2.2.2.1 How does NR works with nonlinear manifolds ?

- How to deal with Newton-Raphson procedures when there is no affine relationship between the nodal displacements and the reduced-coordinates? For instance, if we write $\boldsymbol{d} = \boldsymbol{V}(\boldsymbol{q})$, and $\delta \boldsymbol{d} = \boldsymbol{\Phi} \delta \boldsymbol{q}$, where $\boldsymbol{\Phi} = \dfrac{\partial \boldsymbol{V}}{\partial \boldsymbol{q}}$? If we have $\boldsymbol{q}_n$ and $\Delta \boldsymbol{q}$, we cannot simply make $\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \Delta \boldsymbol{q}$ —we are forced to work with curvilinear coordinates, aren't we ?
- The following papers may prove illuminating in answering this question:
  - "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders"

2.3 Assessment of the ECM in the presence of inertial forces

- Semi-discrete equation of motion $\boldsymbol{M}\dddot{\boldsymbol{d}} + \boldsymbol{K}\ddot{\boldsymbol{d}} = \boldsymbol{0}$.
- Projection onto reduced basis $\boldsymbol{\Phi}$ (n modes).
- Empirical finding: Cubature internal forces: $m = n(n+1)/2$ (term $\boldsymbol{K}\ddot{\boldsymbol{d}}$)
- Example: CUBATURE_INERTIAL_FORCES
- See script CubatureInertialForces.m
- **CONCLUSION**: For n modes, $m = (n+1)n/2$ are needed to integrate internal forces. Likewise, for inertial forces, $m = (n+1)n/2$ are also required. However, they are not the same set. As a consequence, if one wishes to integrate both

  $$m = (n+1)n \tag{2.3.1}$$

  points are needed.
- ECM applied at element level. Changes to be introduced. $f_{int}(\boldsymbol{x}) = w(x)B(x)^T\sigma(x)$ (this is the point approach). In the element approach, we make $f^e_{total} = \sum w(x_i)B(x_i)^T\sigma(x_i)$.

2.4 Alternative to the ECM: the Empirical Quadrature Method by A. Patera

2.4.1 Abstract

The goal of this study is to explore the method called "the Empirical Quadrature", developed by Patera and co-workers Yano (2019); Yano and Patera (2019), which in turn is inspired by the pioneering idea of Ryu and Boyd Ryu and Boyd (2015), which is to address the determination of "Gauss" nodes and positive weights using Linear Programming. We have tested the approach (see Section 2.4.5), and have empirically found that, indeed, one may pose the efficient integration problem as the minimization of a linear function with linear constraints. However, we have observed no apparent advantages over our approach —the ECM. Future work should explore this idea in more complex scenarios (ROM). Perhaps one may exploit the fact that the LP approach allows one to individually control the error in approximating each basis function of the integrand.

2.4.2 Introduction

- The EQM has been developed by A. Patera et co-workers, see Yano (2019); Yano and Patera (2019).
- Starting point: method advocated by Boyd et co-workers: Ryu and Boyd (2015)
- Idea

  > Gauss quadrature is a well-known method for estimating the integral of a **continuous function with respect to a given measure** as a weighted sum of the function evaluated at a set of node points. Gauss quadrature is traditionally developed using orthogonal polynomials. We show that Gauss quadrature can also be obtained as the solution to an **infinite-dimensional linear program** (LP): minimize the nth moment among all nonnegative measures that match the 0 through nâ1 moments of the given measure. While this infinite-dimensional LP provides no computational advantage in the traditional setting of integration on the real line, it can be used to construct Gauss-like quadratures in more general settings, including arbitrary domains in multiple dimensions.

- What is the concept of "measure" in mathematics ? According to Gautschi (2004), see pag. 12, a measure is "differential" in the expression of an integral —when it is continuous. However, it may be discrete as well—the length of an interval, for instance. Wikipedia tells us that "In mathematical analysis, a measure on a set is a systematic way to assign a number to each suitable subset of that set, intuitively interpreted as its size. In this sense, a **measure is a generalization of the concepts of length, area, and volume**. A particularly important example is the Lebesgue measure on a Euclidean space, which assigns the conventional length, area, and volume of Euclidean geometry to suitable subsets of the n-dimensional Euclidean space Rn. For instance, the Lebesgue measure of the interval [0, 1] in the real numbers is its length in the everyday sense of the word, specifically"
- And the statement:

  > "minimize the nth moment among all nonnegative measures that match the 0 through nâ1 moments of the given measure" ?

