Accelerating PDE-constrained optimization by model order reduction with error control

Yao Yue
Karl Meerbergen

Report TW 611, March 2012



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Accelerating PDE-constrained optimization by model order reduction with error control

Yao Yue
Karl Meerbergen

Report TW 611, March 2012

Department of Computer Science, K.U.Leuven

Abstract

Design optimization problems are often formulated as PDE-constrained optimization problems where the objective is a function of the output of a large-scale parametric dynamical system, obtained from the discretization of a PDE. To reduce its high computational cost, model order reduction techniques can be used. Two-sided Krylov-Padé type methods are very well suited since also the gradient to the design parameters can be computed accurately at a low cost. In Yue and Meerbergen, International J. of Numerical methods in Engineering, 2012, we embedded model reduction and parametric model order reduction in the damped BFGS method. In the current paper, we present a new provable convergent error-based trust region method that allows to better exploit interpolatory reduced models. Then, we propose two practical algorithms to fit in this framework. For our benchmark problems, we use a simple interpolatory model order reduction method based on two-sided Krylov methods for a single interpolatory point. Numerical experiments from civil engineering show that the new methods outperform damped BFGS accelerated by non-interpolatory reduced models.

Keywords : PDE-constrained optimization, model order reduction, Krylov methods, design optimization, vibrations and structures
MSC : 35Q93, 49M20, 65F10, 65N22, 90C59

ACCELERATING PDE-CONSTRAINED OPTIMIZATION BY MODEL ORDER REDUCTION WITH ERROR CONTROL*

YAO YUE[†§] AND KARL MEERBERGEN[‡§]

Abstract. Design optimization problems are often formulated as PDE-constrained optimization problems where the objective is a function of the output of a large-scale parametric dynamical system, obtained from the discretization of a PDE. To reduce its high computational cost, model order reduction techniques can be used. Two-sided Krylov-Padé type methods are very well suited since also the gradient to the design parameters can be computed accurately at a low cost. In Yue and Meerbergen, International J. of Numerical methods in Engineering, 2012, we embedded model reduction and parametric model order reduction in the damped BFGS method. In the current paper, we present a new provable convergent error-based trust region method that allows to better exploit interpolatory reduced models. Then, we propose two practical algorithms to fit in this framework. For our benchmark problems, we use a simple interpolatory model order reduction method based on two-sided Krylov methods for a single interpolatory point. Numerical experiments from civil engineering show that the new methods outperform damped BFGS accelerated by non-interpolatory reduced models.

Key words. PDE-constrained optimization, model order reduction, Krylov methods, design optimization, vibrations and structures

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

1. Introduction. PDE-constrained optimization arises from the determination of optimal parameters of industrial designs such as insulation panels and airplane engines. A common approach to solve these design optimization problems is to conduct numerical parameter studies via a discretized 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 high number of evaluations of the system output, e.g., by a (numerical) integration over a time interval or a frequency range. Throughout this paper, we use $g(\gamma)$ to represent the objective function, where $\gamma \in \mathbb{C}^l$ represents l design parameters.

To reduce the high computational cost in computing $g(\gamma)$, we can use Model Order Reduction (MOR). MOR has been successfully applied to many different fields such as circuit simulations [11, 19], (vibro) acoustics [15] and MEMS design [12], leading to significant speedups in computing the system output. In [24], we proposed the MOR Framework to combine MOR [4] with design optimization. For each γ value that the optimization algorithm accesses, say $\gamma^{(i)}$, the MOR Framework first fixes γ to $\gamma^{(i)}$ in the parametric system, and then builds a reduced model, from which approximations to g is cheaply computed. Note that two-sided Krylov methods also well approximate the gradients of g [24, 7]. It is also possible to approximate Hessians [7], but this is usually more expensive. Therefore, we use Quasi-Newton methods in this paper.

*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 the Research Council KU Leuven grants CoE EF/05/006 (Optimization in Engineering) and OT/10/038 (Multi-parameter model order reduction and its applications).

[†]yao.yue@cs.kuleuven.be

[‡]karl.meerbergen@cs.kuleuven.be

[§]Department of Computer Science, KU Leuven, Celestijnenlaan 200A, bus 2402, 3001 Heverlee, Belgium.

A major drawback of the MOR Framework is that it requires building a new reduced model from scratch for each γ accessed by optimization. Since also the gradient is well approximated, we expect the subspaces built at $\gamma^{(i)}$ to be rich enough for γ's near $\gamma^{(i)}$. In order to reuse the reduced model built for $\gamma^{(i)}$, we use these subspaces to project the full parameter model in order to obtain a simple interpolatory reduced model, which provides us an objective approximation $\widehat{g}^{(i)}(\gamma)$. We say that the reduced model $\widehat{g}^{(i)}(\gamma)$ interpolates at $\gamma^{(i)}$ if we use the bases generated at $\gamma^{(i)}$ for the projection. Note that a reduced model can also interpolate at multiple points [7], but we do not exploit it in the current paper since a local approximation is usually sufficient for optimization. If we exploit $\widehat{g}^{(i)}(\gamma)$ also for $\gamma \neq \gamma^{(i)}$, we can decrease the number of reduced models needed, and thereby reduce the optimization time. However, by doing this, we may also run into the risk of inaccurate approximation, especially for parameter values far away from $\gamma^{(i)}$.

To achieve accuracy control, we develop a heuristic error estimator of $|g(\gamma) - \widehat{g}^{(i)}(\gamma)|$, namely $e^{(i)}(\gamma)$, for the entire parameter space. With this error bound, we can also safely use $\widehat{g}^{(i)}(\gamma)$ for $\gamma \neq \gamma^{(0)}$. To exploit a reduced model, the classical trust region methods is unsuitable since it requires direct evaluations of g, which is much more expensive than building reduced models. So an interesting topic is whether it is possible to purely rely on surrogate models to achieve convergence. We show that under the "relaxed first order condition", which the reduced models satisfy, we achieve convergence if all iterates satisfy the proposed "error-aware sufficient decrease condition". We also discuss how we can locate such iterates in practical algorithms.

Inspired by the convergence theory, we develop two algorithms: Error-based Trust Region Method (ETR) and 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 iterate, ETR and EP build a surrogate model $\widehat{g}^{(i)}$ interpolated at $\gamma^{(i)}$. To fully exploit $\widehat{g}^{(i)}$, we use $\widehat{g}^{(i)}$ and its error bound to formulate the i-th optimization subproblem and conduct an inner phase optimization for this subproblem.

ETR and EP differ in how they define the optimization subproblem. ETR uses the error bound $e^{(i)}(\gamma)$ to define a "trust region", i.e. only to trust a surrogate model when the relative error 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 runs into several difficulties. For example, around the true optimizer, the gradient of the error bound can be large even if the error bound is small. To overcome these difficulties, we introduce a weighting function for the penalty term. EP exploits a larger region of a surrogate model compared with ETR since it accepts points with large error bound, provided it achieves sufficiently larger objective decrease. However, EP does not necessarily outperform ETR. We observe in numerical tests that EP may require more surrogate models sometimes.

In this paper, we apply ETR and EP to a specific class of applications, namely optimization of second order dynamical systems arsing from structures and vibrations, where we use Krylov-Padé type MOR with a single interpolation frequency to obtain 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. For example, we can also use Parameterized MOR (PMOR) [23, 14, 13], or the so-called *Rational Krylov Methods* [20, 23], which takes multiple interpolation frequencies and parameter values, to generate interpolatory reduced models.

Throughout this paper, we use lower case letters for vectors, upper case 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, and also for the complex conjugate of a complex number. For a nonsingular matrix M, we use M^{-*} to denote $(M^*)^{-1}$. We define the derivative of a parameterized matrix $A(\lambda) = \left[a_{p,q}(\lambda)\right]_{n\times n}$ 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.

2. Description of design optimization problems. In this paper, we focus on parameter studies in the frequency domain. The discretized model is of the form

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

where $\mathcal{L}(\omega,\gamma)$ represents an $n \times n$ matrix parameterized with the frequency ω and l design parameters $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_l) \in \mathbb{C}^l$, $f \in \mathbb{C}^n$ 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 $[\omega_L, \omega_H]$. Mathematically, this optimization problem can be formulated as

$$\min_{\gamma\in\Omega} \; g(\gamma), \qquad \text{where } g(\gamma) = \int_{\omega_L}^{\omega_H} |y(\omega,\gamma)|^2 \,\mathrm{d}\omega \tag{2.2}$$

and Ω 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

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

where we define the right and left state vectors as

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

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

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

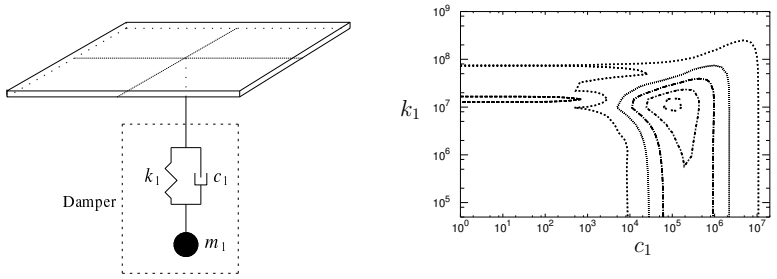
where $K(\gamma)$, $C(\gamma)$ and $M(\gamma)$ are $n \times n$ parameterized matrices representing stiffness, damping and mass, respectively.

