is handled by the empirical interpolant at a reduced computational cost.

The EI method exploits the offline/online computational concept where we decompose the problem into a (possibly very) expensive offline part which affords a cheap online part. In this case, the expensive offline part is in finding the reduced basis and constructing the empirical interpolant. Once the empirical interpolant is found then we use it for the fast online evaluation of (6). One of the main reasons why the empirical interpolant is used for fast online evaluation of (6) is due to its desirable error properties that makes it superior to other interpolation methods, such as polynomial interpolation. In addition, the empirical interpolant avoids many of the pitfalls of high-dimensional interpolation that we would otherwise encounter (see, e.g. [35]).

3.2. Affine parameterization

We would like the kernel to be separable in the mode parameters (λ_x, λ_y) and spatial position (x,y) . For these reasons we will look for a representation of the kernel that has the following form:

$$\begin{aligned} \mathcal{K}(\lambda_x; x) &= a(\lambda_x)f(x), \\ \mathcal{K}(\lambda_y; y) &= a(\lambda_y)f(y). \end{aligned} \quad (12)$$

The functions a and f are the same irrespective of whether the kernel is a function of x or y due to the symmetries of the Hermite–Gauss modes. Using the affine parameterization, the coupling coefficient (6) is:

$$\begin{aligned} k_{nm, nm'}(q_x, q'_x, q_y, q'_y) \\ = a^*(\lambda_x)a(\lambda_y) \iint_{-\infty}^{\infty} f^*(x)A(x, y)f(y)dx dy. \end{aligned} \quad (13)$$

This affine parameterization thus allows us to compute all the parameter-dependent pieces efficiently in the online procedure as all the x – y integrals are performed only once for a given mirror map. In general the kernel will not admit an exact affine decomposition as in (12). Using the EI method, the approximation to the kernels will have the form:

$$\begin{aligned} \mathcal{K}(\lambda_x; x) &\approx \sum_i c_i(\lambda_x)e_i(x), \\ \mathcal{K}(\lambda_y; y) &\approx \sum_i c_i(\lambda_y)e_i(y). \end{aligned} \quad (14)$$

The sum is over the reduced basis elements e_i and coefficients c_i that contain the parameter dependence.

Given a basis $e_i(x)$, the $c_i(\lambda_x)$ in (14) are the solutions to the M -point interpolation problem whereby we require the interpolant to be exactly equal to the kernel at any parameter value λ_x at a set of interpolation nodes $\{X\}_{i=1}^M$:

$$\mathcal{K}(\lambda_x; X_j) = \sum_{i=1}^M c_i(\lambda_x)e_i(X_j) = \sum_{i=1}^M V_{ji} c_i(\lambda_x), \quad (15)$$

where the matrix V is given by

$$V \equiv \begin{pmatrix} e^1(X_1) & e^2(X_1) & \cdots & e^M(X_1) \\ e^1(X_2) & e^2(X_2) & \cdots & e^M(X_2) \\ e^1(X_3) & e^2(X_3) & \cdots & e^M(X_3) \\ \vdots & \vdots & \ddots & \vdots \\ e^1(X_M) & e^2(X_M) & \cdots & e^M(X_M) \end{pmatrix}. \quad (16)$$

Thus we have:

$$c_i(\lambda_x) = \sum_{j=1}^M (V^{-1})_{ij} \mathcal{K}(\lambda_x; X_j). \quad (17)$$

Substituting (17) into (14), the empirical interpolant is:

$$\mathcal{I}_M[\mathcal{K}](\lambda_x; x) = \sum_{j=1}^M \mathcal{K}(\lambda_x; X_j)B_j(x), \quad (18)$$

where:

$$B_j(x) \equiv \sum_{i=1}^M e_i(x)(V^{-1})_{ij} \quad (19)$$

and is independent of λ_x . The special spatial points $\{X_k\}_{k=1}^M$, selected from a discrete set of points along x , as well as the basis can be found using algorithm 1 which is described in the next section.

We note that the kernels $\mathcal{K}(\lambda_x; x)$ appear explicitly on the right-hand side of (18). Because of this, we have to be able to directly evaluate the kernel at the EI nodes $\{X_k\}_{k=1}^M$. Fortunately this is possible in this case as we have closed form expressions for the kernels. If the kernels were solutions to ordinary or partial differential equations that needed to be evaluated numerically then using the empirical interpolant becomes more challenging, however this is not required here (see, e.g., [34, 36–39] for applications of the EI method to ordinary and partial differential equation solvers).

3.3. The EI method algorithm (offline)

The EI method algorithm solves (18) for arbitrary λ_x . While it would be possible in principle to use arbitrary basis functions, such as Lagrange polynomials which are common in interpolation problems [40, 41], we take a different approach that uses only the information contained in the kernels themselves. We will take as our basis a set of M judiciously chosen kernels sampled at points on the parameter space $\{\lambda_x^i\}_{i=1}^M$, where M is equal to the number of basis elements in (18). Because the kernels vary smoothly with λ_x a linear combination of the basis elements will give a good approximation to $\mathcal{K}(\lambda_x; x)$ for any parameter value [34]. We can then build an interpolant using this basis by matching $\mathcal{K}(\lambda_x; x)$ to the span of the basis at a set of M interpolation nodes $\{X_k\}_{k=1}^M$. The EI method algorithm, shown in algorithm 1, provides both the basis and the nodes.

The EI method algorithm uses a greedy procedure to select the reduced basis elements and interpolation nodes. With the greedy algorithm, the basis and interpolant are constructed iterative whereby the interpolant on each iteration

is optimized according to an appropriate error measure. This guarantees that the error of the interpolant is on average decreasing and—as we show in appendix B—that the interpolation error decreases exponentially quickly. We follow algorithm 3.1 of [42], which is outlined in appendix A.

3.4. Reduced order quadrature (online)

Substituting the empirical interpolant (18) into (11) gives the ROQ,

$$\begin{aligned} k_{nm,n'm'}(q_x, q'_x, q_y, q'_y; A) \\ = \sum_{k=1}^M \sum_{l=1}^M \omega_{kl} \mathcal{K}(\lambda_x; X_k) \mathcal{K}(\lambda_y; Y_l), \end{aligned} \quad (20)$$

with the ROQ weights ω_{kl} given by:

$$\omega_{kl} = \sum_{i=1}^L \sum_{j=1}^L W_{ij} A(x_i, y_j) B_k(x_i) B_l(y_j). \quad (21)$$

The ROQ form of the coupling coefficient enables fast online evaluations of the coupling coefficients. Note that because only M^2 operations are required to perform the double sum (20) we expect that the ROQ is faster than the traditional L^2 -term Newton–Cotes integration by a factor of L^2/M^2 provided that $M < L$. We expect in practice that $M \ll L$ due to the exponential convergence of the EI method.

