Using Model Order Reduction for Design Optimization of Structures and Vibrations

Yao Yue
Karl Meerbergen

Katholieke Universiteit Leuven
Department of Computer Science

Using Model Order Reduction for Design Optimization of Structures and Vibrations

Yao Yue
Karl Meerbergen

Department of Computer Science, K.U.Leuven

Abstract

In many engineering problems, the behavior of dynamical systems depends on physical parameters. In design optimization, these parameters are determined so that an objective function is minimized. For applications in vibrations and structures, the objective function depends on the frequency response function over a given frequency range and we optimize it in the parameter space. Due to the large size of the system, numerical optimization is expensive. In this paper, we propose the combination of Quasi-Newton line search optimization methods and Krylov-Padé type algebraic model order reduction techniques to speed up numerical optimization of dynamical systems. We prove that Krylov-Padé type model order reduction allows for fast evaluation of the objective function and its gradient, thanks to the moment matching property for both the objective function and the derivatives towards the parameters. We show that reduced models for the frequency alone lead to significant speed ups. More interestingly, we show that reduced models valid for the frequency range and a line in the parameter space are helpful for the reduction of the computation time for minimax optimization.

Keywords : (Parameterized) Model Order Reduction, Krylov Methods, Quasi-Newton Optimization, Design Optimization, Structures and Vibrations

Using Model Order Reduction for Design Optimization of Structures and Vibrations

Yao Yue and Karl Meerbergen

May 31, 2010

Abstract

In many engineering problems, the behavior of dynamical systems depends on physical parameters. In design optimization, these parameters are determined so that an objective function is minimized. For applications in vibrations and structures, the objective function depends on the frequency response function over a given frequency range and we optimize it in the parameter space. Due to the large size of the system, numerical optimization is expensive. In this paper, we propose the combination of Quasi-Newton line search optimization methods and Krylov-Padé type algebraic model order reduction techniques to speed up numerical optimization of dynamical systems. We prove that Krylov-Padé type model order reduction allows for fast evaluation of the objective function and its gradient, thanks to the moment matching property for both the objective function and the derivatives towards the parameters. We show that reduced models for the frequency alone lead to significant speed ups. More interestingly, we show that reduced models valid for the frequency range and a line in the parameter space are helpful for the reduction of the computation time for minimax optimization.

Keywords: (Parameterized) Model Order Reduction, Krylov Methods, Quasi-Newton Optimization, Design Optimization, Structures and Vibrations

1 Introduction

Numerical parameter studies of vibration problems arising from applications such as the design of airplane engines, insulation panels along motor ways or in houses, and RCL oscillator circuits are often carried out in order to choose the "optimal" values of the parameters to meet design objectives such as reducing noise, and thus can be viewed as optimization problems. These problems are often computationally extremely expensive, since for each single parameter value, an entire Frequency Response Function (FRF) needs to be computed, which by itself is already quite expensive.

The computational cost for FRFs has been dramatically reduced by a factor of ten or more by using Model Order Reduction (MOR) techniques [22]. The goal of MOR is to construct a low order model to approximate the original large-scale model with high accuracy to reduce the computational cost. It has already been successfully applied to many different fields such as circuit simulations [10, 25], (vibro) acoustics [22] and MEMS design [15]. However, little work has been done to introduce MOR into optimization although optimization problems are more expensive in general, because an FRF must be computed on each iteration.

The goal of this paper is the efficient solution of design optimization problems arising from structures and vibrations. The objective of our optimization is to find the values of design parameters so that some norm of the FRF is minimized. When we use the ∞-norm, we have a so-called minimax optimization problem, while with the 2-norm, we minimize some energy norm over a given frequency range. When a sparse linear system solver is directly used for the computation of the FRFs, the gradients are cheaply computed as by-products of the function evaluations. Therefore, Quasi-Newton type methods are good choices for these problems. However, for large scale systems that often occur in real world applications, function evaluations are quite expensive. So, Krylov-Padé type MOR methods are often used for efficient function evaluations since they are suitable for reducing large-scale systems. Traditionally, we fix all the design parameters and do MOR only on the frequency ω to accelerate the evaluations of the corresponding FRF. Recently, Parameterized MOR (PMOR) [31, 14, 30, 8, 17, 11, 12, 20, 19, 18] was introduced to reduce both ω and the design parameters. If we reduce on all parameters, we can use a single reduced model for the computation of all FRFs. However, when the system has many design parameters, it is not practical to do PMOR on all these parameters because 1.) PMOR algorithms with many parameters are tricky and difficult to implement; 2.) The order of the reduced order model should increase if we reduce on more parameters and require similar accuracy, which may make the acceleration by PMOR less efficient. Therefore, we suggest to fix several parameters and do PMOR on the other parameters to

obtain a reduced model that approximates the original model on a subspace of interest. If a parameter is allowed to change in the reduced model, we call it a *free parameter*; otherwise, we call it a *fixed parameter*.

To efficiently increase the performance of Quasi-Newton type optimization, (P)MOR should also be able to approximate gradients. As is well known, Krylov Padé type MOR methods have a moment matching property for the output. More recently, it was proven that the gradients are matched in the interpolation points [13, 1]. However, for optimization problems, a reduced model should be used to compute the gradients of several iteration points other than the interpolation point. We will prove that the first order derivatives also satisfy a moment matching property in two-sided (P)MOR methods, both for free parameters and for fixed parameters. This means that we can compute the derivatives accurately via the reduced model around the interpolation point, and therefore, we can use the reduced model for Quasi-Newton type optimization. Note that Hessians do not have a moment matching property unless we reduce on all parameters, so Newton-type methods are not favorable. Another advantage of Quasi-Newton methods is that they are effective even when the object function is non-smooth [16, 29], as is the case for minimax optimization. To guarantee convergence, we use a backtracking strategy with the Armijo condition [24]. We then propose two general frameworks to solve optimal design problems:

1. *MOR Framework* only takes ω as a free parameter. It builds a new reduced model for each different value of the parameter vectors;

2. *PMOR Framework* also takes the line search direction as a free parameter. It uses the same reduced model for the computation of all FRFs whose design parameters lie on the search direction in the parameter space. It applies to linear systems that are polynomially dependent on design parameters.

When the objective function is smooth as is the case in 2-norm optimization, the probability that the first line search point satisfies the Armijo condition is high for Quasi-Newton type optimization methods within the convex region. This implies that backtracking is not often carried out, so that a reduced model in the search direction can seldom be exploited. Therefore, MOR Framework appears to be more efficient than PMOR Framework. If the objective function is non-smooth, particularly when the optimizer is a kink point as is the case in ∞-norm optimization, a line search step may contain many backtracking steps so that the Quasi-Newton method can greatly benefit from a PMOR reduced model valid for the search direction. PMOR Framework is also more suitable for the systems containing only one design parameter.

The paper is organized as follows. In §2, we formulate the optimization problem, present two benchmark problems, and review well established Quasi-Newton optimization methods. Section 3 discusses the moment matching properties for the derivatives, which are not widely known properties of MOR and PMOR methods. In §4, we fit MOR and PMOR into the Quasi-Newton type optimization methods. We also introduce a two-sided PIMTAP method that is used for matching moments of the gradients. In §5, we present the numerical examples for the proposed methods. We conclude the paper in §6.

Throughout the paper, we use lower case letters to denote vectors, and upper case letters to denote 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. The real part of a complex number z is denoted by $\Re\{z\}$. By convention, $\hat{\cdot}$ in the reduced model corresponds to $\cdot$ in the original model.

2 Optimal Design of Vibration Systems

In this section, we introduce optimal design problems arising from mechanical systems. The usual way to study mechanical systems is to analyze their finite element or finite difference discretized models. If we analyze the resulting ODE models in the frequency domain, we need to solve large-scale algebraic systems, the solution of which is very expensive. Consider the classical vibration system

$$\begin{cases} \big(K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma)\big)x(\omega,\gamma) = f, \\ y(\omega,\gamma) = \ell^* x(\omega,\gamma), \end{cases} \tag{1}$$

where $K(\gamma)$, $C(\gamma)$ and $M(\gamma)$ are parameterized $n \times n$ matrices representing stiffness, damping and mass respectively, $f \in \mathbb{C}^n$ is its excitation, $\ell \in \mathbb{C}^n$ is its output vector, x is the state vector, $\gamma \in \mathbb{C}^\nu$ denotes ν design parameters, ω is the frequency and y is the system output. For any fixed γ_0, $|y(\omega,\gamma_0)|^2$ represents an FRF[1]. The optimization problem of (1) can be formulated as

$$\min_\gamma g(\gamma), \quad \text{with } g(\gamma) = T(y(\omega,\gamma)), \tag{2}$$

---

[1]This is actually a squared FRF, we call it an FRF throughout this paper for simplicity

where $T(y(\omega, \gamma))$ picks out the criterion $g(\gamma)$ that we are interested in from the FRF corresponding to the parameter value γ, such as its average vibration amplitude over a frequency range. Our objective is to minimize $g(\gamma)$; that is, to find the optimal γ value whose corresponding FRF meets our objective best. In this paper, we concentrate on two popular settings of $T(y(\omega, \gamma))$:

- **2-norm:** $g_2(\gamma) = \int_{\omega_L}^{\omega_H} |y(\omega, \gamma)|^2 \, d\omega$, where we minimize the energy norm of the FRF;

- **∞-norm:** $g_\infty(\gamma) \triangleq \max\limits_{\omega \in [\omega_L, \omega_H]} |y(\omega, \gamma)|^2$, where we minimize the highest peak of the FRF.

For a parameterized matrix $A(\gamma) = \big[a_{i,j}(\gamma)\big]_{n \times n}$, we define its derivative $\frac{\partial A(\gamma)}{\partial \gamma_i} = \left[\frac{\partial a_{i,j}(\gamma)}{\partial \gamma_i}\right]_{n \times n}$. Then we can easily derive the formulae for computing function values and derivatives

$$
\begin{aligned}
y &= \ell^* \big(K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma)\big)^{-1} f \\
\frac{\partial y}{\partial \omega} &= \ell^* \big(K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma)\big)^{-1} \big(2\omega M(\gamma) - iC(\gamma)\big)\big(K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma)\big)^{-1} f \\
\frac{\partial y}{\partial \gamma_j} &= \ell^* \big(K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma)\big)^{-1} \left(-\frac{\partial K(\gamma)}{\partial \gamma_j} - i\omega \frac{\partial C(\gamma)}{\partial \gamma_j} + \omega^2 \frac{\partial M(\gamma)}{\partial \gamma_j}\right)\big(K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma)\big)^{-1} f
\end{aligned}
\tag{3}
$$

$$
|y|^2 = y^* y, \qquad \frac{\partial |y|^2}{\partial \omega} = 2\Re\left\{y^* \frac{\partial y}{\partial \omega}\right\}, \qquad \frac{\partial |y|^2}{\partial \gamma_j} = 2\Re\left\{y^* \frac{\partial y}{\partial \gamma_j}\right\}
$$

From the above formulae, we can see that computing y, $\frac{\partial y}{\partial \omega}$ and $\frac{\partial y}{\partial \gamma_i}$ at a given point (ω_0, γ_0) share the same computationally dominant part: the LU factorization of the matrix $K(\gamma_0) + i\omega_0 C(\gamma_0) - \omega_0^2 M(\gamma_0)$ and the linear solves with f and ℓ. Therefore, if we want to use MOR to accelerate this Quasi-Newton type optimization efficiently, we must use the reduced model to compute both y and its derivatives.

