Efficient greedy algorithms for successive constraints methods with high-dimensional parameters

Shun Zhang

Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

Received *****; accepted after revision +++++

Presented by

Abstract

We generalize efficinet greedy algorithms developed in [2] to the Successive Constraints Method (SCM) of computing lower bounds of the coercivity constants. The algorithms are based on a newly developed simple version of SCM [4]. With monotonicity-based algorithm, the amount of work is saved substantially over the standard greedy algorithm. Combined this with the safety check guaranteed adaptively enriching greedy algorithm, the SCM for problems with high dimensional parameter space is now workable and more robust.

To cite this article: S. Zhang, C. R. Acad. Sci. Paris, Ser. I . xxx (201x)

Résumé

Algorithme glouton efficace pour la Méthode par Contraintes Successives (MCS) pour des espaces de paramètres de haute dimension. Dans ce travail, on generalize l'algorithme glouton developé dans [2] pour la Méthode par Contraintes Successives (MCS) qui permet de calculer une borne inférieure de la constante de coercivité. L'algorithme est basé sur la nouvelle version simplifiée de la MCS [4]. Avec cet algorithme, assurant la monotinicité de la constante, beaucoup de travail peut être sauvé en comparant à la version standard. Ici, on combine cette approche avec la version de l'algorithme glouton [2] en rendant la MCS faisable pour des espaces de parametètres de haute dimension et plus robuste.

Pour citer cet article :S. Zhang, C. R. Acad. Sci. Paris, Ser. I . xxx (201x).

1. Introduction

In [2], efficiency greedy algorithms are developed for problems with high-dimensional parameter space. We apply the algorithms to reduced basis and empirical interpolation methods successfully. Another major component of the certified reduced basis methods is an accurate estimation of the lower-bounds

Email address: Shun_Zhang@Brown.edu (Shun Zhang).

of parametric coercivity and inf-sup stability constants. Successive constrains methods (SCM) are often used to get the estimation of such constants. Traditional SCM [3,1] computes the coercivity constants by solving a linear programming problem with two kinds of constraints. The first kind of constraint is from M_α closest points in a set $\mathbb{C}_N$, where eigenvalues and eigenvectors are computed. The second kind of constraint is from M_+ closest points in the train set Ξ. Since the size of Ξ is often large to ensure that Ξ is a fine subset of the parameter domain $\mathcal{D} \in \mathbb{R}^p$, where p is the number of parameters, finding M_+ closest points in Ξ is not a simple task. If Ξ is randomly generated, then a sorting process or some tree structure has to be used, which will be slow and complicated. If Ξ is generated based on a Cartesian grid, it will too coarse or impossible to construct when p is large. Thus, direct applications of the saturation assumption based and adaptively enriching greedy algorithms developed in [2] to SCM are not simple. Fortunately, in [4], a new version of SCM is suggested. For the second kind of constraint, only the parameter itself needs to be kept to ensure the monotonicity of the error estimator. We can discard M_α and use the full $\mathbb{C}_N$ as the first kind of constraint. With this new version of SCM, random generated Ξ can be used and there is no need to find closest points in both $\mathbb{C}_N$ and Ξ. In this paper, with some modification of the method in [4] (the active set of constraints is not tracked), we develop monotonicity-based greedy algorithm and adaptively enriching greedy algorithm for SCM based on the ideas of [2]. The saving of the algorithms are substantial and the new algorithms can be applied to SCM with high dimensional spaces.

2. Monotonicity-based Greedy Algorithm for Successive Constraints Methods

We describe the monotonicity-based greedy algorithm of SCM in this section. This algorithm corresponds to the saturation assumption based greedy algorithm in [2].