The number of operations in (20) can be compressed further still due to the separability of the empirical interpolant (18) into beam parameters λ_x and spatial position x that allows us to exploit the spatial symmetry in the HG modes. The HG modes exhibit spatial symmetry/antisymmetry under reflection about the origin. Hence it is useful to split the x and y dimensions into four equally sized quadrants and perform the ROQ in each quadrant separately. For example, when a HG mode is symmetric between two or four of the quadrants then only two or one set(s) of coefficients $\{\mathcal{K}(\lambda_x; X_k)\}_{k=1}^M$ needs to be computed (and likewise for $\{\mathcal{K}(\lambda_y; Y_l)\}_{l=1}^M$). This will speed up the computation of the ROQ (20) by up to a factor of four. Hence, in practice we need only build the empirical interpolant over one half plane for either positive or negative values of x (or equivalently y); we derive the basis spanning the second half-plane by reflecting the basis about the origin. To ensure that this symmetry is exploitable the data points of the map must be distributed equally and symmetrically about the beam axis ($(x, y) = (0, 0)$). Those points that lie on the x and y axes must also be weighted to take into account they contribute to multiple quadrants when the final sum is computed. In the cases where the map data points are not correctly aligned we found that bilinear interpolation of the data to retrieve symmetric points did not introduce any significant errors. However, higher-order interpolation methods can introduce artefacts to the map data.

4. Exemplary case: near-unstable cavities and control signals

There are several scenarios when modeling tools can benefit heavily from the ROQ method, of particular interest are cases where the simulation time is dominated by the integration time of the mirror surface maps. One such example is an investigation into the feasibility of upgrading the LIGO interferometers with near-unstable arm cavities. The stability of a Fabry–Perot cavity is determined by its length L and radius of curvature (RoC) of each of its mirrors and can be described using the parameter::

$$g = (1 - L/R_{c_{\text{tm}}})(1 - L/R_{c_{\text{em}}}). \quad (22)$$

with $0 \leq g \leq 1$ defining the stable region.

Near-unstable cavities are of interest because they result in larger beam sizes on the cavity mirrors (see also figure 3) which reduces the coating thermal noise [12], one of the limiting noise sources of the detector. One negative aspect of such near-unstable cavities is that the transverse optical mode separation frequency approaches zero as $g \rightarrow 0$ or 1. The mode separation frequency determines the difference in resonance frequency of higher-order modes with respect to the fundamental mode. Thus with a lower separation frequency any defect in the cavity causing scattering into HOMs is suppressed less and can contaminate control signals for that cavity and couple extra noise into the main interferometer output signal. Another potential problem is additional clipping or scattering of the beam on the mirrors due to the larger beam sizes which can result in increased roundtrip losses of the arm cavity. The optimal cavity design must be determined as a trade-off between these degrading effects and the reduction in coating thermal noise. This is a typical task where a numerical model can be employed to search the parameter space. In this case each point in that parameter space corresponds to a different beam size in the cavity which forces a re-computation of the scattering matrices on the mirrors. Thus the new algorithm described in this paper should yield a significant reduction in computing time.

In this section we briefly summarize the results from the simulations and in the following section we provide the details of setting up the model and give an analysis of the performance of the ROQ algorithm. We have implemented the ROQ integration in our open-source simulation tool FINESSE and use the official input parameter files for the LIGO detectors [23]. Below we show the preliminary investigation of the behavior of a single Advanced LIGO like arm cavity with a finesse of 450 where the mirror maps for the mirrors ETM08 and ITM04 were applied to the high reflective (HR) surfaces. The nominal RoCs of ETM08 and ITM04 are 1934 m and 2245 m respectively [29]. Note that we do not report the scientific results of the simulation task which will be published elsewhere. This example is representative for a class of modeling performed regularly for the LIGO commissioning and design and provides us with a concrete and quantitative setup to demonstrate the required steps to use the ROQ algorithm.

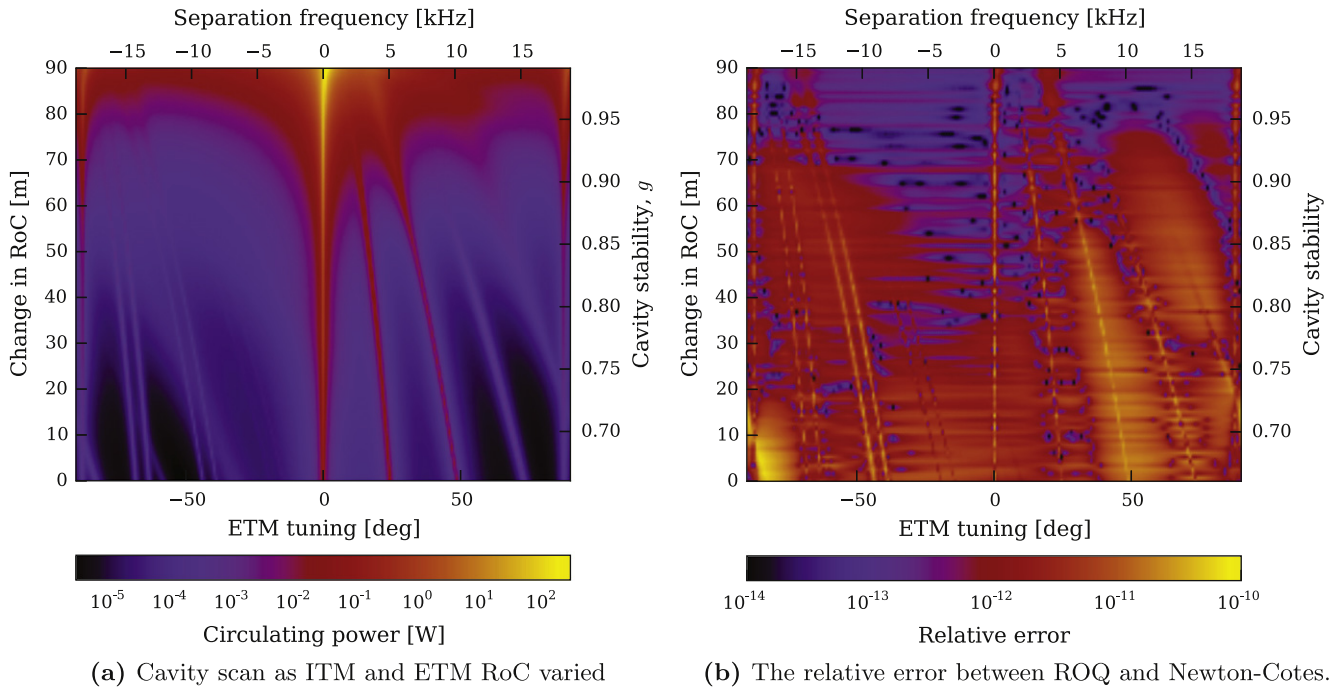


Figure 2. A modeled LIGO cavity power scan as its ITM and ETM mirrors radius of curvature of the mirrors of the ITM and ETM are varied to make the cavity increasingly more unstable. The simulation was ran using $\mathcal{O}_{\max} = 10$ and includes clipping from the finite size of the mirrors and surface imperfections from the ETM08 and ITM04 maps. Figure 2(a) shows how the amount of power scattering into HOM changes as $g \rightarrow 1$. Also visible here is the reduction in the mode separation frequency with increasing instability. The contribution of the TEM_{00} mode has been removed to make the HOM content visible. The reduced basis was built for mode order $\mathcal{O} = 14$, to reduce errors, see figure 7. The difference in this result when using ROQ compared to Newton–Cotes is shown in 2(b).

