

 Open access • Journal Article • DOI:10.1137/120869171

Accelerating Optimization of Parametric Linear Systems by Model Order Reduction

— [Source link](#) 

Yao Yue, Karl Meerbergen

Published on: 27 Jun 2013 - Siam Journal on Optimization (Society for Industrial and Applied Mathematics)

Topics: Surrogate model, Trust region, Optimization problem, Model order reduction and Penalty method

Related papers:

- [A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems](#)
- [Nonlinear Model Reduction via Discrete Empirical Interpolation](#)
- [Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations](#)
- [Progressive construction of a parametric reduced-order model for PDE-constrained optimization](#)
- [Interpolatory Projection Methods for Parameterized Model Reduction](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/accelerating-optimization-of-parametric-linear-systems-by-31hh5chkda>

ACCELERATING OPTIMIZATION OF PARAMETRIC LINEAR SYSTEMS BY MODEL ORDER REDUCTION*

YAO YUE[†] AND KARL MEERBERGEN[†]

Abstract. Design optimization problems are often formulated as an optimization problem whose objective is a function of the output of a large-scale parametric linear system, obtained from the discretization of a PDE. To reduce the high computational cost of the objective and its gradient, model order reduction techniques can be used. This paper uses interpolatory reduced models as surrogate models in an optimization procedure. We replace the standard first-order condition by the relaxed first-order condition, which is more suitable when algebraic reduced models are used as surrogate models. The relaxed first-order condition imposes that the approximation quality of the surrogate model at the interpolation point can be measured and refined and that the surrogate model is equipped with an error bound on the entire parameter space. We propose two optimization algorithms: one uses the error bound to define a trust region and the other penalizes the objective with the error bound. We prove convergence of both methods under mild conditions in a unified framework. These methods are efficient when surrogate models are cheap to build and evaluate, since they only need these operations to achieve convergence. Numerical experiments from civil engineering show good performance of the proposed methods.

Key words. design optimization, model order reduction, Krylov methods, surrogate models, trust region methods, penalty methods, vibrations and structures

AMS subject classifications. 35Q93, 49M20, 65F10, 65N22, 90C59

DOI. 10.1137/120869171

1. Introduction. Design optimization determines the optimal parameters of an industrial design such as insulation panels and airplane engines. A common approach to solve these problems is to conduct numerical parameter studies via a discretized PDE model. One difficulty with this method is the high computational cost. First, discretization results in high-order systems, whose output is expensive to evaluate. Furthermore, the objective function is often some norm of the system output whose computation requires a large number of evaluations of the system output over a time interval or a frequency range. In this paper, we use $g(\gamma)$ to denote the objective function, where $\gamma \in \mathbb{C}^l$ represents l design parameters.

Model order reduction (MOR) [4] has been successfully applied to many different fields, such as circuit simulations [13, 21], (vibro) acoustics [17], and MEMS design [14], leading to significant speedups in computing the system output. There are two rough classes of numerical optimization methods: line search methods and trust region methods. MOR can be used to accelerate both classes of methods. We now discuss three approaches from the literature.

1. In [27], we proposed the MOR framework that builds a reduced model for each γ accessed by the optimization algorithm, say, $\gamma^{(i)}$. A reduced model is then

*Received by the editors March 8, 2012; accepted for publication (in revised form) April 30, 2013; published electronically June 27, 2013. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State Science Policy Office. The research is also partially funded by Research Council KU Leuven grants PFV/10/002 and OT/10/038. The scientific responsibility rests with the authors.

<http://www.siam.org/journals/siopt/23-2/86917.html>

[†]Department of Computer Science, KU Leuven, 3001 Heverlee, Belgium (yue@mpi-magdeburg.mpg.de, karl.meerbergen@cs.kuleuven.be).

used for efficiently estimating $g(\gamma^{(i)})$ and $\nabla g(\gamma^{(i)})$. Two-sided Krylov methods approximate both $g(\gamma^{(i)})$ and $\nabla g(\gamma^{(i)})$ well even for systems depending nonlinearly on γ [27, 8]. Although it is also possible to approximate Hessians [8], this is usually more expensive. Therefore, gradient-based methods such as quasi-Newton methods are very suitable to combine with this type of MOR method. A major drawback of the MOR framework [27] is that it does not fully exploit the subspaces built by MOR. Since also the gradient is well approximated, we expect the subspaces built at $\gamma^{(i)}$ to be rich enough for γ 's near $\gamma^{(i)}$ when the objective function is smooth.

2. An early work for applying MOR to optimization was the trust-region proper orthogonal decomposition (TRPOD) method [5, 12], which embeds reduced models in the classical trust region method. A disadvantage of TRPOD is the high computational cost. Since it uses the classical trust region method, it requires the first-order condition, i.e., $\hat{g}^{(i)}(\gamma^{(i)}) = g(\gamma^{(i)})$ and $\nabla \hat{g}^{(i)}(\gamma^{(i)}) = \nabla g(\gamma^{(i)})$, and therefore needs direct evaluations of g , which are often quite computationally expensive.

3. Parametric model order reduction (PMOR) techniques can also be used. The reduced basis method [22] uses an error estimator to successively refine the reduced model by adding interpolation points in the parameter space where the error estimation is large. The optimization method is then applied to the parametric reduced model. The downside of this method is that a large amount of work is spent in building an accurate reduced model for the entire parameter space, while the optimization method only requires function and gradient evaluations on the optimization path.

These observations lead us to the objectives of this paper: (1) building reduced models according to the requirements of optimization algorithms, (2) fully exploiting the subspaces built by MOR, and (3) proving convergence purely relying on reduced models. In order to reuse the subspaces built for $\gamma^{(i)}$, we use them to project the full-parameter model defined also for $\gamma \neq \gamma^{(i)}$ in order to obtain a simple interpolatory reduced model, which provides us an objective approximation on the entire parameter space. We denote this interpolatory model by $\hat{g}^{(i)}(\gamma)$. We say that $\hat{g}^{(i)}(\gamma)$ interpolates at $\gamma^{(i)}$ since we use the bases generated at $\gamma^{(i)}$ for the projection. Note that an interpolatory reduced model can have multiple interpolation points [8], but we do not exploit this in the current paper since a local approximation is usually sufficient for optimization. By exploiting $\hat{g}^{(i)}(\gamma)$ also for $\gamma \neq \gamma^{(i)}$, we expect to decrease the number of reduced models needed and thereby reduce the computation time. However, by doing this, we may also run into the risk of inaccurate approximations, especially for γ 's far away from $\gamma^{(i)}$. To achieve accuracy control, we develop a heuristic error estimator of $|g(\gamma) - \hat{g}^{(i)}(\gamma)|$, denoted by $e^{(i)}(\gamma)$, for the entire parameter space.

An interesting question is whether it is possible to achieve convergence purely relying on surrogate models. Clearly, if the surrogate models are not accurate enough, this is not possible. Fortunately, a good property of Krylov–Padé-type surrogate models is that successively enlarging the Krylov subspaces can infinitely increase the accuracy of both the objective function and its gradient at the interpolation point. This means that the approximation quality can approach the first-order condition. We summarize this situation in the more general “relaxed first-order condition”: (1) surrogate models are equipped with an error bound on the entire parameter space and (2) the approximation quality at the interpolation point can be infinitely refined. Under this condition and some other mild conditions, we achieve convergence if all iterates satisfy the proposed “error-aware sufficient decrease condition.” We show in section 5 that equipped with a heuristic error bound, Krylov–Padé-type interpolatory reduced models heuristically satisfy the relaxed first-order condition.

Inspired by the proposed convergence theory, we develop two algorithms in section 4: the error-based trust region method (ETR) and the error-based penalty method (EP). The goal of both algorithms is to locate iterates satisfying the error-aware sufficient decrease condition. At the i th iteration, ETR and EP build a surrogate model $\hat{g}^{(i)}$ interpolated at $\gamma^{(i)}$. To fully exploit $\hat{g}^{(i)}$, we use $\hat{g}^{(i)}$ and its error bound to formulate the i th optimization subproblem and conduct an inner phase optimization for this subproblem. ETR and EP differ only in the definition of the optimization subproblem. ETR uses the bound of the relative error to define a “trust region,” i.e., only to trust a surrogate model when this bound is below a tolerance. EP introduces a penalty term based on the error bound $e^{(i)}(\gamma)$. However, simply adding the error bound to the objective function approximation leads to several difficulties in both convergence theory and computational cost. To overcome these difficulties, we introduce a weighting function for the penalty term.

In this paper, we apply ETR and EP to a specific class of applications, namely, optimization of second-order linear systems arising from structures and vibrations, where we use Krylov–Padé-type interpolatory reduced models as surrogate models. Numerical results show the effectiveness of ETR and EP. These results can be generalized to any surrogate model satisfying the relaxed first-order condition, e.g., built by PMOR [26, 15, 16, 8], or rational Krylov methods [23, 26] that take multiple interpolation frequencies and parameter values in interpolating reduced models.

We close the introduction with notation. We use lowercase letters for vectors, uppercase letters for matrices, I for identity matrices, and 0 for zero matrices. We use \cdot^* for the conjugate transpose of a matrix or a vector or for the complex conjugate of a complex number. For a nonsingular matrix M , we use M^{-*} to denote $(M^*)^{-1}$. We define the partial derivative of a parameterized matrix $A(\lambda) = [a_{p,q}(\lambda)]_{n \times n}$ w.r.t. λ_j as $\frac{\partial A(\lambda)}{\partial \lambda_j} = \left[\frac{\partial a_{p,q}(\lambda)}{\partial \lambda_j} \right]_{n \times n}$. The real part of a complex number z is denoted by $\Re\{z\}$. For a variable in the original model, we use the same symbol with a hat for the corresponding variable in the reduced model. The objective function is denoted by $g(\gamma)$. The i th surrogate (reduced) model interpolated at the i th iterate $\gamma^{(i)}$ is denoted by $\hat{g}^{(i)}(\gamma)$. In general, we use the superscript $\cdot^{(i)}$ to indicate the i th optimization subproblem and the subscript \cdot_j for other indices, e.g., the j th iteration in a subproblem or the j th element of a vector. Following the notation system of [10], we denote constants by κ with different subscripts, e.g., κ_{bt} for the backtracking factor and κ_s for the sufficient decrease factor in the Armijo condition.

2. Description of design optimization problems. In this paper, we focus on parameter studies in the frequency domain. The linear system that we study is

$$(2.1) \quad \begin{cases} \mathcal{L}(\omega, \gamma)x(\omega, \gamma) = f, \\ y(\omega, \gamma) = \ell^*x(\omega, \gamma), \end{cases}$$

where $\mathcal{L}(\omega, \gamma)$ represents a large-scale sparse $n \times n$ matrix parameterized with the frequency ω and l design parameters $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_l) \in \mathbb{C}^l$, $f \in \mathbb{C}^n$ the excitation, $x \in \mathbb{C}^n$ the state vector, $\ell \in \mathbb{C}^n$ the output vector, and y the output. The objective function of interest in this paper is the energy norm of the system output over the frequency range $[\omega_L, \omega_H]$. The mathematical formulation of this problem is

$$(2.2) \quad \min_{\gamma \in \Gamma} g(\gamma), \quad \text{where } g(\gamma) = \int_{\omega_L}^{\omega_H} |y(\omega, \gamma)|^2 \phi(\omega) \, d\omega,$$

$\phi(\omega)$ is a scalar function, and Γ represents the feasible region. The numerical integration of $|y|^2$ is of high computational cost since y has to be computed for many values of ω , each requiring a large sparse matrix factorization.

The input-output behavior of system (2.1) is described by the transfer function

$$(2.3) \quad y(\omega, \gamma) = \ell^* \mathcal{L}(\omega, \gamma)^{-1} f = (\mathcal{L}(\omega, \gamma)^{-*} \ell)^* f,$$

where we define the right and left state vectors as¹

$$(2.4) \quad x(\omega, \gamma) \triangleq \mathcal{L}(\omega, \gamma)^{-1} f \quad \text{and} \quad \xi(\omega, \gamma) \triangleq \mathcal{L}(\omega, \gamma)^{-*} \ell.$$

In structures and vibrations, the following second-order system resulting from the discretization of the governing PDEs is of particular interest:

$$(2.5) \quad \begin{cases} (K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma))x(\omega, \gamma) = f, \\ y(\omega, \gamma) = \ell^* x(\omega, \gamma), \end{cases}$$

where $K(\gamma)$, $C(\gamma)$, and $M(\gamma)$ are $n \times n$ parameterized matrices representing stiffness, damping, and mass, respectively.

2.1. An example: Floor damper optimization. In this application, we consider the design of a tuned mass damper (TMD) whose function is to alleviate the vibration of a simply supported floor inside a building located near a highway. Its conceptual model is shown in Figure 2.1(a). We used a discrete Kirchhoff triangular shell element model [9] for the plate and obtained the following discrete system:

$$(2.6) \quad \begin{cases} (K_0 + (k_1 + i\omega c_1)K_1 - \omega^2 M_0) x = f, \\ y = \ell^* x, \end{cases}$$

where $K_0 = \begin{bmatrix} (1 + ic_p)K_f & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{n \times n}$, $M_0 = \begin{bmatrix} M_f & 0 \\ 0 & m_1 \end{bmatrix} \in \mathbb{C}^{n \times n}$, and $K_1 \in \mathbb{C}^{n \times n}$. The matrix K_1 with four nonzero entries describes the interaction between the damper and its attached element of the plate, K_f and $M_f \in \mathbb{C}^{(n-1) \times (n-1)}$ are the stiffness matrix and the mass matrix obtained from the shell element model of the floor, respectively, c_p represents the proportional damping ratio of the floor, and k_1 , c_1 , and m_1 denote the stiffness, the damping coefficient, and the mass of the damper, respectively. The input vector f represents a unit vertical base excitation, and the output vector ℓ is used to select the degree of freedom corresponding to the center of the plate. In computing the objective g in (2.2), we use $\phi(\omega) = \omega^4$. In this problem, our design parameters are k_1 and c_1 , namely, $\gamma = (k_1, c_1)$, and the constraints are $k_1 > 0$ and $c_1 > 0$. The contour plot of $g(\gamma)$ is shown in Figure 2.1(b).