In the remainder of this section, we give two examples of the general system (1).

2.1 Acoustic Cavity Optimization with One Design Parameter

First, we discuss an optimal design problem of a unit cubic acoustic cavity with one design parameter. We imposed homogeneous boundary conditions on all faces except one where we imposed an admittance boundary condition. The PDE model of this problem is

$$
\begin{cases}
-\nabla^2 u + \left(\frac{\omega}{c}\right)^2 u = f, & \\
\frac{\partial u}{\partial n} + i\omega\gamma\, u = 0, & \text{for } u \in \Gamma_1, \\
u = 0, & \text{for } u \in \Gamma_2,
\end{cases}
\tag{4}
$$

where u, ω, c, f, n, and γ denote displacement, the frequency, the speed of sound, the excitation, the normal direction, and the admittance ratio, respectively, Γ_1 denotes the face with admittance boundary condition and Γ_2 denotes the five other faces. We discretized the unit cube with finite differences and analyzed the resulting system in the frequency domain. The frequency range of interest is $[\omega_L, \omega_H] = [9.5\,\text{Hz}, 11.5\,\text{Hz}]$. The system output, $y(\omega, \gamma)$, is the displacement of a given point inside the cavity. The state-variable description of this dynamical system is

$$
\begin{cases}
\big(K + i\omega\gamma C - \omega^2 M\big)x(\omega, \gamma) = f, \\
y(\omega, \gamma) = \ell^* x(\omega, \gamma).
\end{cases}
\tag{5}
$$

For any fixed γ_0, $|y(\omega, \gamma_0)|^2$ defines an FRF. We show three FRFs with different γ values in Fig. 1. Our design objective is to choose γ to minimize some norm of the FRF.

2.1.1 ∞-norm optimization

The design objective in this setting is to minimize the highest peak of the FRF by choosing the optimal admittance ratio γ:

$$
\min_\gamma g_\infty(\gamma), \quad \text{where } g_\infty(\gamma) \triangleq \max_{[\omega_L, \omega_H]} |y(\omega, \gamma)|^2.
\tag{6}
$$

We can see from Fig. 1 that when γ changes, both the value and the position of the local maxima change, and even the number of local maxima may change. To study how the local maxima change w.r.t γ, we consider the necessary condition $\frac{\partial |y|^2}{\partial \omega} = 0$. This condition implicitly defines *Local MAximum Curves* (*LMAC*s) and *Local MInimum Curves* (*LMIC*s). Fig. 2 shows the LMACs and LMICs projected onto the
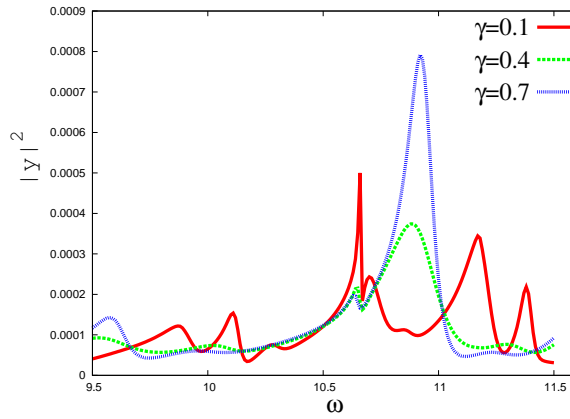
Figure 1: FRFs with different γ values

contour plot[2]. Note that an LMAC or LMIC may discontinue at certain points because of its interaction with other LMACs or LMICs.

To better understand this optimization problem, we project the LMACs onto the $|y|^2 - \gamma$ plane and this projection is shown in Fig. 3. In practice, these *Projected LMACs* (*PLMACs*) appear to be convex when γ is around the optimizer. The *Projected Global MAximum Curve* (*PGMAC*) in Fig. 3 is the maximum of all the PLMACs and it usually has non-differentiable kink points when different PLMACs intersect. Although a PLMAC may discontinue at some point, the PGMAC is defined everywhere except for $\gamma = 0$ since the global maximum of an FRF always exists provided that the FRFs do not have vertical asymptotes, which correspondes to $\gamma \neq 0$ in this example. Furthermore, the PGMAC is continuous because all PLMACs are continuous inside their defining domains.
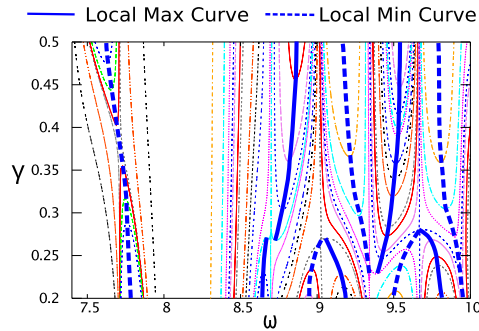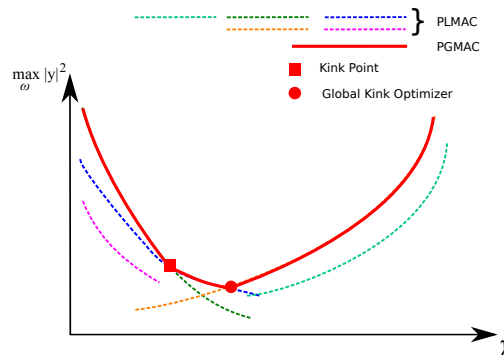

Figure 2: Contour Plot of $|y(\omega, \gamma)|^2$

Figure 3: Relationship Between Global Optimizer and Local Optimizers

---

[2]Because the contour plot for the frequency range [9.5 Hz, 11.5 Hz] is messy, we use another frequency range

2.1.2 2-norm optimization

If we use the 2-norm, the optimization problem can be formulated as

$$\min_{\gamma} g_2(\gamma), \quad \text{where } g_2(\gamma_0) \triangleq \int_{\omega_L}^{\omega_H} |y(\omega, \gamma_0)|^2 \, d\omega. \tag{7}$$
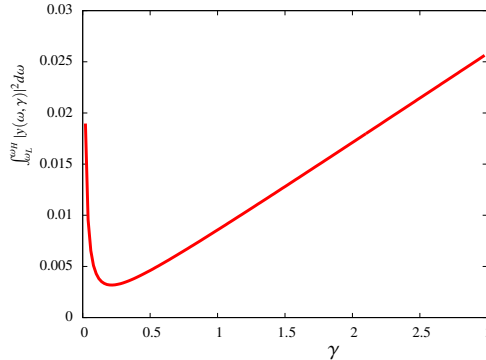

Figure 4: Plot of $g_2(\gamma)$ in 2-norm Optimization

When $\gamma \neq 0$, the corresponding FRF has no vertical asymptote and is differentiable. Therefore, the derivative of $g_2(\gamma)$ can be computed as

$$\frac{dg_2}{d\gamma} = \int_{\omega_L}^{\omega_H} 2\Re\{y^* \frac{\partial y}{\partial \gamma}\} \, d\omega. \tag{8}$$

As is shown in Fig. 4, $g_2(\gamma)$ is smoother than $g_\infty(\gamma)$ and this fact makes it easier to optimize.

2.2 Floor Damper Optimization with Two Design Parameters

In this application, we consider a floor damper whose function is to alleviate the vibration of the floor inside a building located near a highway. Its conceptual model is shown in Fig. 5. The floor is 10m×10m×0.3m in size. Its Young's modulus, Poisson's ratio, proportional damping ratio and density are 30 GPa, 0.3, 0.1 and 2500 kg/m³, respectively. The damper can be modeled as a classical stiffness-damping-mass system. Its mass m_1 is 400 kg, and our objective is to minimize the 2-norm or the ∞-norm of the output over the frequency range [2 Hz, 100 Hz] by choosing the stiffness k_1 and the damping factor c_1 of the damper.
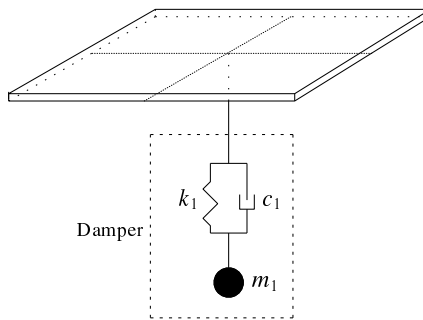
Figure 5: The Conceptual Model of the Floor Damper Optimization Problem

We use a DKT shell element finite element model [7] for the plate and obtain the following discrete system:

$$\begin{cases} \left(K_0 + (k_1 + i\omega c_1)K_1 - \omega^2 M\right) x = f, \\ \qquad\qquad\qquad\qquad\qquad y = \ell^* x, \end{cases} \tag{9}$$

where $K_0 = \begin{bmatrix} K_f & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{n \times n}$, $M_0 = \begin{bmatrix} M_f & 0 \\ 0 & m_1 \end{bmatrix} \in \mathbb{C}^{n \times n}$, and $K_1 \in \mathbb{C}^{n \times n}$. The matrix K_1 with three 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 stiffness matrix and mass matrix obtained from the shell element model of the floor respectively, and m_1 denotes the mass of the damper. The contour plot of the ∞-norm of $|y|^2$ is shown in Fig 6.

5

Figure 6: Contour Plot of $\max_{\omega \in [\omega_L, \omega_H]} |y(\omega, \gamma)|^2$ in the Floor Damper Optimization Problem

2.3 Optimization Algorithm

We divide both the 2-norm optimization and the ∞-norm optimization into two phases: the Outer Phase and the Inner Phase.

- **Outer Phase:** $\min_{\gamma} g(\gamma)$.

- **Inner Phase:** $g(\gamma) = \begin{cases} \max\limits_{\omega_L \leq \omega \leq \omega_H} |y(\omega, \gamma)|^2, & (\infty\text{-norm optimization}), \\ \int_{\omega_L}^{\omega_H} |y(\omega, \gamma)|^2 \, \mathrm{d}\omega, & (2\text{-norm optimization}). \end{cases}$

The Outer Phase optimizes $g(\gamma)$ in the parameter space, while to get the function value and its gradient for a given γ, an Inner Phase is conducted.
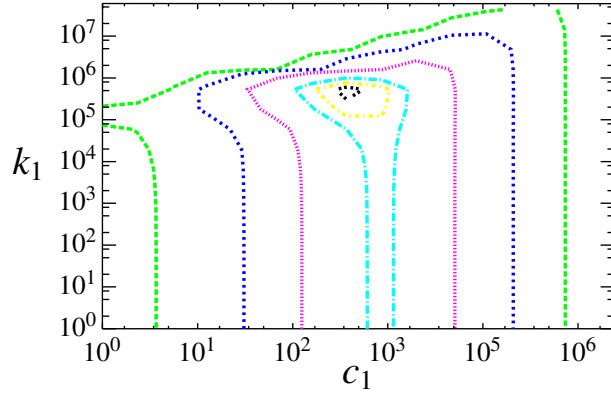
2.3.1 Inner Phase

Here we assume that for the Inner Phase discussed, γ is fixed to γ_0.