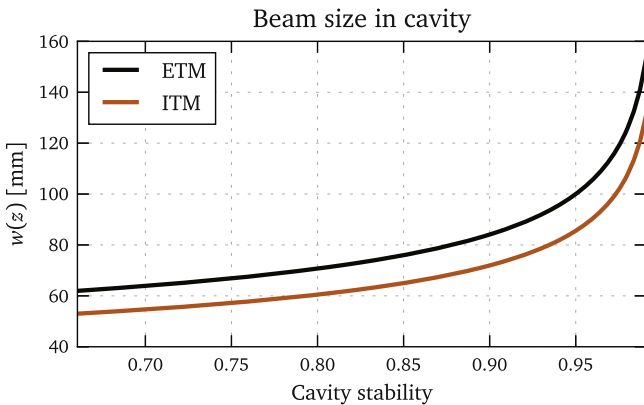


Figure 3. The beam size on the ITM and ETM of a LIGO cavity as a function of cavity stability parameter as the mirror RoCs are tuned.

Modeling the LIGO cavity for differing stabilities involves varying the RoC of both the ITM and ETM. The resulting change in $w(z)$ at each surface means the scattering matrices will need to be recomputed for each state we choose. To view the HOM content in the cavity created by the scattering a *cavity scan* can be performed, displacing one of the cavity mirrors along the cavity axis on the order of the wavelength of the laser light, $\lambda = 1064 \text{ nm}$, to change the resonance condition of the cavity. We have performed the simulations using $\mathcal{O} = 10$ with Newton–Cotes integration and our ROQ method. The results for cavity scans at different RoCs are shown in figure 2(a). The dominant mode is the fundamental TEM_{00} whose resonance defines the zero tuning,

the power in the TEM_{00} mode has been removed from this plot to better show the lower power HOM content. For more stable cavities (at the bottom of the plot in figure 2(a)) the HOMs are well separated and not resonant at the same time as the TEM_{00} . As the RoC is increased, the stability is reduced and the HOMs can be seen to converge and eventually become resonant at a tuning of 0. At a stability of $g \approx 0.98$ the cavity mode begins to break down significantly and many modes become resonant. The effect of this on a sensing and control signal used for a Pound–Drever–Hall control system is shown in figure 4, where for increasingly unstable cavities the error signal becomes degraded, showing an offset to the nominal zero crossing, a reduced slope and overall asymmetry around the center. The complete investigation into the feasibility of such cavities is beyond the scope of this paper, it includes amongst other issues the quantitative comparison of the control noise from the degradation of the control signals with the reduced thermal noise. The simulation task described above is sufficient to provide a test case for our ROQ method.

5. Application and performance of simulations using ROQ

In this section we provide a detailed and complete recipe for setting up and using the ROQ for the LIGO example, using FINESSE and PYKAT, and discuss the performance, in terms of speed and accuracy, of our method. The description should be sufficient for the reader to implement our method for their

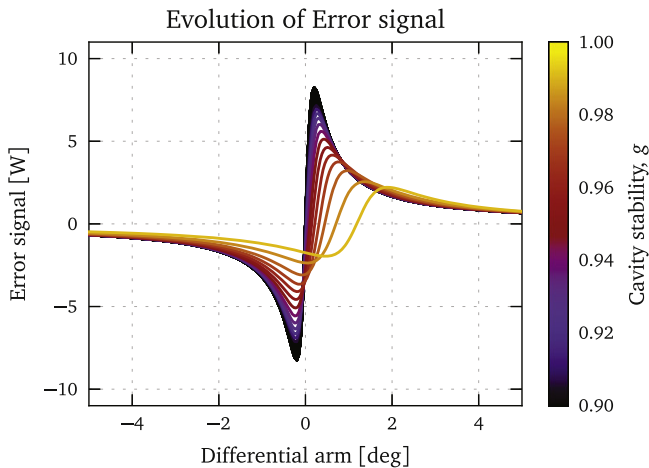


Figure 4. The Pound–Drever–Hall error signal for the LIGO cavity modelled in figure 2. A significant change in zero-crossing position and shape can be seen as the stability of the cavity is reduced ($g \rightarrow 1$).

own optical setup. In this section we include the costly ‘off-line’ procedure of building the empirical interpolant for completeness. As part of the ongoing code development of FINESSE and PYKAT we intend to pre-generate the empirical interpolants, suitable for a wide range of problems, so that the typical user should not need to run the costly offline building of the interpolant into their simulations.

5.1. Computing the ITM and ETM empirical interpolants

Firstly the range of beam parameters for the simulation must be determined. Once this is known a training set can be constructed and the empirical interpolant can be computed. The surface distortions that are of interest are those on the HR surfaces of a LIGO arm cavity mirror. We will require two empirical interpolants, one for the ITM HR surface and one for the ETM HR surface due to the differing beam parameters at each mirror. The beam parameter range that the training sets should span are determined by varying the RoC of the ITM and ETM to include the range of cavity stabilities which we want to model. The beam parameter ranges are shown in figure 5. The required ranges are $4.7 \text{ m} < w_0 < 12.0 \text{ mm}$ for ITM and ETM, and $2.11 \text{ km} < z < 2.20 \text{ km}$ for the ITM and $-1.88 \text{ km} < z < -1.79 \text{ km}$ for the ETM, up to a maximum optical mode order of $\mathcal{O} = 20$, Newton–Cotes degree of 6, $L = 1199$. For this example we fix the maximum tolerable error of the empirical interpolant to $\epsilon = 10^{-14}$.

Using these ranges the method described in section 3.1 can be used to produce the empirical interpolants. The offline computation of the basis can have significant computational cost. For very wide parameter ranges the memory required to store the training sets can quickly exceed that of typical machines. For the above parameters, with 100 sample points each in the w_0 and z range, up to $\mathcal{O} = 14$ and $\epsilon = 10^{-14}$ approximately 7 GB of memory was required. Running this method on machines with less memory is possible by storing the training set on a hard drive using a suitable data storage format such as HDF5 for access. Computation time of the

empirical interpolant is then limited by the read and write times of the media. Using a MacBook pro 2012 model which contains a 2.7 GHz Intel core i7 with 8 GB of RAM generating the ITM and ETM reduced basis and empirical interpolant takes ≈ 4 h each. The number of elements in the final reduced basis for the ITM and ETM were $N = 30$ and $N = 29$ respectively. In figure 8 the convergence of the empirical interpolant error with respect to the acceptable empirical interpolant error. One can see that the empirical interpolant error converges exponentially as described in appendix B.