Given an affine bilinear form $a(u, v; \boldsymbol{\mu}) = \sum_{i=1}^{Q} \Theta_i(\boldsymbol{\mu}) a_i(u, v)$, $u, v \in X^{fe}$ and $\boldsymbol{\mu} \in \mathcal{D}$. A representation of the coercivity constant is $\alpha^{fe}(\boldsymbol{\mu}) = \inf_{w \in X^{fe}} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|^2_{X_h}} = \inf_{w \in X^{fe}} \sum_{i=1}^{Q} \Theta_i(\boldsymbol{\mu}) \frac{a_i(w,w;\boldsymbol{\mu})}{\|w\|^2_{X_h}}$. Then

$$\alpha^{fe}(\boldsymbol{\mu}) = \min_{\mathbf{y} \in \mathcal{Y}} \mathcal{I}(\boldsymbol{\mu}, \mathbf{y}) \quad \text{where} \quad \mathcal{I}(\boldsymbol{\mu}, \mathbf{y}) = \sum_{i=1}^{Q} \Theta_i(\boldsymbol{\mu}) y_i,$$
$$\mathcal{Y} = \left\{ \boldsymbol{y} = (y_1, \cdots, y_Q) \in \mathbb{R}^Q \mid \exists\, w \in X^{fe} \text{ s.t. } y_i = \frac{a_i(w,w)}{\|w\|^2_{X_h}}, 1 \le i \le Q \right\}.$$

Lower and upper bounds of $\alpha^{fe}(\boldsymbol{\mu})$ can be found by building two sets $\mathcal{Y}_{UB}$ and $\mathcal{Y}_{LB}$ such that $\mathcal{Y}_{UB} \subset \mathcal{Y} \subset \mathcal{Y}_{LB}$. Define $\alpha_{LB}(\boldsymbol{\mu}) = \min_{\mathbf{y} \in \mathcal{Y}_{LB}} \mathcal{I}(\boldsymbol{\mu}, \mathbf{y})$ and $\alpha_{UB}(\boldsymbol{\mu}) = \min_{\mathbf{y} \in \mathcal{Y}_{UB}} \mathcal{I}(\boldsymbol{\mu}, \mathbf{y})$. We introduce a parameter set $\mathbb{C}_N = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_N\}$, which is a subset of $\mathcal{D}$ with N parameter vectors. For each $\boldsymbol{\mu}_i \in \mathbb{C}_N$, we solve a generalized eigenvalue problem $a(w_i, v; \boldsymbol{\mu}_i) = \alpha^{fe}(\boldsymbol{\mu}_i)(w_i, v)_{X^{fe}}$, $v \in X^{fe}$ with $\alpha^{fe}(\boldsymbol{\mu}_i)$ is the smallest eigenvalue and $w_i \in X^{fe}$ is the corresponding eigenvector. Then $\mathcal{Y}_{UB}(\mathbb{C}_N)$ is defined by

$$\mathcal{Y}_{UB}(\mathbb{C}_N) = \{\mathbf{y}^*(\boldsymbol{\mu}_i), 1 \le i \le N\} \quad \text{with} \quad \mathbf{y}^*(\boldsymbol{\mu}) = \arg\min_{\mathbf{y} \in \mathcal{Y}} \mathcal{I}(\boldsymbol{\mu}, \mathbf{y}), \tag{2.1}$$

It's clear that $\mathcal{Y}_{UB}(\mathbb{C}_N) \subset \mathcal{Y}$. An online step of computing $\alpha^N_{UB} = \min_{\mathbf{y} \in \mathcal{Y}_{LB}(\mathbb{C}_N)} \mathcal{I}(\boldsymbol{\mu}, \mathbf{y})$, is independent of the degree of freedom of X^{fe}, but get a point in $\mathbb{C}_N$ offline involving an eigenvalue problem in the finite element spaces. For $\mathcal{Y}_{LB}$, we define a box $\mathcal{B} = \Pi_{i=1}^{Q} [\sigma_i^-, \sigma_i^+] \subset \mathbb{R}^Q$, where $\sigma_i^- = \inf_{v \in X^{fe}} \frac{a_i(v,v)}{\|\mathbf{v}\|^2_{X^{fe}}}$ and $\sigma_i^+ = \sup_{v \in X^{fe}} \frac{a_i(v,v)}{\|\mathbf{v}\|^2_{X^{fe}}}$ for $1 \le i \le Q$. Obviously, $\mathcal{Y}$ is a subset of $\mathcal{B}$. For any $\boldsymbol{\mu} \in \Xi$, defined an $\alpha_{saved}(\boldsymbol{\mu})$ with initialization 0. Define