2.2. Optimization algorithm. To choose a suitable optimization algorithm, we first analyze the formulae for $g(\gamma)$ and its derivatives. Using the formulae

$$(2.7) \quad \begin{aligned} y &= \ell^* \mathcal{L}(\omega, \gamma)^{-1} f, & |y|^2 &= y^* y, \\ \frac{\partial y}{\partial \omega} &= -\ell^* \mathcal{L}(\omega, \gamma)^{-1} \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \omega} \mathcal{L}(\omega, \gamma)^{-1} f, & \frac{\partial |y|^2}{\partial \omega} &= 2\Re \left\{ y^* \frac{\partial y}{\partial \omega} \right\}, \\ \frac{\partial y}{\partial \gamma_j} &= -\ell^* \mathcal{L}(\omega, \gamma)^{-1} \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} \mathcal{L}(\omega, \gamma)^{-1} f, & \frac{\partial |y|^2}{\partial \gamma_j} &= 2\Re \left\{ y^* \frac{\partial y}{\partial \gamma_j} \right\}, \end{aligned}$$

we resort to the trapezoidal rule [11] to perform the numerical integration in computing $g(\gamma) = \int_{\omega_L}^{\omega_H} |y(\omega, \gamma)|^2 d\omega$ and $\frac{\partial g(\gamma)}{\partial \gamma_j} = \int_{\omega_L}^{\omega_H} \frac{\partial |y(\omega, \gamma)|^2}{\partial \gamma_j} d\omega$.

¹We use ξ to denote the left state vector, although it is commonly used to denote adjoint variables.

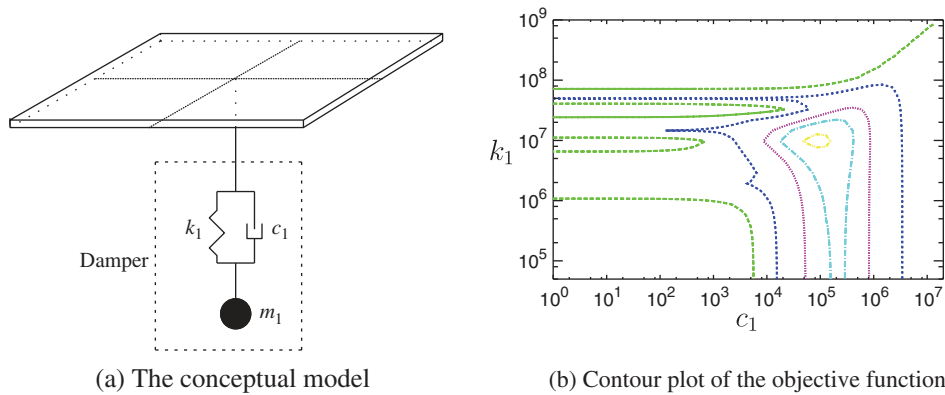


FIG. 2.1. *The floor damper optimization problem. For this plot, we used a discretized model of dimension 280 in order to generate the plot in a short time.*

From (2.7), we can see that because system (2.5) is linear and the output depends linearly on the state vector, computing ∇y is cheap after we have computed y since both computations share the same large sparse matrix factorization. Therefore, gradient-based optimization methods are preferred.

For design optimization problem (2.2), we consider only bound constraints in this paper, i.e., Γ is a hyperrectangle. Our approach consists of first using cotangent barrier functions that we will discuss in section 2.2.1 to convert (2.2) to an unconstrained optimization problem, and then solving it using the damped BFGS method with a backtracking strategy. We require all line search steps to satisfy the Armijo condition,

$$(2.8) \quad g(x_k + \alpha_k p_k) \leq g(x_k) + \kappa_s \alpha_k \nabla g(x_k)^T p_k, \quad \nabla g(x_k)^T p_k < 0,$$

where g is the objective function, x_k is the current iterate, p_k is the current search direction, $\alpha_k > 0$ is the step length, and $\kappa_s \in (0, 1)$ is a constant, which we assign to 10^{-4} in all our numerical tests. The Armijo condition forces sufficient decrease in the objective function for each iteration and plays an important role in convergence [20]. Therefore, we also require it to hold for line searches used elsewhere in the paper, e.g., for the optimization subproblem in ETR and EP discussed in section 3.

2.2.1. Cotangent barrier. In design optimization, the design parameters must take on values that carry physical meaning. For example, k_1 and c_1 in model (2.6) should always be positive and should not take on values that are too small. Therefore, we use the following cotangent barrier function for all design parameters in our applications:

$$(2.9) \quad B_j(\gamma_j) = \begin{cases} 0, & \gamma_j \geq \gamma_j^{\text{th}} & \text{(interval of interest),} \\ \rho_{B_j} \cot\left(\frac{\pi}{2\gamma_j^{\text{th}}}\gamma_j\right), & 0 < \gamma_j < \gamma_j^{\text{th}} & \text{(interval of barrier),} \\ \infty, & \gamma_j \leq 0 & \text{(forbidden interval),} \end{cases}$$

where γ_j^{th} is the threshold value and $\rho_{B_j} > 0$ is a weighting term. In numerical tests, we actually solve

$$(2.10) \quad \min_{\gamma} g(\gamma) + \sum_{j=1}^l B_j(\gamma_j),$$

where the objective function is smooth over the interior of the feasible domain $\text{int}(\Gamma)$ and equals $g(\gamma)$ for the region that we are interested in. Due to the Armijo condition that we use for line searches, we never accept points lying outside $\text{int}(\Omega)$. When a barrier term is active, the gradient of the barrier term helps us to return to the domain of interest. However, whenever the barrier term B_j is active for the optimizer obtained, we should increase ρ_{B_j} to obtain a stronger penalty term and/or increase γ_j^{th} to enlarge the barrier domain, and then we gradually decrease these parameters until the penalty term is inactive inside the domain of interest. Although we may need to repeat this procedure, we cannot run into an infinite loop if $\|\nabla g(\gamma)\|$ is bounded from above and we allow ρ_{B_j} and γ_j^{th} to increase without bound. For all our numerical tests, we used $\rho_{B_j} = 1$ and $\gamma_j^{\text{th}} = 2\pi$ and never needed to increase their values.

3. Using error-aware surrogate models for design optimization. Recall from the introduction that simply building a MOR reduced model for each parameter value accessed by optimization does not fully exploit the effort spent in building the reduced model. A more efficient way is to build an interpolatory reduced model based on the bases generated by MOR, which can be used as a surrogate model in optimization. Since building a surrogate model is much more expensive than evaluating it, we want to exploit each surrogate model as much as possible in a region around the interpolation point. This corresponds well to the concept of a trust region method. In this section, we consider a more general setting: the convergence theory and algorithms of unconstrained optimization using surrogate models with the following properties: (1) surrogate models are equipped with an error bound on the entire parameter space, (2) the approximation quality at the interpolation point can be infinitely refined, and (3) surrogate models are smooth with finite gradient everywhere.

This section is organized as follows. In section 3.1, we will review the first-order condition, analyze the difficulty in meeting it in MOR-based design optimization, and propose a relaxed first-order condition, which does not require exact matches of the objective value and its gradient at the interpolation point but does require an error bound on the objective and the possibility to refine the surrogate models. To achieve convergence, it is common practice to require a sufficient decrease for every iteration. However, the existing sufficient decrease conditions require directly evaluating $g(\gamma)$, which is too expensive. Therefore, in section 3.2, under the relaxed first-order condition, we will present an error-aware sufficient decrease condition that relies only on surrogate models. Roughly speaking, the general idea of this condition is to require the approximate objective decrease achieved by the next iterate to be at least that achieved by an approximate generalized Cauchy point, similarly to what the standard trust region method does. The difference lies in that we never compute the true objective decrease. We will prove that under mild conditions, convergence is guaranteed if all iterates satisfy the error-aware sufficient decrease condition. We will propose a practical procedure that takes both convergence and efficiency into account in section 3.3.

3.1. The relaxed first-order condition. In this section, we first give a brief introduction to the standard trust region method, where the first-order condition plays a key role, and then propose the relaxed first-order condition to overcome the difficulties we have with the first-order condition in MOR-based optimization.

For the i th iteration, the standard trust region method builds a surrogate model $\widehat{g}^{(i)}(\gamma)$ to approximate the exact model $g(\gamma)$ around the current iterate $\gamma^{(i)}$. A classical surrogate model is the quadric determined by the function value, the gradient, and

the (approximate) Hessian at $\gamma^{(i)}$. Other types of surrogate models [1] also exist. The standard trust region method only trusts $\widehat{g}^{(i)}(\gamma)$ within a trust radius Δ_i , so it formulates the i th trust region subproblem as

$$(3.1) \quad \min_s \widehat{g}^{(i)}(\gamma^{(i)} + s) \quad \text{s.t.} \quad \|s\| \leq \Delta_i.$$

In practice, we normally do not target at locating an accurate optimizer of (3.1) but only ask for a sufficient decrease in the objective $\widehat{g}^{(i)}(\gamma)$. Suppose the i th iteration returns $\gamma^{(i+1)}$ as its solution. To test whether it is good enough, we compute

$$(3.2) \quad \rho_i = \frac{g(\gamma^{(i)}) - g(\gamma^{(i+1)})}{\widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma^{(i+1)})}.$$

Based on this value we determine whether we accept $\gamma^{(i+1)}$, enlarge or keep the trust radius, and continue with the $(i+1)$ st optimization subproblem, or we shrink the trust radius and solve the i th trust region subproblem again.

For the standard trust region method to work, we normally require the following first-order condition to hold.

CONDITION 1 (first-order condition).

$$(3.3) \quad \widehat{g}^{(i)}(\gamma^{(i)}) = g(\gamma^{(i)}) \quad \text{and} \quad \nabla \widehat{g}^{(i)}(\gamma^{(i)}) = \nabla g(\gamma^{(i)}).$$

Under the first-order condition, $g(\gamma^{(i)} + s) - \widehat{g}^{(i)}(\gamma^{(i)} + s) = o(s)$ for smooth $g(\gamma)$ and $\widehat{g}^{(i)}(\gamma)$, which means that $\widehat{g}^{(i)}(\gamma)$ is a good approximation of $g(\gamma)$ in the neighborhood of $\gamma^{(i)}$ if $g(\gamma)$ is smooth and $\nabla g(\gamma^{(i)})$ is finite. Therefore, the average approximation error inside the trust region approaches zero when the trust radius approaches zero, which enables the standard trust region method to reduce the approximation error below any positive number simply by reducing the trust radius.

For some applications, however, evaluating $g(\gamma)$ directly is quite expensive and thus both obtaining a model satisfying condition (3.3) and computing ρ_i in (3.2) are unrealistic. Constructing a surrogate model is much cheaper, but the first-order condition (3.3) cannot hold. Therefore, in order to use such a surrogate model in optimization, we must be able to measure its approximation accuracy at $\gamma^{(i)}$ and improve its accuracy at request. Furthermore, to safely use the surrogate model for $\gamma \neq \gamma^{(i)}$, we actually need to measure the approximation accuracy of g on the entire parameter space. See section 5 for an example of such a surrogate model. We summarize the above properties mathematically in Condition 2.

CONDITION 2 (relaxed first-order condition).

C1. Error bounds $e^{(i)}(\gamma)$ and $e_g^{(i)}$ are available for $\widehat{g}^{(i)}(\gamma)$ and $\nabla \widehat{g}^{(i)}(\gamma^{(i)})$, i.e.,

$$(3.4) \quad |\widehat{g}^{(i)}(\gamma) - g(\gamma)| \leq e^{(i)}(\gamma) \quad \text{and} \quad \|\nabla \widehat{g}^{(i)}(\gamma^{(i)}) - \nabla g(\gamma^{(i)})\| \leq e_g^{(i)}.$$

C2. The surrogate model can be infinitely refined at $\gamma^{(i)}$, i.e., for any given $\tau_g > 0$ and $\tau_{\nabla g} > 0$, there exists a surrogate model $\widehat{g}^{(i)}$ satisfying

$$(3.5) \quad \frac{e^{(i)}(\gamma^{(i)})}{\widehat{g}^{(i)}(\gamma^{(i)})} \leq \tau_g \quad \text{and} \quad \frac{e_g^{(i)}}{\|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\|} \leq \tau_{\nabla g}.$$

Lemma 3.1 shows that when $\tau_{\nabla g}$ is small, $\nabla \widehat{g}^{(i)}(\gamma^{(i)})$ approximates $\nabla g(\gamma^{(i)})$ well in both its norm and direction.

LEMMA 3.1. Denote the angle between $\nabla\hat{g}^{(i)}(\gamma^{(i)})$ and $\nabla g(\gamma^{(i)})$ with ϑ . Then, under the relaxed first-order condition, we have

$$(3.6) \quad 1 - \tau_{\nabla g} \leq \frac{\|\nabla g(\gamma^{(i)})\|}{\|\nabla\hat{g}^{(i)}(\gamma^{(i)})\|} \leq 1 + \tau_{\nabla g}$$

$$(3.7) \quad \text{and} \quad \cos \vartheta \geq \frac{1 - \tau_{\nabla g}^2}{\sqrt{1 + \tau_{\nabla g}^2}}.$$

Proof. Statement (3.6) directly follows from (3.4), (3.5), and

$$\begin{aligned} \|\nabla g(\gamma^{(i)})\| &\geq \|\nabla\hat{g}^{(i)}(\gamma^{(i)})\| - \|\nabla\hat{g}^{(i)}(\gamma^{(i)}) - \nabla g(\gamma^{(i)})\|, \\ \|\nabla g(\gamma^{(i)})\| &\leq \|\nabla\hat{g}^{(i)}(\gamma^{(i)})\| + \|\nabla\hat{g}^{(i)}(\gamma^{(i)}) - \nabla g(\gamma^{(i)})\|. \end{aligned}$$

Statement (3.7) holds because

$$\begin{aligned} \cos \vartheta &= \frac{\|\nabla\hat{g}^{(i)}(\gamma^{(i)})\|^2 + \|\nabla g(\gamma^{(i)})\|^2 - \|\nabla\hat{g}^{(i)}(\gamma^{(i)}) - \nabla g(\gamma^{(i)})\|^2}{2\|\nabla\hat{g}^{(i)}(\gamma^{(i)})\|\|\nabla g(\gamma^{(i)})\|} \\ &\geq \frac{\|\nabla\hat{g}^{(i)}(\gamma^{(i)})\|^2 - (e_g^{(i)})^2}{\|\nabla\hat{g}^{(i)}(\gamma^{(i)})\|\sqrt{\|\nabla\hat{g}^{(i)}(\gamma^{(i)})\|^2 + (e_g^{(i)})^2}} \geq \frac{1 - \tau_{\nabla g}^2}{\sqrt{1 + \tau_{\nabla g}^2}}. \quad \square \end{aligned}$$

In the rest of the paper, we assume that ϑ is acute, i.e., $\tau_{\nabla g} < 1$, which is always possible under the relaxed first-order condition due to C2.