5.2. Producing the ROQ weights

Once the empirical interpolant has been computed for both ITM and ETM HR surfaces the ROQ weights (21) can be computed by convoluting the mirror maps with the interpolant. The surface maps that we have chosen are the measured surface distortions of the (uncoated) test masses currently installed at the LIGO Livingston observatory, shown in figure 1. The maps contain $L \approx 1200$ samples and we can expect a theoretical speed-up of $L^2/N^2 \approx 1200^2/30^2 = 1600$ from using ROQ over Newton–Cotes. These maps include an aperture, \mathcal{A} , and the variation in surface height in meters, $z(x, y)$. Thus to calculate the HOM scattering on reflection from one of these mirrors with (6) the distortion term is:

$$A(x, y) = \mathcal{A}(x, y)e^{2ikz(x, y)}, \quad (23)$$

where $\mathcal{A}(x, y)$ is 1 if $\sqrt{x^2 + y^2} < 0.16 \text{ m}$ and 0 otherwise, and k is the wavenumber of the incident optical field.

Using (23) with equation (21) (with a Newton–Cotes rule of the same degree the empirical interpolant was generated with) the ROQ weights can be computed for each map shown in figure 1. This computational cost is proportional to the number of elements in the empirical interpolant, M , and the number of samples in the map, L^2 . For the LIGO maps this takes ≈ 10 s on our 2012 MacBook Pro. The resulting ROQ rule for the maps can be visualized as shown in figure 6: the amplitude of the ROQ weights map out the aperture and the phase of the weights varies for different maps because of the different surface structure. We note that there are non-zero ROQ weights associated with points in the region where the mirror maps are zero outside the aperture of the mirror. While this may be counter intuitive, it is a consequence of the fact that the empirical interpolant nodes lie within the full x – y plane and, that they are constructed without any knowledge of the mirror maps: the weights still receive no contribution from the region where $A(x, y) = 0$ as this region does not contribute to the sum in (21). However, the ROQ uses information about the kernels (7) over the entire region, including where $A(x, y) = 0$. The computation of these ROQ weights need only be performed once for each map, unless the range of beam parameters required for the empirical interpolant are changed.

We verify that the process of generating the ROQ rule has worked correctly by computing the scattering matrices with ROQ and Newton–Cotes across the parameter space. We

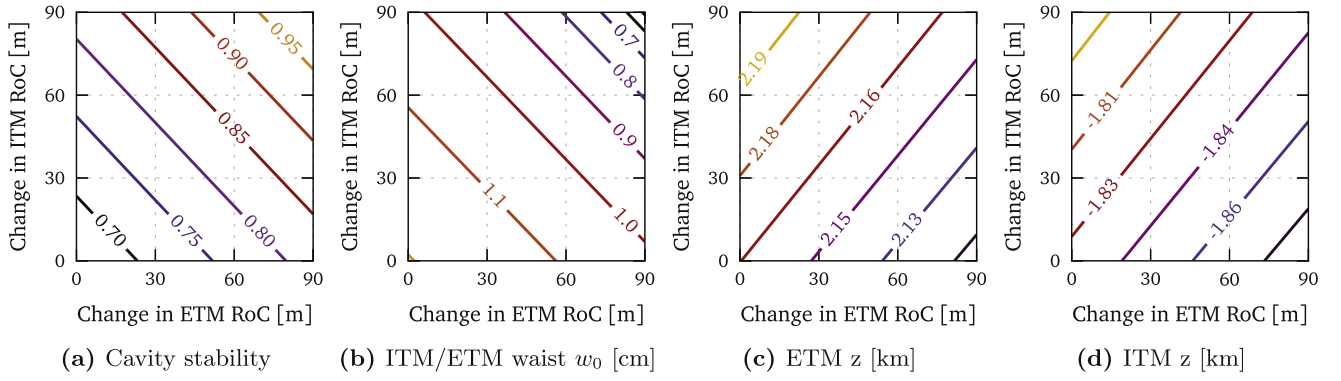


Figure 5. Range of beam parameters needed to model a change in curvature from 0 m to 90 m at the ITM and the ETM. In order to utilize the ROQ to cover this parameter space, the empirical interpolant needs to be constructed using a training set made from kernels (7) densely covering this space.