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 the floor inside a building located near a highway. Its conceptual model is shown in Fig. 2.1(a). We used a discrete Kirchhoff triangular shell element finite element model [8] for the plate and obtained the following discrete system:

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

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 m_1 denotes the mass of the damper. The input vector f represents a unit vertical base excitation, and the output vector ℓ is used to select the degree of freedom in the center of the plate. The contour plot of the $g(\gamma)$ is shown in Fig. 2.1(b).



(a) The conceptual model (b) Contour plot of the objective function

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.*

2.2. Optimization algorithm. To choose a suitable optimization algorithm, we first analyze the formulae for $g(\gamma)$ and its derivatives. Using the following formulae for function value and derivatives of $|y(\omega,\gamma)|^2$

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

we resort to the Trapezoidal rule [10] to perform the numerical integration in computing $g(\gamma) = \int_{\omega_L}^{\omega_H} |y(\omega,\gamma)|^2 \,\mathrm{d}\omega$ and $\frac{\partial g(\gamma)}{\partial \gamma_j} = \int_{\omega_L}^{\omega_H} \frac{|y(\omega,\gamma)|^2}{\partial \gamma_j} \,\mathrm{d}\omega$.

From (2.7), we can see that 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 only consider bound constraints in this paper. Our approach consists of first using cotangent barrier functions that we will discuss in §2.2.1 to convert (2.2) to an unconstrained optimization problem, and then solve it using the damped BFGS method with a backtracking strategy. We require all line search steps to satisfy the Armijo condition:

$$g(x_k + \alpha_k p_k) \le g(x_k) + \kappa_s \alpha_k \nabla g(x_k)^T p_k, \qquad \nabla g(x_k)^T p_k < 0, \tag{2.8}$$

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 [18]. 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 later in §3.

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

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

where γ_j^{th} is the threshold value. In numerical tests, we actually solve

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

where the objective function is smooth over the interior of the feasible domain $\text{int}(\Omega)$ and equals $g(\gamma)$ for the region that we are really interested in. Because of the Armijo condition used, 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 the domain of interest.

3. Using error-aware surrogate models for design optimization. The two most important directions in numerical optimization are line search methods and trust region methods [18, 9]. In our case, we want to exploit each surrogate model around its interpolation point, which seems to fit well with the concept of trust region methods. Therefore, we first give a brief introduction to the standard trust region method, and then follow its spirit to develop our convergence theory and algorithms.

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 formulates the i-th trust region subproblem as

$$\min_{s} \; \widehat{g}^{(i)}(\gamma^{(i)} + s), \qquad \text{s.t. } \|s\| \leq \Delta_i. \tag{3.1}$$

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 function $\widehat{g}(\gamma)$.

Suppose the i-th iteration returns $\gamma_{\text{cand}}^{(i+1)}$ as a candidate for the next iterate. To test whether this point is "good enough", we compute

$$\rho_i = \frac{g(\gamma^{(i)}) - g(\gamma_{\text{cand}}^{(i+1)})}{\widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma_{\text{cand}}^{(i+1)})}. \tag{3.2}$$

Based on this value we determine whether we use $\gamma_{\text{cand}}^{(i+1)}$ or $\gamma^{(i)}$ for $\gamma^{(i+1)}$, and whether we enlarge, keep, or shrink the trust radius for the $(i+1)$-st iteration.

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

CONDITION 1. *First Order Condition:*

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

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 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. See §4 for an example of such a surrogate model. We summarize the above properties mathematically in Condition 2.

CONDITION 2. *Relaxed first order condition.*

C1. *Relatively cheap error bounds $e^{(i)}(\gamma)$ and $e_g^{(i)}$ are available, i.e.*

$$|\widehat{g}^{(i)}(\gamma)-g(\gamma)|\le e^{(i)}(\gamma) \qquad \textit{and} \qquad \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})-\nabla g(\gamma^{(i)})\|\le e_g^{(i)}; \tag{3.4}$$

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*

$$\frac{e^{(i)}(\gamma^{(i)})}{\widehat{g}^{(i)}(\gamma^{(i)})}\le\tau_g \qquad \textit{and} \qquad \frac{e_g^{(i)}}{\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|}\le\tau_{\nabla g}. \tag{3.5}$$

C3. *We have three levels of fidelity. The true objective is $g(\gamma)$, so $g(\gamma)$ and $\nabla g(\gamma)$ provides the highest level of fidelity. When we generate $\widehat{g}^{(i)}$, we impose conditions* (3.4) *and* (3.5)*, so $\widehat{g}^{(i)}(\gamma^{(i)})$ and $\nabla\widehat{g}^{(i)}(\gamma^{(i)})$ are of high fidelity. For $\gamma\neq\gamma^{(i)}$, however, we normally only know that $\widehat{g}^{(i)}(\gamma)$ is a good approximation in a neighborhood of $\gamma^{(i)}$ due to the high fidelity of $\widehat{g}^{(i)}(\gamma^{(i)})$ and $\nabla\widehat{g}^{(i)}(\gamma^{(i)})$, so $\widehat{g}^{(i)}(\gamma)$ is of low fidelity for $\gamma\neq\gamma^{(i)}$.*

C4. *We have three levels of computational cost. Directly computing $g(\gamma)$ and $\nabla g(\gamma)$ is very expensive, which we want to avoid. Generating $\widehat{g}^{(i)}$ is expensive, so we want to reduce the number of surrogate models required. Evaluation of $\widehat{g}^{i}(\gamma)$ and $\nabla\widehat{g}^{i}(\gamma)$ is cheap.*

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 norm and direction.

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

$$1-\tau_{\nabla g}\le\frac{\|\nabla g(\gamma^{(i)})\|}{\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|}\le 1+\tau_{\nabla g}, \tag{3.6}$$

and

$$\cos\vartheta\ge\frac{1-\tau_{\nabla g}^2}{\sqrt{1+\tau_{\nabla g}^2}}. \tag{3.7}$$

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

$$\begin{aligned}\|\nabla g(\gamma^{(i)})\|&\ge\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|-\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})-\nabla g(\gamma^{(i)})\|,\\ \|\nabla g(\gamma^{(i)})\|&\le\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|+\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})-\nabla g(\gamma^{(i)})\|.\end{aligned}$$

Statement (3.7) holds because

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

In the rest of the paper, we assume that ϑ is acute, i.e. $\tau_{\nabla g} < 1$.

The general idea to exploit surrogate models under the relaxed first order condition is that, for the i-th iteration, we first build the i-th surrogate model interpolated at $\gamma^{(i)}$, and then use its output $\widehat{g}^{(i)}(\gamma)$ and the error bound function $e^{(i)}(\gamma)$ to construct the objective function $F^{(i)}(\gamma)$ and the constraint $c^{(i)}(\gamma)$ for the i-th subproblem

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

In this paper, we assume that $F^{(i)}(\gamma) = \widehat{g}^{(i)}(\gamma)$ holds in a neighborhood of $\gamma^{(i)}$. See §3.2.1 and §3.2.2 for how ETR and EP choose $F^{(i)}(\gamma)$ and $c^{(i)}(\gamma)$.

Under the relaxed first order condition, we prefer to fully exploit each surrogate model to reduce the number of surrogate models required, since generating a surrogate model $\widehat{g}^{(i)}$ is much more expensive than evaluating it. Therefore, we use an inner phase optimization, e.g., employing a Quasi-Newton method, for each optimization subproblem. We do not need to solve the optimization subproblem (3.8) accurately. We terminate an iteration when we (heuristically) conjecture that continuing further contributes little to the whole picture. See §3.2.1 and §3.2.2 for more information.

To achieve convergence, it is common practice to require each optimization subproblem to achieve *sufficient decrease*. However, the existing sufficient decrease conditions require directly evaluating $g(\gamma)$, which is too expensive under the relaxed first order condition. Therefore, we will develop an error-aware sufficient decrease condition that is suitable for surrogate models satisfying the relaxed first order condition inn §3.1. We will prove that under this condition as well as several mild assumptions on the objective function and the constraint, convergence is guaranteed. Then in §3.2, inspired by the convergence theory, we propose ETR and EP, whose goal is to efficiently locate a solution satisfying the error-aware sufficient decrease condition.

3.1. An error-aware sufficient decrease condition for convergence under the relaxed first order condition. In our optimization subproblem (3.8), like in standard trust region subproblems, the Armijo condition cannot be directly used since the angle between $\gamma_{\text{cand}}^{(i+1)} - \gamma^{(i)}$ and the minus gradient direction may be not acute. Therefore, we use two steps to guarantee sufficient decrease, and thereby convergence.

1. Look in a descent direction and find a point with sufficient decrease, which is normally known as the generalized Cauchy point.

2. Require the objective decrease achieved by the solution of the optimization subproblem to be at least as much as that achieved by the generalized Cauchy point.