For ∞-norm optimization, the Inner Phase computes the maximum of a given FRF over the interval $[\omega_L, \omega_H]$. As is shown in Fig. 1, an FRF may have several local maxima, and to locate the global maximum, we first do a coarse grid search, and then use the point with largest $|y|^2$ among the grid points as the starting point of a one dimensional Quasi-Newton search, which approximates $\left(|y(\omega_{k+1}, \gamma)|^2 \big|_{\gamma=\gamma_0} \right)''$ by $\frac{\left(|y(\omega_{k+1}, \gamma)|^2 \big|_{\gamma=\gamma_0} \right)' - \left(|y(\omega_k, \gamma)|^2 \big|_{\gamma=\gamma_0} \right)'}{\omega_{k+1} - \omega_k}$. It is possible, although rare, that an FRF reaches its global maximum value at two or more different frequency values. In this case, our algorithm only targets at locating one of them. The Inner Phase for ∞-norm optimization is expensive because we need to compute y on a large number of grid points to avoid missing peaks and we also need to compute y and $\frac{\partial y}{\partial \omega}$ for all Quasi-Newton iteration points.

For 2-norm optimization, the Inner Phase computes the integrals of $|y(\omega, \gamma_0)|^2$ and $\frac{\partial |y(\omega, \gamma)|^2}{\partial \gamma_i}\big|_{\gamma=\gamma_0}$ $(i = 1, 2, \ldots, l)$ over the interval $[\omega_L, \omega_H]$ to compute $g(\gamma_0)$ and $\nabla g|_{\gamma=\gamma_0}$. We use the Trapezoidal rule [9] for these numerical integrations. The Inner Phase for 2-norm optimization is also expensive because we need to compute $|y(\omega, \gamma_0)|^2$ and $\frac{\partial |y(\omega, \gamma)|^2}{\partial \gamma_i}\big|_{\gamma=\gamma_0}$ $(i = 1, 2, \ldots, l)$ on a large number of ω values to guarantee the accuracy of the numerical integration.

2.3.2 Outer Phase

For ∞-norm optimization, the optimizer is often a kink point as is shown in Fig. 3. Therefore, we cannot use the usual stopping criterion $\left\| \nabla g(\gamma)|_{\gamma=\gamma_i} \right\| \leq \tau$, where γ_i represents the point returned by the i-th iteration and τ is a tolerance. Instead, we use the stopping criterion $\|\gamma_i - \gamma_{i-1}\| \leq \tau$. We can expect the Outer Phase of ∞-norm optimization to converge slower than 2-norm optimization since non-smooth optimization problems are more difficult and require more computational effort.

For the acoustic cavity optimization, we use a one-dimensional Quasi-Newton optimization method for the Outer Phase; for the floor damper optimization, we have two parameters and we have freedom

to choose which Quasi-Newton algorithm to use. The BFGS* method, which is the most popular Quasi-Newton type method, is not suitable for our application, because to make the BFGS method work, the curvature condition must be satisfied [24], i.e. the underlying function must be a quadric, but in our application, the objective function is only strictly convex in a small region around the optimizer, but semi-convex in a much larger region as is shown in Fig. 6. Thus, a violation of the curvature condition is likely to happen and the BFGS method may break down. Therefore, we chose to use the damped BFGS method that forces the curvature condition to be satisfied [24]. The damped BFGS method updates the approximate Hessian B_k in the following way.

Algorithm 2.1 (Update of B_k at the k-th Step of the Damped BFGS Method [24])
  1. Define $s_k = \gamma_{k+1} - \gamma_k$, $t_k = \nabla g(\gamma_{k+1}) - \nabla g(\gamma_k)$ and

$$\theta_k = \begin{cases} 1 & \text{if } s_k^T t_k \geq 0.2 s_k^T B_k s_k \\ \frac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T t_k} & \text{if } s_k^T t_k < 0.2 s_k^T B_k s_k \end{cases}$$

  Compute $r_k = \theta_k t_k + (1 - \theta_k) B_k s_k$.
  2. Update Hessian approximation B_k as

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{r_k r_k^T}{s_k^T r_k}$$

As was discussed in §2.1, the objective function in ∞-norm optimization often has non-differentiable kink points, which are cross-points of two or more PLMACs. Our algorithm only locates one of these PLMACs and use its derivatives for the Quasi-Newton type optimization.

2.3.3   Convergence

All Quasi-Newton steps in our algorithm use a backtracking strategy with the Armijo condition, which guarantees the convergence of our algorithm [24]. To enlarge the convergence region, when the inner product of the Quasi-Newton direction with the gradient direction is negative, we reverse the Quasi-Newton direction to ensure that the Armijo condition can be met.

3   Derivative Computations via Krylov Padé Type MOR

As was mentioned in Section 2, using MOR to accelerate parameter optimization problems can only be effective if also the derivatives are computed via the reduced model. In this section, we discuss the moment matching properties of the derivatives between the original model and the reduced model. We divide this problem into two categories: computation of derivatives w.r.t *free* variables in the reduced model and computation of derivatives w.r.t *fixed* variables in the reduced model. We will show that we have moment matching properties for the low order derivatives towards free variables for both one-sided and two-sided MOR methods, while we have moment matching properties for the first order derivatives towards fixed variables only for two-sided MOR methods.

3.1   Moment Matching Properties of Function Values

Consider the following system with l free parameters $p_1, p_2, \ldots, p_l$[3]:

$$\begin{cases} A(p_1, p_2, \ldots, p_l)x = b, \\ \qquad\qquad y = \ell^* x, \end{cases} \tag{10}$$

where $A(p_1, p_2, \ldots, p_l) : \mathbb{C}^l \to \mathbb{C}^{n \times n}$ is a matrix function, $b \in \mathbb{C}^n$ is the system input vector and $\ell \in \mathbb{C}^n$ is the system output vector.

Krylov-Padé Type MOR generates the base vectors of two subspaces: a right Krylov subspace $\mathcal{V}$ related to $A(p_1, p_2, \ldots, p_l)$ and b, and a left subspace $\mathcal{W}$ related to $A^*(p_1, p_2, \ldots, p_l)$ and ℓ. We postpone the details of the algorithms to generate these base vectors for §4.

Assume that we have generated $V, W \in \mathbb{C}^{n \times k}$, whose column vectors span $\mathcal{V}$ and $\mathcal{W}$ respectively. Define $\hat{A}(p_1, p_2, \ldots, p_l) = W^* A(p_1, p_2, \ldots, p_l)V$, $\hat{b} = W^* b$ and $\hat{\ell} = V^* \ell$. Then, we can construct the following reduced model:

$$\begin{cases} \hat{A}(p_1, p_2, \ldots, p_l)\hat{x} = \hat{b}, \\ \qquad\qquad \hat{y} = \hat{\ell}^* \hat{x}, \end{cases} \tag{11}$$

---

[3] In this paper, we set set $p_1 = \omega$ and $p_2 = \gamma_1, p_3 = \gamma_2, \ldots, p_l = \gamma_{l-1}$, where $\gamma_1, \gamma_2, \ldots, \gamma_{l-1}$ are $l-1$ free design parameters.

The key feature of Krylov methods is *moment matching*. Let the Taylor expansion of y be

$$y = \ell^* A(p_1, p_2, \ldots, p_l)^{-1} b = (A(p_1, p_2, \ldots, p_l)^{-*}\ell)^* b \triangleq \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} m(i_1, i_2, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l}, \quad (12)$$

where $m(i_1, i_2, \ldots, i_l)$ is defined as the $(i_1, i_2, \ldots, i_l)$-th moment of y. Similarly, we can define the $(i_1, i_2, \ldots, i_l)$-th moment of $\hat{y}$ and denote it by $\hat{m}(i_1, i_2, \ldots, i_l)$. We say that the $(i_1, i_2, \ldots, i_l)$-th moments of y and $\hat{y}$ match if

$$m(i_1, i_2, \ldots, i_l) = \hat{m}(i_1, i_2, \ldots, i_l).$$

Similar to (12), we can also expand x and $t \triangleq A(p_1, p_2, \ldots, p_l)^{-*}\ell$:

$$x = A(p_1, p_2, \ldots, p_l)^{-1} b \triangleq \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} r(i_1, i_2, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l}, \quad (13)$$

$$t = A(p_1, p_2, \ldots, p_l)^{-*}\ell \triangleq \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} \zeta(i_1, i_2, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l}, \quad (14)$$

We define $r(i_1, i_2, \ldots, i_l)$ as the $(i_1, i_2, \ldots, i_l)$-th moment of x, and $\zeta(i_1, i_2, \ldots, i_l)$ as the $(i_1, i_2, \ldots, i_l)$-th moment of t. We can use these moments to construct *Left Krylov Subspace*s and *Right Krylov Subspace*s.

Definition 1 *The $(i_1, i_2, \ldots, i_l)$-th right Krylov subspace $\mathcal{K}^{(r)}(i_1, i_2, \ldots, i_l)$ is defined as the subspace spanned by all $r(j_1, j_2, \ldots, j_l)$ satisfying $0 \le j_1 \le i_1$, $0 \le j_2 \le i_2$, $\ldots$, $0 \le j_l \le i_l$. Similarly, the $(i_1, i_2, \ldots, i_l)$-th left Krylov subspace $\mathcal{K}^{(l)}(i_1, i_2, \ldots, i_l)$ is defined as the subspace spanned by all $\zeta(j_1, j_2, \ldots, j_l)$ satisfying $0 \le j_1 \le i_1$, $0 \le j_2 \le i_2$, $\ldots$, $0 \le j_l \le i_l$.*

Krylov methods approximate x (or t) with a vector in a right (or left) Krylov subspace with the moment matching property, as is shown in Lemma 1.

Lemma 1 *For system (10) and (11), assume*

$$A(p_1, p_2, \ldots, p_l) = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} \mathring{A}(i_1, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l},$$

where $\mathring{A}(i_1, \ldots, i_l) \in \mathbb{C}^{n \times n}$. Assume that $W^* \mathring{A}(0, 0, \ldots, 0)V$ is nonsingular. Then, if $\mathcal{K}^{(r)}(i_1, i_2, \ldots, i_l) \subseteq \text{colspan}\{V\}$, it follows that $r(i_1, i_2, \ldots, i_l) = V\hat{r}(i_1, i_2, \ldots, i_l)$ and $m(i_1, i_2, \ldots, i_l) = \hat{m}(i_1, i_2, \ldots, i_l)$. Similarly, if $\mathcal{K}^{(l)}(i_1, i_2, \ldots, i_l) \subseteq \text{colspan}\{W\}$, it follows that $\zeta(i_1, i_2, \ldots, i_l) = W\hat{\zeta}(i_1, i_2, \ldots, i_l)$ and $m(i_1, i_2, \ldots, i_l) = \hat{m}(i_1, i_2, \ldots, i_l)$.

See Appendix A for the proof.

If a (P)MOR method generates only the left or the right Krylov subspaces, it is called a *one-sided* method, while if it generates both the left and the right Krylov subspaces, it is called a *two-sided* method. By intuition, we expect two-sided methods to match more moments. We will discuss this topic for the one-parameter case and the two-parameter case in §4.