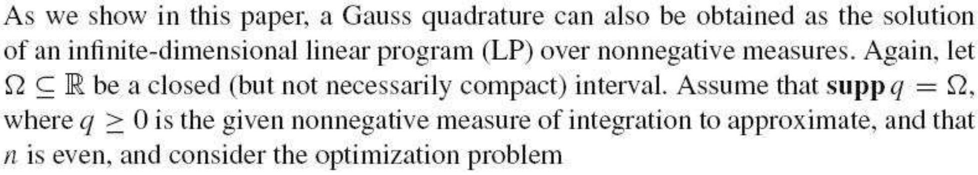
  What is the "nth moment" ? I understand that a moment is the integral of a monomial times a function, isn't it ? According to Wikipedia: "In mathematics, a moment is a specific quantitative measure of the shape of a function. It is used in both mechanics and statistics. If the function represents physical density, then the zeroth moment is the total mass, the first moment divided by the total mass is the center of mass, and the second moment is the rotational inertia. If the function is a probability distribution, then the zeroth moment is the total probability (i.e. one), the first moment is the expected value, the second central moment is the variance, the third standardized moment is the skewness, and the fourth standardized moment is the kurtosis. The mathematical concept is closely related to the concept of moment in physics."

- It can be gleaned from the preceding paragraph that perhaps the advantage lies in the interpretation of the problem. It introduces the concept of "moment". What can this concept be used for empirical functions ? How to define a moment in arbitrary domains, and for arbitrary functions —non-analytical ?
- Then Gautschi (2004) goes on and make a remark which crucial to understand his work. He introduces the notion of "efficient" quadrature rule: "We call a quadrature of order n with N ¡ n nodes efficient; such a quadrature requires fewer function evaluations than its order. The **linear equations for the weights of an efficient quadrature have more equations than variables**; these

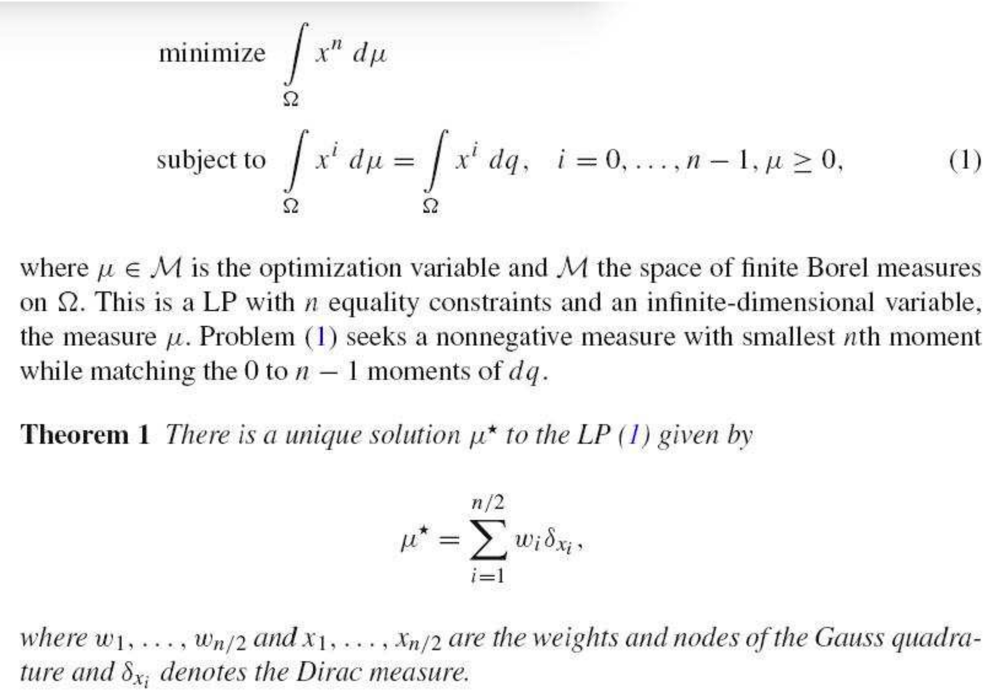
equations are not solvable unless the nodes are chosen very carefully". In 1814 Gauss [11] discovered the first efficient quadrature, which is now called a Gauss quadrature. A Gauss quadrature of order n requires only N = n/2 nodes (for n even). And then he poses the central claim of the paper: a **Gauss quadrature can also be obtained as the solution of an infinite-dimensional linear program (LP) over nonnegative measures**.

2.4.3 Gauss Quadrature Via Linear Programming

- What is **supp** q = Ω ? "In mathematics, the support of a real-valued function f is the subset of the domain containing those elements which are not mapped to zero. If the domain of f is a topological space, the support of f is instead defined as the smallest closed set containing all points not mapped to zero."