$$\mathcal{Y}_{LB}(\boldsymbol{\mu}; \mathbb{C}_N) = \left\{ \mathbf{y} \in \mathcal{B} \,|\mathcal{I}(\boldsymbol{\mu}', \mathbf{y}) \ge \alpha^{fe}(\boldsymbol{\mu}'), \forall \boldsymbol{\mu}' \in \mathbb{C}_N, \quad \mathcal{I}(\boldsymbol{\mu}, \mathbf{y}) \ge \alpha_{saved}(\boldsymbol{\mu}) \right\}. \tag{2.2}$$

It's well known that if we keep $\mathcal{I}(\boldsymbol{\mu}, \mathbf{y}) \geq \alpha_{saved}(\boldsymbol{\mu})$ as a constraint in $\mathcal{Y}_{LB}$, then the updated $\alpha_{LB}^N(\boldsymbol{\mu}) = \min_{\mathbf{y} \in \mathcal{Y}_{LB}(\boldsymbol{\mu}; \mathbb{C}_N)} \mathcal{I}(\boldsymbol{\mu}, \mathbf{y}) \geq \alpha_{saved}(\boldsymbol{\mu})$. Each time, if $\alpha_{LB}^N(\boldsymbol{\mu})$ is computed for $\boldsymbol{\mu}$, we update its corresponding value in $\alpha_{saved}(\boldsymbol{\mu})$. Define an error estimator $\eta(\boldsymbol{\mu}; \mathbb{C}_N) = \frac{\alpha_{UB}^N(\boldsymbol{\mu}) - \alpha_{LB}^N(\boldsymbol{\mu})}{\alpha_{UB}^N(\boldsymbol{\mu})}$. It's easy to see $0 \leq \eta(\boldsymbol{\mu}; \mathbb{C}_N) < 1$ and $\eta(\boldsymbol{\mu}; \mathbb{C}_N)$ is non-increasing with respect to N. The saturation assumption in [2] is now a proved monotonicity.

For each parameter value $\boldsymbol{\mu} \in \Xi$, we create an error profile $\eta_{saved}(\boldsymbol{\mu})$ with initial values 1 and a coercivity profile $\alpha_{saved}(\boldsymbol{\mu})$ with initial values 0. Now suppose $\mathbb{C}_N$ is determined and we want to find the next sample $\boldsymbol{\mu}_{N+1} = \arg\max_{\boldsymbol{\mu} \in \Xi} \eta(\boldsymbol{\mu}; \mathbb{C}_N)$. We keep updating a temporary maximum when $\boldsymbol{\mu}$ runs through Ξ, until the whole Ξ is searched. Since η is not increasing, if for some parameter $\boldsymbol{\mu} \in \Xi$, its saved error estimation $\eta_{saved}(\boldsymbol{\mu})$ is less than the current temporary maximum, this $\boldsymbol{\mu}$ will not be chosen in this loop. Thus, the computation of $\eta(\boldsymbol{\mu})$ can be skipped and we leave the values of $\eta_{saved}(\boldsymbol{\mu})$ and $\alpha_{saved}(\boldsymbol{\mu})$ untouched. Otherwise, we update those values and compare the updated error estimator with the current temporary maximum to see whether an update is needed. Due to the monotonicity of the error estimator, the saving of the algorithm is substantial.

3. Adaptively Enriching Greedy Algorithm for Successive Constraints Methods

A detailed pseudo-code of the adaptively enriching greedy algorithm for the reduced basis and empirical interpolation methods can be found in [2]. For the new version of SCM discussed in [4] and the above section, only some small adaptions are needed to develop the adaptively enriching greedy algorithm for SCM. For completeness, we describe the main ideas here.