This procedure works well for the standard trust region method, but needs modifications to be efficient for our problem since it requires directly evaluating g at the interpolation point, the generalized Cauchy point and the solution, which are too expensive under the relaxed first order condition. Therefore, we will develop a convergence theory totally relying on the surrogate models.

In §3.1.1, we define the *approximate generalized Cauchy point*, prove that it is well-defined, and give the minimal objective decrease on $\widehat{g}^{(i)}$ it achieves. Then in

§3.1.2, we give an error-aware sufficient decrease condition, and prove that if it is satisfied for all iterates, we achieve convergence on g. In §3.1.3, we discuss how to locate such iterates and also how to further reduce the computational cost.

3.1.1. The approximate generalized Cauchy point and sufficient objective decrease. To obtain the approximate generalized Cauchy point, we first conduct line search on a descent direction

$$\gamma = \gamma^{(i)} + \alpha p^{(i)}. \tag{3.9}$$

where the search direction $p^{(i)}$ satisfies $\nabla\widehat{g}^{(i)}(\gamma^{(i)})^T p^{(i)} < 0$ and $\|p^{(i)}\| = 1$, and the step length satisfies $\alpha \geq 0$. If we denote the initial step length by $\alpha_0^{(i)}$ and the backtracking factor with $\kappa_{bt} \in (0,1)$, the j-th iterate for this line search is represented with $\gamma^{(i)}[j] = \gamma^{(i)} + \kappa_{bt}^j \alpha_0^{(i)} p^{(i)}$. We accept $\gamma^{(i)}[j]$ if it satisfies both the constraint

$$c^{(i)}(\gamma) \geq 0 \tag{3.10}$$

and the Armijo condition (2.8)

$$\widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma) \geq \kappa_s \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \|\gamma^{(i)} - \gamma\| \cos\Phi^{(i)}, \tag{3.11}$$

where $\Phi^{(i)}$ represents the angle between $-\nabla\widehat{g}^{(i)}$ and $p^{(i)}$. We define the $\gamma^{(i)}[j]$ that satisfies both conditions (3.10) and (3.11) with the smallest j as the approximate generalized Cauchy point $\gamma_{\mathrm{AGC}}^{(i)}$. If we use line search methods to solve the optimization subproblems, locating $\gamma_{\mathrm{AGC}}^{(i)}$ introduces no extra computational cost since it is actually the first iterate of the i-th optimization subproblem.

To achieve convergence, we require $\alpha_0^{(i)}$ to be not too small w.r.t. $\|\widehat{g}^{(i)}(\gamma^{(i)})\|$, This means for each i, there exists a positive number $\kappa_n^{(i)}$ with

$$\alpha_0^{(i)} \geq \kappa_n^{(i)} \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|. \tag{3.12}$$

satisfying $\kappa_n^{(i)} \geq \kappa_n > 0$ for all i's. If we use a Quasi-Newton method to generate this step, we have $\alpha_0^{(i)} = \|(B_0^{(i)})^{-1}\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| \geq (\lambda_{\max}^{(i)})^{-1}\|\widehat{g}^{(i)}(\gamma^{(i)})\|$, where $B_0^{(i)}$ is the Hessian approximation at the first step of 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 approach infinity, there exists a κ_n satisfying $\kappa_n^{(i)} \geq \kappa_n > 0$. Even if this is not true, we can circumvent this difficulty by resetting $B_0^{(i)}$ to a multiple of the identity matrix when $\lambda_{\max}^{(i)}$ becomes too large.

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

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 $u_{\widehat{g}}^{(i)} \geq 0$ 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, $\Phi^{(i)} < \frac{\pi}{2}$, $\kappa_s \in (0,1)$, $c^{(i)}(\gamma^{(i)}) > 0$, $u_c^{(i)} > 0$, $\kappa_n^{(i)} > 0$ and $u_{\widehat{g}}^{(i)} > 0$, we have*

1. *all γ's on the search direction* (3.9) *with*

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

satisfy both conditions (3.10) *and* (3.11)*;*

2. *the approximate generalized Cauchy point* $\gamma^{(i)}_{\mathrm{AGC}}$ *satisfies*

$$\|\gamma^{(i)}_{\mathrm{AGC}} - \gamma^{(i)}\| \geq \min\left\{\kappa^{(i)}_{\nabla g}\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|, \kappa_{bt}\frac{c^{(i)}(\gamma^{(i)})}{u^{(i)}_c}\right\}, \tag{3.14}$$

where $\kappa^{(i)}_{\nabla g} = \min\left\{\kappa^{(i)}_n, \dfrac{\kappa_{bt}(1-\kappa_s)\cos\Phi^{(i)}}{u^{(i)}_{\widehat{g}}}\right\}$;

3. *the lower bound of the decrease in* $\widehat{g}^{(i)}$ *achieved by* $\widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}})$ *is given by*

$$\begin{aligned}&\widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}})\\ &\geq \left(\kappa_s\cos\Phi^{(i)}\right)\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|\min\left\{\kappa^{(i)}_{\nabla g}\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|, \kappa_{bt}\frac{c^{(i)}(\gamma^{(i)})}{u^{(i)}_c}\right\}.\end{aligned} \tag{3.15}$$

Proof.

1. If $\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| = 0$, condition (3.13) becomes $\gamma = \gamma^{(i)}$, so conditions (3.10) and (3.11) 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

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

(i) Equation (3.16) has at least one solution since otherwise, $\nabla_{p^{(i)}}\widehat{g}^{(i)}(\gamma) < -\kappa_s\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|\cos\Phi^{(i)}$ 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, which 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 with γ, satisfies

$$\|\gamma^{(i)} - \gamma\| \geq \frac{\left|\nabla_{p^{(i)}}\widehat{g}^{(i)}(\gamma^{(i)}) - \nabla_{p^{(i)}}\widehat{g}^{(i)}(\gamma)\right|}{u^{(i)}_{\widehat{g}}} = \frac{(1-\kappa_s)\cos\Phi^{(i)}\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|}{u^{(i)}_{\widehat{g}}}, \tag{3.17}$$

which means that (3.11) holds for all γ's on the search direction (3.9) satisfying

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

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

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

which means that condition (3.10) holds for all γ's on the search direction (3.9) satisfying $\|\gamma^{(i)} - \gamma^{(i)}_c\| \leq \frac{c^{(i)}(\gamma^{(i)})}{u^{(i)}_c}$. Based on arguments above, statement (3.13) holds.

2. If the initial point $\gamma^{(i)} + \alpha^{(i)}_0 p^{(i)}$ satisfies both (3.10) and (3.11), the step length is at least $\kappa^{(i)}_n\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|$. Otherwise, we have two possibilities. If backtracking stops before entering the region defined by (3.13), then (3.14) must hold; otherwise, the backtracking iterate before the accepted iterate violates (3.13), so the accepted iterate satisfies (3.14).

3. A direct result of (3.14) and (3.11). ∎

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.

3.1.2. An error-aware sufficient decrease condition and convergence for the "ideal" case. Although $\gamma_{\mathrm{AGC}}^{(i)}$ is a generalized Cauchy point for $\widehat{g}^{(i)}$, not necessarily a true generalized Cauchy point for g, we will prove that in an "ideal" case, it is sufficient to use it to achieve the convergence of g. Here, the "ideal" case means that all iterations satisfy the following *error-aware sufficient decrease condition*.

CONDITION 3. *Error-aware sufficient decrease condition*

$$\widehat{g}^{(\gamma_{\mathrm{cand}}^{(i+1)})}(\gamma_{\mathrm{cand}}^{(i+1)}) \leq \widehat{g}^{(i)}(\gamma_{\mathrm{AGC}}^{(i)}). \tag{3.19}$$

If this condition is satisfied, we assign $\widehat{g}^{(i+1)} = \widehat{g}^{(\gamma_{\mathrm{cand}}^{(i+1)})}$ and $\gamma^{(i+1)} = \gamma_{\mathrm{cand}}^{(i+1)}$.

THEOREM 3.3. *Assume all conditions for Theorem 3.2 hold. Assume further that* $\Phi^{(i)} \leq \Phi < \dfrac{\pi}{2}$, $c^{(i)}(\gamma^{(i)}) \geq c_l > 0$, $0 < u_c^{(i)} \leq u_c$, $0 < u_{\widehat{g}}^{(i)} \leq u_{\widehat{g}}$, $\kappa_n^{(i)} \geq \kappa_n > 0$ *for all i's,* $\tau_{\nabla g} < 1$, $\widehat{g}^{(0)}(\gamma^{(0)})$ *is finite and* $\widehat{g}^{(i)}(\gamma)$ *is bounded from below for all i's. Then if the error-aware sufficient decrease condition* (3.19) *holds for all i, i.e.* $\widehat{g}^{(i+1)}(\gamma^{(i+1)}) \leq \widehat{g}^{(i)}(\gamma_{\mathrm{AGC}}^{(i)})$, *we have* $\lim\limits_{i\to\infty} \|\nabla g(\gamma^{(i)})\| = 0$.