When we use surrogate models, we can define three levels of accuracy. The true objective is $g(\gamma)$, so $g(\gamma)$ and $\nabla g(\gamma)$ provide the highest level of accuracy. When we generate $\hat{g}^{(i)}$, we impose conditions (3.4) and (3.5), so $\hat{g}^{(i)}(\gamma^{(i)})$ and $\nabla\hat{g}^{(i)}(\gamma^{(i)})$ are of high accuracy. For $\gamma \neq \gamma^{(i)}$, however, we normally only know that $\hat{g}^{(i)}(\gamma)$ is a good approximation in a neighborhood of $\gamma^{(i)}$ due to the high accuracy of $\hat{g}^{(i)}(\gamma^{(i)})$ and $\nabla\hat{g}^{(i)}(\gamma^{(i)})$, so $\hat{g}^{(i)}(\gamma)$ is of low accuracy for $\gamma \neq \gamma^{(i)}$.

To develop efficient algorithms for our situation, we also make the following assumptions for the computational cost.

ASSUMPTION 1 (three levels of computational cost). Directly computing $g(\gamma)$ and $\nabla g(\gamma)$ is very expensive, which we want to avoid. Generating $\hat{g}^{(i)}$ is expensive, so we prefer to use fewer surrogate models. Evaluating $\hat{g}^{(i)}(\gamma)$ and $\nabla\hat{g}^{(i)}(\gamma)$ is cheap.

Under Assumption 1, we prefer to fully exploit each surrogate model to reduce the number of surrogate models required. Therefore, at the i th iteration, we first build the i th surrogate model interpolated at $\gamma^{(i)}$, and then we use its output $\hat{g}^{(i)}(\gamma)$ and the error bound function $e^{(i)}(\gamma)$ to construct the i th subproblem

$$(3.8) \quad \min_{\gamma} F^{(i)}(\gamma) \quad \text{s.t.} \quad c^{(i)}(\gamma) \geq 0.$$

$F^{(i)}$ and $c^{(i)}$ are formulated to locate a γ satisfying the error-aware sufficient decrease condition that we will propose in section 3.2, which is the key in achieving convergence. However, to fully exploit surrogate models, we aim at a larger decrease. Therefore, we use an inner phase optimization, e.g., employing a quasi-Newton method confined to the feasible domain, for solving (3.8). Normally, $F^{(i)}$ equals to or is an upper bound of $\hat{g}^{(i)}$. In this paper, we assume that $F^{(i)}(\gamma) = \hat{g}^{(i)}(\gamma)$ holds in a neighborhood of $\gamma^{(i)}$ because in this case, $\nabla F^{(i)}(\gamma^{(i)}) \rightarrow 0$ means $\nabla\hat{g}^{(i)}(\gamma^{(i)}) \rightarrow 0$, and $\nabla g(\gamma^{(i)}) \rightarrow 0$ due to (3.6) since $\tau_{\nabla g} < 1$. We postpone the details in formulating $F^{(i)}(\gamma)$ and $c^{(i)}(\gamma)$ to sections 4.1 and 4.2. For the convergence theory, we need only $\hat{g}^{(i)}(\gamma)$ and $c^{(i)}(\gamma)$.

3.2. An error-aware sufficient decrease condition for convergence under the relaxed first-order condition. Following the spirit of the standard trust region method, we use the following three steps to locate a solution of (3.8) and check the sufficient decrease: (1) Look in a descent direction and find a point with sufficient decrease in the surrogate model $\widehat{g}^{(i)}$, which is known as the approximate generalized Cauchy point. (2) Locate a solution of (3.8), under the condition that the decrease in the surrogate model $\widehat{g}^{(i)}$ achieved by it is no less than that achieved by the approximate generalized Cauchy point. (3) Test the sufficient decrease condition that leads to the convergence of the original model g .

3.2.1. The approximate generalized Cauchy point and sufficient objective decrease. Given a descent direction $p^{(i)}$ satisfying $\nabla\widehat{g}^{(i)}(\gamma^{(i)})^T p^{(i)} < 0$ and $\|p^{(i)}\| = 1$, the approximate generalized Cauchy point, which we denote by $\gamma_{\text{AGC}}^{(i)}$, is defined as a point that achieves sufficient decrease in the surrogate model $\widehat{g}^{(i)}$ on

$$(3.9) \quad \gamma = \gamma^{(i)} + \alpha p^{(i)} \quad (\alpha \geq 0).$$

For simplicity, this paper only considers obtaining $\gamma_{\text{AGC}}^{(i)}$ by backtracking-Armijo searches. Under this setting, $\gamma_{\text{AGC}}^{(i)}$ should satisfy three conditions:

1. It should satisfy the Armijo condition (2.8) on $\widehat{g}^{(i)}$,

$$(3.10) \quad \widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma) \geq \kappa_s \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \|\gamma^{(i)} - \gamma\| \cos \Phi^{(i)},$$

where $\Phi^{(i)}$ represents the angle between $-\nabla\widehat{g}^{(i)}$ and $p^{(i)}$.

2. It should satisfy the constraint

$$(3.11) \quad c^{(i)}(\gamma) \geq 0.$$

3. The initial step length of the backtracking, which is denoted by $\alpha_0^{(i)}$, should not be too small w.r.t. $\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|$; more specifically, there exists a $\kappa_n > 0$ satisfying

$$(3.12) \quad \alpha_0^{(i)} \geq \kappa_n \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \quad \text{for all } i\text{'s.}$$

For quasi-Newton methods, $\alpha_0^{(i)} = \|(B_0^{(i)})^{-1} \nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \geq (\lambda_{\max}^{(i)})^{-1} \|\widehat{g}^{(i)}(\gamma^{(i)})\|$, where $B_0^{(i)}$ is the initial Hessian approximation for the i th optimization subproblem and $\lambda_{\max}^{(i)} > 0$ is the largest eigenvalue of $B_0^{(i)}$. As long as $\lambda_{\max}^{(i)}$ does not grow without bound, we can set $0 < \kappa_n \leq \min_i \{(\lambda_{\max}^{(i)})^{-1}\}$. So when $\lambda_{\max}^{(i)}$ becomes too large, we can assign $\lambda_{\max}^{(i)}$ to $h_0 > 0$ by resetting $B_0^{(i)}$ to $h_0 I$.

PROCEDURE 1. Locate $\gamma_{\text{AGC}}^{(i)}$ on a given descent search direction $p^{(i)}$.

1. Assign $j = 0$ and choose $\kappa_{bt} \in (0, 1)$. Find an $\alpha_0^{(i)}$ satisfying (3.12).
2. Compute $\gamma = \gamma^{(i)} + \alpha_j^{(i)} p^{(i)}$. Check whether γ satisfies both (3.10) and (3.11).
 - 2.1. If both conditions hold, assign $\gamma_{\text{AGC}}^{(i)} = \gamma$.
 - 2.2. Otherwise, assign $\alpha_{j+1}^{(i)} = \kappa_{bt} \alpha_j^{(i)}$, $j = j + 1$, and go back to step 2.

If we use an Armijo line search to solve the optimization subproblems, $\gamma_{\text{AGC}}^{(i)}$ is simply the first iterate of the Armijo line search.

Theorem 3.2 shows that under mild conditions, we can find the $\gamma_{\text{AGC}}^{(i)}$ within finite steps. It also gives the minimal objective decrease on $\widehat{g}^{(i)}$ achieved by $\gamma_{\text{AGC}}^{(i)}$.

THEOREM 3.2. Assume that $c^{(i)}$ and $\nabla\widehat{g}^{(i)}$ are Lipschitz continuous, i.e., there exist $u_c^{(i)} \geq 0$ and $u_g^{(i)} \geq 0$ such that $|c^{(i)}(\gamma) - c^{(i)}(\gamma^{(0)})| \leq u_c^{(i)} \|\gamma - \gamma^{(0)}\|$ and

$\|\nabla\widehat{g}^{(i)}(\gamma) - \nabla\widehat{g}^{(i)}(\gamma^{(0)})\| \leq u_{\nabla\widehat{g}}^{(i)}\|\gamma - \gamma^{(0)}\|$. Then, if $\widehat{g}^{(i)}(\gamma)$ is bounded from below, $\Phi^{(i)} < \frac{\pi}{2}$, $\kappa_s \in (0, 1)$, $c^{(i)}(\gamma^{(i)}) > 0$, $u_c^{(i)} > 0$, $\kappa_n > 0$, and $u_{\widehat{g}}^{(i)} > 0$, we have the following:

1. Both conditions (3.11) and (3.10) hold for all γ 's of the form (3.9) that satisfy

$$(3.13) \quad \|\gamma - \gamma^{(i)}\| \leq \min \left\{ \frac{(1 - \kappa_s) \cos \Phi^{(i)} \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|}{u_{\widehat{g}}^{(i)}}, \frac{c^{(i)}(\gamma^{(i)})}{u_c^{(i)}} \right\}.$$

2. The approximate generalized Cauchy point $\gamma_{\text{AGC}}^{(i)}$ satisfies

$$(3.14) \quad \|\gamma_{\text{AGC}}^{(i)} - \gamma^{(i)}\| \geq \min \left\{ \kappa_{\nabla g}^{(i)} \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|, \kappa_{bt} \frac{c^{(i)}(\gamma^{(i)})}{u_c^{(i)}} \right\},$$

where $\kappa_{\nabla g}^{(i)} = \min \left\{ \kappa_n, \frac{\kappa_{bt}(1 - \kappa_s) \cos \Phi^{(i)}}{u_{\widehat{g}}^{(i)}} \right\}$.

3. A lower bound of the decrease in $\widehat{g}^{(i)}$ achieved by $\widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)})$ is given by

$$(3.15) \quad \widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)}) \geq \left(\kappa_s \cos \Phi^{(i)} \right) \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \min \left\{ \kappa_{\nabla g}^{(i)} \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|, \kappa_{bt} \frac{c^{(i)}(\gamma^{(i)})}{u_c^{(i)}} \right\}.$$

Proof. 1. If $\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| = 0$, condition (3.13) becomes $\gamma = \gamma^{(i)}$, so conditions (3.11) and (3.10) hold. Now we consider the case where $\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \neq 0$. For the direction $p^{(i)}$ in (3.9), we have $\nabla_{p^{(i)}}\widehat{g}^{(i)}(\gamma^{(i)}) = -\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \cos \Phi^{(i)}$. Consider the equation

$$(3.16) \quad \nabla_{p^{(i)}}\widehat{g}^{(i)}(\gamma) = -\kappa_s \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \cos \Phi^{(i)}.$$

(i) Equation (3.16) has at least one solution. We prove this by contradiction. If (3.16) has no solution, $\nabla_{p^{(i)}}\widehat{g}^{(i)}(\gamma) < -\kappa_s \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \cos \Phi^{(i)}$ holds for all γ 's on the search line, indicating that $\widehat{g}^{(i)}(\gamma)$ approaches $-\infty$ as the step length α approaches ∞ according to Lagrange's mean value theorem. This contradicts the assumption that $\widehat{g}^{(i)}(\gamma)$ is bounded from below.

(ii) The Lipschitz continuity assumption on $\nabla\widehat{g}^{(i)}$ indicates that a solution of (3.16), which we denote by γ , satisfies

$$(3.17) \quad \|\gamma^{(i)} - \gamma\| \geq \frac{|\nabla_{p^{(i)}}\widehat{g}^{(i)}(\gamma^{(i)}) - \nabla_{p^{(i)}}\widehat{g}^{(i)}(\gamma)|}{u_{\widehat{g}}^{(i)}} = \frac{(1 - \kappa_s) \cos \Phi^{(i)} \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|}{u_{\widehat{g}}^{(i)}},$$

which means that (3.10) holds for all γ 's on the search direction (3.9) satisfying

$$(3.18) \quad \|\gamma^{(i)} - \gamma\| \leq \frac{(1 - \kappa_s) \cos \Phi^{(i)} \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|}{u_{\widehat{g}}^{(i)}}$$

according to Lagrange's mean value theorem.

Similarly, due to the Lipschitz continuity of $c^{(i)}$, a solution of $c^{(i)}(\gamma) = 0$ must satisfy

$$\|\gamma^{(i)} - \gamma_c^{(i)}\| \geq \frac{|c^{(i)}(\gamma^{(i)}) - c^{(i)}(\gamma_c^{(i)})|}{u_c^{(i)}} = \frac{c^{(i)}(\gamma^{(i)})}{u_c^{(i)}},$$

which means that condition (3.11) holds for all γ 's on the search direction (3.9) satisfying $\|\gamma^{(i)} - \gamma_c^{(i)}\| \leq \frac{c^{(i)}(\gamma^{(i)})}{u_c^{(i)}}$. Based on arguments above, statement (3.13) holds.

2. First, if the initial point $\gamma^{(i)} + \alpha_0^{(i)} d^{(i)}$ satisfies both (3.11) and (3.12), its distance to $\gamma^{(i)}$ is at least $\min \left\{ \kappa_{\nabla g}^{(i)} \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\|, \kappa_{bt} \frac{c^{(i)}(\gamma^{(i)})}{u_c^{(i)}} \right\}$. Then, if backtracking occurs, it must stop before

$$(3.19) \quad \|\gamma - \gamma^{(i)}\| \leq \kappa_{bt} \min \left\{ \frac{(1 - \kappa_s) \cos \Phi^{(i)} \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\|}{u_{\nabla \widehat{g}}^{(i)}}, \frac{c^{(i)}(\gamma^{(i)})}{u_c^{(i)}} \right\}$$

holds, since otherwise the previous backtracking point would satisfy condition (3.13) and thus be already accepted as $\gamma_{\text{AGC}}^{(i)}$. Therefore, (3.19) cannot hold for $\gamma_{\text{AGC}}^{(i)}$. According to these two arguments, $\gamma_{\text{AGC}}^{(i)}$ satisfies (3.14).

3. This is a direct result of (3.14) and (3.10). \square

The assumptions in Theorem 3.2 are not very restrictive. If $c^{(i)}$ and $\nabla \widehat{g}^{(i)}$ are not Lipschitz continuous, they can change abruptly, and gradient-based optimization algorithms may run into problems. If $c^{(i)}(\gamma^{(i)}) \leq 0$, the optimization subproblem is not properly defined. The assumption that $\widehat{g}^{(i)}(\gamma)$ is bounded from below makes sense for physical applications, e.g., zero is a natural bound for an energy function.