3.2 Moment Matching Property of Derivatives w.r.t Free Variables

Theorem 2 *Let $(\xi, \hat{\xi})$ be $(y, \hat{y})$, $(x, V\hat{x})$ or $(t, W\hat{t})$ in the system pair (10) and (11). Let*

$$\xi = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} \xi(i_1, i_2, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l},$$

$$\hat{\xi} = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} \hat{\xi}(i_1, i_2, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l}.$$

Then the $(i_1, i_2, \ldots, i_l)$-th moments of $\frac{\partial^{r_1 + r_2 + \cdots + r_l} \xi}{\partial p_1^{r_1} \partial p_2^{r_2} \ldots \partial p_l^{r_l}}$ and $\frac{\partial^{r_1 + r_2 + \cdots + r_l} \hat{\xi}}{\partial p_1^{r_1} \partial p_2^{r_2} \ldots \partial p_l^{r_l}}$ match if and only if the $(i_1 + r_1, i_2 + r_2, \ldots, i_l + r_l)$-th moments of ξ and $\hat{\xi}$ match for $i_1 \ge 0$, $i_2 \ge 0$, $\ldots$, $i_l \ge 0$.

Proof:

Let

$$\frac{\partial^{r_1+r_2+\cdots+r_l}\xi}{\partial p_1^{r_1}\partial p_2^{r_2}\cdots\partial p_l^{r_l}} = \sum_{i_1=0}^{\infty}\sum_{i_2=0}^{\infty}\cdots\sum_{i_l=0}^{\infty} u(i_1,i_2,\ldots,i_l)s_1^{i_1}s_2^{i_2}\cdots s_l^{i_l},$$

$$\frac{\partial^{r_1+r_2+\cdots+r_l}\hat{\xi}}{\partial s_1^{r_1}\partial s_2^{r_2}\cdots\partial s_l^{r_l}} = \sum_{i_1=0}^{\infty}\sum_{i_2=0}^{\infty}\cdots\sum_{i_l=0}^{\infty} \hat{u}(i_1,i_2,\ldots,i_l)p_1^{i_1}p_2^{i_2}\cdots p_l^{i_l}.$$

It is easy to derive

$$u(i_1,i_2,\ldots,i_l) = \left[\prod_{j_1=i_1+1}^{i_1+r_1} j_1 \prod_{j_2=i_2+1}^{i_2+r_2} j_2 \quad,\ldots, \prod_{j_l=i_l+1}^{i_l+r_l} j_l\right] \xi(i_1+r_1,i_2+r_2,\ldots,i_l+r_l),$$

$$\hat{u}(i_1,i_2,\ldots,i_l) = \left[\prod_{j_1=i_1+1}^{i_1+r_1} j_1 \prod_{j_2=i_2+1}^{i_2+r_2} j_2 \quad,\ldots, \prod_{j_l=i_l+1}^{i_l+r_l} j_l\right] \hat{\xi}(i_1+r_1,i_2+r_2,\ldots,i_l+r_l).$$

When $i_1 \geq 0$, $i_2 \geq 0$, ..., $i_l \geq 0$, the coefficient between the brackets are nonzero, and $\xi(i_1+r_1,i_2+r_2,\ldots,i_l+r_l) = \hat{\xi}(i_1+r_1,i_2+r_2,\ldots,i_l+r_l)$ is equivalent to $u(i_1,i_2,\ldots,i_l) = \hat{u}(i_1,i_2,\ldots,i_l)$. $\square$

For reduced models that contain only one free variable like in the acoustic cavity model (5), we have the following corollary.

Corollary 3 *Under the conditions of Theorem 2, if both (10) and (11) have only one parameter s and the first k moments of ξ and $\hat{\xi}$ w.r.t s match, the first $k-r$ moments of $\frac{\partial^r \xi}{\partial s^r}$ and $\frac{\partial^r \hat{\xi}}{\partial s^r}$ $(r < k)$ w.r.t s match. If no other moments of ξ and $\hat{\xi}$ w.r.t s match, exactly $k-r$ moments of $\frac{\partial^r \xi}{\partial s^r}$ and $\frac{\partial^r \hat{\xi}}{\partial s^r}$ w.r.t s match.*

We illustrate Theorem 2 with a system parameterized with s and λ in Fig. 7, where a solid circle at (i,j) means the (i,j)-th moment is matched.

(a) Moment matching pattern for y

(b) Moment matching pattern for $\frac{\partial y}{\partial s}$

(c) Moment matching pattern for $\frac{\partial y}{\partial \lambda}$
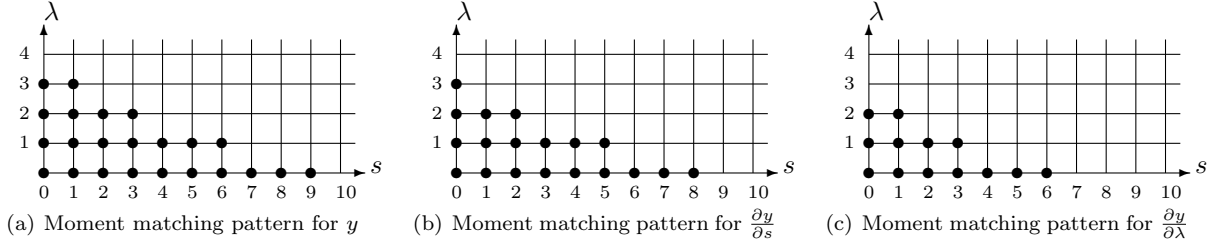
Figure 7: Illustrations of the Derivative Computation of a System Containing Two Parameters.

3.3 Moment Matching Properties the First Order Derivatives w.r.t. Fixed Variables in the Reduced Model

Assume that we also have a fixed parameter q, which is free to change in the original model but is fixed to q_0 in the reduced model.

$$\begin{cases} A(p_1,p_2,\ldots,p_l|q)x = b, \\ \qquad\qquad y = \ell^*x. \end{cases} \tag{15}$$

The first order derivative w.r.t q can be computed as

$$\left.\frac{\partial y}{\partial q}\right|_{q=q_0} = -\ell^* A(p_1,p_2,\ldots,p_l|q=q_0)^{-1}\left.\frac{\partial A(p_1,p_2,\ldots,p_l|q)}{\partial q}\right|_{q=q_0} A(p_1,p_2,\ldots,p_l|q=q_0)^{-1}b. \tag{16}$$

To see the moment matching properties between $\left.\frac{\partial y}{\partial q}\right|_{q=q_0}$ and $\left.\frac{\partial \hat{y}}{\partial q}\right|_{q=q_0}$, we prove the following more general theorem.

Theorem 4 *Consider the system pair (10) and (11). Assume*

$$A(p_1,p_2,\ldots,p_l) = \sum_{i_1=0}^{\infty}\sum_{i_2=0}^{\infty}\cdots\sum_{i_l=0}^{\infty} \mathring{A}(i_1,\ldots,i_l)p_1^{i_1}p_2^{i_2}\cdots p_l^{i_l},$$

where $\mathring{A}(i_1, \ldots, i_l) \in \mathbb{C}^{n \times n}$ and $W^* \mathring{A}(0, 0, \ldots, 0)V$ is nonsingular. Let $B(p_1, p_2, \ldots, p_l)$ be a matrix function and define $\hat{B}(p_1, p_2, \ldots, p_l) = W^* B(p_1, p_2, \ldots, p_l)V$. Then, if $\mathcal{K}^{(r)}(i_1, i_2, \ldots, i_l) \subseteq \mathrm{colspan}\{V\}$ and $\mathcal{K}^{(l)}(i_1, i_2, \ldots, i_l) \subseteq \mathrm{colspan}\{W\}$, it follows that the $(i_1, i_2, \ldots, i_l)$-th moment of

$$\ell^* A(p_1, p_2, \ldots, p_l)^{-1} B(p_1, p_2, \ldots, p_l) A(p_1, p_2, \ldots, p_l)^{-1} b, \tag{17}$$

denoted by $m^B(i_1, i_2, \ldots, i_l)$, matches the $(i_1, i_2, \ldots, i_l)$-th moment of

$$\hat{\ell}^* \hat{A}(p_1, p_2, \ldots, p_l)^{-1} \hat{B}(p_1, p_2, \ldots, p_l) \hat{A}(p_1, p_2, \ldots, p_l)^{-1} \hat{b}, \tag{18}$$

denoted by $\hat{m}^B(i_1, i_2, \ldots, i_l)$.

Proof:

According to (13) and (14)

$$\ell^* A(p_1, p_2, \ldots, p_l)^{-1} = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \cdots \sum_{i_l=0}^{\infty} \zeta^*(i_0, i_1, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l} \tag{19}$$

$$A(p_1, p_2, \ldots, p_l)^{-1} b = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \cdots \sum_{i_l=0}^{\infty} r(i_0, i_1, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l} \tag{20}$$

Let the Taylor expansion of $B(p_1, p_2, \ldots, p_l)$ be

$$B(p_1, p_2, \ldots, p_l) = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \cdots \sum_{i_l=0}^{\infty} \mathring{B}(i_0, i_1, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l}. \tag{21}$$

As (17) is the product of (19), (21) and (20),

$$m^B(i_1, i_2, \ldots, i_l)$$
$$= \sum_{\alpha_1+\beta_1+\delta_1=i_1} \sum_{\alpha_2+\beta_2+\delta_2=i_2} \cdots \sum_{\alpha_l+\beta_l+\delta_l=i_l} \zeta^*(\alpha_1, \alpha_2, \ldots, \alpha_l) \mathring{B}(\beta_1, \beta_2, \ldots, \beta_l) r(\delta_1, \delta_2, \ldots, \delta_l) \tag{22}$$

According to Lemma 1, for all $\alpha_1 \leq i_1, \alpha_2 \leq i_2, \ldots \alpha_l \leq i_l, \delta_1 \leq i_1, \delta_2 \leq i_2, \ldots, \delta_l \leq i_l$ and arbitrary $\beta_1, \beta_2, \ldots, \beta_l$, we have

$$\zeta^*(\alpha_1, \alpha_2, \ldots, \alpha_l) \mathring{B}(\beta_1, \beta_2, \ldots, \beta_l) r(\delta_1, \delta_2, \ldots, \delta_l)$$
$$= \hat{\zeta}^*(\alpha_1, \alpha_2, \ldots, \alpha_l) W^* \mathring{B}(\beta_1, \beta_2, \ldots, \beta_l) V \hat{r}(\delta_1, \delta_2, \ldots, \delta_l)$$
$$= \hat{\zeta}^*(\alpha_1, \alpha_2, \ldots, \alpha_l) \hat{\mathring{B}}(\beta_1, \beta_2, \ldots, \beta_l) \hat{r}(\delta_1, \delta_2, \ldots, \delta_l).$$

Therefore, all terms in (22) are matched by the reduced model and

$$m^B(i_1, i_2, \ldots, i_l) = \hat{m}^B(i_1, i_2, \ldots, i_l).$$

$\square$

The moment matching properties for derivatives are direct results of Theorem 4 because the right hand side of (16) is independent of q and thus, it is only a special case of (17). Note that sometimes more moments may be matched in Theorem 4, such as the cases when $\mathring{B}(0, 0, \ldots, 0) = 0$. Although Theorem 4 also holds for free parameters, Theorem 2 gives more precise results. For example, if we reduce a system with no design parameter and both the left and the right Krylov subspaces are of order k, we know from Theorem 5 in §4 that the first $2k$ moments of y and $\hat{y}$ match, and then according to Theorem 2, the first $2k - 1$ moments of the first order derivative match, but Theorem 4 can only tell us that the first k moments match.