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

Since $\kappa_{\nabla g}^{(i)} = \min\left\{\kappa_n^{(i)}, \dfrac{\kappa_{bt}(1-\kappa_s)\cos\Phi^{(i)}}{u_{\widehat{g}}^{(i)}}\right\} \geq \min\left\{\kappa_n, \dfrac{\kappa_{bt}(1-\kappa_s)\cos\Phi}{u_{\widehat{g}}}\right\} \triangleq \kappa_{\nabla g}$,
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 $\lim\limits_{i\to\infty} \|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| = 0$ by contradiction. Assume there would exist a $\varepsilon \in (0, \|\nabla\widehat{g}^{(0)}(\gamma^{(0)})\|)$ 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,

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

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

All assumptions in Theorem 3.2 are not very restrictive except that all iterates satisfy (3.19). If $c^{(i)}(\gamma^{(i)})$ and $\kappa_n^{(i)}$ do not approach zero, and $u_c^{(i)}$, $u_{\widehat{g}}^{(i)}$ and $\lambda_{\max}^{(i)}$ do not approach $+\infty$, then c_l, κ_n, u_c, $u_{\widehat{g}}$ and $\lambda_{\max}$ 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., reset $B_0^{(i)}$ to a multiple of the identity matrix for Quasi-Newton methods. Requiring all iterates to satisfy (3.19) is too optimistic, and now we turn to discuss what to do when condition (3.19) fails.

3.1.3. Practical considerations and convergence for the general case.

3.1.3.1. Refine the approximate generalized Cauchy point by backtracking. 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, the sufficient decrease condition (3.19) can be too restrictive sometimes. To see this, define the i-th acceptable region as $\mathcal{S}^{(i)} = \left\{\gamma_{\text{cand}}^{(i+1)} \,\middle|\, \widehat{g}^{(\gamma_{\text{cand}}^{(i+1)})}(\gamma_{\text{cand}}^{(i+1)}) \le \widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)})\right\}$. The region $\mathcal{S}^{(i)}$ should be somehow large if $\nabla\widehat{g}^{(i)}(\gamma^{(i)})$ is not small. However, in the following two extreme cases, we have $\mathcal{S}^{(i)} = \emptyset$.

1. $\widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)}) < \min\limits_{\gamma} g(\gamma)$ and $\widehat{g}^{(\gamma_{\text{cand}}^{(i+1)})}(\gamma_{\text{cand}}^{(i+1)}) > g(\gamma_{\text{cand}}^{(i+1)})$.
2. $\gamma_{\text{AGC}}^{(i)}$ is the global optimizer of $\widehat{g}^{(i)}$ and $\widehat{g}^{(\gamma_{\text{cand}}^{(i+1)})}(\gamma_{\text{cand}}^{(i+1)}) > \widehat{g}^{(i)}(\gamma_{\text{cand}}^{(i+1)})$. In this case, the i-th optimization subproblem returns $\gamma_{\text{AGC}}^{(i)}$ as $\gamma_{\text{cand}}^{(i+1)}$ and the sufficient decrease condition becomes $\widehat{g}^{(\gamma_{\text{cand}}^{(i+1)})}(\gamma_{\text{cand}}^{(i+1)}) \le \widehat{g}^{(i)}(\gamma_{\text{cand}}^{(i+1)})$. Therefore, $\mathcal{S}^{(i)} = \emptyset$.

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

For such cases, we have to use a less restrictive error-aware sufficient decrease condition. Fortunately, such a condition can be obtained simply by conducting backtracking on $\gamma_{\text{AGC}}^{(i)}$ since when $\gamma_{\text{AGC}}^{(i)}$ is sufficiently close to $\gamma^{(i)}$, we expect backtracking leads to a $\gamma_{\text{AGC}}^{(i)}$ with smaller error and smaller decrease in $\widehat{g}^{(i)}$. Therefore, we assign

$$\gamma_{\text{AGC}}^{(i)} \Leftarrow \gamma^{(i)} + \kappa_{\text{bt}}^{j(\gamma_{\text{AGC}}^{(i)})}(\gamma_{\text{AGC}}^{(i)} - \gamma^{(i)}) \tag{3.20}$$

where $j(\gamma_{\text{AGC}}^{(i)})$ is the smallest positive integer with which the updated $\gamma_{\text{AGC}}^{(i)}$ satisfies both conditions (3.10) and (3.11). 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 achieve convergence 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. If (3.19) is still too restrictive, we can conduct further backtracking on $\gamma_{\text{AGC}}^{(i)}$.

COROLLARY 3.4. *Under all assumptions of Theorems 3.2 and 3.3, if all iterates satisfy* (3.19)*, where we allow to conduct finite steps of backtracking for* $\gamma_{\text{AGC}}^{(i)}$ *as in* (3.20)*, we still have* $\lim\limits_{i\to\infty} \|\nabla g(\gamma^{(i)})\| = 0$.

3.1.3.2. On the failure of the error-aware sufficient decrease conditions (3.19). When condition (3.19) fails, we can use the following two strategies.

S1. Conduct backtracking on $\gamma_{\text{AGC}}^{(i)}$ to make condition (3.19) 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 (3.19) is too restrictive, we define

$$\varrho^{(i)} = \frac{\widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}})}{\widehat{g}^{(i)}(\gamma^{(i)}) - \widehat{g}^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}})}. \tag{3.21}$$

Since $\widehat{g}^{(i)}(\gamma^{(i)}) > \widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}}) \geq \widehat{g}^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}})$ due to the sufficient decrease condition[1], we impose on the optimization subproblem, we have $\varrho^{(i)} \in (0, 1]$.

If condition (3.19) fails, we have the following possibilities.

1. When $\widehat{g}^{(\gamma^{(i+1)}_{\mathrm{cand}})}(\gamma^{(i+1)}_{\mathrm{cand}}) > \widehat{g}^{(i)}(\gamma^{(i)})$, S1 cannot not work since $\widehat{g}^{(i)}(\gamma^{(i)}) > \widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}})$ always holds. Therefore, we use S2.
2. When $\varrho^{(i)}$ is not sufficiently small, condition (3.19) is too restrictive since $\gamma^{(i)}_{\mathrm{AGC}}$ has achieved similar decrease as the optimizer. Therefore, we use S1.
3. When $\varrho^{(i)}$ is sufficiently small, condition (3.19) is not too restrictive, but $\gamma^{(i+1)}_{\mathrm{cand}}$ is not an adequate optimizer. Therefore, we use S2.

3.1.3.3. A means to reduce the computational cost. Directly checking (3.19) requires building a new surrogate model $\widehat{g}^{(\gamma^{(i+1)}_{\mathrm{cand}})}$. If the test fails, the surrogate model $\widehat{g}^{(\gamma^{(i+1)}_{\mathrm{cand}})}$ cannot be used to define the next optimization subproblem, which is a waste of computational cost. Therefore, if we can foresee the test failure without generating $\widehat{g}^{(\gamma^{(i+1)}_{\mathrm{cand}})}$, we can avoid this waste. Now we show that this is possible for some cases.

LEMMA 3.5.

1. *A sufficient condition for* (3.19) *is*

$$\widehat{g}^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}}) + e^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}}) + e^{(\gamma^{(i+1)}_{\mathrm{cand}})}(\gamma^{(i+1)}_{\mathrm{cand}}) \leq \widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}}). \tag{3.22}$$

2. *A necessary condition for* (3.19) *is*

$$\widehat{g}^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}}) - e^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}}) - e^{(\gamma^{(i+1)}_{\mathrm{cand}})}(\gamma^{(i+1)}_{\mathrm{cand}}) \leq \widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}}). \tag{3.23}$$

Proof. Lemma 3.5 follows due to (3.19) and

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

Note that $e^{(\gamma^{(i+1)}_{\mathrm{cand}})}(\gamma^{(i+1)}_{\mathrm{cand}})$ must be small since otherwise, $\widehat{g}^{(\gamma^{(i+1)}_{\mathrm{cand}})}$ is not sufficiently accurate for the $(i+1)$-st optimization subproblem. Therefore, we reject $\gamma^{(i+1)}_{\mathrm{cand}}$ if

$$\widehat{g}^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}}) - e^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}}) > \widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}}). \tag{3.24}$$

Similarly, we accept $\gamma^{(i+1)}_{\mathrm{cand}}$ as $\gamma^{(i+1)}$ when

$$\widehat{g}^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}}) + e^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}}) < \widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}}), \tag{3.25}$$

since if we require $e^{(i+1)}(\gamma^{(i+1)})$ to be sufficiently small at the generation of $\widehat{g}^{(i+1)}$, condition (3.22) will hold. Beyond these two cases, we have to generate $\widehat{g}^{(\gamma^{(i+1)}_{\mathrm{cand}})}$ to check the sufficient decrease condition (3.19).

[1] $\widehat{g}^{(i)}(\gamma^{(i)}_{\mathrm{AGC}}) = \widehat{g}^{(i)}(\gamma^{(i+1)}_{\mathrm{cand}})$ only when $\gamma^{(i+1)}_{\mathrm{cand}} = \gamma^{(i)}_{\mathrm{AGC}}$.