Like mentioned in [2], even with the above monotonicity-based greedy algorithm, we still face some problems. It's hard to determine the size of the train set Ξ. If the size of Ξ is too small, then it might not be a fine enough subset of $\mathcal{D}$. On the other hand, if the size of Ξ is too large, then the each searching of the greedy algorithm is very expansive. To ensure the SCM is good enough with a reasonable size of Ξ, a "safety check" step is added at the end of the algorithm, that is, we test the quality of $\mathbb{C}_N$ by a large number of parameters to see if the resulting error estimators are smaller than the tolerance. If not, new points are added into $\mathbb{C}_N$ and the safety check step is re-done until the set $\mathbb{C}_N$ passes the "safety check". After having a $\mathbb{C}_N$, some points in Ξ are already smaller then the tolerance, thus will never be chosen. Those points can be removed from Ξ and new random points can be added into Ξ to make the size of Ξ a constant. For the unchanged part of Ξ, the monotonicity algorithm should be used to save wokload.

4. Numerical Tests

For the thermal block problem [4], $\nabla \cdot (\kappa \nabla u) = 0$ in $[0,1]^2$, $u = 0$ on $\Gamma_{top} = \{x \in (0,1), y = 1\}$, $\kappa \nabla u \cdot \mathbf{n} = 0$ on $\Gamma_{side} = \{x = 0 \text{ and } x = 1, y \in (0,1)\}$, $\kappa \nabla u \cdot \mathbf{n} = 1$ on $\Gamma_{base} = \{x \in (0,1), y = 0\}$. The domain is decomposed into 9 subdomains: $R_k = (\frac{i-1}{3}, \frac{i}{3}) \times (\frac{j-1}{3}, \frac{j}{3})$, for $i = 1, 2, 3$, $j = 1, 2, 3$, and $k = 3(i-1) + j$. The diffusion constant κ is set to be $5^{2\mu_k - 1}$, for $x \in R_k$, $k = 1, 2, \cdots, 8$, and $\kappa_9 = 1$ for $x \in R_9$, where $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_8) \in [0,1]^8$. The H^1-norm is used as the underlying norm. For monotonicity-based greedy algorithm, we use a Ξ with $10'000$ random points and set tolerance 0.2. From Fig. 1, except for the first step, the amount of points whose error estimators are actually computed are very low. After $N = 20$, only about 1.6% points need to be computed.

For the adaptively enriching greedy algorithm, we keep the size of Ξ to be $10'000$ and set tolerance 0.2. $100'000$ points are used to check the quality of SCM. Fig. 2 shows the convergence behavior, some jumps

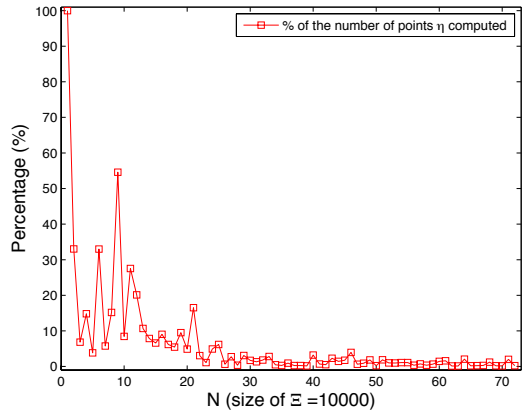

Figure 1. Percentage of work at each step N using Monotonicity-based Algorithm.