4 Two Frameworks to Use MOR in Accelerating Optimization

As was analyzed in §2, parameter optimization of large scale linear systems is computationally expensive. It requires a large amount of function and derivative evaluations of a large-scale system. Our idea is to use two-sided Krylov-Padé type (P)MOR to reduce the computational cost since it can accurately compute both the function values and gradients thanks to the moment matching properties. In this section, we first give a brief introduction to Krylov-Padé type (P)MOR methods and then propose two frameworks that apply them to Quasi-Newton type optimization algorithms.

4.1 Background: Krylov Subspace

Given a matrix $A \in \mathbb{C}^{n \times n}$ and a vector $b \in \mathbb{C}^n$, the k-th *Krylov subspace* is defined as

$$\mathcal{K}_k(A, b) = \text{span}\{b, Ab, A^2 b, \ldots, A^{k-1} b\}, \quad (k \leq n \text{ and in most cases } k \ll n).$$

It plays an important role in linear equations solving, algebraic eigenvalue problems, MOR, etc. Krylov subspace methods are very suitable for large scale problems because only matrix-vector multiplication is required and consequently, the generation of Krylov subspaces is computationally relatively cheap. However, explicitly computing the Krylov vectors $A^i b$ $(0 \leq i \leq k-1)$ to generate the Krylov subspace is numerically unstable because these vectors tend very quickly to become almost linearly dependent. The *Arnoldi process* [2] is a numerically stable scheme that generates an orthonormal basis for $\mathcal{K}_k(A, b)$.

In this paper, the term "Krylov subspace" is also used in a more general sense, such as the generalization of Krylov subspace for second order systems and parameterized systems.

First, we examine the simplest case: Krylov-Padé type MOR on a first order linear system with one parameter

$$\begin{cases} (K - \alpha M)x = b, \\ y = \ell^* x, \end{cases} \tag{23}$$

where $K, M \in \mathbb{C}^{n \times n}$, $b, \ell \in \mathbb{C}^n$, and K is nonsingular.

For this system,

$$y = \ell^* (K - \alpha M)^{-1} b = \sum_{i=0}^{\infty} \left[\ell^* (K^{-1} M)^i K^{-1} b \right] \alpha^i, \tag{24}$$

and according to Definition (12), the i-th moment m_i is

$$m_i = \ell^* \left[(K^{-1} M)^i K^{-1} b \right] = \left[(K^{-*} M^*)^i K^{-*} \ell \right]^* b. \tag{25}$$

Similarly, we can also derive the i-th moment of x and t according to (13) and (14)

$$r_i = (K^{-1} M)^i K^{-1} b, \quad \zeta_i = (K^{-*} M^*)^i K^{-*} \ell. \tag{26}$$

Based on Definition 1, the k-th right Krylov subspace is $\mathcal{K}_k(K^{-1} M, K^{-1} b)$ and the k-th left Krylov subspace is $\mathcal{K}_k(K^{-*} M^*, K^{-*} \ell)$. Then according to Lemma 1, we can use them to build a reduced model with moment matching property. In general, we have the following theorem.

Theorem 5 *For the first order system (23), if* $\text{colspan}\{V_k\} \supseteq \mathcal{K}_{k_1}(K^{-1} M, K^{-1} b)$, $\text{colspan}\{W_k\} \supseteq \mathcal{K}_{k_2}(K^{-*} M^*, K^{-*} \ell)$ $(k_1, k_2 \leq k)$ *and* $W_k^* K V_k$ *is nonsingular, we have that*

1. $(K^{-1} M)^i K^{-1} b = V_k (\hat{K}^{-1} \hat{M})^i \hat{K}^{-1} \hat{b}$ *for all* $0 \leq i \leq k_1 - 1$;

2. $(K^{-*} M^*)^i K^{-*} \ell = W_k (\hat{K}^{-*} \hat{M}^*)^i \hat{K}^{-*} \hat{\ell}$ *for all* $0 \leq i \leq k_2 - 1$;

3. $\hat{m}_i = m_i$ *for all* $0 \leq i \leq k_1 + k_2 - 1$.

See [26] for a proof of Theorem 5. Due to Theorem 5, $\hat{y}$ is a Padé type approximation of y [10]. So when α is small, $\hat{y}$ is a good approximation of y.

Shifting is important for real world applications. For example, if we want to compute $y(\alpha)$ for several α's clustered around σ (not an eigenvalue) far away from the origin, the first equation of the linear system (23) is often preconditioned by left multiplying $(K - \sigma M)^{-1}$ to shift the interpolation point from the origin to σ to increase accuracy [21]. Note that when K is singular, shift is required to make the Arnoldi process applicable.

4.2 MOR Framework

In this part, we develop the MOR Framework that computes one reduced model that fixes all design parameters for the solution of each point in the parameter space, which uses a whole Inner Phase to calculate the 2-norm or the ∞-norm of the corresponding FRF. We first introduce SOAR, which is a modification of the Arnoldi process for second order systems. Then we discuss the details of the MOR Framework.

4.2.1 SOAR

To reduce the second order system

$$\begin{cases} (K + i\omega C - \omega^2 M)x = f, \\ y = \ell^* x, \end{cases} \qquad (K, C, M \in \mathbb{C}^{n \times n} \text{ and } l, f \in \mathbb{C}^n) \tag{27}$$

we transform it to the first order system

$$\begin{cases} \left(\begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix} - \omega \begin{bmatrix} -iC & M \\ I & 0 \end{bmatrix} \right) \begin{bmatrix} x \\ \omega x \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \\ y = [\ell^*, 0] \begin{bmatrix} x \\ \omega x \end{bmatrix}, \end{cases} \tag{28}$$

on which an Arnoldi process can be applied. The disadvantage of this approach is that: 1.) the dimension of the system order is doubled; 2.) the second order system is reduced to a first order system whose physical meaning is not so clear. Second Order ARnoldi (SOAR) [28, 5, 4, 3, 26] is an improvement of this approach. Rather than reducing the equivalent first order system, SOAR directly reduce (27) to a low order second order system. SOAR is therefore called *structure preserving*. SOAR actually projects the second order system to a *second order Krylov subspace* [5, 4, 3, 26], which contains the subspace generated by the Arnoldi process. This means: 1.) the SOAR reduced model is expected to be at least as accurate as the Arnoldi reduced model; 2.) SOAR inherits the moment matching properties from the Arnoldi Process: if the left second order Krylov subspace is of order k_1 and the right second order Krylov subspace is of order k_2, then the first $k_1 + k_2$ moments of y and $\hat{y}$ match.

4.2.2 MOR Framework: One Reduced Model for Each Inner Phase

Since in an Inner Phase, all design parameters are fixed, we can use SOAR to reduce on ω. This method is supposed to be efficient since the computational cost of an Inner Phase is high. For ∞-norm optimization, it includes

- Computing y for all grid points in the coarse grid search;

- Computing y and $\frac{\partial y}{\partial \omega}$ for each step in the one-dimensional Quasi-Newton search;

- Computing ∇g $(g(\gamma) = \max\limits_{[\omega_L, \omega_H]} |y(\omega, \gamma)|^2)$ at the optimizer found.

For 2-norm optimization, it includes

- Computing y and $\frac{\partial y}{\partial \gamma_i}$ $(i = 1, 2, \ldots, l)$ for all interpolation points in numerical integration.

The MOR Framework use a SOAR reduced model for each Inner Phase so that all these computations within an Inner Phase can be done with a single reduced model. Therefore, the optimization performance is expected to increase.

4.3 PMOR Framework

PMOR is a natural extension of MOR to accelerate the solution of linear systems with design parameters. It can reduce a system on a larger subspace than MOR. In this subsection, we first give a brief introduction to PIMTAP, a representative of PMOR methods. To use PMOR for computing derivatives, we then propose two-sided PIMTAP and discuss its moment matching properties. Finally we propose the PMOR Framework.

4.3.1 PIMTAP

PMOR is complicated both in complexity and numerical stability. Pioneering work includes [31, 14, 30, 8, 17, 11, 12]. PIMTAP (Parameterized Interconnect Macromodeling via a Two-directional Arnoldi Process) [20, 19, 18] provides a flexible way for reducing linear systems with multiple parameters. The linear system that PIMTAP reduces in the two-parameter case is:

$$\begin{cases} \big(G_0 + \lambda G_1 + s(C_0 + \lambda C_1)\big)x = b, \\ \qquad\qquad\qquad y = \ell^* x, \end{cases} \qquad (G_0, G_1, C_0, C_1 \in \mathbb{C}^{n \times n}, \, b, \ell \in \mathbb{C}^n). \tag{29}$$

Let the Taylor series of x in terms of s and λ be $x = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} r_i^j s^i \lambda^j$ $(r_i^j \in \mathbb{C}^n)$, then according to (13), r_i^j is the (i,j)-th 2-parameter moment of x. Substituting x with this Taylor series in (29) and comparing the left hand side with the right hand side leads to the following recursive relationship of the moments:

$$\begin{cases} r_i^j = 0, & \text{if } i < 0 \text{ or } j < 0; \\ r_i^j = G_0^{-1} b & \text{if } i = 0 \text{ and } j = 0; \\ r_i^j = -G_0^{-1}(C_0 r_{i-1}^j + G_1 r_i^{j-1} + C_1 r_{i-1}^{j-1}), & \text{otherwise.} \end{cases} \tag{30}$$

PIMTAP projects (29) to a subspace $\mathcal{V}$ spanned by a selection of the moments. According to Definition 1, the (p,q)-th right Krylov subspace is $\mathcal{V}(p,q) = \mathrm{span}\{r_i^j \,|\, i = 0 : p-1, j = 0 : q-1\}$, If we define $r_{[i]}^{[j]}$, $G_{[j]}$ and $C_{[j]}$ as

$$r_{[i]}^{[j]} = \begin{bmatrix} r_{i-1}^0 \\ r_{i-1}^1 \\ \vdots \\ r_{i-1}^{j-1} \end{bmatrix}, \quad G_{[j]} = \underbrace{\begin{bmatrix} G_0 & & & \\ G_1 & G_0 & & \\ & G_1 & G_0 & \\ & & \ddots & \ddots & \\ & & & G_1 & G_0 \end{bmatrix}}_{j \text{ blocks}}, \quad C_{[j]} = \underbrace{\begin{bmatrix} C_0 & & & \\ C_1 & C_0 & & \\ & C_1 & C_0 & \\ & & \ddots & \ddots & \\ & & & C_1 & C_0 \end{bmatrix}}_{j \text{ blocks}},$$

the recursive relationship (30) could be rearranged as

$$r_{[i]}^{[j]} = -G_{[j]}^{-1} C_{[j]} r_{[i-1]}^{[j]}, \quad \text{for all } i > 1. \tag{31}$$

From the relationship (31), we can see that $r_{[1]}^{[j]}, r_{[2]}^{[j]}, \ldots, r_{[k]}^{[j]}$ span $\mathcal{K}_k \left\{ -G_{[j]}^{-1} C_{[j]}, r_{[1]}^{[j]} \right\}$, which can be generated with the standard Arnoldi process. We can see that each Krylov vector of $\mathcal{K}_k \left\{ -G_{[j]}^{-1} C_{[j]}, r_{[1]}^{[j]} \right\}$ can be split into j base vectors of the (k,j)-th right Krylov subspace $\mathcal{K}^{(r)}(k,j)$. Thus, we can obtain all $k \times j$ base vectors of $\mathcal{K}^{(r)}(k,j)$. In this way, PIMTAP can generate a rectangular moment matching pattern such as the one shown in Fig 8(a). Non-rectangular moment matching patterns are sometimes also required because in some applications, the high order cross-term moments are not so important. An extreme example is the method proposed in [14] that matches none of the cross-term moments but is still accurate enough for its application [14]. However, for Krylov-Padé type methods, the moment matching pattern should be the union of several rectangles such as the one shown in Fig. 8(b), to have moment matching properties due to Lemma 1. This kind of patterns could be expressed by a vector $p \in \mathbb{N}^k$ satisfying $0 < p(i) \le p(j)$ when $i > j$, meaning that r_i^j is matched if and only if $j \le k$ and $i \le p(j) - 1$. For example, the moment matching vector of the example in Fig. 8(b) is $(10, 7, 4, 2)$. For non-rectangular patterns, PIMTAP uses TAP [18], which can be regarded as an efficient rearrangement of several Arnoldi processes, to generate the Krylov subspaces.