3.1.3.4. A practical working procedure. Now we combine what we have discussed in §3.1.3.2 and §3.1.3.3 and give the following practical working procedure, which both guarantees convergence and achieves efficiency.

PROCEDURE 1.

1. *First check condition* (3.25). *If it holds, accept* $\gamma_{\text{cand}}^{(i+1)}$ *as* $\gamma^{(i+1)}$.
2. *Otherwise, check condition* (3.24). *If it holds, reject* $\gamma_{\text{cand}}^{(i+1)}$, *and use S2 on Page 11 to update* $\gamma_{\text{cand}}^{(i+1)}$.
3. *If both conditions* (3.24) *and* (3.25) *fail, check whether* $\varrho^{(i)}$ *is sufficiently small. If not, conduct backtracking for* $\gamma_{\text{AGC}}^{(i)}$ *and go to 1.*
4. *Otherwise, generate* $\widehat{g}^{(\gamma_{\text{cand}}^{(i+1)})}$ *and check condition* (3.19). *If condition* (3.19) *holds, accept* $\gamma_{\text{cand}}^{(i+1)}$ *as* $\gamma^{(i+1)}$.
5. *Otherwise, reject* $\gamma_{\text{cand}}^{(i+1)}$ *and use S2 on Page 11 to update* $\gamma_{\text{cand}}^{(i+1)}$.

3.2. 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. Otherwise, we may waste surrogate models. So an efficient algorithm should have a mechanism to make condition (3.25) easy to satisfy. Since after possible backtracking(s) on $\gamma_{\text{AGC}}^{(i)}$, $\widehat{g}^{(i)}(\gamma_{\text{cand}}^{(i+1)}) < \widehat{g}^{(i)}(\gamma_{\text{AGC}}^{(i)})$ must hold, ETR requires the error term $e^{(i)}(\gamma_{\text{cand}}^{(i+1)})$ to be small compared with these two values in order to increase to possibility that (3.25) holds. EP considers $\widehat{g}^{(i)}(\gamma_{\text{cand}}^{(i+1)}) + e^{(i)}(\gamma_{\text{cand}}^{(i+1)})$ as a whole and introduces a penalty term to achieve error control.

3.2.1. Error-based Trust Region method (ETR). ETR defines the i-th optimization subproblem as

$$\min_{\gamma} \ \widehat{g}^{(i)}(\gamma) \qquad \text{s.t.} \ \frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} \le \epsilon_L. \quad (\epsilon_L > 0) \tag{3.26}$$

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

$$\frac{e^{(i)}(\gamma^{(i)})}{\widehat{g}^{(i)}(\gamma^{(i)})} \le \delta\epsilon_L, \qquad (\delta \in (0,1)) \tag{3.27}$$

which means that the interpolation point must strictly satisfy the constraint, i.e. $c^{(i)}(\gamma^{(i)}) > 0$. If condition (3.27) 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 (3.26). 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 (3.26) on the boundary, we use the following stopping criterion:

$$\|\nabla\widehat{g}^{(i)}(\gamma)\| \le \tau_{sub} \qquad \text{or} \qquad \beta\epsilon_L \le \frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} \le \epsilon_L, \qquad (\tau_{sub} \ge 0,\, \beta \in (\delta, 1)), \tag{3.28}$$

where δ is defined in (3.27). 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. Fig. 3.1 shows four possible optimization paths for ETR.

We summarize the working flow of ETR in Algorithm 2.

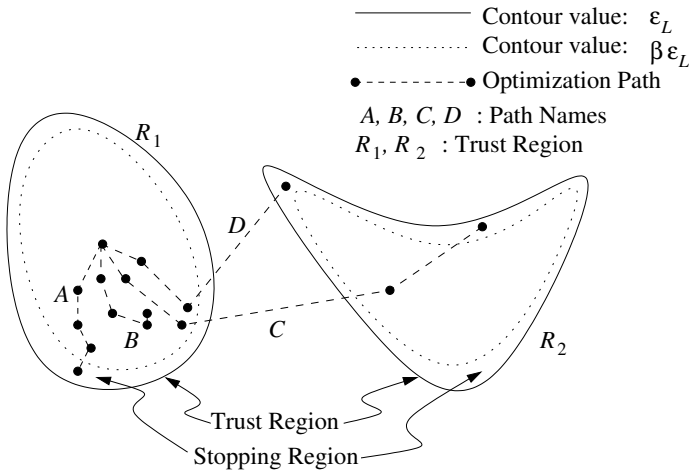


FIG. 3.1. *Four possible optimization paths for ETR. Paths A, C and D terminate when they reach a point too close to the boundary, while path B terminates when it reaches a stationary point. Paths C and D show that ETR may enter another trust region without noticing this.*

ALGORITHM 2. *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 relative error bound of the optimizer* τ_{opt}, *compute* $\epsilon_L = \frac{\tau_{opt}}{1+\tau_{opt}}$.
2. *Generate the i-th surrogate model* $\widehat{g}^{(i)}$ *satisfying* $\frac{e^{(i)}(\gamma^{(i)})}{\widehat{g}^{(i)}(\gamma^{(i)})} \leq \delta\epsilon_L$, $e_g^{(i)} < \tau$ *and* $\frac{e_g^{(i)}}{\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\|} \leq \tau_{\nabla g}$. *If* $\|\nabla\widehat{g}^{(i)}(\gamma^{(i)})\| + e_g^{(i)} \leq \tau$, *return* $\gamma^{(i)}$ *and stop.*
3. *Minimize* $\widehat{g}^{(i)}(\gamma)$ *with the stopping criterion* (3.28)*; let the optimizer be* $\gamma_{\text{cand}}^{(i+1)}$.
4. *Use Procedure 1 to determine how to continue.*

The choice of ϵ_L follows from the fact that

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

where we use $\frac{g(\gamma^{(k)})}{\widehat{g}^{(k)}(\gamma^{(k)})} \geq 1 - \epsilon_L$, a formula that can be proven similar to (3.6).

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

3.2.2. Error-based Penalty Method (EP). Sometimes, the constraint in ETR is too restrictive since a γ value satisfying (3.25) is excluded from being exploited when $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} > \epsilon$. Since (3.25) is cheap to verify, we may further reduce optimization time if we accept such points. A straightforward approach is to minimize the left hand size of (3.25), i.e. $b^{(i)}(\gamma) = \widehat{g}^{(i)}(\gamma) + e^{(i)}(\gamma)$. But this leads to the following problems: 1.) the penalty term $e^{(i)}(\gamma)$ can be non-convex; 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, $\widehat{b}^{(i)}$ is a worse approximation of

g than $\widehat{g}^{(i)}(\gamma^{(i)})$, especially when the error bound $e^{(i)}(\gamma)$ overestimates the true error; 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

$$w(\epsilon) = 0 \text{ when } \epsilon \leq \epsilon_L, \quad w(\epsilon) = 1, \text{ when } \epsilon \geq \epsilon_H, \quad (\epsilon_L < \epsilon_H). \tag{3.29}$$

We define the i-th optimization subproblem as

$$\min_{\gamma} \quad F^{(i)}(\gamma) = \widehat{g}^{(i)}(\gamma) + w\left(\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)}\right) e^{(i)}(\gamma). \tag{3.30}$$

The objective $F^{(i)}(\gamma)$ equals $\widehat{g}^{(i)}(\gamma)$ when the relative error $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} \leq \epsilon_L$, and equals $b^{(i)}(\gamma)$ when $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} \geq \epsilon_H$. As in ETR, we also require $\widehat{g}^{(i)}$ to satisfy (3.27). 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 non-zero 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; 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 $e^{(i)}(\gamma)$; 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:

$$\|\nabla F^{(i)}(\gamma)\| \leq \tau_{sub} \ (\tau_{sub} \geq 0) \quad \text{or} \quad \frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} > \epsilon_L \text{ holds for } \mu \text{ successive steps.} \tag{3.31}$$

Since μ strongly impacts the algorithm 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 §5 for more discussion. Fig. 3.2 shows four possible optimization paths for EP.