3.2.2. An error-aware sufficient decrease condition and convergence for the “ideal” case. Although $\gamma_{\text{AGC}}^{(i)}$ is used to define a sufficient decrease in the surrogate model $\widehat{g}^{(i)}$ rather than in the true model g , we now show that we can use it to formulate an *error-aware sufficient decrease condition* and prove that if this condition is satisfied by all iterations, which we call the ideal case, we attain convergence to the minimizer of the original model g under some mild conditions.

CONDITION 3. *The error-aware sufficient decrease condition is*

$$(3.20) \quad \widehat{g}^{(i+1)}(\gamma^{(i+1)}) \leq \widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)}),$$

where $\widehat{g}^{(i)}$ denotes a surrogate model interpolated at $\gamma^{(i)}$ satisfying (3.5) with $\tau_{\nabla g} < 1$.

This condition is computationally feasible to check since it needs no evaluation of g but needs only one generation of a new surrogate model $\widehat{g}^{(i+1)}$ and two solves of the surrogate models. If this condition is satisfied, we accept $\gamma^{(i+1)}$ and $\widehat{g}^{(i+1)}$ and use them to formulate the $(i+1)$ st optimization subproblem. Theorem 3.3 shows that under mild conditions, we achieve convergence if we accept $\gamma^{(i+1)}$ only when it satisfies condition (3.20). We will discuss in section 3.2.3 what we do when condition (3.20) fails.

THEOREM 3.3. *Assume that (1) all conditions for Theorem 3.2 hold, (2) the relaxed first-order condition (Condition 2) holds, and (3) $\Phi^{(i)} \leq \Phi < \frac{\pi}{2}$, $c^{(i)}(\gamma^{(i)}) \geq c_l > 0$, $0 < u_c^{(i)} \leq u_c$, $0 < u_g^{(i)} \leq u_g$ for all i 's, $\tau_{\nabla g} < 1$, $\kappa_n > 0$, $\widehat{g}^{(0)}(\gamma^{(0)})$ is finite and $\widehat{g}^{(i)}(\gamma)$ is bounded from below for all i 's. Then if the error-aware sufficient decrease condition (3.20) holds for all i , we have $\lim_{i \rightarrow \infty} \|\nabla g(\gamma^{(i)})\| = 0$.*

Proof. According to the assumptions and (3.15),

$$\begin{aligned} & \widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i+1)}(\gamma^{(i+1)}) \\ & \geq \left(\kappa_s \cos \Phi^{(i)} \right) \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\| \min \left\{ \kappa_{\nabla g}^{(i)} \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\|, \kappa_{bt} \frac{c^{(i)}(\gamma^{(i)})}{u_c^{(i)}} \right\}. \end{aligned}$$

So,
$$\widehat{g}^{(0)}(\gamma^{(0)}) - \widehat{g}^{(k)}(\gamma^{(k)}) \geq \sum_{i=0}^{k-1} \left(\kappa_s \cos \Phi^{(i)} \right) \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\| \min \left\{ \kappa_{\nabla g}^{(i)} \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\|, \kappa_{bt} \frac{c^{(i)}(\gamma^{(i)})}{u_c^{(i)}} \right\}.$$

Since $\kappa_{\nabla g}^{(i)} = \min \left\{ \kappa_n, \frac{\kappa_{bt}(1 - \kappa_s) \cos \Phi^{(i)}}{u_{\widehat{g}}^{(i)}} \right\} \geq \min \left\{ \kappa_n, \frac{\kappa_{bt}(1 - \kappa_s) \cos \Phi}{u_{\widehat{g}}} \right\} \triangleq \kappa_{\nabla g}$,

we have

$$\widehat{g}^{(0)}(\gamma^{(0)}) - \widehat{g}^{(k)}(\gamma^{(k)}) \geq \kappa_s \cos \Phi \sum_{i=0}^{k-1} \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\| \min \left\{ \kappa_{\nabla g} \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\|, \kappa_{bt} \frac{c_l}{u_c} \right\}.$$

Now we prove $\lim_{i \rightarrow \infty} \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\| = 0$ by contradiction. Assume there would exist an $\varepsilon \in (0, \|\nabla \widehat{g}^{(0)}(\gamma^{(0)})\|)$ and a subsequence $\nu(j)$ satisfying $\|\nabla \widehat{g}^{\nu(j)}(\gamma^{\nu(j)})\| \geq \varepsilon$ for all $j \in \mathbb{N}$ with $\nu(0) = 0$. Then,

$$\lim_{k \rightarrow \infty} \widehat{g}^{(0)}(\gamma^{(0)}) - \widehat{g}^{\nu(k)}(\gamma^{\nu(k)}) \geq \lim_{k \rightarrow \infty} \kappa_s \cos \Phi \sum_{j=0}^{k-1} \varepsilon \min \left\{ \kappa_{\nabla g} \varepsilon, \kappa_{bt} \frac{c_l}{u_c} \right\} = +\infty,$$

contradicting the fact that $\widehat{g}^{(0)}(\gamma^{(0)})$ is finite and $\widehat{g}^{(k)}(\gamma)$ is bounded from below. Therefore, $\lim_{i \rightarrow \infty} \|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\| = 0$. According to (3.6), $\lim_{i \rightarrow \infty} \|\nabla g(\gamma^{(i)})\| = 0$ since $\tau_{\nabla g} < 1$. \square

All assumptions in Theorem 3.2 are not very restrictive except that all iterates satisfy (3.20). If $c^{(i)}(\gamma^{(i)})$ does not approach zero, and $u_c^{(i)}$, $u_{\widehat{g}}^{(i)}$, and $\lambda_{\max}^{(i)}$ do not approach $+\infty$, then c_l , u_c , $u_{\widehat{g}}$, and λ_{\max} are well defined. Although $\Phi^{(i)}$ may approach $\frac{\pi}{2}$, we can change the search direction $p^{(i)}$ when $\Phi^{(i)}$ exceeds Φ to avoid this problem, e.g., force $\Phi^{(i)} = 0$ by resetting $B_0^{(i)}$ to a multiple of the identity matrix for quasi-Newton methods. Requiring all iterates to satisfy (3.20) is too optimistic; we now discuss what to do when condition (3.20) fails.

3.2.3. Practical considerations and convergence for the general case.

Refine the approximate generalized Cauchy point by backtracking. Now we show that even when the error-aware sufficient decrease condition (3.20) fails, we do not necessarily need to reject $\gamma^{(i+1)}$ since condition (3.20) can be too restrictive. A good sufficient decrease condition should not be too restrictive. For example, the Armijo condition (2.8) in line searches usually employs such small κ_s value as 10^{-4} . However, condition (3.20) can be too restrictive sometimes, i.e., the decrease of $\widehat{g}^{(i)}$ achieved by $\gamma_{\text{AGC}}^{(i)}$ is relatively large. Here, we give two extreme examples:

1. $\widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)}) < \min_{\gamma} g(\gamma)$ and $\widehat{g}^{(i+1)}(\gamma^{(i+1)}) > g(\gamma^{(i+1)})$.
2. $\gamma_{\text{AGC}}^{(i)}$ is the global minimizer of $\widehat{g}^{(i)}$ and $\widehat{g}^{(i+1)}(\gamma^{(i+1)}) > \widehat{g}^{(i)}(\gamma^{(i+1)})$.

Note that these cases can even happen to surrogate models with high accuracy.

To deal with such difficulties, we have to relax condition (3.20). Fortunately, we can relax condition (3.20) simply by conducting backtracking on $\gamma_{\text{AGC}}^{(i)}$ since when $\gamma_{\text{AGC}}^{(i)}$ is sufficiently close to $\gamma^{(i)}$, we expect that backtracking leads to a $\gamma_{\text{AGC}}^{(i)}$ with smaller error and smaller decrease in $\widehat{g}^{(i)}$. More specifically, we assign

$$(3.21) \quad \gamma_{\text{AGC}}^{(i)} \Leftarrow \gamma^{(i)} + \kappa_{bt}(\gamma_{\text{AGC}}^{(i)} - \gamma^{(i)})$$

until both conditions (3.11) and (3.10) hold. Backtracking on $\gamma_{\text{AGC}}^{(i)}$ is cheap since no extra surrogate model is required. With this updated $\gamma_{\text{AGC}}^{(i)}$, we can still attain convergence to the minimizer of g since if we assign $\alpha_0^{(i)} = \|\gamma_{\text{AGC}}^{(i)} - \gamma^{(i)}\|$, Theorems 3.2 and 3.3 apply with no change.

COROLLARY 3.4. *Assume that all assumptions of Theorems 3.3 hold. Then, if all iterates satisfy (3.20), where we allow to conduct finite steps of backtracking for $\gamma_{\text{AGC}}^{(i)}$ as in (3.21), we have $\lim_{i \rightarrow \infty} \|\nabla g(\gamma^{(i)})\| = 0$.*

Cure the failure of the error-aware sufficient decrease condition (3.20). When condition (3.20) fails, we can use the following two strategies:

- S1. Conduct backtracking on $\gamma_{\text{AGC}}^{(i)}$ to make condition (3.20) easier to satisfy. This option is always preferred since conducting backtracking on $\gamma_{\text{AGC}}^{(i)}$ is cheap.
- S2. Reformulate the i th optimization subproblem and solve it again. Here we have several possibilities: (1) Reduce τ and $\tau_{\nabla g}$ and refine the surrogate model according to these new tolerances; we can use this strategy for both ETR and EP. (2) Use a more restrictive constraint $c^{(i)}$; we use this strategy in ETR. (3) introduce stronger error-based penalty in $F^{(i)}$; we use this strategy in EP.

We now explain how to choose between these two strategies in different situations. To judge whether condition (3.20) is too restrictive, we define

$$(3.22) \quad \varrho^{(i)} = \frac{\widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)})}{\widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma^{(i+1)})}.$$

Since $\widehat{g}^{(i)}(\gamma^{(i)}) > \widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)}) \geq \widehat{g}^{(i)}(\gamma^{(i+1)})$ due to the sufficient decrease condition² that we impose on the optimization subproblem, we have $0 < \varrho^{(i)} \leq 1$.

If condition (3.20) fails, we have the following possibilities:

1. When $\widehat{g}^{(i+1)}(\gamma^{(i+1)}) > \widehat{g}^{(i)}(\gamma^{(i)})$, S1 cannot work since $\widehat{g}^{(i)}(\gamma^{(i)}) > \widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)})$ always holds. Therefore, we use S2.
2. When $\varrho^{(i)}$ is not sufficiently small, condition (3.20) is too restrictive since $\gamma_{\text{AGC}}^{(i)}$ has achieved similar decrease as the optimizer. Therefore, we use S1.
3. When $\varrho^{(i)}$ is sufficiently small, condition (3.20) is not too restrictive. Thus $\gamma^{(i+1)}$ is not an adequate optimizer. Therefore, we use S2.

A means to reduce the computational cost. Directly checking the error-aware sufficient decrease condition (3.20) requires building a new surrogate model $\widehat{g}^{(i+1)}$. If the test fails, the surrogate model $\widehat{g}^{(i+1)}$ cannot be used to define the next optimization subproblem, which is a waste of computational cost. In this section, we will develop sufficient and necessary conditions for the error-aware sufficient decrease condition, which do not require an additional surrogate model to be built.

LEMMA 3.5. *Under the relaxed first-order condition (Condition 2),*

1. *a sufficient condition for (3.20) is*

$$(3.23) \quad \widehat{g}^{(i)}(\gamma^{(i+1)}) + e^{(i)}(\gamma^{(i+1)}) + e^{(i+1)}(\gamma^{(i+1)}) \leq \widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)});$$

2. *a necessary condition for (3.20) is*

$$(3.24) \quad \widehat{g}^{(i)}(\gamma^{(i+1)}) - e^{(i)}(\gamma^{(i+1)}) - e^{(i+1)}(\gamma^{(i+1)}) \leq \widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)}).$$

² $\widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)}) = \widehat{g}^{(i)}(\gamma^{(i+1)})$ only when $\gamma^{(i+1)} = \gamma_{\text{AGC}}^{(i)}$.

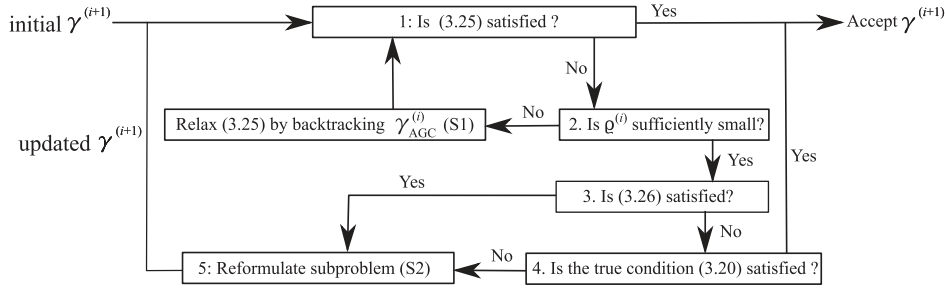


FIG. 3.1. The proposed working procedure in checking condition (3.20).

Proof. Lemma 3.5 follows due to (3.20) and

$$\begin{aligned} \widehat{g}^{(i+1)}(\gamma^{(i+1)}) &\leq g(\gamma^{(i+1)}) + e^{(i+1)}(\gamma^{(i+1)}) \\ &\leq \widehat{g}^{(i)}(\gamma^{(i+1)}) + e^{(i)}(\gamma^{(i+1)}) + e^{(i+1)}(\gamma^{(i+1)}), \\ \widehat{g}^{(i+1)}(\gamma^{(i+1)}) &\geq g(\gamma^{(i+1)}) - e^{(i+1)}(\gamma^{(i+1)}) \\ &\geq \widehat{g}^{(i)}(\gamma^{(i+1)}) - e^{(i)}(\gamma^{(i+1)}) - e^{(i+1)}(\gamma^{(i+1)}). \quad \square \end{aligned}$$

Under the relaxed first-order condition, it is safe to accept $\gamma^{(i+1)}$ when

$$(3.25) \quad \widehat{g}^{(i)}(\gamma^{(i+1)}) + e^{(i)}(\gamma^{(i+1)}) < \widehat{g}^{(i)}(\gamma_{AGC}^{(i)}),$$

since if at the generation of $\widehat{g}^{(i+1)}$, we require $e^{(i+1)}(\gamma^{(i+1)})$ to be sufficiently smaller than the difference between the right-hand side and left-hand side of (3.25), condition (3.23) will hold. Similarly, we assume that condition (3.24) does not hold when

$$(3.26) \quad \widehat{g}^{(i)}(\gamma^{(i+1)}) - e^{(i)}(\gamma^{(i+1)}) > \widehat{g}^{(i)}(\gamma_{AGC}^{(i)}),$$

because the satisfactions of both (3.26) and (3.24) may require a large $e^{(i+1)}(\gamma^{(i+1)})$, which leads to an inaccurate surrogate model for the next iteration.