(a) The moment matching pattern of PIMTAP via Arnoldi Process. The matched moments spans $\mathcal{V}(10, 4)$.

(b) An example of moment matching pattern for PIMTAP via TAP. The moment matching vector is $(10, 7, 4, 2)$.
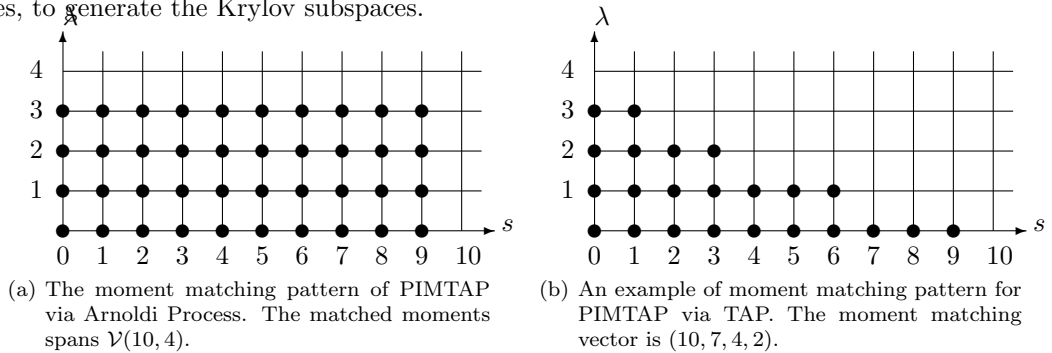
Figure 8: Illustrations of the moment matching patterns of PIMTAP.

4.3.2 Two-sided PIMTAP

Since PIMTAP can be viewed as a rearrangement of several Arnoldi Processes, we can expect that two sided methods also work for PIMTAP. Expand $t \triangleq \left(G_0 + \lambda G_1 + s(C_0 + \lambda C_1) \right)^{-*} \ell$ as $\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \zeta_i^j s^i \lambda^j$, where ζ_i^j is the (i,j)-th moment of t due to (14). Define $\zeta_{[i]}^{[j]} = \left[(\zeta_{-1}^0)^T, (\zeta_{-1}^1)^T, \ldots, (\zeta_{-1}^{j-1})^T \right]^T$. The recursive

relationship of $\zeta_{[i]}^{[j]}$ is $\zeta_{[i]}^{[j]} = -G_{[j]}^{-*}C_{[j]}^{*}\zeta_{[i-1]}^{[j]}$. Therefore, $\zeta_{[1]}^{[j]}, \zeta_{[2]}^{[j]}, \ldots, \zeta_{[k]}^{[j]}$ spans $\mathcal{K}_k\{-G_{[j]}^{-*}C_{[j]}^{*}, \zeta_{[1]}^{[j]}\}$, with whose Krylov vectors we can obtain the base vectors of the (k,j)-th left Krylov subspace $\mathcal{K}^{(l)}(k,j)$.

Using the PIMTAP to obtain $V \in \mathbb{C}^{n \times k}$ and $W \in \mathbb{C}^{n \times k}$, whose column vectors form orthonormal bases for the Krylov subspaces specified by the left and right moment matching pattern respectively, we obtain a two-sided PIMTAP reduced order model

$$\begin{cases} (\hat{G}_0 + \lambda\hat{G}_1 + s(\hat{C}_0 + \lambda\hat{C}_1))x = b, \\ \hat{y} = \hat{\ell}^*x, \end{cases} \tag{32}$$

where $\hat{G}_0 = W^*G_0V$, $\hat{G}_1 = W^*G_1V$, $\hat{C}_0 = W^*C_0V$, $\hat{C}_1 = W^*C_1V$, $\hat{b} = W^*b$, $\hat{\ell} = V^*\ell$. The following theorem shows the moment matching properties of two-sided PIMTAP.

Theorem 6 *Assume $\hat{G}$ in (32) to be nonsingular, then, the $(i-1, j-1)$-th moment of y in the two-sided PIMTAP reduced model and the original model match if any of the following three conditions is satisfied:*

1. *$\mathcal{K}^{(l)}(i,j) \subseteq \text{colspan}\{W\}$ or $\mathcal{K}^{(r)}(i,j) \subseteq \text{colspan}\{V\}$;*

2. *there exists $i_l, i_r \geq 0$, such that the $\mathcal{K}^{(l)}(i_l,j) \subseteq \text{colspan}\{W\}$, $\mathcal{K}^{(r)}(i_r,j) \subseteq \text{colspan}\{V\}$ and $i_l + i_r = i$;*

3. *there exists $j_l, j_r \geq 0$, such that the $\mathcal{K}^{(l)}(i,j_l) \subseteq \text{colspan}\{W\}$, $\mathcal{K}^{(r)}(i,j_r) \subseteq \text{colspan}\{V\}$ and $j_l + j_r = j$.*

Proof:

1. A direct deduction of Lemma 1.

2. A direct deduction of the last statement of Theorem 5.

3. Change the roles of λ and ω in PIMTAP. Define $r_{h[i]}^{[j]} = \left[(r_0^{j-1})^T, (r_1^{j-1})^T, \ldots, (r_{i-1}^{j-1})^T\right]^T$ and we can find a relationship similar to (31), which we denote as $r_{h[i]}^{[j]} = -G_{h[j]}^{-1}C_{h[j]}r_{h[i]}^{[j-1]}$, for all $i > 1$. This relationship provides an alternative implementation to generate the 2-parameter Krylov subspace. Both of the two implementations generate the same subspace, the subspace spanned by the moments to be matched. Using the result in case 2 for these alternative implementations for $\mathcal{K}^{(l)}(i,j_l)$ and $\mathcal{K}^{(r)}(i,j_r)$, the $(i-1, j-1)$-th moments of y and $\hat{y}$ match according to Theorem 5.

$\square$

Theorem 6 shows that if the left Krylov space and the right Krylov space use the same rectangular moment matching pattern that matches k moments, then $3k$ moments of y are matched. When the moment matching pattern is not rectangular, the number of matched moments of y is less than $3k$. Fig 9 gives an example where the left Krylov subspace and the right Krylov subspace have different moment matching patterns. Actually, using the same moment matching pattern for both Krylov subspaces is favorable because according to Theorem 4, more moments of the first order derivatives are matched. This is what we do in PMOR Framework.

(a) Moment matching pattern for the left Krylov subspace

(b) Moment matching pattern for the right Krylov subspace

(c) Moment matching pattern for y
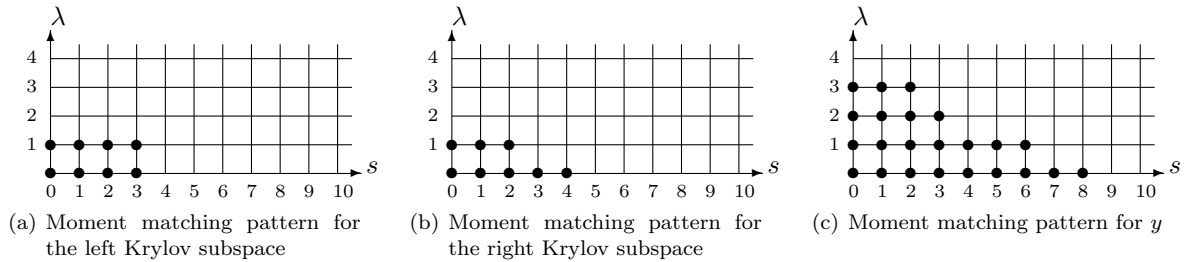
Figure 9: Illustrations of the moment matching property of two-sided PIMTAP

4.3.3 PMOR Framework: One Reduced Model for Each Line Search Iteration in the Outer Phase

Since PIMTAP can reduce on more than one parameter, it can accelerate the optimization performance more efficiently. For the acoustic cavity optimization problem, we can reduce on both $i\omega\gamma$ and $-\omega^2$ and the reduced system can be used for the whole optimization. In general, however, we may have several parameters and it is impractical to reduce on all of them. An alternative is to reduce on a selection of parameters or linear combinations of them. To accelerate the Inner Phase, we must reduce on ω. We have

the freedom to choose design parameters or linear combinations of the design parameters to reduce. One choice to integrate PIMTAP and Outer Phase optimization is to use line search optimization algorithm and perform PIMTAP reduction on the search direction. Assume that the optimizer at iteration j is $\gamma^{(j)}$ and the search direction for the next step is $d^{(j)}$. System (5) is reformulated for $\gamma = \gamma^{(j)} + \alpha d^{(j)}$ as

$$\begin{cases} \left(K(\gamma^{(j)} + \alpha d^{(j)}) + i\omega C(\gamma^{(j)} + \alpha d^{(j)}) - \omega^2 M(\gamma^{(j)} + \alpha d^{(j)})\right)x(\omega, \gamma^{(j)} + \alpha d^{(j)}) = f, \\ y(\omega, \gamma) = \ell^* x(\omega, \gamma^{(j)} + \alpha d^{(j)}). \end{cases} \quad (33)$$

Assume that $K(\gamma), C(\gamma)$ and $M(\gamma)$ are all matrix polynomials of γ whose degrees are less than or equal to q. We expand $K(\gamma^{(j)} + \alpha d^{(j)}) = \sum_0^q K_i \alpha^i$, $C(\gamma^{(j)} + \alpha d^{(j)}) = \sum_0^q C_i \alpha^i$, $M(\gamma^{(j)} + \alpha d^{(j)}) = \sum_0^q M_i \alpha^i$, linearize the system (33) similar as what we have done in (28), and obtain the following linearized system:

$$\begin{cases} \left(\tilde{K} + \alpha \tilde{C} + i\omega \tilde{M}\right)\tilde{x} = f, \\ y = \tilde{\ell}^* \tilde{x}. \end{cases} \quad (34)$$

where

$$\tilde{K} = \left[\begin{array}{c|c|c|c} K_0 & 0_{n\times(q-1)n} & C_0 & 0_{n\times qn} \\ \hline & I_{2qn\times 2qn} \end{array}\right],$$

$$\tilde{x} = \begin{bmatrix} x \\ \alpha x \\ \vdots \\ \alpha^{q-1}x \\ \hline i\omega x \\ i\omega\alpha x \\ \vdots \\ i\omega\alpha^q x \end{bmatrix}, \tilde{\ell} = \begin{bmatrix} \ell \\ 0 \\ \vdots \\ 0 \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$\tilde{C} = \left[\begin{array}{ccccc|ccccc|c} K_1 & K_2 & \dots & K_q & C_1 & C_2 & \dots & C_q & 0_{n\times n} \\ \hline -I_{(q-1)n\times(q-1)n} & & & & & & & \\ \hline & & & 0_{qn\times(2q+1)n} & & & & \\ \hline & & & & & & -I_{n\times n} \end{array}\right],$$

$$\tilde{M} = \left[\begin{array}{c|ccc} 0_{n\times qn} & M_0 & \dots & M_q \\ \hline 0_{(q-1)n\times(2q+1)n} & & & \\ \hline -I_{qn\times qn} & & & \\ \hline 0_{n\times(2q+1)n} & & & \end{array}\right].$$

The linearization is not unique, and for specific problems, it is possible to find a linearization with lower order. For the linearization (34), if K_0 is nonsingular, then $\tilde{K}$ is also nonsingular and we can use two-sided PIMTAP to reduce on α and $i\omega$. According to Theorem 3 and Theorem 4, we can compute g and ∇g via the reduced model with moment matching properties. Therefore, we can expect two-sided PIMTAP to work well in accelerating a line search iteration in the Outer Phase.