ALGORITHM 3. *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 $\widehat{g}^{(i)}$ satisfying $\frac{e^{(i)}(\gamma^{(i)})}{\widehat{g}^{(i)}(\gamma^{(i)})} \leq \delta\epsilon_L$, $e_g^{(i)} < \tau$ and $\frac{e_g^{(i)}}{\|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\|} \leq \tau_{\nabla g}$. If $\|\nabla \widehat{g}^{(i)}(\gamma^{(i)})\| + e_g^{(i)} \leq \tau$, return $\gamma^{(i)}$ and stop.*
3. *Minimize $F^{(i)}(\gamma)$ defined by (3.30) using the stopping criterion (3.31); let the optimizer be $\gamma_{\text{cand}}^{(i+1)}$.*
4. *Use Procedure 1 to determine how to continue.*

The convergence of EP follows from Procedure 1. When Procedure 1 asks to reconstruct the optimization subproblem, $\widehat{g}^{(i)}$ is certain to be not accurate enough at

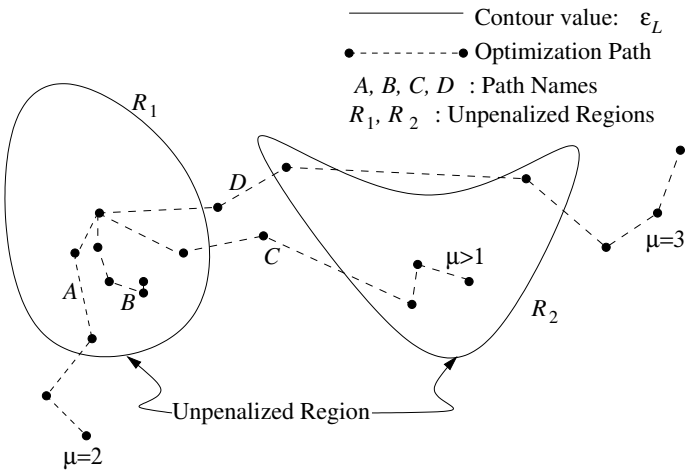


FIG. 3.2. *Four possible optimization paths for EP. Paths B and C terminate at stationary points in the unpenalized region, while paths A and D terminate when they stay in the penalized region for μ successive steps.*

$\gamma_{\text{cand}}^{(i+1)}$, but it is not fully penalized there. So if we do not refine $\widehat{g}^{(i)}$, we should decrease ϵ_H below the relative error at $\gamma_{\text{cand}}^{(i+1)}$ to ensure that $\gamma_{\text{cand}}^{(i+1)}$ will not be accepted again.

3.3. Implementation details for ETR and EP.

1. *Large-step backtracking.* In numerical tests, we divide the backtracking factor κ_{bt} by $\left(1+0.05\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)}\right)$ when $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} > \epsilon_L$. When $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)} > \epsilon_L$, ETR uses backtracking in order to return to the trust region. It is reasonable to use a large backtracking steps when $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)}$ 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 $h_0 I$. 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.

4. 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 MOR methods generate two $n\times k$ ($k \le n$ 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

$$\begin{cases} \widehat{\mathcal{L}}(\omega,\gamma)z = \widehat{f}, \\ \widehat{y} = \widehat{\ell}^* z, \end{cases} \tag{4.1}$$

where $\widehat{\mathcal{L}}(\omega,\gamma) = W_k^*\mathcal{L}(\omega,\gamma)V_k$, $\widehat{f} = W_k^* f$ and $\widehat{\ell} = V_k^*\ell$. The left state vector of system (4.1) is $\zeta(\omega,\gamma) \triangleq \widehat{\mathcal{L}}(\omega,\gamma)^{-*}\widehat{\ell}$, 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) [22, 5, 21] method.

4.1. SOAR and the MOR Framework. In the MOR Framework [24], 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:

$$\begin{cases} \big(\widehat{K} + i(\omega-\omega^{(i)})\widehat{C} - (\omega-\omega^{(i)})^2\widehat{M}\big) z(\omega-\omega^{(i)},\gamma^{(i)}) = \widehat{f}, \\ \widehat{y}(\omega-\omega^{(i)},\gamma^{(i)}) = \widehat{\ell}^* z(\omega-\omega^{(i)},\gamma^{(i)}), \end{cases} \tag{4.2}$$

where $\widehat{K} = W^*\Big(K(\gamma^{(i)}) + i\omega^{(i)}C(\gamma^{(i)}) - (\omega^{(i)})^2M(\gamma^{(i)})\Big)V$, $\widehat{C} = W^*\Big(C(\gamma^{(i)}) - 2\omega^{(i)}M(\gamma^{(i)})\Big)V$, $\widehat{M} = W^*M(\gamma^{(i)})V$, $\widehat{f} = W_k^*f$ and $\widehat{\ell} = V_k^*\ell$. The computationally dominant part of SOAR is the LU factorization of the large-scale sparse matrix $\Big(K(\gamma^{(i)}) + i\omega^{(i)}C(\gamma^{(i)}) - (\omega^{(i)})^2M(\gamma^{(i)})\Big)$. See [24] for more details.

No design parameter is not preserved in (4.2). However, if we use the bases W_k and V_k for the general system (2.1) as we did in (4.1), we obtain a parametric reduced model, which can be seen as an interpolatory reduced model for 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 (4.1) 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

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

where we call $m\left(j|\omega^{(i)},\gamma^{(i)}\right)$ and $\widehat{m}\left(j|\omega^{(i)},\gamma^{(i)}\right)$ 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 $m\left(j|\omega^{(i)},\gamma^{(i)}\right) = \widehat{m}\left(j|\omega^{(i)},\gamma^{(i)}\right)$. 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 $\frac{\partial\widehat{y}}{\partial\gamma_q}$ $(1\le q\le l)$. Theorem 4.1 [24] summarizes the moment matching properties of SOAR.

THEOREM 4.1. *Assume that SOAR generates W_k and V_k with the interpolation point $(\omega^{(i)},\gamma^{(i)})$ and $\Big(K(\gamma^{(i)}) + i\omega^{(i)}C(\gamma^{(i)}) - (\omega^{(i)})^2M(\gamma^{(i)})\Big)$ is nonsingular. Then for the order k generalized SOAR reduced model* (4.1) *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 $\frac{\partial\widehat{y}(\omega,\gamma)}{\partial\gamma_q}\Big|_{\gamma=\gamma^{(i)}}$ $(1\le q\le l)$ match.*

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

According to Theorem 4.1, 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 $\omega^{(i)} = \frac{\omega_L+\omega_H}{2}$.

4.2. An approximate error bound. Finding a tight and computationally cheap error bound for the mismatch between the outputs of (2.1) and (4.1) is still an open problem. With $\gamma = \gamma^{(0)}$, error bounds are developed in [6, 17]. However, for

$\gamma \neq \gamma^{(0)}$, no error bounds exist according 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

$$r_r^{(i)}(\omega,\gamma) = \mathcal{L}(\omega,\gamma)V_k^{(i)}z^{(i)}(\omega,\gamma) - f, \quad r_l^{(i)}(\omega,\gamma) = \mathcal{L}^*(\omega,\gamma)W_k^{(i)}\zeta^{(i)}(\omega,\gamma) - \ell. \tag{4.3}$$

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

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

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

$$\begin{aligned}|y|^2 &= |\widehat{y}^{(i)} + (y - \widehat{y}^{(i)})|^2 \le (|\widehat{y}^{(i)}| + |y - \widehat{y}^{(i)}|)^2 = |\widehat{y}^{(i)}|^2 + 2|\widehat{y}^{(i)}||y - \widehat{y}^{(i)}| + |y - \widehat{y}^{(i)}|^2 \\ &\approx |\widehat{y}^{(i)}|^2 + 2|\widehat{y}^{(i)}|e_y^{(i)} + {e_y^{(i)}}^2 \triangleq b_{y^2}^{(i)},\end{aligned}$$

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

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

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

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

Then we derive $$\frac{\partial e^{(i)}(\gamma)}{\partial \gamma_j} = 2\int_{\omega_L}^{\omega_H}\frac{\partial |\widehat{y}^{(i)}|}{\partial \gamma_j}e_y^{(i)} + \left(|\widehat{y}^{(i)}| + e_y^{(i)}\right)\frac{\partial e_y^{(i)}}{\partial \gamma_j}.$$

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

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

$$
\begin{aligned}
&= \left| (W_k\zeta)^* \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j} V_k z - \xi^* \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j} x \right| \\
&= \left| (W_k\zeta)^* \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j} V_k z - (W_k\zeta)^* \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j} x + (W_k\zeta)^* \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j} x - \xi^* \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j} x \right| \\
&\le \left| (W_k\zeta)^* \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j} \mathcal{L}(\omega,\gamma)^{-1} r_r^{(i)} \right| + \left| r_l^{(i)*} \mathcal{L}(\omega,\gamma)^{-1} \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j} x \right| \\
&\le \left\| \mathcal{L}(\omega,\gamma)^{-*} \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j}^* W_k\zeta \right\| \left\| r_r^{(i)} \right\| + \left\| r_l^{(i)} \right\| \left\| \mathcal{L}(\omega,\gamma)^{-1} \frac{\partial \mathcal{L}(\omega,\gamma)}{\partial \gamma_j} x \right\| \\
&\approx \left\| \widehat{\mathcal{L}}^{(i)}(\omega,\gamma)^{-*} \frac{\partial \widehat{\mathcal{L}}^{(i)}(\omega,\gamma)}{\partial \gamma_j}^* \zeta \right\| \left\| r_r^{(i)} \right\| + \left\| r_l^{(i)} \right\| \left\| \widehat{\mathcal{L}}^{(i)}(\omega,\gamma)^{-1} \frac{\partial \widehat{\mathcal{L}}^{(i)}(\omega,\gamma)}{\partial \gamma_j} z \right\| \\
&\triangleq b_{grad_j}(\omega,\gamma).
\end{aligned}
$$

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 multiplies 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 since the error bound is only heuristic and normally overestimated, so 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.