3.3. A practical working procedure. Now, based on what we have discussed in section 3.2, we give a practical working procedure shown in Figure 3.1 and Procedure 2, which takes both convergence and efficiency into account.

PROCEDURE 2.

1. We first check condition (3.25) since it is cheap to check and lies on the shortest path to acceptance. If it holds, we accept $\gamma^{(i+1)}$ and go to the next iteration.

2. Otherwise, if condition (3.25) fails, we have two options: S1 and S2. We choose S1 whenever feasible because it requires no additional surrogate model. Therefore, we check whether $\rho^{(i)}$ is sufficiently small. If not, conduct backtracking for $\gamma_{AGC}^{(i)}$ (S1) and go to step 1.

3. Otherwise, if $\rho^{(i)}$ is not sufficiently small, we check the necessary condition (3.26). If condition (3.26) holds, we are almost sure that $\gamma^{(i+1)}$ is not a good point. So in this case, we reformulate the optimization subproblem and solve it again (S2) rather than build a new surrogate model to check (3.20).

4. Otherwise, if condition (3.26) fails, we build a new surrogate model $\widehat{g}^{(i+1)}$ and check condition (3.20). If it holds, accept $\gamma^{(i+1)}$.

5. Otherwise, reject $\gamma^{(i+1)}$ and use S2 to update $\gamma^{(i+1)}$.

TABLE 3.1
Parameters for solving the subproblem.

Name	Meaning	Range	Suggested value
κ_{bt}	backtracking factor	(0, 1)	0.3–0.7
κ_s	sufficient decrease coefficient	(0, 1)	very small, e.g., 10^{-4}
κ_n	tolerance on the scaling	(0, 1)	very small, e.g., 10^{-5}
Φ	tolerance on the search angle	$(0, \frac{\pi}{2})$	very close to $\frac{\pi}{2}$
$\rho^{(i)}$	to choose between S1 and S2	(0,1)	close to 0, e.g., 0.1

In Procedure 2, we build a new surrogate model only if we reach step 4, and we fail to use it to formulate the next optimization subproblem only if we reach step 5. We finalize this section by summarizing the parameters used by Procedure 2 in Table 3.1.

4. Two design optimization algorithms based on the relaxed first-order condition. In terms of computational cost, the ideal case is that condition (3.25) always holds. So an efficient algorithm should have a mechanism to make condition (3.25) easy to satisfy. Since after possible backtracking(s) on $\gamma_{AGC}^{(i)}$, $\widehat{g}^{(i)}(\gamma^{(i+1)}) < \widehat{g}^{(i)}(\gamma_{AGC}^{(i)})$ must hold, ETR requires $e^{(i)}(\gamma^{(i+1)})$ to be small compared with these two values in order to increase the possibility that (3.25) holds. EP considers $\widehat{g}^{(i)}(\gamma^{(i+1)}) + e^{(i)}(\gamma^{(i+1)})$ as a whole and introduces a penalty term to achieve error control.

4.1. ETR method. ETR defines the i th optimization subproblem as

$$(4.1) \quad \min_{\gamma} \widehat{g}^{(i)}(\gamma) \quad \text{s.t.} \quad \frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} \leq \epsilon_L \quad (\epsilon_L > 0).$$

To ensure that the feasible domain is not empty, ETR requires

$$(4.2) \quad \frac{e^{(i)}(\gamma^{(i)})}{\widehat{g}^{(i)}(\gamma^{(i)})} \leq \delta \epsilon_L \quad (\delta \in (0, 1)),$$

which means that the interpolation point must strictly satisfy the constraint in ETR subproblem (4.1). If condition (4.2) does not hold, we refine the surrogate model.

We use a line search method, such as a quasi-Newton method, to solve the trust region subproblem (4.1). To stay inside the trust region, we do backtracking not only when the Armijo condition (2.8) is violated, but also when the current γ lies outside the trust region, i.e., $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} > \epsilon_L$. Since we are not interested in accurately locating an optimizer of (4.1) on the boundary, we use the following stopping criterion:

$$(4.3) \quad \|\nabla \widehat{g}^{(i)}(\gamma)\| \leq \tau_{sub} \quad \text{or} \quad \beta \epsilon_L \leq \frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} \leq \epsilon_L \quad (\tau_{sub} \geq 0, \beta \in (\delta, 1)),$$

where δ is defined in (4.2). This means that ETR terminates the i th optimization subproblem either when an accurate stationary point is located or when the current iterate is too close to the trust region boundary. In the latter case, we heuristically assume that it is difficult to further decrease the objective inside the trust region. Figure 4.1 shows four possible optimization paths for ETR.

We summarize the working flow of ETR in Algorithm 3.

ALGORITHM 3 (ETR).

1. *Initialization:* let $i = 0$, $\tau \geq \tau_{sub} \geq 0$, $\delta \in (0, 1)$, $\beta \in (\delta, 1)$, choose the initial point $\gamma^{(0)}$, $\kappa_s \in (0, 1)$, $\tau_{\nabla g} \in (0, 1)$. Specify a desired tolerance on the relative error bound at the optimizer τ_{opt} , compute $\epsilon_L = \frac{\tau_{opt}}{1 + \tau_{opt}}$.

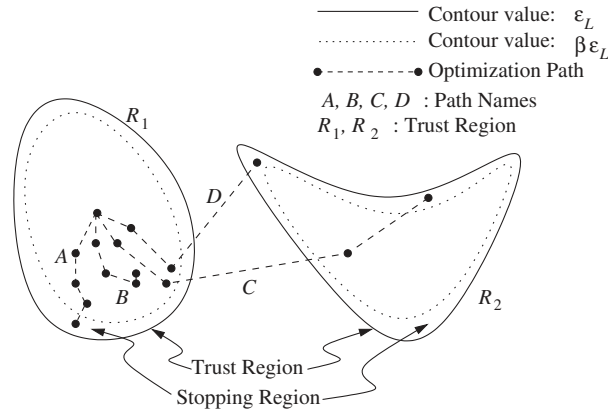


FIG. 4.1. Four possible optimization paths for ETR. Paths A, C, and D terminate when they reach a point too close to the boundary, while path B terminates when it reaches a stationary point. Paths C and D show that ETR may enter another trust region without noticing this. The contour lines shown are for the relative error bound.

2. Generate the i th surrogate model $\hat{g}^{(i)}$ satisfying $\frac{e^{(i)}(\gamma^{(i)})}{\hat{g}^{(i)}(\gamma^{(i)})} \leq \delta\epsilon_L$, $e_g^{(i)} < \tau$, and $\frac{e_g^{(i)}}{\|\nabla\hat{g}^{(i)}(\gamma^{(i)})\|} \leq \tau_{\nabla g}$. If $\|\nabla\hat{g}^{(i)}(\gamma^{(i)})\| + e_g^{(i)} \leq \tau$, return $\gamma^{(i)}$ and stop.
3. Minimize $\hat{g}^{(i)}(\gamma)$ with the stopping criterion (4.3); let the optimizer be $\gamma^{(i+1)}$.
4. Use Procedure 2 to determine how to continue.

The choice of ϵ_L follows from the fact that

$$\frac{|\hat{g}^{(k)}(\gamma^{(k)}) - g(\gamma^{(k)})|}{g(\gamma^{(k)})} = \frac{|\hat{g}^{(k)}(\gamma^{(k)}) - g(\gamma^{(k)})| \hat{g}^{(k)}(\gamma^{(k)})}{\hat{g}^{(k)}(\gamma^{(k)}) g(\gamma^{(k)})} < \frac{\epsilon_L}{1 - \epsilon_L} = \tau_{\text{opt}},$$

where we use $\frac{g(\gamma^{(k)})}{\hat{g}^{(k)}(\gamma^{(k)})} \geq 1 - \epsilon_L$, a formula that can be proved similar to (3.6). This relationship enables us to set τ_{opt} , the tolerance on the relative error bound at the optimizer a priori.

The convergence of ETR follows from Procedure 2. When Procedure 2 asks to reconstruct the optimization subproblem, we know that the current trust region is too large for $\hat{g}^{(i)}$. So if we do not to refine $\hat{g}^{(i)}$, we decrease ϵ_L below the relative error obtained by $\gamma^{(i+1)}$, since otherwise it is highly possible that we reach the same point.

4.2. EP method. Sometimes, the constraint in ETR is too restrictive since a γ value satisfying (3.25) is excluded from being exploited when $\frac{e^{(i)}(\gamma)}{\hat{g}^{(i)}(\gamma)} > \epsilon$. Since (3.25) is cheap to verify, we may further reduce the computation time if we accept such points. A straightforward approach is to minimize the left-hand side of (3.25), i.e., $b^{(i)}(\gamma) = \hat{g}^{(i)}(\gamma) + e^{(i)}(\gamma)$. But this leads to the following problems: (1) the penalty term $e^{(i)}(\gamma)$ can be nonconvex; (2) at the optimizer γ_{opt} , $\|\nabla e^{(i)}(\gamma_{\text{opt}})\|$ can be large even when $e^{(i)}(\gamma_{\text{opt}})$ is small; (3) locally, $\hat{b}^{(i)}$ is a worse approximation of g than $\hat{g}^{(i)}(\gamma^{(i)})$, especially when the error bound $e^{(i)}(\gamma)$ overestimates the true error; and (4) computing $\nabla e^{(i)}(\gamma)$ can be expensive, especially when the parameter space is of high dimension. Therefore, we introduce a weighting function for the penalty term. Let $w(\epsilon)$ be a differentiable and monotonically increasing function satisfying

$$(4.4) \quad w(\epsilon) = 0 \text{ when } \epsilon \leq \epsilon_L, \quad w(\epsilon) = 1, \text{ when } \epsilon \geq \epsilon_H \quad (\epsilon_L < \epsilon_H).$$

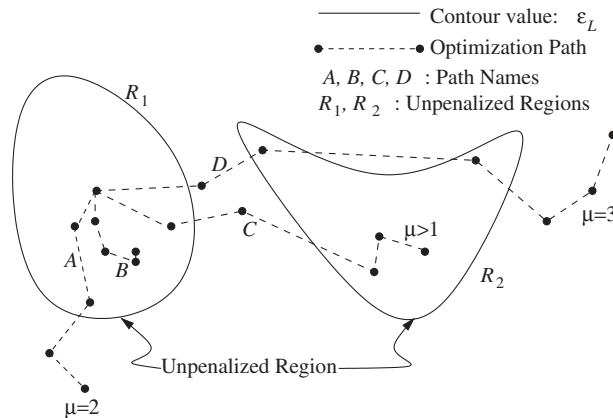


FIG. 4.2. Four possible optimization paths for EP. Paths B and C terminate at stationary points in the unpenalized region, while paths A and D terminate when they stay in the penalized region for μ successive steps. The contour lines shown are for the relative error bound.

See (6.1) for an example of $w(\epsilon)$. We define the i th optimization subproblem as

$$(4.5) \quad \min_{\gamma} \quad F^{(i)}(\gamma) = \widehat{g}^{(i)}(\gamma) + w\left(\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)}\right) e^{(i)}(\gamma).$$

So $F^{(i)}(\gamma)$ equals $\widehat{g}^{(i)}(\gamma)$ when the relative error estimator $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} \leq \epsilon_L$ and equals $b^{(i)}(\gamma)$ when $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} \geq \epsilon_H$. As in ETR, we require $\widehat{g}^{(i)}$ to satisfy (4.2). Therefore, if both $e^{(i)}(\gamma)$ and $\widehat{g}^{(i)}(\gamma)$ are positive continuous functions, there exists a neighborhood around $\gamma^{(i)}$, where $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} \leq \epsilon_L$ holds and $F^{(i)}(\gamma) = \widehat{g}^{(i)}(\gamma)$. We call this region the *unpenalized region* and call the region with nonzero penalty the *penalized region*.

Thanks to the weighting function w , (1) $F^{(i)}(\gamma)$ is convex around $\gamma^{(i)}$ as long as $\widehat{g}^{(i)}(\gamma)$ is and (2) we avoid computing $\nabla e^{(i)}(\gamma)$ within the unpenalized region. When we enter the penalized region, however, if we want to continue we need computing $\nabla F^{(i)}(\gamma)$, which requires $\nabla e^{(i)}(\gamma)$. It is possible that after a few iterations, we return to the unpenalized region since $\nabla F^{(i)}(\gamma)$ also takes $\nabla e^{(i)}(\gamma)$ into account for the penalized region. However, if we stay within the penalized region for too many—say, μ —successive steps, we prefer to terminate the current optimization subproblem since (1) computing $\nabla e^{(i)}(\gamma)$ is normally more expensive than computing $\widehat{g}^{(i)}(\gamma)$ and $e^{(i)}(\gamma)$, and (2) locating γ with $\nabla F^{(i)}(\gamma) = 0$ but $\nabla \widehat{g}^{(i)}(\gamma) \neq 0$ is beyond our true interest.

Therefore, we use the following stopping criterion for EP:

$$(4.6) \quad \|\nabla F^{(i)}(\gamma)\| \leq \tau_{sub} \quad (\tau_{sub} \geq 0) \quad \text{or} \quad \frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} > \epsilon_L \quad \text{holds for } \mu \text{ successive steps.}$$

Since μ strongly impacts the algorithm's behavior, we sometimes explicitly specify it in the algorithm acronym as EP(μ). Note that for $\mu = 1$, we never compute $\nabla e^{(i)}(\gamma)$, which is desirable if computing $\nabla e^{(i)}(\gamma)$ is expensive. Larger μ may save the total number of iterations, but it also results in higher computational cost per iteration. See section 6 for more discussions. Figure 4.2 shows four possible optimization paths for EP.

ALGORITHM 4 (EP).