For the floor damper model (9), $K_0 = \begin{bmatrix} K_f & 0 \\ 0 & 0 \end{bmatrix}$ is singular, so we shift the interpolation point to ω_0 first to make MOR possible. In the j-th iteration, given the current point $\left(k_1^{(j)}, c_1^{(j)}\right)$ and the search direction $\left(d_1^{(j)}, d_2^{(j)}\right)$, the system (9) can be rewritten as

$$\begin{cases} \left(\begin{bmatrix} K_{11} & 0 \\ 0 & I \end{bmatrix} + \alpha \begin{bmatrix} C_{11} & C_{12} \\ 0 & 0 \end{bmatrix} + i(\omega - \omega_0)\begin{bmatrix} M_{11} & M_{12} \\ -I & 0 \end{bmatrix}\right)\begin{bmatrix} x \\ i(\omega - \omega_0)x \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \\ \qquad\qquad\qquad\qquad y = [\ell^*, 0^T]\begin{bmatrix} x \\ i(\omega - \omega_0)x \end{bmatrix}. \end{cases} \quad (35)$$

where $K_{11} = K_0 + k_1^{(j)}K_1 + i\omega_0 c_1^{(j)}K_1 - \omega_0^2 M$, $C_{11} = d_1^{(j)}K_1 + i\omega_0 d_2^{(j)}K_1$, $C_{12} = d_2^{(j)}K_1$, $M_{11} = c_1^{(j)}K_1 + 2i\omega_0 M$ and $M_{12} = M$.

5 Numerical Results and Analysis

In this section, we apply the MOR Framework and the PMOR Framework developed in §4 to the two design optimization problems described in §2.

5.1 Acoustic Cavity Design Problem

For the acoustic cavity design problem, we have implemented both the MOR Framework and the PMOR Framework for ∞-norm optimization. We use the direct method that uses the original model for all computations as a reference to show the effectivity of the two frameworks. Table 1 shows the numerical

results when the finite difference model has 15625 DOFs. Table 2 indicates that MOR is effective even when applied to a relatively small scale system with 216 DOFs. In both examples, the grid search step length, the frequency range and the backtracking factor is set to 0.5 Hz, [9.5 Hz,11.5 Hz] and 0.5 respectively. The initial value is 0.3.

Table 1: Numerical Results of the Order 15625 Model

	Direct method	MOR Framework	PMOR Framework
Matrix size	15625	20	$(10, 8, 5, 3, 1)^4$
Optimizer computed	(10.1245, 0.2671)	(10.1241, 0.2670)	(10.1245, 0.2671)
CPU time	2764s	158s	15s

Table 2: Numerical Results of the Order 216 Model

	Direct method	MOR Framework	PMOR Framework
Matrix size	216	30	(20, 10, 5, 3, 1)
Optimizer computed	(9.44428, 0.260166)	(9.44436, 0.260147)	(9.44428, 0.260182)
CPU time	1.28s	1.21s	0.24s

In both test cases, PMOR Framework is cheaper but locates the optimizer more accurately. This implies that the PMOR Framework outperforms the MOR Framework when the system has only one design parameter since only one reduced model is needed for the whole optimization.

5.2   Floor Damper Design Problem

In this section, we applied the MOR Framework and the PMOR Framework for both ∞-norm optimization and 2-norm optimization of the floor damper design problem. We used two different finite element models for the floor damper: a 280-order model obtained from a 10×10 uniform mesh and a 29800-order model obtained from a 100×100 uniform mesh.

First, we consider the ∞-norm optimization for the floor damper design problem. For the optimization of both the 280-order model and the 29800-order model, the frequency range is $[2\,\text{Hz}, 100\,\text{Hz}]$ and the grid search interval is 4 Hz respectively. The backtracking factor is set to 0.5. The initial value is $(10^6\,\text{N/m}, 10^4\,\text{Ns/m})$.

Table 3: Numerical Results of ∞-norm Optimization of an Order 280 Model

	Direct method	MOR Framework	PMOR Framework
Matrix size	280	25	(15, 10, 8, 5)
Optimizer computed	(1470571.461, 4351.660974)	(1470571.324, 4351.660851)	(1470571.353, 4351.660598)
Optimized value	1161.56029	1161.559454	1161.559633
Number of Iterations	21	21	21
Total Backtracking Steps	104	104	104
CPU time	162s	27s	14s

Table 3 shows that both the MOR Framework and the PMOR Framework work well in reducing the optimization time. For a relatively large problem shown in Table 4, the direct method needs several days to run and we do not complete its computation, while for SOAR and PIMTAP, we need only about an hour to finish the optimization, which is good news for actual design in industry. From Table 3 and Table 4, we can see that ∞-norm optimization involves many backtracking steps, which makes the PMOR Framework more efficient than the MOR Framework.

Now we turn to the 2-norm optimization problem. The problem setting is the same as what we use for ∞-norm optimization except that we use 200 uniform grid points to carry out a numerical integration based on the Trapezoidal rule.

---

[4]For PMOR Framework, we show the moment matching pattern of two-sided PIMTAP instead to be more specific.

Table 4: Numerical Results of ∞-norm Optimization of an Order 29800 Model

	Direct method	MOR Framework	PMOR Framework
Matrix size	29800	25	(15, 10, 8, 5)
Optimizer computed	—	(1479151.998, 4432.836792)	(1479151.616, 4432.873736)
Optimized value	—	126.9130782	126.9129419
Number of iterations	—	16	17
Total Backtracking Steps	—	59	76
CPU time	Several Days	2521s	1248s

Table 5: Numerical Results of 2-norm Optimization of an Order 280 Model

	Direct method	MOR Framework	PMOR Framework
Matrix size	280	25	(15, 10, 8, 5)
Optimizer computed	(1489489.106, 3542.277609)	(1489489.106, 3542.277608)	(1489489.106, 3542.277613)
Optimized value	18340.65632	18340.65632	18340.65632
Number of iterations	17	17	17
Total Backtracking Steps	6	6	6
CPU time	27s	7s	8s

The numerical results in Table 5 and Table 6 show that for 2-norm optimization, the MOR Framework locates more accurate optimizers in less time. This is because backtracking steps are seldom carried out and the freedom of the PMOR Framework in the search direction, at the expense of increasing the order of the reduced model, is seldom exploited.

The results in Table 3 and Table 5 indicates that 2-norm optimization cost less time than ∞-norm optimization. On the one hand, the optimization of the 2-norm converges faster because its objective function is smoother. To show this, we show $\|\gamma_k - \gamma_{k-1}\|$ for the example of Table 3 and the example of Table 5 in Fig. 10. On the other hand, for 2 norm optimization, backtracking does not occur frequently when the objective function is convex and smooth, which means that most line search steps include only one iteration step. For ∞-norm optimization, however, the optimizer is almost always a kink point[5]. Around the kink optimizer, a Quasi-Newton type method may have a too large initial step length, in which case it needs many backtracking steps to satisfy the Armijo condition. So, the last steps of the ∞-norm optimization are very expensive. This problem can easily be alleviated by using the PMOR Framework since in that case, all backtracking steps use the same reduced model, which is relatively cheap. Our experience also shows that if 2-norm optimization start from a point in the non-convex region, the number of backtracking steps increases, which may make the PMOR Framework outperform the MOR Framework.
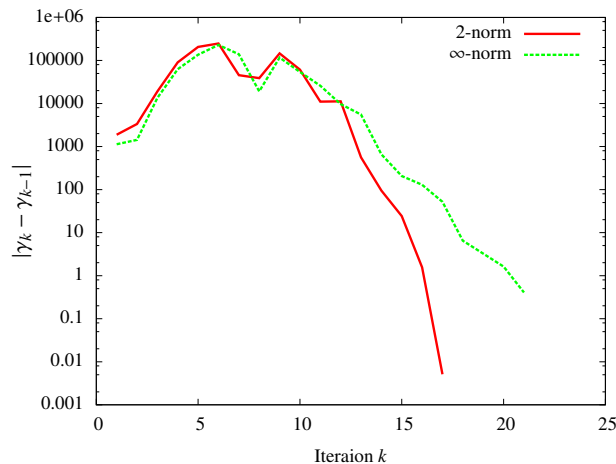
Figure 10: Comparison of Convergence Rate for 2-norm and ∞-norm Optimization

In our numerical results, we chose the sizes for the reduced model so that the Krylov methods just start to converge and the errors between the two frameworks are comparable. These errors lead to some differences in the converging process, such as the number of iterations required to converge. Actually, if we increase these sizes a little bit, the reduced models become much more accurate and these differences completely disappear.

---

[5]The optimizer is also possible to be a saddle point, but this is rare as it indicates two LMACs have the same gradient at the cross-point.

Table 6: Numerical Results of 2-norm Optimization of an Order 29800 Model

	Direct method	MOR Framework	PMOR Framework
Matrix size	29800	25	(15, 10, 8, 5)
Optimizer computed	—	(1504079.109, 3582.064791)	(1504079.109, 3582.064785)
Optimized value	—	2011.873086	2011.873086
Number of iterations	—	17	17
Total Backtracking Steps	—	2	2
CPU time	Several Days	980s	1211s

5.3 MOR Eases the Choice of Optimization Parameters

For both 2-norm optimization and ∞-norm optimization, we have to deal with the trade-off between the optimization performance and the computational cost in the Inner Phase. In the 2-norm case, we need to use many integration points to guarantee the accuracy of the numerical integration, which is computationally expensive when the order of the problem is high. For the ∞-norm optimization, the step length of the grid search should be fine enough to avoid missing the global optimizer.

How to choose the number of integration points or grid points is by itself a complicated optimization problem. But if we use MOR, all evaluations in an Inner Phase are computed with the same reduced model and even if we use a fine grid to increase accuracy, the computational cost is not significantly increased compared with the direct methods. In our numerical examples, we use relatively few grid points to make a fair comparison between the direct method and MOR methods.