4.2.1. 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 Fig. 4.1. Fig. 4.1 shows that 1.) the reduced model approximates the original model well in a large domain; 2.) the contour lines for the relative error estimator $\frac{e^{(i)}(\gamma)}{\widehat{g}^{(i)}(\gamma)}$ can be convex, non-convex or even disconnected.

4.3. Combining SOAR with ETR and EP. SOAR reduced models serve as suitable surrogate models for ETR and EP because they satisfy all four items of the relaxed first order condition (Condition 2 on Page 6):

For C1. See §4.2 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. With $k = n$ (assume W_k and V_k are of full rank), we obtain an exact model.

For C3. According to Theorem 4.1, $\widehat{g}^{(i)}(\gamma^{(i)})$ and $\nabla \widehat{g}^{(i)}(\gamma)\big|_{\gamma=\gamma^{(i)}}$ are good approximations of $g(\gamma^{(i)})$ and $\nabla g(\gamma)|_{\gamma=\gamma^{(i)}}$ with error of orders $\int_{\omega_L}^{\omega_H} \mathcal{O}\left((\omega-\omega^{(i)})^{2k}\right) \mathrm{d}\omega$ and $\int_{\omega_L}^{\omega_H} \mathcal{O}\left((\omega-\omega^{(i)})^{k}\right) \mathrm{d}\omega$, respectively. Although we lose moment matching properties for $\gamma \neq \gamma^{(i)}$, if both g and $\widehat{g}^{(i)}$ are smooth, the reduced model should approximate the original model well for γ's close enough to $\gamma^{(i)}$ since $\widehat{g}^{(i)}(\gamma^{(i)})$ and $\nabla \widehat{g}^{(i)}(\gamma)\big|_{\gamma=\gamma^{(i)}}$ are well approximated by $g(\gamma^{(i)})$ and $\nabla g(\gamma)|_{\gamma=\gamma^{(i)}}$, respectively.

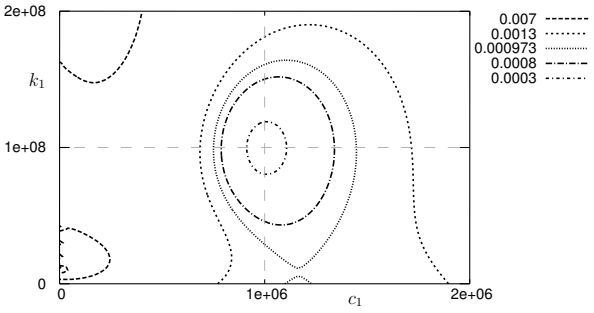


FIG. 4.1. *Contour plot of the relative error estimator for the floor damper optimization problem. For this plot, we used a discretized model of dimension 280 as the original model, while a SOAR reduced model of dimension 7 interpolated at* $(k_1, c_1) = (10^8, 10^6)$ *as the surrogate model. We used a coarse mesh for discretization in order to generate an accurate plot in a short time.*

For C4. Computing $g(\gamma)$ and $\nabla g(\gamma)$ requires large-scale sparse LU factorizations of $\mathcal{L}(\omega, \gamma)$ at all ω values accessed by the numerical integrator; building $\widehat{g}^{(i)}$ only requires the large-scale LU factorization of $\mathcal{L}(\omega, \gamma)$ for the interpolation point $(\omega^{(i)}, \gamma^{(i)})$; evaluating $\widehat{g}^{(i)}(\gamma)$ requires no additional large-scale sparse matrix factorization.

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

All our codes are implemented with the C++ package GLAS [16]. We solve large-scale sparse linear equations with MUMPS [2] and small-scale dense linear equations with LAPACK [3]. All numerical tests are run using a DELL Latitude E6400 with Intel® Core™2 Duo 2.66GHz CPU and 4GB of main memory.

5.1. Choice of parameters for ETR and EP in the MOR scenario.

1. We recommend to use a relatively large order for reduced models since in that way, we can take larger steps and use less reduced models for optimization.

2. It is preferable not to use too small ϵ_L and ϵ_H since the proposed error bound is not very tight. Using a too small ϵ_L will result in too small steps and slows down the convergence We have 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 (3.29) as

$$w(\epsilon) = \begin{cases} 0, & \text{when } \epsilon \leq 0.1, \\ 2\left(10\epsilon - 1\right)^2, & \text{when } 0.1 < \epsilon \leq 0.15, \\ 1 - 2\left(2 - 10\epsilon\right)^2, & \text{when } 0.15 < \epsilon \leq 0.2, \\ 1, & \text{when } \epsilon > 0.2. \end{cases} \tag{5.1}$$

3. We choose δ to be sufficiently below 1 to prevent the trust/unpenalized region to be 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 barrier threshold γ_j^{th} is chosen so that the barrier term is only active for physically unrealistic γ_j values. Our numerical tests use $\gamma_j^{th} = 2\pi$ $(j = 1, 2, \ldots, l)$.

5. 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 his happens more often when we use HAT. For all numerical tests, we set $h_0 = 10^{-5}$ and $\lambda_\tau^{(\max)} = 1$.

5.2. Problem I: floor damper optimization. See §2.1 for the problem description. The floor is 10m×10m×0.3m 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 29799 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 factor c_1. The frequency range of interest is [2 rad/s, 200 rad/s]. The feasible domain is $[\,1\,\mathrm{N/m},\ 10^{10}\,\mathrm{N/m}\,]\times[\,1\,\mathrm{Ns/m},\ 10^{10}\,\mathrm{Ns/m}\,]$ and the initial point is $[\,k_1^{(0)},\ c_1^{(0)}\,]=[\,10^7\,\mathrm{N/m},\ 10^7\mathrm{Ns/m}\,]$. In numerical integration, we used 151 interpolation frequencies for $\widehat{g}^{(i)}(\gamma)$ and $\nabla\widehat{g}^{(i)}(\gamma)$, 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.

TABLE 5.1

Numerical results of the floor damper optimization problem. In this table, Ni and Nm represent the total number of sub-iterations and the total number of reduced model needed, respectively; "Optimum" is the objective value of the located optimizer computed via the original model *g; the "Relative Error" shown is computed as* $\frac{|g(\gamma_{\mathrm{opt}})-\hat{g}(\gamma_{\mathrm{opt}})|}{g(\gamma_{\mathrm{opt}})}$*, where* $\hat{g}$ *is the last reduced model generated and* γ_{opt} *is the located optimizer.*

	Order	Optimum	CPU Time	Relative Error	Ni	Nm
ETR	7	1.316206122e+10	220s	2.626620741e-09	69	20
	10	1.316206122e+10	35s	2.424547354e-11	14	3
ETR + HAT	7	1.316206122e+10	192s	2.571925506e-09	61	20
	10	1.316206122e+10	37s	4.696182922e-11	15	3
EP(1)	7	1.316206119e+10	129s	2.616067046e-09	32	16
	10	1.316206122e+10	41s	1.729372585e-11	10	4
EP(1) + HAT	7	1.316206119e+10	141s	2.594100758e-09	35	17
	10	1.316206122e+10	40s	3.049540945e-11	9	4
EP(2)	7	1.316206122e+10	169s	2.585171968e-09	48	17
	10	1.316206122e+10	45s	3.963751122e-11	14	4
EP(2) + HAT	7	1.316206122e+10	194s	2.593526614e-09	103	20
	10	1.316206122e+10	44s	9.38070038e-11	13	4
EP(μ) ($\mu\geq8$)	10	1.316206123e+10	58s	7.083806541e-11	20	4
ETR & EP(μ) ($\mu\geq1$)	12	1.316206122e+10	25s	3.87328218e-11	9	2
The MOR Framework	7	1.316206122e+10	89s	2.6865095e-09	9	9
Direct Method	29800	1.316206122e+10	7202s	—	9	0