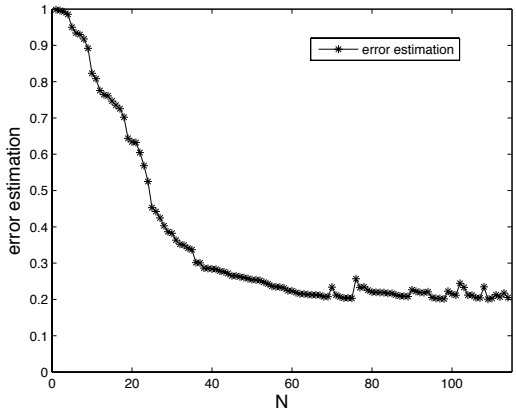

Figure 2. Convergence behavior of Adaptively Enriching Algorithm

after $N = 70$ are the new bad points discovered by the safety check step. Percentage of work (effected at each step N) w.r.t. the size of Ξ and of the number of points remained in the train set (at each step N) is shown in Fig. 3. Before $N = 70$, it's basically the monotonicity-based algorithm, very small number of points are thrown away, and the percentage of points whose error estimators are computed is low. After 70 more points in $\mathbb{C}_N$, several rounds of fresh random points are enriched, and then 10 rounds of new points to pass the "safety check". Compared to the pure monotonicity-based algorithm, a larger N is obtained, but the quality of $\mathbb{C}_N$ is more guaranteed.

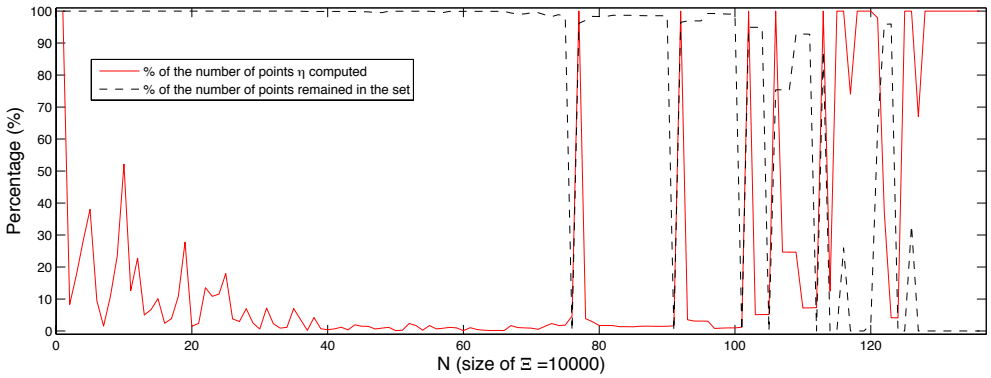

Figure 3. Percentage of work (effected at each step N) w.r.t. the size of Ξ and of the number of points remained in the train set (at each step N) of the adaptively enriching greedy algorithm

Remark 1 *In [4], an active set of the first type of constraint is used. We discard this condition in our algorithm. The computational time is reasonable from the numerical results. Due to the monotonicity-based algorithm, at the beginning, when N is small, error estimators are computed for relatively large potion of the train set, the computational costs are acceptable at this stage; when N is relatively big (in our numerical test $N > 20$), error estimators are computed on only a very small part of the whole Ξ, so the computational costs are still reasonable even the active constraint set is not tracked. For the adaptively*

enriching greedy algorithm, when new random samples are enriched into Ξ, no history information of active constraints is available, so we have to use the full $\mathbb{C}_N$ as the set of constraints.

On the online procedure, like mentioned in [4], all points in $\mathbb{C}_N$ are used as constraints, and the computational time is acceptable.

References

[1] Y. Chen, J. S. Hesthaven, Y. Maday, and J. Rodriguez, *A monotonic evaluation of lower bounds for inf-sup stability constants in the frame of reduced basis approximations*, C. R. Acad. Sci. Paris, Ser. I, 346 (2008), pp. 1295–1300.

[2] J. S. Hesthaven, B. Stamm, and S. Zhang, *Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods*, Brown University DAM-Scientific Computing Tech Report 2011-15, submitted to M2AN.

[3] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera, *A successive constrainst linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants*, C. R. Acad. Sci. Paris, Ser. I, 345 (2007), pp. 473–478.

[4] S. Vallaghe, A. Le-Hyaric, M. Fouquemberg, C. Prud'homme, *A successive constrainst method with minimal offline constraints for lower bounds of parametric coercivity constant*, C.R. Acad. Sci. Paris, Ser. I, 2011, submitted.