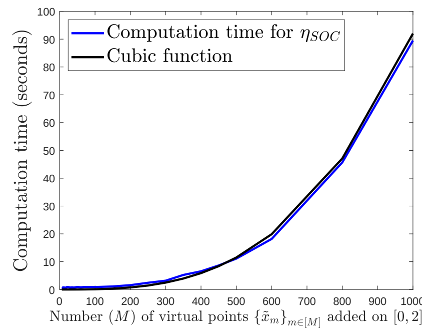
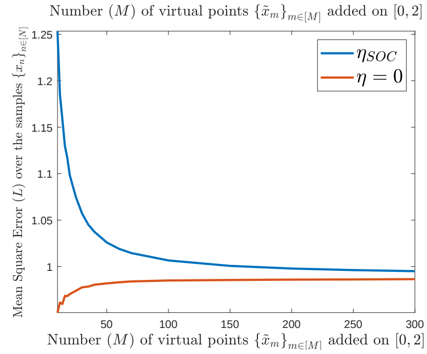
We thank the reviewers for their commitment and valuable insights despite the difficult times. We use Ri below to refer to the i^{th} reviewer. Questions/remarks are indicated by **Q** with reviewer identifiers in parentheses, answers are denoted by **A**. To refer to line X in the submission we use the shorthand 'lX'. Our new figures are located on the r.h.s.

We briefly recall our **primary focus** (l55-58): to propose a flexible optimization framework capable of handling jointly general hard shape constraints (expressible as affine inequalities over derivatives on compact sets) with rich function classes (RKHSs). To the best of our knowledge, our approach is the first in this direction with guarantees. We are thus less concerned about high-dimensional scalability questions, though we explicitly acknowledge it (l226) and provide practical algorithms which allow a benign control of the computations in moderate dimension (l227-231, l258-259). We note that specialized SOC solvers (instead of CVXGEN which we used for illustration) can provide additional speed-up.

Q (R1, R3): Table 1 (SOCP vs PDCD: comparable performance). R3: It would be nice to also demonstrate empirically that JQR violates the imposed non-crossing constraints. **A**: We answer these 2 questions jointly. Following Sangnier et al. 2016, a JQR method is considered to be favorable if (i) the technique gives comparable results in terms of pinball loss (see our Table 1), and (ii) it violates the imposed shape constraints less often (SOCP respects it by construction, whereas PDCD often produces crossings as it can be seen in the last column of Table 1 of Sangnier et al. 2016).





Q (R1, R3): higher dimensionality, soft shape constraint inducing regularizers. **A**: In higher dimensions, compact coverings and our technique can still be applied, though η would be larger. Soft-constrained solutions (e.g. PDCD) might run faster but again without guarantees.

Q (R1, R2): Role of virtual points (R1) and computational complexity (R2) are not discussed. **A**: The complexity is $O((P+N+M)^3)$ in the worst case (l226). As even a kernel ridge regression (KRR) scales cubicly one can not expect in general better behavior with additional hard shape constraints. We provide the computational times for the reviewers associated to KRR with monotonicity (Section B) on the r.h.s. In practice, recycling the N sample points among the M virtual centers effectively reduces (l228-231, l259) the number of coefficients to be determined (see $f_{\boldsymbol{\eta},q}$ in Theorem (ii)) and hence the computational time.

Q (R1): Prior work for shape-constrained GPs could be added, like SK (A. Solin & M. Kok., 2019). **A**: We cite from the GP literature C. Agrell (2019) who handles shape constraints $a \le \mathcal{L}f \le b$ in GPs in a soft fashion where $\mathcal{L}$ is a linear operator. SK tackles equality constraints ($f(\mathbf{x}) = 0$) on the boundary of the domain of a GP in a hard way. Though SK's constraint (equality of only function values on boundary) and its handling (computing the eigendecomposition of the Laplace operator) are quite different from ours, we are happy to refer to it for the sake of completeness.

Q (R2): The authors give only examples where 0-order differential constraints are imposed. **A**: We consider higher order constraints in our examples in economics (Fig. 1(a): 0-1st order; Fig. 1(b): 0-1-2nd order), analysis of aircraft trajectories (Fig. 2, 0-1st order), and KRR with monotonicity (Fig. 4(a), 1st order).

Q (R2): Could a representer theorem be achieved by setting $\eta = 0$? **A**: Yes, however the choice of $\eta = 0$ would correspond to the discretization (6) which does not ensure shape constraints in a hard way on the K_i-s.

Q (R3): Significance of (9)? **A**: (9) is a computable bound (footnote 5) and can be applied as an alternative stopping criterion for the number of virtual points to add. In practice, we use the strategy detailed in l227-231 which works reliably.

Q (R3, R4): How is (8) solved in practice? **A**: For instance in the JQR example (4)-(5), with a radial kernel $k(\mathbf{x},\mathbf{y}) = k_0(\|\mathbf{x}-\mathbf{y}\|_{\mathcal{X}})$ and k_0 monotonically decreasing (such as the Gaussian kernel) (8) simplifies (l220) to $\eta_i = \sup_{\mathbf{u}\in\mathbb{B}_{\|\cdot\|_{\mathcal{X}}}(\mathbf{0},1)}\sqrt{|2k_0(0) - 2k_0(\delta_i\|\mathbf{u}\|_{\mathcal{X}})|} = \sqrt{|2k_0(0) - 2k_0(\delta_i)|}$; hence η_i can be computed analytically. Similar computation can be carried out for higher order derivatives. For more general kernels, estimating η_i-s can be also done by sampling uniformly $\mathbf{u}$ in the unit ball.

Q (R3, R4): It would be interesting to see the effects of the choice of δ (or M) and compare it with $\eta = 0$. **A**: The objective values for $\eta = 0$ are always below the optimal value of the original problem, while that of with $\eta > 0$ are above. We provide an illustration (r.h.s.) that as M is increasing the objective values get closer to each other.

Q (R4): How the virtual points $\mathbf{x}_{i,m}$-s are chosen in practice? **A**: According to our experiences, besides the recycling trick (l227-231), choosing the $\mathbf{x}_{i,m}$-s to form an approximately uniform grid is a safe and reliable choice as it implies uniformly small δ_i, thus low bound on η_i (l222), and hence tighter guarantee (see (10)).

We hope that we have answered all the questions of the reviewers.