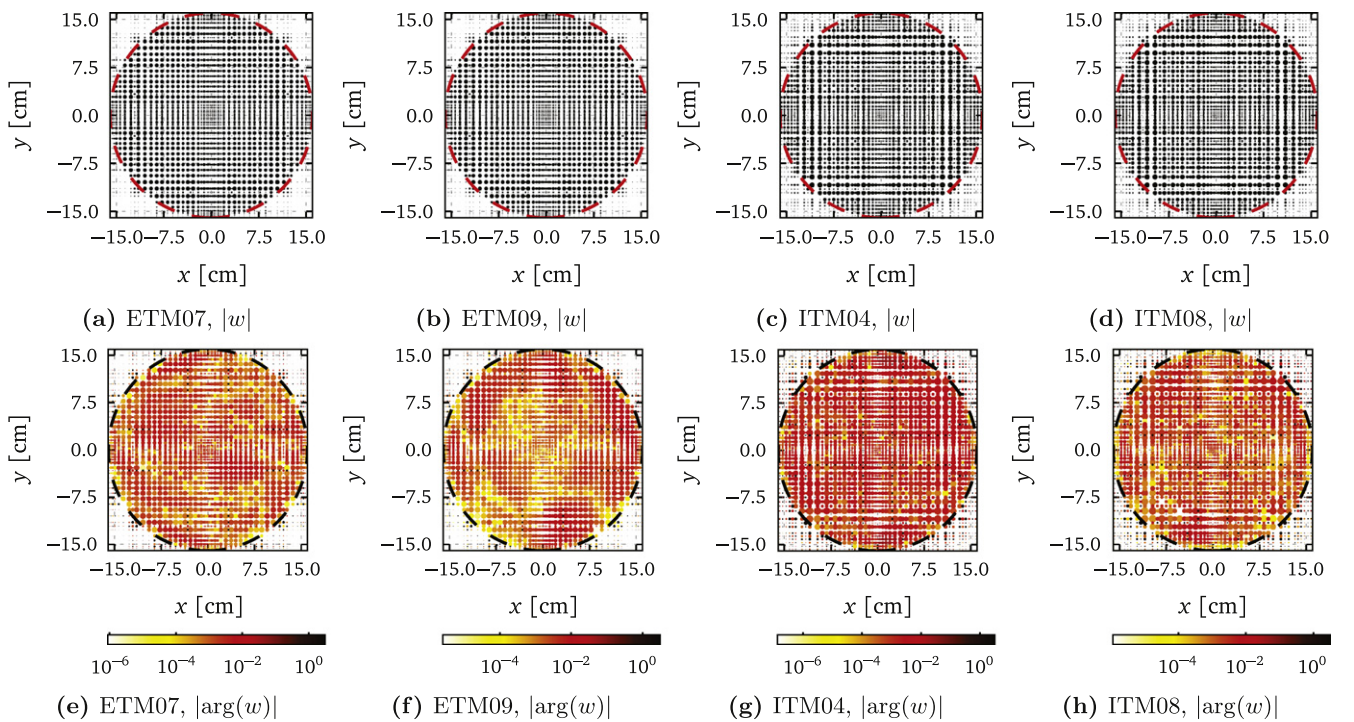


Figure 6. Visualization of the final quadrature rule, with the absolute and argument values of the ROQ weights (21) generated for each of the maps as shown in figure 1. The top plots show the absolute value: the size of each point is proportional to $|w|$ and the center of each point lies on a specific empirical interpolation node in the x - y plane (X_i, Y_j) (see (15) and (20)). The bottom plots show $\log_{10}(\arg(w))$. The dashed line on each plot shows the mirror surface boundary; outside the boundary the mirror map data is equal to zero.

compute $\hat{k}(q; \text{ETM07})$ with $\mathcal{O}_{\max} = 10$ using ROQ and then again using Newton–Cotes integration. Computing the relative error between each element of these two matrices the maximum error can be taken for q values spanning the requested q parameter range. Figure 9 shows how the final error of the empirical interpolant, σ_M , propagates into an error in the scattering matrix. This shows the maximum (solid line) and minimum (dashed line) errors for any element in the scattering matrix between the two methods. From this it can be seen that building a more accurate empirical interpolant results in smaller maximum errors in the scattering matrix. Now, using the most accurate reduced basis the maximum

relative error is shown in figure 7 over the q space, where the white dashed box shows the boundaries of the parameters in the training set. Overall the method successfully computed a ROQ rule that accurately reproduced the Newton–Cotes results for scattering up to $\mathcal{O} = 10$. It should be noted that the largest errors, e.g. as seen in figure 9, do not represent the full parameter space but occur only at smallest z and largest w_0 . It was also found that building a basis including a higher maximum HOM, for example basis of order 14 for scattering computations up to order 10, significantly improved the accuracy of the ROQ. Using an reduced basis constructed for order 14 rather than order 10 only increased the number of

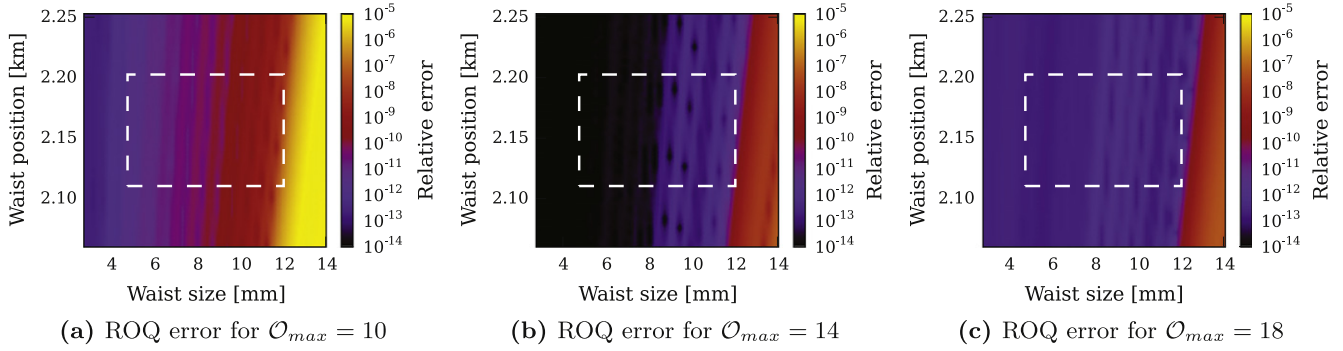


Figure 7. Maximum relative error between the scattering matrices computed for the ETM07 surface map, with ROQ and using Newton–Cotes, for mode orders up to $\mathcal{O}_{\max} = 18$. The dashed white area represents the beam-parameter region over which training sets were generated. The subplots illustrate how using an ROQ built for a larger \mathcal{O}_{\max} scattering reduces the maximum error significantly.

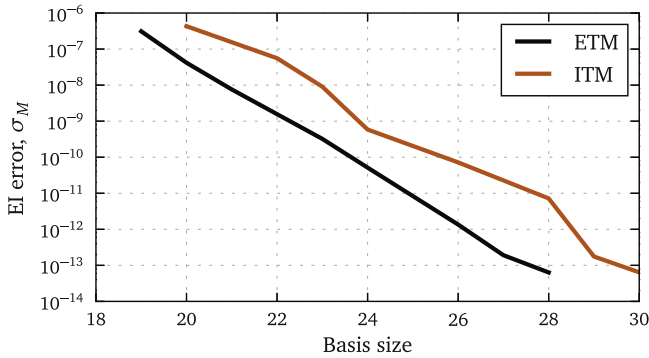


Figure 8. Empirical interpolant error as a function of the number of basis elements selected by the greedy algorithm (algorithm 1) for the example described in section 5.1. As expected from the error analysis in appendix B, the empirical interpolant error displays exponential convergence with the basis size.

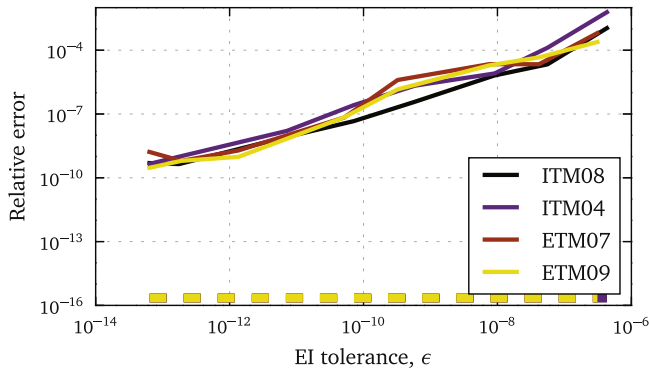


Figure 9. Relative error in the scattering matrices computed using the ROQ and Newton–Cotes integration (with $\mathcal{O}_{\max} = 10$) as a function of the empirical interpolant tolerance ϵ . The empirical interpolant was built for maximum coupling $\mathcal{O}_{\max} = 14$. The error is the minimum (dashed lines) and maximum (solid lines) over the parameter space with which the empirical interpolant was built for.

elements in the basis by 2, thus not significantly degrading any speed improvements. It can also be seen in figure 7 that ROQ extrapolates beyond the originally requested q parameter space and does not instantly fail for evaluations outside of it. A gradual decrease in the accuracy can be seen when using larger w_0 values.