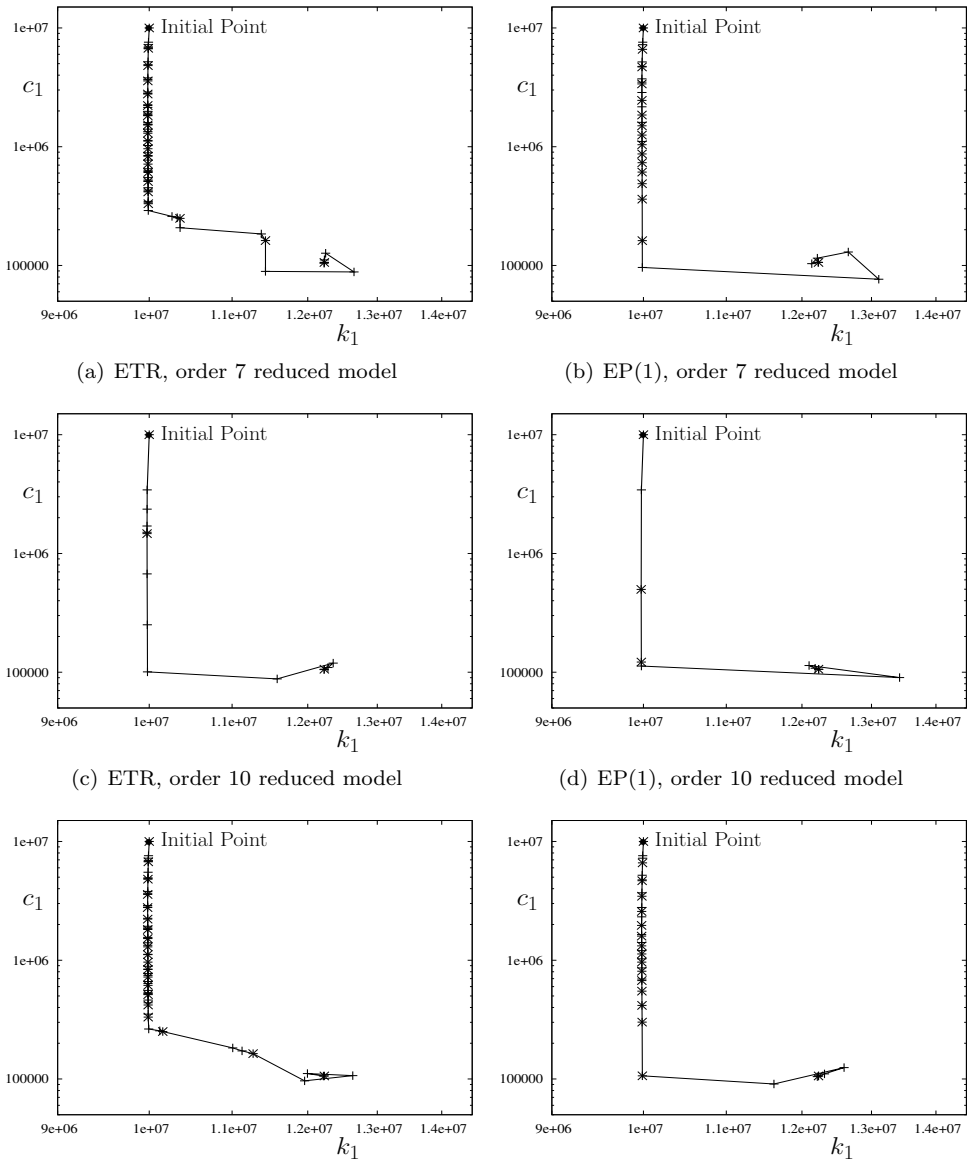
Now we analyze the numerical results shown in Table 5.1.

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 ETR can only take relatively small steps. From Fig. 5.1(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, Fig. 5.1(b) shows that this phenomenon is not so severe as in ETR.

3. When the reduced models are of size 10, ETR is a little bit more efficient than EP(1). In this case, the reduced models are more accurate. Compare Fig. 5.1(c) with Fig. 5.1(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. Compare Fig. 5.1(a) and Fig. 5.1(b) with Fig. 5.1(e) and Fig. 5.1(f), we see



(a) ETR, order 7 reduced model

(b) EP(1), order 7 reduced model

(c) ETR, order 10 reduced model

(d) EP(1), order 10 reduced model

(e) ETR+ HAT, order 7 reduced model

(f) EP(1) + HAT, order 7 reduced model

FIG. 5.1. *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 ∗.*

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 Ni) and more CPU time since locating a stationary point outside the unpenalized region is beyond our true interest.

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. Fur-

ther 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 [24] lacks error control, but costs more time than ETR and EP with the reduced model of size or 10 or 12. The direct method is very time consuming.

5.3. 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 meters 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

$$\begin{cases} \Big(K_0 + i\omega C_0 + \sum_{i=1}^{4}(k_i + i\omega c_i)K_i - \omega^2 M_0\Big)x = f, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y = \ell^* x, \end{cases} \tag{5.2}$$

where K_0 and M_0 are obtained from a finite element model with 25962 DOFs, $C_0 = 0.1003M_0 + 0.0001591K_0$ represents Rayleigh damping, K_i is a matrix with four non-zero entries that represents the interaction between the i-th TMD and the foot bridge, 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 $\ell = 10^6\ell$. In numerical integration, we used 151 interpolation frequencies in computing $\widehat{g}^{(i)}(\gamma)$ 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 the angular frequency range $[\omega_L, \omega_H]$ =[0 rad/s, 10π rad/s].

TABLE 5.2
Numerical results of the footbridge damper optimization problem. See Table 5.1 for explanations for Ni, Nm, "Optimum" and "Relative Error".

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

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

From Table 5.2, we deduce similar conclusions as in §5.2 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 seems to be more effective 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 only adds a rank two change to the Hessian approximation.

3. EP(2) seems to work better for this problem. We observe 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 model needed.

5.4. 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 high accuracy, ETR can outperform EP(1) since EP(1) may take too large steps, in which case we need to generate additional reduced models. 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.

6. Conclusion. This paper presented two methods, ETR and EP, to fully exploit surrogate models with error bounds available in PDE-constrained 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 prove 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 [24]. 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 vibration response 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. The scientific responsibility rests with its authors.

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*, Structural and Multidisciplinary Optimization, 15 (1998), pp. 16–23.

[2] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM Journal of Matrix Analysis and Applications, 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*, Society for Industrial and Applied Mathematics, Philadelphia, PA, third ed., 1999.

[4] A. C. Antoulas and D. C. Sorensen, *Approximation of large-scale dynamical systems: an overview*, International Journal of Applied Mathematics and Computer Science, 11 (2001), pp. 1093–1121.

[5] Z. Bai and Y. Su, *Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method*, SIAM Journal on Scientific Computing, 26 (2005), pp. 1692–1709.

[6] Z. Bai and Q. Ye, *Error estimation of the Padé approximation of transfer functions via the Lanczos process*, ETNA, 7 (1998), pp. 1–17.

[7] Ulrike Baur, Christopher Beattie, Peter Benner, and Serkan Gugercin, *Interpolatory projection methods for parameterized model reduction*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2489–2518.

[8] C. R. Calladine, *Theory of Shell Structures*, Cambridge University Press, Cambridge, New York, New Rochelle, Melbourne, Sydney, 1983.

[9] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint, *Trust Region Methods*, SIAM/MPS, 2000.

[10] Philip J. Davis and Philip Rabinowitz, *Methods of Numerical Integration*, Dover Publications, Mineola, New York, 2nd ed., 2007.

[11] 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.

[12] Jeong Sam Han, Evgenii B. Rudnyi, and Jan G. Korvink, *Efficient optimization of transient dynamic problems in MEMS devices using model order reduction*, Journal of Micromechanics and Microengineering, 15 (2005), pp. 822–832.

[13] Yung-Ta Li, Zhaojun Bai, and Yangfeng Su, *A two-directional Arnoldi process and its application to parametric model order reduction*, Journal of Computational and Applied Mathematics, 226 (2009), pp. 10–21.

[14] Yung-Ta Li, Zhaojun Bai, Yangfeng Su, and Xuan Zeng, *Model order reduction of parameterized interconnect networks via a Two-Directional Arnoldi process*, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 27 (2008), pp. 1571–1582.

[15] M. Malhotra and P. M. Pinsky, *Efficient computation of multi-frequency far-field solutions of the Helmholtz equation using Padé approximations*, Journal of Computational Acoustics, 8 (2000), pp. 223–240.

[16] Karl Meerbergen, *GLAS (Generic Linear Algebra Software)*. `http://people.cs.kuleuven.be/~karl.meerbergen/glas/`.

[17] ———, *The solution of parametrized symmetric linear systems*, SIAM journal on matrix analysis and applications, 24 (2003), pp. 1038–1059.

[18] Jorge Nocedal and Stephen Wright, *Numerical Optimization*, Springer, New York, 2nd ed., 2006.

[19] Altan Odabasioglu, Mustafa Celik, and Lawrence 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, USA, 1997, IEEE Computer Society, pp. 58–65.

[20] Axel Ruhe and Daniel Skoogh, *Rational Krylov algorithms for eigenvalue computation and model reduction*, in Applied Parallel Computing Large Scale Scientific and Industrial Problems, Lecture Notes in Computational Science and Engineering, Vol. 1541, Erik Elmroth Bo Kågström, Jack Dongarra and Jerzy Wasniewski, eds., Dordrecht, Heidelberg, London, New York, 1998, Springer, pp. 491–502.

[21] Behnam. Salimbahrami and Boris Lohmann, *Order reduction of large scale second-order systems using Krylov subspace methods*, Linear Algebra and its Applications, 415 (2006), pp. 385–405.

[22] T. Z. U. J. SU and RR CRAIG, *Model reduction and control of flexible structures using Krylov vectors*, Journal of guidance, control, and dynamics, 14 (1991), pp. 260–267.

[23] D. S. Weile, E. Michielssen, and K. Gallivan, *Reduced-order modeling of multiscreen frequency-selective surfaces using Krylov-based rational interpolation*, IEEE transactions on antennas and propagation, 49 (2001), pp. 801–813.

[24] Yao Yue and Karl Meerbergen, *Using Krylov-Padé model order reduction for accelerating design optimization of structures and vibrations in the frequency domain*, International Journal for Numerical Methods in Engineering, (Accepted).