1. *Initialization:* let $i = 0$, $\tau \geq \tau_{sub} \geq 0$, $\delta \in (0, 1)$, μ a positive integer, choose the initial point $\gamma^{(0)}$, $\kappa_s \in (0, 1)$, $\tau_{\nabla g} \in (0, 1)$. Specify a desired relative error bound of the optimizer τ_{opt} , compute $\epsilon_L = \frac{\tau_{opt}}{1+\tau_{opt}}$ and choose $\epsilon_H \in (\epsilon_L, 1)$.
2. Generate the i th surrogate model $\hat{g}^{(i)}$ satisfying $\frac{e_g^{(i)}(\gamma^{(i)})}{\hat{g}^{(i)}(\gamma^{(i)})} \leq \delta\epsilon_L$, $e_g^{(i)} < \tau$, and $\frac{e_g^{(i)}}{\|\nabla \hat{g}^{(i)}(\gamma^{(i)})\|} \leq \tau_{\nabla g}$. If $\|\nabla \hat{g}^{(i)}(\gamma^{(i)})\| + e_g^{(i)} \leq \tau$, return $\gamma^{(i)}$ and stop.
3. Minimize $F^{(i)}(\gamma)$ defined by (4.5) using the stopping criterion (4.6); let the optimizer be $\gamma^{(i+1)}$.
4. Use Procedure 2 to determine how to continue.

The convergence of EP follows from Procedure 2. When Procedure 2 asks to reconstruct the optimization subproblem, $\hat{g}^{(i)}$ is certain to be not accurate enough at $\gamma^{(i+1)}$, but it is not fully penalized there. So if we do not refine $\hat{g}^{(i)}$, we should decrease ϵ_H below the relative error at $\gamma^{(i+1)}$ to ensure that $\gamma^{(i+1)}$ will not be accepted again.

4.3. Implementation details for ETR and EP.

1. *Large-step backtracking.* In numerical tests, we divide the backtracking factor κ_{bt} by $(1 + 0.05 \frac{e^{(i)}(\gamma)}{\hat{g}^{(i)}(\gamma)})$ when $\frac{e^{(i)}(\gamma)}{\hat{g}^{(i)}(\gamma)} > \epsilon_L$. When $\frac{e^{(i)}(\gamma)}{\hat{g}^{(i)}(\gamma)} > \epsilon_L$, ETR uses backtracking in order to return to the trust region. It is reasonable to use a large backtracking step when $\frac{e^{(i)}(\gamma)}{\hat{g}^{(i)}(\gamma)}$ is too large to reduce the number of backtracking steps required. For EP, we can also use this technique to quickly escape the region with high error.

2. *Hessian approximation transfer (HAT).* For the initial Hessian approximation for the i th optimization subproblem $B_0^{(i)}$, it is common practice to take a multiple of the identity matrix, which we denote by h_0I . For ETR and EP, another possibility is HAT, i.e., using the Hessian approximation trained by the i th iteration as the initial Hessian approximation of the $(i + 1)$ st iteration. For EP, we use $\nabla \hat{g}^{(i)}$ rather than $\nabla F^{(i)}$ to train the Hessian approximation for HAT.

5. **Combining MOR with ETR and EP.** In this section, we first introduce Krylov–Padé-type MOR and then discuss how to combine MOR with ETR and EP.

Two-sided Krylov–Padé-type MOR methods generate two $n \times k$ ($k \leq n$, and typically $k \ll n$) matrices W_k and V_k with orthogonal columns, approximate the right state variable x as $V_k z$ ($z \in \mathbb{C}^k$), and force the residual $(\mathcal{L}(\omega, \gamma)V_k z - f)$ orthogonal to the range of W_k , i.e., $W_k^*(\mathcal{L}(\omega, \gamma)V_k z - f) = 0$, to obtain the reduced model

$$(5.1) \quad \begin{cases} \hat{\mathcal{L}}(\omega, \gamma)z = \hat{f}, \\ \hat{y} = \hat{\ell}^*z, \end{cases}$$

where $\hat{\mathcal{L}}(\omega, \gamma) = W_k^* \mathcal{L}(\omega, \gamma) V_k$, $\hat{f} = W_k^* f$, and $\hat{\ell} = V_k^* \ell$. The left state vector of system (5.1) is $\zeta(\omega, \gamma) \triangleq \hat{\mathcal{L}}(\omega, \gamma)^{-*} \hat{\ell}$, from which the left state vector of system (2.1) ξ is approximated as $\xi \approx W_k \zeta$.

The many different Krylov–Padé-type MOR methods use different strategies in generating W_k and V_k , but they all satisfy the *moment matching properties*. In this paper, we concentrate on the SOAR (second-order Arnoldi) [25, 6, 24] method.

5.1. **SOAR and the MOR framework.** In the MOR framework [27], when the optimization algorithm needs the function value and gradient at $\gamma^{(i)}$, we fix $\gamma = \gamma^{(i)}$ and generate a reduced model for system (2.5). To reduce the second-order system, we resort to the two-sided SOAR method to generate W_k and V_k , from which we derive the following reduced model:

$$(5.2) \quad \begin{cases} (\widehat{K} + i(\omega - \omega^{(i)})\widehat{C} - (\omega - \omega^{(i)})^2\widehat{M})z(\omega - \omega^{(i)}, \gamma^{(i)}) = \widehat{f}, \\ \widehat{y}(\omega - \omega^{(i)}, \gamma^{(i)}) = \widehat{\ell}^*z(\omega - \omega^{(i)}, \gamma^{(i)}), \end{cases}$$

where $\widehat{K} = W^*(K(\gamma^{(i)}) + i\omega^{(i)}C(\gamma^{(i)}) - (\omega^{(i)})^2M(\gamma^{(i)}))V$, $\widehat{C} = W^*(C(\gamma^{(i)}) - 2\omega^{(i)}M(\gamma^{(i)}))V$, $\widehat{M} = W^*M(\gamma^{(i)})V$, $\widehat{f} = W_k^*f$, and $\widehat{\ell} = V_k^*\ell$. The computationally dominant part of SOAR is the LU factorization of the large-scale sparse matrix $(K(\gamma^{(i)}) + i\omega^{(i)}C(\gamma^{(i)}) - (\omega^{(i)})^2M(\gamma^{(i)}))$. See [27] for more details.

No design parameter is preserved in (5.2). However, if we use the bases W_k and V_k for the general system (2.1) as we did in (5.1), we obtain a parametric reduced model, which can be seen as an interpolatory reduced model using a single interpolation point $\gamma^{(i)}$. At the interpolation point $\gamma^{(i)}$, an important relationship between the original model (2.5) and the SOAR reduced model (5.1) is the *moment matching property*. Let the Taylor expansion of $y(\omega, \gamma^{(i)})$ and $\widehat{y}^{(i)}(\omega, \gamma^{(i)})$ around $\omega^{(i)}$ be

$$y(\omega, \gamma^{(i)}) \triangleq \sum_{j=0}^{\infty} m(j|\omega^{(i)}, \gamma^{(i)}) (\omega - \omega^{(i)})^j, \quad \widehat{y}^{(i)}(\omega, \gamma^{(i)}) \triangleq \sum_{j=0}^{\infty} \widehat{m}(j|\omega^{(i)}, \gamma^{(i)}) (\omega - \omega^{(i)})^j,$$

where we call $m(j|\omega^{(i)}, \gamma^{(i)})$ and $\widehat{m}(j|\omega^{(i)}, \gamma^{(i)})$ the j th moments of y and \widehat{y} with the interpolation point $(\omega^{(i)}, \gamma^{(i)})$, respectively. We say the j th moments of y and \widehat{y} with the interpolation point $(\omega^{(i)}, \gamma^{(i)})$ match if $m(j|\omega^{(i)}, \gamma^{(i)}) = \widehat{m}(j|\omega^{(i)}, \gamma^{(i)})$. Similarly we can define the moments of $\frac{\partial y}{\partial \omega}$, $\frac{\partial \widehat{y}}{\partial \omega}$, $\frac{\partial y}{\partial \gamma_q}$, and $\frac{\partial \widehat{y}}{\partial \gamma_q}$ ($1 \leq q \leq l$). Theorem 5.1 [27] summarizes the moment matching properties of SOAR.

THEOREM 5.1. *Assume that SOAR generates W_k and V_k with the interpolation point $(\omega^{(i)}, \gamma^{(i)})$ and $(K(\gamma^{(i)}) + i\omega^{(i)}C(\gamma^{(i)}) - (\omega^{(i)})^2M(\gamma^{(i)}))$ is nonsingular. Then for the order k generalized SOAR reduced model (5.1) and the order n original model (2.1), we have the following moment matching properties:*

1. *The first $2k$ moments of $y(\omega, \gamma^{(i)})$ and $\widehat{y}^{(i)}(\omega, \gamma^{(i)})$ match.*
2. *The first $2k - 1$ moments of $\frac{\partial y(\omega, \gamma)}{\partial \omega} \Big|_{\gamma=\gamma^{(i)}}$ and $\frac{\partial \widehat{y}^{(i)}(\omega, \gamma)}{\partial \omega} \Big|_{\gamma=\gamma^{(i)}}$ match.*
3. *The first k moments of $\frac{\partial y(\omega, \gamma)}{\partial \gamma_q} \Big|_{\gamma=\gamma^{(i)}}$ and $\frac{\partial \widehat{y}^{(i)}(\omega, \gamma)}{\partial \gamma_q} \Big|_{\gamma=\gamma^{(i)}}$ ($1 \leq q \leq l$) match.*

All moments here are with the interpolation point $(\omega^{(i)}, \gamma^{(i)})$.

According to Theorem 5.1, we see that $\widehat{y}^{(i)}(\omega^{(i)}, \gamma^{(i)}) = y(\omega^{(i)}, \gamma^{(i)})$ and also expect $\widehat{y}^{(i)}(\omega, \gamma^{(i)})$ to approximate $y(\omega, \gamma^{(i)})$ well for frequency values around $\omega^{(i)}$. Therefore, we prefer a frequency shift $\omega^{(i)}$ within the frequency range of interest. In our numerical tests, we always take $\omega^{(i)} = \frac{\omega_L + \omega_H}{2}$.

5.2. An approximate error bound. Finding a tight and computationally cheap error bound for (3.4) in the relaxed first-order condition (Condition 2) is normally challenging. For a MOR reduced model that is valid only for a specific parameter value, error bounds are developed in [7, 18]. However, for an interpolatory reduced model defined on the entire parameter space, no error bound exists, to the knowledge of the authors. Therefore, we introduce a heuristic error bound to make ETR and EP applicable. Define the residual of the right and left state variables as

$$(5.3) \quad r_r^{(i)}(\omega, \gamma) = \mathcal{L}(\omega, \gamma)V_k^{(i)}z^{(i)}(\omega, \gamma) - f, \quad r_l^{(i)}(\omega, \gamma) = \mathcal{L}^*(\omega, \gamma)W_k^{(i)}\zeta^{(i)}(\omega, \gamma) - \ell.$$

An approximate error bound for $\widehat{y}^{(i)}$ can be derived as follows:

$$\begin{aligned} |\widehat{y}^{(i)} - y^{(i)}| &= \left| \ell^* \mathcal{L}^{-1}(\mathcal{L}V_k^{(i)}z^{(i)} - f) \right| = \left| \ell^* \mathcal{L}(\omega, \gamma)^{-1} r_r^{(i)} \right| \leq \|\ell^* \mathcal{L}(\omega, \gamma)^{-1}\| \|r_r^{(i)}\| \\ &= \|\mathcal{L}(\omega, \gamma)^{-*} \ell\| \|r_r^{(i)}\| \approx \|\widehat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \widehat{\ell}^{(i)}\| \|r_r^{(i)}\| \triangleq e_y^{(i)}. \end{aligned}$$

Then we can deduce an approximate upper bound of $|y|^2$:

$$\begin{aligned} |y|^2 &= |\hat{y}^{(i)} + (y - \hat{y}^{(i)})|^2 \leq (|\hat{y}^{(i)}| + |y - \hat{y}^{(i)}|)^2 \\ &= |\hat{y}^{(i)}|^2 + 2|\hat{y}^{(i)}||y - \hat{y}^{(i)}| + |y - \hat{y}^{(i)}|^2 \approx |\hat{y}^{(i)}|^2 + 2|\hat{y}^{(i)}|e_y^{(i)} + e_y^{(i)2} \triangleq b_y^{(i)}, \end{aligned}$$

based on which we define the surrogate model and its heuristic error bound

$$\hat{g}^{(i)}(\gamma) = \int_{\omega_L}^{\omega_H} |\hat{y}^{(i)}|^2 d\omega, \quad e^{(i)}(\gamma) = \int_{\omega_L}^{\omega_H} 2|\hat{y}^{(i)}|e_y^{(i)} + e_y^{(i)2} d\omega.$$

The formula for computing $\frac{\partial \hat{g}^{(i)}(\gamma)}{\partial \gamma_j}$ is similar to (2.7). To compute $\frac{\partial e^{(i)}(\gamma)}{\partial \gamma_j}$, we deduce

$$\begin{aligned} \frac{\partial \hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \hat{\ell}^{(i)}}{\partial \gamma_j} &= -\hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \frac{\partial \hat{\mathcal{L}}^{(i)}(\omega, \gamma)^*}{\partial \gamma_j} \hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \hat{\ell}^{(i)}, \\ \frac{\partial r_r^{(i)}(\omega, \gamma)}{\partial \gamma_j} &= \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} V_k^{(i)} \hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-1} \hat{f}^{(i)} \\ &\quad - \mathcal{L}(\omega, \gamma) V_k^{(i)} \hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-1} \frac{\partial \hat{\mathcal{L}}^{(i)}(\omega, \gamma)}{\partial \gamma_j} \hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-1} \hat{f}^{(i)}, \\ \frac{\partial \|\hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \hat{\ell}^{(i)}\|}{\partial \gamma_j} &= \frac{1}{\|\hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \hat{\ell}^{(i)}\|} \Re \left\{ \left(\hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \hat{\ell}^{(i)} \right)^* \frac{\partial \hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \hat{\ell}^{(i)}}{\partial \gamma_j} \right\}, \\ \frac{\partial \|r_r^{(i)}(\omega, \gamma)\|}{\partial \gamma_j} &= \frac{1}{\|r_r^{(i)}(\omega, \gamma)\|} \Re \left\{ \left(r_r^{(i)}(\omega, \gamma) \right)^* \frac{\partial r_r^{(i)}(\omega, \gamma)}{\partial \gamma_j} \right\}, \\ \frac{\partial e_y^{(i)}}{\partial \gamma_j} &= \frac{\partial \|\hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \hat{\ell}^{(i)}\|}{\partial \gamma_j} \|r_r^{(i)}(\omega, \gamma)\| + \|\hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \hat{\ell}^{(i)}\| \frac{\partial \|r_r^{(i)}(\omega, \gamma)\|}{\partial \gamma_j}. \end{aligned}$$