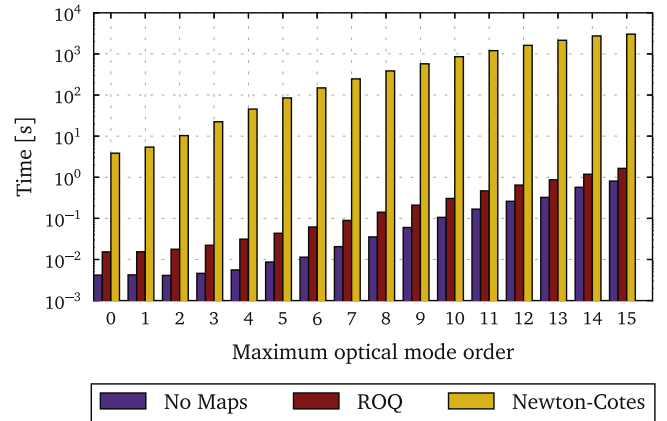


Figure 10. Time taken for a single run of FINESSE to model the steady state optical fields in an LIGO cavity with surface maps on both the ITM and ETM HR surfaces.

5.3. Performance

The time taken to run these FINESSE simulations as \mathcal{O} is increased is shown in figure 10 demonstrating how much more efficient it is to use ROQ over Newton–Cotes for the computation of scattering matrices. Here, the timing of running the entirety of FINESSE is used—rather than just the ROQ method—because there are additional timing improvements from having to read and handle significantly less data with ROQ: from the $L \times L$ maps down to $M \times M$ ROQ weights. This smaller amount of data comes with the typical benefits of fitting better into processor caches, smaller memory footprint and reduced disk read times. The case with no maps being used is also shown to illustrate how much time is spent in FINESSE doing calculations not involving maps—which becomes the dominant computational cost when using ROQ. A significant improvement is also found for order zero scattering where only one scattering integral needs to be calculated. This is partly time saved from having to read larger data from the disk and manipulating it in memory. The one-time offline cost for each map for a particular empirical interpolant, thus is not included in the timing shown in the figure 10 (5 s for each map in this example). The overall speed-up achieved can be seen in figure 11, reaching ≈ 2700 times faster to run the entire simulation at $\mathcal{O} = 10$. The overall speed-up then

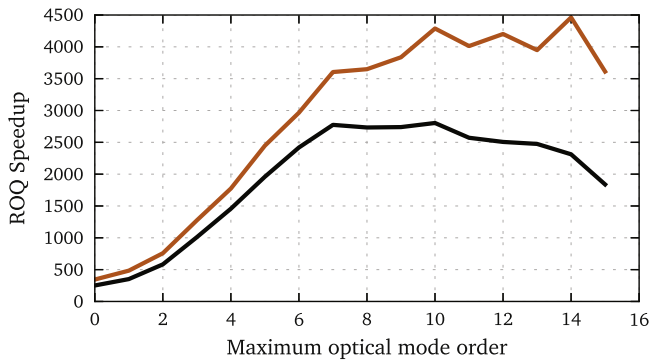


Figure 11. The speed-up achieved using ROQ compared to Newton–Cotes as a function of mode order using the timing values in figure 10. The red line shows the speed-up if the time for initialization and post-processing is subtracted from both times for Newton–Cotes and ROQ. This demonstrates the improvements just for the computational cost relating to map scattering calculations.

begins to drop slightly as the base time taken to run the rest of FINESSE becomes larger. The color trace in figure 11 shows the speed-up if this base time is removed, again showing an impressive speed-up peaking at 4000 times faster.

Using ROQ enables us to perform such modeling tasks with a far greater efficiency. Running the model to compute the output seen in figure 2(a) required computing 100 different scattering matrices for the various changes in RoC. This took 20.5 h to compute with Newton–Cotes and 18 min with ROQ. Note that the effective speed-up in this case is less than the large values in figure 11 because here we have included the total runtime of the simulation. This includes the initialization and running of the other aspects of FINESSE which took ≈ 17 min. The actual time taken for just the ROQ calculation is ≈ 30 s thus a speed-up in the ROQ vs Newton–Cotes is ≈ 2500 . The difference in the final result between ROQ and Newton–Cotes is shown in terms of relative error in figure 2(b). We have prepared the ROQ input for this example such that the error is significantly lower than 1 ppm (relative error of 10^{-6}) thereby showing that ROQ can be both much faster and still sufficiently accurate.

6. Conclusion

Numerical modeling of optical systems plays a vital role for the design and commissioning of precision interferometers. The typical use of the simulation software in this area requires rapid iterations of many simulation runs and manual fine tuning as modeling progresses, which is not well suited for large computer clusters. The scope of current investigations is often limited by the required computation time and thus the development of fast and flexible tools is a priority. Current problems in precision interferometers, such as LIGO, involve the investigation of laser beam shape distortions and their effect on the interferometer signal. Frequency-domain simulations using Gaussian modes to describe the beam properties have emerged as fast and flexible tools. However, the computation of the scattering matrix for mirror surface distortions—effectively an overlap integral of measured surface data with Hermite–Gauss

modes—has shown to be a limiting factor in improving the computational speed of such tools. A significant reduction in computational time of current numerical tools is required for more efficient in-depth modeling of interferometers including more realistic features such as clipping, optical defects, thermal distortions and parametric instabilities. In this work we have demonstrated how the EI method can be used to generate an optimised quadrature rule for paraxial optical scattering calculations, known as a ROQ. Our method removes the prohibitive computational cost of computing the scattering by speeding up the calculation of the steady state optical fields in a LIGO arm cavity by up to a factor of 2750, reducing simulation times from days to minutes. Using an exemplary simulation task of near-unstable arm cavities for the LIGO interferometers we have demonstrated that our method is both accurate and fast for a typical modeling scenario where imperfections in the interferometer have a significant impact on optical performance. We have provided a complete recipe to recreate and use the new algorithm and provide an open source implementation in our general-purpose simulation software FINESSE. Importantly, the ROD method is generic and can be applied to any optical scattering problem for any surface distortion data.

Acknowledgments

We would like to thank GariLynn Billingsley for providing the Advanced LIGO mirror surface maps and for advice and support on using them. We would also like to thank Kent Blackburn, Peter Deiner, Scott Field and Chad Galley for useful discussions and encouragement throughout this project. Some of the computations were carried out using the high performance computing resources provided by Louisiana State University (<http://www.hpc.lsu.edu>) and the Extreme Science and Engineering Discovery Environment (XSEDE) [43], which is supported by National Science Foundation grant number ACI-1053575. This work has been supported by the Science and Technology Facilities Council (STFC) Consolidated Grant ST/K000845/1 and the U.S. National Science Foundation under cooperative agreement NSF-PHY-0757058. This document has been assigned the LIGO Laboratory document number LIGO-P1500128.