5.4 Open problem: Determination of the size of the reduced model

An open problem is how large the dimensions of subspaces should be for practical use in numerical optimization. In this paper, we have shown that accurate results can be approximated well with the results obtained from MOR. In practice, however, the number of vectors should be determined by an error estimate. One possibility is to use error estimations as in [6], or use the residual norm for the systems

$$
\begin{aligned}
(K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma))x &= f \\
(K(\gamma) + i\omega C(\gamma) - \omega^2 M(\gamma))^* t &= \ell
\end{aligned}
$$

The computed solutions are then the exact solution of a perturbed system. Such residual norms are cheaply computed. See, e.g., [27][22][23]. The development of practical heuristics is future work.

6 Conclusion

In this paper, we have presented a Quasi-Newton based line search optimization method with backtracking strategy for the solution of optimization problems arising from design optimization of vibrations and structures. We used (P)MOR in order to reduce the computational cost. We hereby compare MOR reduced models for the frequency response function with fixed parameters, and PMOR reduced models for the frequency and the search direction. When the objective function is the 2-norm of the frequency response function, the best choice are reduced models for the frequency only. For the ∞-norm of the frequency response function, backtracking in the search direction is frequently required, in which case advantage can be made of a reduced model in the frequency and the line search direction. For systems with only one design parameter, PMOR reduced models are usually better choices.

Acknowledgements

A Proof of Lemma 1

Proof:

From (13), we derive

$$b = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} A(p_1, p_2, \ldots, p_l) r(i_1, i_2, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l}, \tag{36}$$

$$\hat{b} = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} \hat{A}(p_1, p_2, \ldots, p_l) \hat{r}(i_1, i_2, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l}. \tag{37}$$

As $\hat{A}(p_1, p_2, \ldots, p_l) = W^* A(p_1, p_2, \ldots, p_l) V$ and $\hat{b} = W^* b$, the equation (37) can be rewritten as

$$W^* b = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} W^* A(p_1, p_2, \ldots, p_l) V \hat{r}(i_1, i_2, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l}. \tag{38}$$

Multiplying (36) by W^* on the left gives

$$W^* b = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} W^* A(p_1, p_2, \ldots, p_l) r(i_1, i_2, \ldots, i_l) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l}, \tag{39}$$

$(38) - (39)$ gives,

$$\sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} W^* A(p_1, p_2, \ldots, p_l) \left(V \hat{r}(i_1, i_2, \ldots, i_l) - r(i_1, i_2, \ldots, i_l) \right) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l} = 0,$$

$$\sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} \sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} \ldots \sum_{j_l=0}^{\infty} W^* \mathring{A}(j_1, j_2, \ldots, j_l) \left(V \hat{r}(i_1, i_2, \ldots, i_l) - r(i_1, i_2, \ldots, i_l) \right) p_1^{i_1+j_1} p_2^{i_2+j_2} \ldots p_l^{i_l+j_l} = 0,$$

$$\sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \ldots \sum_{i_l=0}^{\infty} \sum_{j_1=0}^{i_1} \sum_{j_2=0}^{i_2} \ldots \sum_{j_l=0}^{i_l} W^* \mathring{A}(i_1-j_1, i_2-j_2, \ldots, i_l-j_l) \left(V \hat{r}(j_1, j_2, \ldots, j_l) - r(j_1, j_2, \ldots, j_l) \right) p_1^{i_1} p_2^{i_2} \ldots p_l^{i_l} = 0.$$

The fact that the Taylor series equals to zero means for all $i_1, i_2, \ldots, i_l \geq 0$,

$$\sum_{j_1=0}^{i_1} \sum_{j_2=0}^{i_2} \ldots \sum_{j_l=0}^{i_l} W^* \mathring{A}(i_1-j_1, i_2-j_2, \ldots, i_l-j_l) \left(V \hat{r}(j_1, j_2, \ldots, j_l) - r(j_1, j_2, \ldots, j_l) \right) = 0. \tag{40}$$

Now we prove $r(i_1, i_2, \ldots, i_l) = V \hat{r}(i_1, i_2, \ldots, i_l)$ by induction.

1. For $i_1 = i_2 = \ldots = i_l = 0$, we have

$$W^* \mathring{A}(0, 0, \ldots, 0) \left(V \hat{r}(0, 0, \ldots, 0) - r(0, 0, \ldots, 0) \right) = 0.$$

Since $r(0, 0, \ldots, 0) \in \text{colspan}\{V\}$, there exists $\xi(0, 0, \ldots, 0) \in \mathbb{C}^k$ such that $r(0, 0, \ldots, 0) = V \xi(0, 0, \ldots, 0)$. Therefore,

$$W^* \mathring{A}(0, 0, \ldots, 0) V \left(\hat{r}(0, 0, \ldots, 0) - \xi(0, 0, \ldots, 0) \right) = 0.$$

As $W^* \mathring{A}(0, 0, \ldots, 0) V$ is nonsingular, $\hat{r}(0, 0, \ldots, 0) = \xi(0, 0, \ldots, 0)$ and

$$r(0, 0, \ldots, 0) = V \hat{r}(0, 0, \ldots, 0).$$

2. Suppose $r(j_1, j_2, \ldots, j_l) = V \hat{r}(j_1, j_2, \ldots, j_l)$ for all $j_1 \leq i_1$, $j_2 \leq i_2, \ldots, j_l \leq i_l$ except $j_1 = i_1$, $j_2 = i_2, \ldots, j_l = i_l$. Under this assumption, the equation (40) becomes

$$W^* \mathring{A}(0, 0, \ldots, 0) \left(V \hat{r}(i_1, i_2, \ldots, i_l) - r(i_1, i_2, \ldots, i_l) \right) = 0. \tag{41}$$

Like the proof in 1, we have $r(i_1, i_2, \ldots, i_l) = V \hat{r}(i_1, i_2, \ldots, i_l)$.

As $r(i_1, i_2, \ldots, i_l) = V \hat{r}(i_1, i_2, \ldots, i_l)$,

$$m(i_1, i_2, \ldots, i_l) = \ell^* r(i_1, i_2, \ldots, i_l) = \ell^* V \hat{r}(i_1, i_2, \ldots, i_l) = \hat{\ell} \hat{r}(i_1, i_2, \ldots, i_l) = \hat{m}(i_1, i_2, \ldots, i_l).$$

Similarly, we can prove the second half of the lemma.

$\square$

References

[1] A. Antoulas, C. Beattie, and S. Gugercin. Interpolatory model reduction of large-scale dynamical systems. In *Efficient Modeling and Control of Large-Scale Systems*. Springer-Verlag, 2010.

[2] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9(17):17–29, 1951.

[3] Z. Bai, K. Meerbergen, and Y. Su. Arnoldi methods for structure-preserving dimension reduction of second-order dynamical systems. In *Dimension reduction of large-scale systems*, pages 173–189. Springer, 2005.

[4] 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(5):1692–1709, 2005, doi:10.1137/040605552.

[5] Z. Bai and Y. Su. SOAR: a second-order Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications*, 26(3):640–659, 2005, doi:10.1137/S0895479803438523.

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

[7] C. R. Calladine. *Theory of Shell Structures*. Cambridge University Press, 1989.

[8] L. Daniel, O. C. Siong, L. S. Chay, K. H. Lee, and J. White. A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 23(5):678–693, 2004, doi:10.1109/TCAD.2004.826583.

[9] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Dover Publications, 2nd edition, 2007.

[10] P. Feldman and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Computer-Aided Design*, 14(5):639–649, 1995, doi:10.1109/43.384428.

[11] L. Feng. Parameter independent model order reduction. *Mathematics and Computers in Simulation*, 68:221–234, 2005, doi:10.1088/0960-1317/20/4/045030.

[12] L. H. Feng, E. B. Rudnyi, and J. G. Korvink. Preserving the film coefficient as a parameter in the compact thermal model for fast electrothermal simulation. *IEEE Trans. Computer-Aided Design*, 24(12):1838–1847, 2005, doi:10.1109/TCAD.2005.861422.

[13] S. Gugercin, A. C. Antoulas, and C. Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM journal on matrix analysis and applications*, 30(2):609–638, 2008, doi:10.1137/060666123.

[14] P. Gunupudi and M. Nakhla. Multi-dimensional model reduction of VLSI interconnects. In *Custom Integrated Circuits Conference, 2000. Proceedings of the IEEE 2000*, pages 499–502, 2000.

[15] J. S. Han, E. B. Rudnyi, and J. G. Korvink. Efficient optimization of transient dynamic problems in MEMS devices using model order reduction. *Journal of Micromechanics and Microengineering*, 15(4):822–832, 2005, doi:10.1088/0960-1317/15/4/021.

[16] C. Lemaréchal. Numerical experiments in nonsmooth optimization. In E. Nurminski, editor, *Progress in Nondifferentiable Optimization*, pages 61–84, 1982.

[17] X. Li, P. Li, and L. T. Pileggi. Parameterized interconnect order reduction with explicit-and-implicit multi-parameter moment matching for inter/intra-die variations. In *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 806–812. IEEE Computer Society Washington, DC, USA, 2005.

[18] Y.-T. Li, Z. Bai, and Y. Su. A two-directional Arnoldi process and its application to parametric model order reduction. *Journal of Computational and Applied Mathematics*, 226(1):10–21, 2009, doi:10.1016/j.cam.2008.05.059.

[19] Y.-T. Li, Z. Bai, Y. Su, and X. Zeng. Parameterized model order reduction via a two-directional Arnoldi process. In *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, pages 868–873, 2007.

[20] Y.-T. Li, Z. Bai, Y. Su, and X. 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(9):1571–1582, 2008, doi:10.1109/TCAD.2008.927768.

[21] K. Meerbergen. The solution of parametrized symmetric linear systems. *SIAM journal on matrix analysis and applications*, 24(4):1038–1059, 2003, doi:10.1137/S0895479800380386.

[22] K. Meerbergen. Fast frequency response computation for Rayleigh damping. *International Journal for Numerical Methods in Engineering*, 73(1):96–106, 2008, doi:10.1002/nme.2058.

[23] K. Meerbergen and Z. Bai. The Lanczos method for parameterized symmetric linear systems with multiple right-hand sides. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1642–1662, 2010, doi:10.1137/08073144X.

[24] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[25] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macro-modeling algorithm. In *ICCAD '97: Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design*, pages 58–65, Washington, DC, USA, 1997. IEEE Computer Society.

[26] B. Salimbahrami and B. Lohmann. Order reduction of large scale second-order systems using Krylov subspace methods. *Linear Algebra and its Applications*, 415(2-3):385–405, 2006, doi:10.1016/j.laa.2004.12.013.

[27] V. Simoncini and F. Perotti. On the numerical solution of $(\lambda^2 A + \lambda B + C)x = b$ and application to structural dynamics. *SIAM Journal on Scientific Computing*, 23(6):1875–1897, 2002, doi:10.1137/S1064827501383373.

[28] T. Z. U. J. SU and R. CRAIG. Model reduction and control of flexible structures using Krylov vectors. *Journal of guidance, control, and dynamics*, 14(2):260–267, 1991, doi:10.2514/3.20636.

[29] J. Vlček and Lukšan. Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. *Journal of Optimization Theory and Applications*, 111(2):407–430, 2001, doi:10.1023/A:1011990503369.

[30] 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(5):801–813, 2001, doi:10.1109/8.929635.

[31] D. S. Weile, E. Michielssen, and E. Grimme. A method for generating rational interpolant reduced order models of two-parameter linear systems. *Applied Mathematics Letters*, 12(5):93–102, 1999, doi:10.1016/S0893-9659(99)00063-4.