Then we derive

$$\frac{\partial e^{(i)}(\gamma)}{\partial \gamma_j} = 2 \int_{\omega_L}^{\omega_H} \frac{\partial |\hat{y}^{(i)}|}{\partial \gamma_j} e_y^{(i)} + (|\hat{y}^{(i)}| + e_y^{(i)}) \frac{\partial e_y^{(i)}}{\partial \gamma_j}.$$

To obtain $e_g^{(i)}$, which is an upper bound of $\|\nabla \hat{g}^{(i)}(\gamma^{(i)}) - \nabla g(\gamma^{(i)})\|$, we derive

$$\begin{aligned} &\left| \frac{\partial \hat{y}}{\partial \gamma_j} - \frac{\partial y}{\partial \gamma_j} \right| \\ &= \left| \hat{\ell}^* \hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-1} \frac{\partial \hat{\mathcal{L}}^{(i)}(\omega, \gamma)}{\partial \gamma_j} \hat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-1} \hat{f} - \ell^* \mathcal{L}^{(i)}(\omega, \gamma)^{-1} \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} \mathcal{L}^{(i)}(\omega, \gamma)^{-1} f \right| \\ &= \left| (W_k \zeta)^* \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} V_k z - \xi^* \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} x \right| \\ &= \left| (W_k \zeta)^* \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} V_k z - (W_k \zeta)^* \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} x + (W_k \zeta)^* \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} x - \xi^* \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} x \right| \\ &\leq \left| (W_k \zeta)^* \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} \mathcal{L}(\omega, \gamma)^{-1} r_r^{(i)} \right| + \left| r_l^{(i)*} \mathcal{L}(\omega, \gamma)^{-1} \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} x \right| \\ &\leq \left\| \mathcal{L}(\omega, \gamma)^{-*} \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} W_k \zeta \right\| \left\| r_r^{(i)} \right\| + \left\| r_l^{(i)} \right\| \left\| \mathcal{L}(\omega, \gamma)^{-1} \frac{\partial \mathcal{L}(\omega, \gamma)}{\partial \gamma_j} x \right\| \end{aligned}$$

$$\approx \left\| \widehat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-*} \frac{\partial \widehat{\mathcal{L}}^{(i)}(\omega, \gamma)^*}{\partial \gamma_j} \zeta \left\| \left\| r_r^{(i)} \right\| + \left\| r_l^{(i)} \right\| \right\| \left\| \widehat{\mathcal{L}}^{(i)}(\omega, \gamma)^{-1} \frac{\partial \widehat{\mathcal{L}}^{(i)}(\omega, \gamma)}{\partial \gamma_j} z \right\|$$

$$\triangleq b_{grad_j}(\omega, \gamma).$$

Then we define $e_g^{(i)} \triangleq \sqrt{\sum_{j=1}^l \left(\int_{\omega_L}^{\omega_H} b_{grad_j}(\omega, \gamma^{(j)}) \right)^2}$.

The computations of $\widehat{g}^{(i)}(\gamma)$ and $\nabla \widehat{g}^{(i)}(\gamma)$ are cheap since only the reduced order model is involved. For $e^{(i)}(\gamma)$, however, the computation involves large-scale matrix-vector multiplications for all ω 's accessed by the numerical integrator. The computations for $\nabla e^{(i)}(\gamma)$ and $e_g^{(i)}$ are even more expensive since the above computations are needed for all l dimensions. Therefore, we use a fine grid for integrating $\widehat{g}^{(i)}(\gamma)$ and $\nabla \widehat{g}^{(i)}(\gamma)$ since their accuracy is important for optimization. For $e^{(i)}(\gamma)$, $\nabla e^{(i)}(\gamma)$ and $e_g^{(i)}$, however, we use a coarse grid to reduce the computational cost because as the error bound is only heuristic and normally overestimated, we do not need to compute it with full accuracy. To reuse the small-scale dense LU factorizations, the coarse grid points are chosen among the fine grid points.

5.2.1. An illustration: Applying the error bound to the floor damper optimization problem. Using the heuristic error bound developed above, we plot the contour lines of the relative error estimator $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)}$ for the floor damper optimization problem in Figure 5.1. Figure 5.1 shows that (1) the reduced model approximates the original model well in a large domain and (2) the contour lines for the relative error estimator $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)}$ can be convex, nonconvex, or even disconnected.

5.3. Combining SOAR with ETR and EP. SOAR reduced models serve as suitable surrogate models for ETR and EP because they satisfy the relaxed first-order condition (Condition 2) and Assumption 1:

For C1. See section 5.2 for the error bounds.

For C2. If the reduced model is not accurate enough, we can increase k and compute more Krylov vectors to enlarge subspaces W_k and V_k . This leads to more accurate approximations for g and ∇g due to moment matching properties. According to Theo-

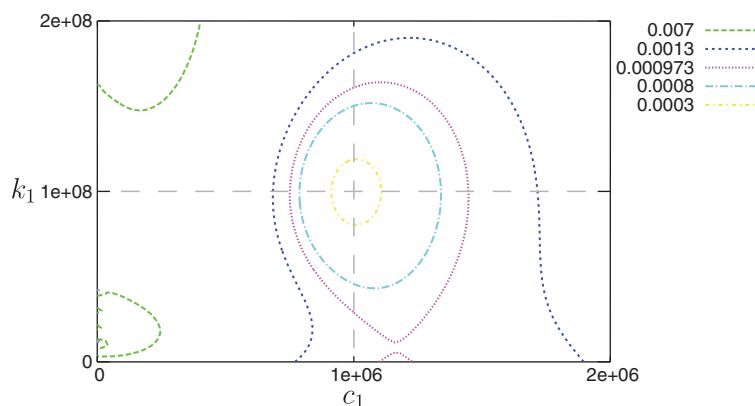


FIG. 5.1. Contour plot of the relative error estimator for the floor damper optimization problem. For this plot, we used a discretized model of dimension 280 as the original model and a SOAR reduced model of dimension 7 interpolated at $(k_1, c_1) = (10^8, 10^6)$ as the surrogate model. We used a coarse mesh for discretization in order to generate an accurate plot in a short time.

rem 5.1, $\widehat{g}^{(i)}(\gamma^{(i)})$ and $\nabla\widehat{g}^{(i)}(\gamma^{(i)})$ are good approximations of $g(\gamma^{(i)})$ and $\nabla g(\gamma)|_{\gamma=\gamma^{(i)}}$ with errors of orders $\int_{\omega_L}^{\omega_H} \mathcal{O}((\omega - \omega^{(i)})^{2k}) d\omega$ and $\int_{\omega_L}^{\omega_H} \mathcal{O}((\omega - \omega^{(i)})^k) d\omega$, respectively. With $k = n$ (assume W_k and V_k are of full rank), we obtain an exact model.

For Assumption 1. Computing $g(\gamma)$ and $\nabla g(\gamma)$ requires LU factorizations of the large-scale sparse matrix $\mathcal{L}(\omega, \gamma)$ at all ω values accessed by the numerical integrator; building $\widehat{g}^{(i)}$ requires only one LU factorization of $\mathcal{L}(\omega, \gamma)$ at the interpolation point $(\omega^{(i)}, \gamma^{(i)})$; evaluating $\widehat{g}^{(i)}(\gamma)$ requires no additional factorization of $\mathcal{L}(\omega, \gamma)$.

6. Numerical results. In this section, we apply ETR and EP to two design optimization problems arising from structures and vibrations.

All our codes were implemented with the C++ package GLAS [19]. We solved large-scale sparse linear equations with MUMPS [2] and small-scale dense linear equations with LAPACK [3]. All numerical tests were run using a DELL Latitude E6400 with Intel Core 2 Duo 2.66 GHz CPU and 4 GB of main memory.

6.1. Choice of parameters for ETR and EP in the MOR scenario. Recall the parameters listed in Table 3.1 for solving the subproblem. Here, we discuss the other parameters required for ETR, EP, and SOAR.

1. For the order of the reduced model k , we do not have an automatic procedure to determine it a priori. Normally, we rely on experience to choose k . Fortunately, even when k is chosen too small, i.e., when the reduced model is not accurate enough, we can refine it by enlarging k . This strategy is our implementation of S2 for Krylov–Padé-type reduced models. Actually, we recommend use of a relatively large k , since in that way we expect to take larger steps and use less reduced models for optimization and to reduce the possibility that we need to increase k . In our numerical tests, we never needed to increase k .

2. It is preferable not to use too-small ϵ_L and ϵ_H since the heuristic error bound proposed is not very tight. Using a too-small ϵ_L will result in too-small steps and slow down the convergence. We experienced in numerical tests that the true relative error at the optimizer is usually much lower than ϵ_L . For all our numerical tests, we chose $\epsilon_L = 0.1$, $\epsilon_H = 0.2$, and the weighting function $w(\epsilon)$ defined in (4.4) as

$$(6.1) \quad w(\epsilon) = \begin{cases} 0 & \text{when } \epsilon \leq 0.1, \\ 2(10\epsilon - 1)^2 & \text{when } 0.1 < \epsilon \leq 0.15, \\ 1 - 2(2 - 10\epsilon)^2 & \text{when } 0.15 < \epsilon \leq 0.2, \\ 1 & \text{when } \epsilon > 0.2. \end{cases}$$

3. We choose δ to be sufficiently smaller than 1 to prevent the trust/unpenalized region from being too small. For β , however, we set it just slightly below 1 to fully exploit the surrogate model. In our numerical tests, we used $\delta = 0.1$ and $\beta = 0.95$.

4. The initial Hessian scaling h_0 is chosen based on the expected step length and the order of the gradient. When the largest eigenvalue of the Hessian approximation is larger than $\lambda_\tau^{(\max)}$, we reset it to $h_0 I$. Numerical results show that this happens more often when we use HAT. For all numerical tests, we set $h_0 = 10^{-5}$ and $\lambda_\tau^{(\max)} = 1$. Note that when a quasi-Newton method is used, κ_n and Φ (Table 3.1) depend on h_0 and therefore need not be set.

6.2. Problem I: Floor damper optimization. See section 2.1 for the problem descriptions. The floor is $10 \times 10 \times 0.3$ m in size. Its Young’s modulus, Poisson’s ratio, and density are 30 GPa, 0.3, and 2500 kg/m^3 , respectively. We used a 100×100 uniform mesh for the floor and obtained a model with 29,799 degrees of

TABLE 6.1

Numerical results of the floor damper optimization problem. In this table, “Order” represents the dimension of the system matrix; “Optimum” is the objective value of the located optimizer computed via the original model g ; the “Relative Error” shown is computed as $|g(\gamma_{\text{opt}}) - \hat{g}(\gamma_{\text{opt}})|/g(\gamma_{\text{opt}})$, where \hat{g} is the last reduced model generated and γ_{opt} is the located optimizer; N_i and N_m represent the total number of subiterations and the total number of reduced models needed, respectively.

	Order	Optimum	CPU time	Relative error	N_i	N_m
ETR	7	1.316206122e+10	220 s	2.626620741e-09	69	20
	10	1.316206122e+10	35 s	2.424547354e-11	14	3
ETR + HAT	7	1.316206122e+10	192 s	2.571925506e-09	61	20
	10	1.316206122e+10	37 s	4.696182922e-11	15	3
EP(1)	7	1.316206119e+10	129 s	2.616067046e-09	32	16
	10	1.316206122e+10	41 s	1.729372585e-11	10	4
EP(1) + HAT	7	1.316206119e+10	141 s	2.594100758e-09	35	17
	10	1.316206122e+10	40 s	3.049540945e-11	9	4
EP(2)	7	1.316206122e+10	169 s	2.585171968e-09	48	17
	10	1.316206122e+10	45 s	3.963751122e-11	14	4
EP(2) + HAT	7	1.316206122e+10	194 s	2.593526614e-09	103	20
	10	1.316206122e+10	44 s	9.38070038e-11	13	4
EP(μ) ($\mu \geq 8$)	10	1.316206123e+10	58 s	7.083806541e-11	20	4
ETR & EP(μ) ($\mu \geq 1$)	12	1.316206122e+10	25 s	3.87328218e-11	9	2
MOR framework [27]	7	1.316206122e+10	89 s	2.6865095e-09	9	9
Direct method	29800	1.316206122e+10	7202 s	—	9	0

freedom (DOFs). The damper is 3750 kg in weight, and it is described by a classical stiffness-damping-mass model. The design parameters are its stiffness k_1 and damping coefficient c_1 . The frequency range of interest is $[2 \text{ rad/s}, 200 \text{ rad/s}]$. The feasible domain is $[1 \text{ N/m}, 10^{10} \text{ N/m}] \times [1 \text{ Ns/m}, 10^{10} \text{ Ns/m}]$ and the initial point is $[k_1^{(0)}, c_1^{(0)}] = [10^7 \text{ N/m}, 10^7 \text{ Ns/m}]$. In numerical integration, we used 151 interpolation frequencies for $\hat{g}^{(i)}(\gamma)$ and $\nabla \hat{g}^{(i)}(\gamma)$ and 11 interpolation frequencies for $e^{(i)}(\gamma)$, $\nabla e^{(i)}(\gamma)$, and $e_g^{(i)}$. Since K_0 in (2.6) is singular, we first shift the interpolating frequency to $\omega^{(i)} = \frac{\omega_L + \omega_H}{2}$ to make MOR applicable.

Now we analyze the numerical results shown in Table 6.1.

1. Most methods listed take different paths, but all of them converge to the same optimizer. This shows the effectiveness of the optimization algorithms used.

2. When the reduced models are of size 7, EP(1) is more efficient than ETR. In this case, the reduced models are not so accurate and the trust regions are small, which means that ETR can take only relatively small steps. From Figure 6.1(a), we can see that for each reduced model, ETR first takes relatively large steps, but the steps become smaller when ETR approaches the boundary of the trust region. For EP(1), however, Figure 6.1(b) shows that this phenomenon is not so severe as in ETR.

3. When the reduced models are of size 10, ETR is slightly more efficient than EP(1). In this case, the reduced models are more accurate. Comparing Figure 6.1(c) with Figure 6.1(d), we see that EP(1) still converges faster, but since the unpenalized region of the second reduced model is by chance relatively small, it needs one more reduced model than ETR.