Appendix A. EI method algorithm

The first input to the algorithm is a *training space* of kernels—distributed on the parameter space λ_x —and the associated set of parameters. This training space is denoted by $T = \{\lambda_x^k, \mathcal{K}(\lambda_x^k; x)\}_{i=1}^N$ and should be densely populated enough to represent the full space of kernels as faithfully as possible. Hence it is important that $1 \ll N$. The second input is the desired maximum error of the interpolant ϵ . We find that the L^∞ norm is a robust error measure for the empirical interpolant and hence ϵ corresponds to the largest tolerable difference between the empirical interpolant and any kernel in the training set T .

The algorithm is initialized on steps 3 and 4 by setting the zeroth order interpolant to be zero, and defining the zeroth order interpolation error to be infinite. The greedy algorithm proceeds as follows: we identify the basis element on iteration i to be the $\mathcal{K}(\lambda_x; x) \in \mathcal{T}$ that maximizes the L^∞ norm with the interpolant from the previous iteration, $\mathcal{I}_{i-1}[\mathcal{K}](\lambda_x; x)$. This is performed in steps 7 and 8. On step 9 we select X_i , the i th interpolation node, by selecting the position at which the largest error occurs, and adding that position to the set of interpolation nodes. By definition, the interpolant is equal to the underlying function at the interpolation nodes and so the error at X_i —which is the largest error on the current iteration—is removed. On step 10 we normalize the basis function. This ensures that the matrix (16) is well conditioned. On steps 11 and 12 we compute (16) and (19), which are used to construct the empirical interpolant (18). Finally, on step 13 we compute the interpolation error σ_i between the interpolant on the current iteration $\mathcal{I}_i[\mathcal{K}](\lambda_x; x)$ and $\mathcal{K}(\lambda_x; x) \in \mathcal{T}$ as in step 7. The procedure is repeated until $\sigma_i \leq \epsilon$.

Once the interpolant for $\mathcal{K}(\lambda_x; x)$ is found, the equivalent interpolant for $\mathcal{K}(\lambda_y; y)$ is obtained trivially from $\mathcal{I}_M[\mathcal{K}](\lambda_x; x)$ by setting $x \rightarrow y$.

Algorithm 1. EI method algorithm: the EI method algorithm builds an interpolant for the kernels (7) iteratively using a greedy procedure. On each iteration the current interpolant is validated against a ‘training set’ \mathcal{T} of kernels and the worst interpolation error is identified. The interpolant is then updated so that it describes the worst-error point perfectly. This is repeated until the worst error is less than or equal to a user specified tolerance ϵ .

```

1: Input:  $\mathcal{T} = \{\lambda_x^k, \mathcal{K}(\lambda_x^k; x)\}_{k=1}^N$  and  $\epsilon$ 
2: Set  $i = 0$ 
3: Set  $\mathcal{I}_0[\mathcal{K}](\lambda_x; x) = 0$ 
4: Set  $\sigma_0 = \infty$ 
5: While  $\sigma_i \geq \epsilon$ 
6:    $i \rightarrow i + 1$ 
7:    $\lambda_x^i = \arg \max_{\lambda_x \in \mathcal{T}} \|\mathcal{K}(\lambda_x; x) - \mathcal{I}_{i-1}[\mathcal{K}](\lambda_x; x)\|_{L^\infty}$ 
8:    $\xi_i(x) = \mathcal{K}(\lambda_x^i; x)$ 
9:    $X_i = \arg \max_x |\xi_i(x) - \mathcal{I}_{i-1}[\xi_i](x)|$ 
10:   $e_i(x) = \frac{\xi_i(x) - \mathcal{I}_{i-1}[\xi_i](x)}{\xi_i(X_i) - \mathcal{I}_{i-1}[\xi_i](X_i)}$ 
11:   $V_{lm} = e_l(X_m) l \leq i, m \leq i$ 
12:   $B_m(x) = \sum_l e_l(x) (V^{-1})_{lm} l \leq i, m \leq i$ 
13:   $\sigma_i = \max_{\lambda_x \in \mathcal{T}} \|\mathcal{K}(\lambda_x; x) - \mathcal{I}_i[\mathcal{K}](\lambda_x; x)\|_{L^\infty}$ 
14: end while
15: Output: Interpolation matrix  $\{B_j(x)\}_{j=1}^M$  and interpolation nodes  $\{X_j\}_{j=1}^M$ . The equivalent interpolant for  $\mathcal{K}(\lambda_y; y)$  is obtained trivially from  $\{B_j(x)\}_{j=1}^M$  and  $\{X_j\}_{j=1}^M$  by setting  $x \rightarrow y$  and  $X \rightarrow Y$ .
```

Appendix B. Error bounds on the empirical interpolant

Here, we briefly remark on some of the error properties of the EI method. A more detailed error analysis of the empirical

interpolant can be found in [24]. For our purposes the empirical interpolant possess a highly desirable property, namely exponential convergence to the desired accuracy ϵ . It can be shown [24] (though we do not do so here) that there exists constants $C > 0$, $c_1 > 0$ and $\alpha > 0$ such that for any function f the empirical interpolant satisfies

$$\|f - \mathcal{I}_M[f]\|_{L^\infty} \leq \sqrt{2C} \Lambda \exp[-c_1 M^\alpha]. \quad (\text{B1})$$

Where Λ is the ‘Lebesgue constant’: $\Lambda = \|\mathcal{I}_M[f]\|_{L^\infty}$.

This states that under the reasonable assumption that there exists an order M interpolant that allows for exponential convergence, then the EI method will ensure that we converge to this interpolant exponentially quickly. This is an important property as it means that the order of the interpolant, M , tends to be small for practical purposes. In addition, because the quantity on the right-hand side $\sqrt{2C} \Lambda \exp[-c_1 M^\alpha]$ is set to a user specified tolerance ϵ then we can set an *a priori* upper bound on the *worst-fit* of the interpolant. However, one must still verify that the interpolant describes functions outside the training *a posteriori*, though the error bound should still be satisfied provided that the training set was dense enough. In fact, it can be shown [27] that the EI method is a near optimal solution to the Kolmogorov n -width problem in which one seeks to find the best M -dimensional (linear) approximation to a space of functions.

It is important to recall that in this paper we are interpolating the integral kernels (7) which are a function of six free parameters λ_x : two indices n and n' and two complex beam parameters q_x and q'_x . Had we not used the EI method, we would have had to find an alternative way of expressing the λ_x -dependent coefficients in (14). Consider, for example, a case in which we had used tensor-product splines to describe the coefficients: Using a grid of just ten points in each of the six parameters in λ_x would result in an order 10^6 spline which would surely be computationally expensive to evaluate. Furthermore, there would be no guarantee of its accuracy or convergence to a desired accuracy.

References

- [1] Grote H and (The LIGO Scientific Collaboration) 2008 The status of GEO 600 *Class. Quantum Grav.* **25** 114043
- [2] Sigg D and (The LIGO Scientific Collaboration) 2008 Status of the LIGO detectors *Class. Quantum Grav.* **25** 114041
- [3] Acernese F et al 2008 Virgo status *Class. Quantum Grav.* **25** 184001
- [4] Takahashi R and (The TAMA Collaboration) 2004 Status of TAMA300 *Class. Quantum Grav.* **21** S403
- [5] Abbott B and (The LIGO Scientific Collaboration) 2009 Observation of a kilogram-scale oscillator near its quantum ground state *New J. Phys.* **11** 073032
- [6] Aasi J et al (Advanced LIGO) 2015 *Class. Quantum Grav.* **32** 074001
- [7] Acernese F et al 2015 Advanced virgo: a second-generation interferometric gravitational wave detector *Class. Quantum Grav.* **32** 024001
- [8] STAIC 2000 Software tools for advanced interferometer configurations (<http://phys.ufl.edu/LIGO/LIGO/STAIC.html> accessed: 18-11-2014)

- [9] GWIC 2013 Numerical modelling tools for gravitational wave detectors (<https://gwic.ligo.org/simulations> accessed: 18-11-2014)
- [10] Caron B *et al* 1999 SIESTA, a time domain, general purpose simulation program for the VIRGO experiment *Astropart. Phys.* **10** 369–86
- [11] Bhawal B, Cella G, Evans M, Klimenko S, Maros E, Mohanty S D, Rakhmanov M, Savage R L and Yamamoto H 2000 Ligo end-to-end simulation programme *AIP Conf. Proc.* **523** 469–70
- [12] Vinet J-Y 2009 On special optical modes and thermal issues in advanced gravitational wave interferometric detectors *Living Rev. Relativ.* **12** 5
- [13] Vinet J-Y, Hello P, Man C N and Brillat A 1992 A high accuracy method for the simulation of non-ideal optical cavities *J. Phys. I* **2** 1287–303
- [14] Beausoleil R G *et al* 2003 Model of thermal wave-front distortion in interferometric gravitational-wave detectors. i. thermal focusing *J. Opt. Soc. Am. B* **20** 1247–68
- [15] ET Science Team 2010 Einstein gravitational wave telescope conceptual design study *Technical Report* (EGO) ET-0106C-10 (<https://tds.ego-gw.it/itf/tds/>)
- [16] Schibli T R *et al* 2008 Optical frequency comb with submillihertz linewidth and more than 10 w average power *Nat. Photonics* **2** 355–9
- [17] Dickerson S M, Hogan J M, Sugarbaker A, Johnson D M S and Kasevich M A 2013 Multi-axis inertial sensing with long-time point source atom interferometry *Phys. Rev. Lett.* **111** 083001
- [18] Freise A and Strain K 2010 Interferometer techniques for gravitational-wave detection *Living Rev. Relativ.* **13** 1
- [19] Braginsky V B, Strigin S E and Vyatchanin S P 2001 Parametric oscillatory instability in Fabry–Perot interferometer *Phys. Lett. A* **287** 331–8
- [20] Evans M, Barsotti L and Fritschel P 2010 A general approach to optomechanical parametric instabilities *Phys. Lett. A* **374** 665–71
- [21] Bond C, Fulda P, Brown D and Freise A 2013 Investigation of beam clipping in the power recycling cavity of Advanced LIGO using FINESSE *Technical Report LIGO Document* (LIGO DCC) T1300954 (<https://dcc.ligo.org/LIGO-T1300954>)
- [22] Freise A *et al* 2004 Frequency-domain interferometer simulation with higher-order spatial modes *Class. Quantum Grav.* **21** S1067–74
- [23] FINESSE input files for Advanced LIGO (<https://dcc.ligo.org/LIGO-L1300231> accessed: 13-07-2015)
- [24] Antil H, Field S E, Herrmann F, Nochetto R H and Tiglio M 2013 Two-step greedy algorithm for reduced order quadratures *J. Sci. Comput.* **57** 604–37
- [25] Canizares P *et al* 2015 Accelerated gravitational wave parameter estimation with reduced order modeling *Phys. Rev. Lett.* **114** 071104
- [26] Freise A and Brown D, PYKAT webpage (<http://gwoptics.org/pykat/> accessed: 13-07-2015)
- [27] Maday Y, Nguyen N C, Patera A T and Pau S H 2009 A general multipurpose interpolation procedure: the magic points *Commun. Pure Appl. Anal.* **8** 383–404
- [28] Kogelnik H 1965 On the propagation of Gaussian beams of light through lenslike media including those with a loss or gain variation *Appl. Opt.* **4** 1562–9
- [29] Billingsley G, LIGO core optics reference page (<https://galaxy.ligo.caltech.edu/optics/> accessed: 13-07-2015)
- [30] Bond C Z 2014 How to stay in shape: overcoming beam and mirror distortions in advanced gravitational wave interferometers (<http://theses.bham.ac.uk/5223/>)
- [31] Bayer-Helms F 1984 Coupling coefficients of an incident wave and the modes of spherical optical resonator in the case of mismatching and misalignment *Appl. Opt.* **23** 1369–80
- [32] Vinet J Y and (The Virgo Collaboration) 2001 *The Virgo Book of Physics: Optics and Related Topics* (Virgo: EGO) (<http://cascina.virgo.infn.it/vpb/>)
- [33] Ralston A and Rabinowitz P 2001 *A First Course in Numerical Analysis (Dover Books on Mathematics)* (New York: Dover)
- [34] Barrault M, Maday Y, Nguyen N C and Patera A T 2004 An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations *C. R. Math.* **339** 667–672
- [35] Hesthaven J S, Stamm B and Zhang S 2014 Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods *ESAIM: Math. Modelling Numer. Anal.* **48** 259–83
- [36] Gunzburger M D, Peterson J S and Shadid J N 2007 Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data *Comput. Methods Appl. Mech. Eng.* **196** 1030–47
- [37] Kirk B S, Peterson J W, Stogner R H and Carey G F 2006 libMesh: a C++ Library for parallel adaptive mesh refinement/coarsening simulations *Eng. Comput.* **22** 237–54
- [38] Antil H, Heinkenschloss M and Sorensen D C 2014 Application of the discrete empirical interpolation method to reduced order modeling of nonlinear and parametric systems *Reduced Order Methods for Modeling and Computational Reduction* vol 9 (New York: Springer) pp 101–136
- [39] Chaturantabut S and Sorensen D C 2010 Nonlinear model reduction via discrete empirical interpolation *SIAM J. Sci. Comput.* **32** 2737–64
- [40] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 *Numerical Recipes 3 Edition The Art of Scientific Computing* (New York, USA: Cambridge University Press)
- [41] Atkinson A 1989 *An Introduction to Numerical Analysis* 2nd edn (New York: Wiley)
- [42] Aanonsen T Ø 2009 Empirical interpolation with application to reduced basis approximations *Masters Thesis* (Norwegian University of Science and Technology)
- [43] Towns J *et al* 2014 Xsede: accelerating scientific discovery *Comput. Sci. Eng.* **16** 62–74