4. Comparing Figure 6.1(a) and Figure 6.1(b) with Figure 6.1(e) and Figure 6.1(f), we see HAT leads to smoother convergence paths, but not necessarily better performance.

5. EP with $\mu \geq 8$ locates an accurate optimizer for each optimization subproblem. However, this leads to slower convergence (larger N_i) and more CPU time since locating a stationary point outside the unpenalized region is beyond our true interest.

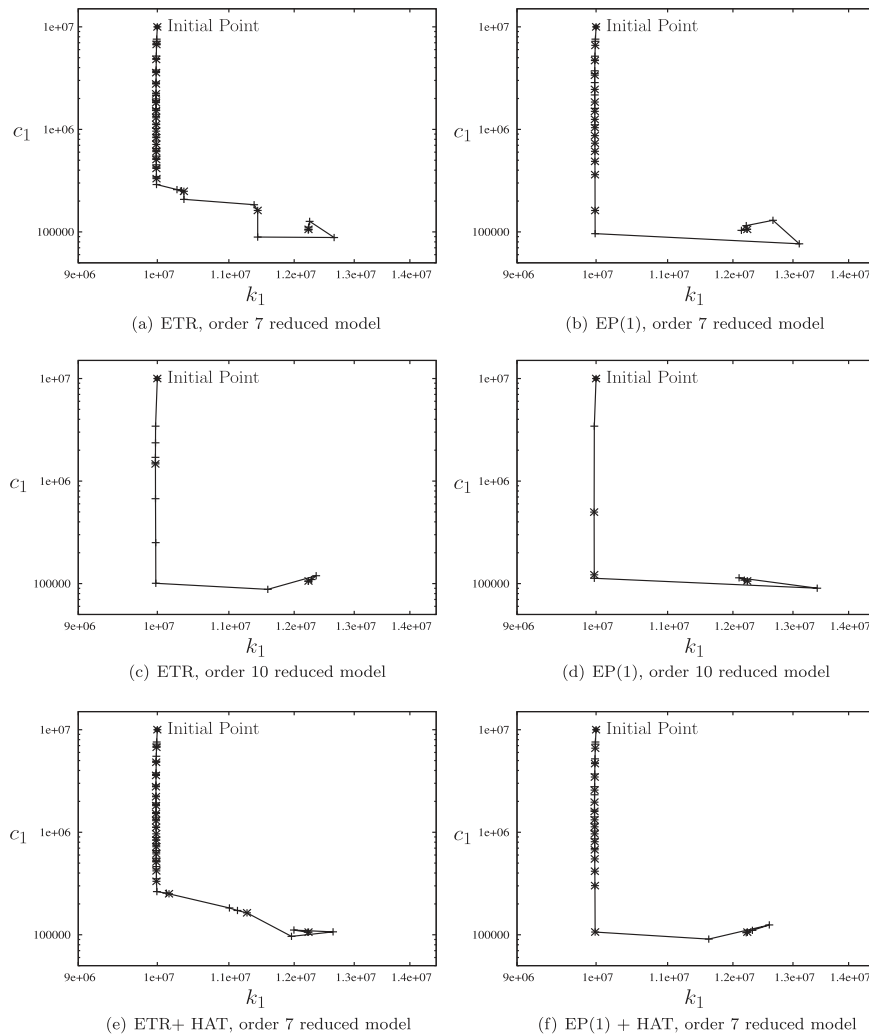


FIG. 6.1. Convergence paths for several cases in the floor damper optimization problem. Iterates of all optimization subproblems are marked with + and the interpolation points for the surrogate models are marked with *.

6. When the reduced models are of size 12, ETR and EP turn out to be the same since the reduced models are very accurate: only one reduced model is required for locating the optimizer and another reduced model is used to check its gradient. Further increasing the order of the reduced models would lead to longer CPU time, since both the generation and the evaluation of reduced models become more expensive.

7. For comparison, we also list the numerical results of the MOR framework and the direct method, which uses sparse solvers for all points accessed by the integrator. The MOR framework [27] lacks error control but costs more time than ETR and EP with the reduced model of size 10 or 12. The direct method is very time-consuming.

6.3. Problem II: Footbridge damper optimization. In this problem, we optimize the stiffnesses and damping coefficients of four TMDs to reduce the vibration of a footbridge located over the Dijle river in Mechelen (Belgium). It is about 31.354 m

in length and the four TMDs are located at nodes at 11.299 m, 19.314 m, 10.549 m, and 20.309 m, respectively, each of which is 40.72 kg in weight.

The discretized model describing the above dynamic system is

$$(6.2) \quad \begin{cases} \left(K_0 + i\omega C_0 + \sum_{i=1}^4 (k_i + i\omega c_i) K_i - \omega^2 M_0 \right) x = f, \\ y = \ell^* x, \end{cases}$$

where K_0 and M_0 are obtained from a finite element model with 25,962 DOFs, $C_0 = 0.1003M_0 + 0.0001591K_0$ represents Rayleigh damping, K_i is a matrix with four nonzero entries that represents the interaction between the i th TMD and the footbridge, the input vector f represents a unit excitation at the center span, and the output vector ℓ picks out the vibration at the center span. To avoid numerical difficulties, we set $\ell = 10^6 \bar{\ell}$. In computing the objective g in (2.2), we use $\phi(\omega) = 1$. In numerical integration, we used 151 interpolation frequencies in computing $\hat{g}^{(i)}(\gamma)$ and $\nabla g^{(i)}(\gamma)$, and 11 interpolation frequencies in computing $e^{(i)}(\gamma)$, $\nabla e^{(i)}(\gamma)$, and $e_g^{(i)}$. Our design objective is to locate the optimal stiffness k_i and damping coefficient c_i for each damper in the sense of minimizing the energy norm of y over $[\omega_L, \omega_H] = [0 \text{ rad/s}, 10\pi \text{ rad/s}]$.

To get the relative error at the optimizer γ_{opt} , we used direct solves of $y(\omega, \gamma_{\text{opt}})$ to compute $g(\gamma_{\text{opt}})$. A single evaluation of g without using MOR costs about 540 s.

From Table 6.2, we deduce similar conclusions as in section 6.2 except for a few cases:

1. EP(1) appears to be much slower than ETR when $k = 18$ and HAT is not used. The reason may be that EP(1) unluckily follows a path that converges slowly.
2. HAT works better for this problem. A possible explanation is that it takes more iterations to train an adequate Hessian approximation when we optimize on more design parameters since each damped BFGS iteration adds only a rank two change to the Hessian approximation.
3. EP(2) seems to work better for this problem. We observed that in this problem, the probability that EP(2) returns from the penalized region to the unpenalized region is much higher, which reduces the number of reduced models needed.

TABLE 6.2
Numerical results of the footbridge damper optimization problem.

	Order	Optimum	CPU time	Relative error	Ni	Nm
ETR	18	24.77751668	705 s	1.313402699e-12	116	16
	20	24.78594112	205 s	4.465054737e-12	75	3
ETR + HAT	18	24.7775165	588 s	8.806680611e-13	82	17
	20	24.7775165	200 s	3.70147281e-12	58	5
EP(1)	18	24.77751694	1245 s	1.033587692e-11	148	25
	20	24.7762798	295 s	3.730479306e-12	96	3
EP(1) + HAT	18	24.7775165	481 s	3.056098836e-12	67	19
	20	24.7775165	304 s	8.247767536e-12	65	4
EP(2)	18	24.77751799	542 s	7.982648976e-12	80	11
	20	24.7775166	190 s	7.944365742e-12	63	3
EP(2)+ HAT	18	24.7812096	985 s	5.484359479e-12	123	17
	20	24.77627985	189 s	3.11547216e-12	67	3
ETR & EP(μ) ($\mu \geq 1$)	32	24.80397266	194 s	1.092742827e-11	82	2
MOR framework [27]	12	24.77751651	879 s	2.023983216e-07	70	70

6.4. Remarks on ETR and EP. For reduced models with relatively low accuracy, EP(1) outperforms ETR since larger steps are allowed. For reduced models with higher accuracy, ETR can outperform EP(1) since it may require fewer reduced models than EP(1). Whether EP with $\mu > 1$ performs well is problem dependent, but at least for the cases where the gradient of the error bound is too expensive to compute, we do not recommend this method.

7. Conclusions and further discussions. This paper presented two methods, ETR and EP, to fully exploit surrogate models equipped with error bounds in optimization. They are of particular interest to surrogate models that are expensive to build, but with function values and an error bound cheaply available. Under the relaxed first-order condition, both methods are proved to converge to a stationary point and allow for an approximation tolerance specified a priori. As an example, we developed a heuristic error bound for Krylov–Padé-type reduced models and used them as surrogate models for ETR and EP. Numerical results show that ETR and EP are less expensive than the MOR framework [27]. They are also more robust than the MOR framework since they take the approximation error into account.

Acknowledgments. The authors would like to thank the following industrial partners for their important contributions to the study of the footbridge: Waterwegen en Zeekanaal NV, Emotec/Emergo-Group, Herbosch-Kiere, and the Flemish Government, Department of Mobility and Public Works Division Steel Structures.

REFERENCES

- [1] N. M. ALEXANDROV, J. E. DENNIS, R. M. LEWIS, AND V. TORCZON, *A trust-region framework for managing the use of approximation models in optimization*, Struct. Multidisciplinary Optim., 15 (1998), pp. 16–23.
- [2] P. R. AMESTOY, I. S. DUFF, J. KOSTER, AND J.-Y. L’EXCELLENT, *A fully asynchronous multi-frontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41.
- [3] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users’ Guide*, 3rd ed., SIAM, Philadelphia, 1999.
- [4] A. C. ANTOULAS AND D. C. SORENSEN, *Approximation of large-scale dynamical systems: An overview*, Int. J. Appl. Math. Comput. Sci., 11 (2001), pp. 1093–1121.
- [5] E. ARIAN, M. FAHL, AND E. W. SACHS, *Managing POD models by optimization methods*, in Proceedings of the 41st IEEE Conference on Decision and Control, vol. 3, 2002, pp. 3300–3305.
- [6] Z. BAI AND Y. SU, *Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method*, SIAM J. Sci. Comput., 26 (2005), pp. 1692–1709.
- [7] Z. BAI AND Q. YE, *Error estimation of the Padé approximation of transfer functions via the Lanczos process*, Electron. Trans. Numer. Anal., 7 (1998), pp. 1–17.
- [8] U. BAUR, C. BEATTIE, P. BENNER, AND S. GUGERCIN, *Interpolatory projection methods for parameterized model reduction*, SIAM J. Sci. Comput., 33 (2011), pp. 2489–2518.
- [9] C. R. CALLADINE, *Theory of Shell Structures*, Cambridge University Press, New York, 1983.
- [10] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Trust Region Methods*, MPS/SIAM Ser. Optim., SIAM, Philadelphia, 2000.
- [11] P. J. DAVIS AND P. RABINOWITZ, *Methods of Numerical Integration*, 2nd ed., Dover Publications, Mineola, New York, 2007.
- [12] M. FAHL AND E. W. SACHS, *Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition*, in Large-Scale PDE-Constrained Optimization, L. T. Biegler, M. Heinkenschloss, O. Ghattas, and B. Bloemen Waanders, eds., Lecture Notes in Comput. Sci. Engrg. 30, Springer, Berlin, 2003, pp. 268–280.
- [13] P. FELDMAN AND R. W. FREUND, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, IEEE Trans. Computer-Aided Design, 14 (1995), pp. 639–649.

- [14] J. S. HAN, E. B. RUDNYI, AND J. G. KORVINK, *Efficient optimization of transient dynamic problems in MEMS devices using model order reduction*, J. Micromechanics Microengineering, 15 (2005), pp. 822–832.
- [15] Y.-T. LI, Z. BAI, Y. SU, AND X. ZENG, *Model order reduction of parameterized interconnect networks via a two-directional Arnoldi process*, IEEE Trans. Computer-Aided Design Integrated Circuits Systems, 27 (2008), pp. 1571–1582.
- [16] Y.-T. LI, Z. BAI, AND Y. SU, *A two-directional Arnoldi process and its application to parametric model order reduction*, J. Comput. Appl. Math., 226 (2009), pp. 10–21.
- [17] M. MALHOTRA AND P. M. PINSKY, *Efficient computation of multi-frequency far-field solutions of the Helmholtz equation using Padé approximations*, J. Comput. Acoustics, 8 (2000), pp. 223–240.
- [18] K. MEERBERGEN, *The solution of parametrized symmetric linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1038–1059.
- [19] K. MEERBERGEN, GLAS (Generic Linear Algebra Software), <http://people.cs.kuleuven.be/~karl.meerbergen/glas>.
- [20] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.
- [21] A. ODABASIOGLU, M. CELIK, AND L. T. PILEGGI, *PRIMA: Passive reduced-order interconnect macromodeling algorithm*, in ICCAD '97: Proceedings of the 1997 IEEE/ACM International Conference on Computer-Aided Design, Washington, DC, IEEE Computer Society, 1997, pp. 58–65.
- [22] G. ROZZA, D. B. P. HUYNH, AND A. T. PATERA, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Arch. Comput. Methods Engrg., 15 (2007), pp. 229–275.
- [23] A. RUHE AND D. SKOOGH, *Rational Krylov algorithms for eigenvalue computation and model reduction*, in Applied Parallel Computing Large Scale Scientific and Industrial Problems, Lect. Notes Comput. Sci. Engrg. 1541, E. Elmroth B. Kågström, J. Dongarra, and J. Wasniewski, eds., Springer, New York, 1998, pp. 491–502.
- [24] B. SALIMBAHRAMI AND B. LOHMANN, *Order reduction of large scale second-order systems using Krylov subspace methods*, Linear Algebra Appl., 415 (2006), pp. 385–405.
- [25] T.-J. SU AND R. R. CRAIG, *Model reduction and control of flexible structures using Krylov vectors*, J. Guidance Control Dynamics, 14 (1991), pp. 260–267.
- [26] D. S. WEILE, E. MICHELSEN, AND K. GALLIVAN, *Reduced-order modeling of multiscreen frequency-selective surfaces using Krylov-based rational interpolation*, IEEE Trans. Antennas and Propagation, 49 (2001), pp. 801–813.
- [27] Y. YUE AND K. MEERBERGEN, *Using Krylov-Padé model order reduction for accelerating design optimization of structures and vibrations in the frequency domain*, Internat. J. Numer. Methods Engrg., 90 (2012), pp. 1207–1232.