As we show in this paper, a Gauss quadrature can also be obtained as the solution of an infinite-dimensional linear program (LP) over nonnegative measures. Again, let $\Omega \subseteq \mathbb{R}$ be a closed (but not necessarily compact) interval. Assume that **supp** $q = \Omega$, where $q \geq 0$ is the given nonnegative measure of integration to approximate, and that n is even, and consider the optimization problem



$$\begin{aligned} &\text{minimize} \quad \int_\Omega x^n \, d\mu \\ &\text{subject to} \quad \int_\Omega x^i \, d\mu = \int_\Omega x^i \, dq, \quad i = 0, \ldots, n-1, \mu \geq 0, \end{aligned} \qquad (1)$$

where $\mu \in \mathcal{M}$ is the optimization variable and $\mathcal{M}$ the space of finite Borel measures on Ω. This is a LP with n equality constraints and an infinite-dimensional variable, the measure μ. Problem (1) seeks a nonnegative measure with smallest nth moment while matching the 0 to $n-1$ moments of dq.

Theorem 1 *There is a unique solution $\mu^\star$ to the LP (1) given by*

$$\mu^\star = \sum_{i=1}^{n/2} w_i \delta_{x_i},$$

where $w_1, \ldots, w_{n/2}$ and $x_1, \ldots, x_{n/2}$ are the weights and nodes of the Gauss quadrature and δ_{x_i} denotes the Dirac measure.

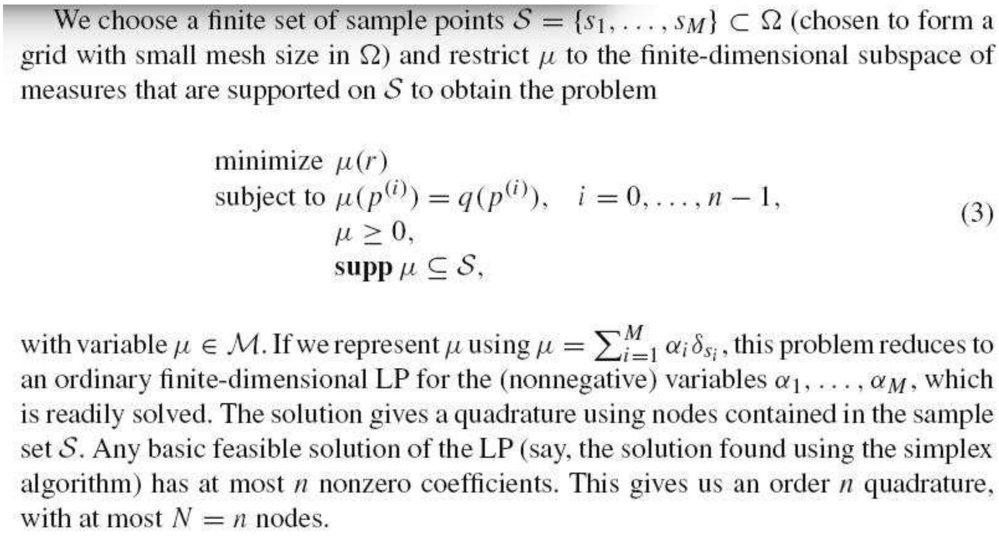
- Let us try to disclose bit by bit the above statement. For us, the closed interval will be [-1,1]. What I don't understand is what is $q \geq 0$ as a nonnegative measure of integration... Can we ignore this detail ?
- What is the space of finite Borel measures ? Wikipedia answer is not helpful at all —skipt it.
- Be that as it may, later on the author says that it is possible to pose the problem as a l1-minimization problem. In this regard, he says his idea is similar to that in Candès et al. (2006) —a paper with about 16000 citations !!!!
- Let us go on with the generalization of the Gauss Quadrature. Read section "Extensions of Gauss Quadrature Via Linear Programming". Monomial are replaced by arbitrary functions, the set is

defined in $\mathbb{R}^{d\times}$, they defined "sensitivity" functions, among other novelties. Then they talk about, again, about a Borel measure, whose support is Ω. The end statement is outright unpalatable. In view of this, let us turn our attention to the implementation per se.

- Now we are in item 4.

> On the other hand, our interest is limited to cases with d fixed and quite small, say, 2 or 3, in which case there are effective methods for solving (2). Several standard methods can solve such infinite-dimensional optimization problems when d is small. One approach focuses on the dual problem, which has a finite number of variables but an infinite number of constraints and so is called a semi-infinite program [4,10]. A cutting-planemethod can be used to solve the dual, fromwhichwe can construct a solution of the original (primal) problem. There are also algorithms that resemble the simplex or exchange method that directly solve the original problem [3,13].

(d is 2 or 3 for us). What is a semi-infinite LP program ? No idea. Let s proceed. "For the sake of completeness we describe a simple but effective method for solving (2) when d is small, say, 2 or 3. Our description is informal; for formal descriptions of an algorithm to solve the infinite-dimesional LP we refer the reader to the references cited earlier."



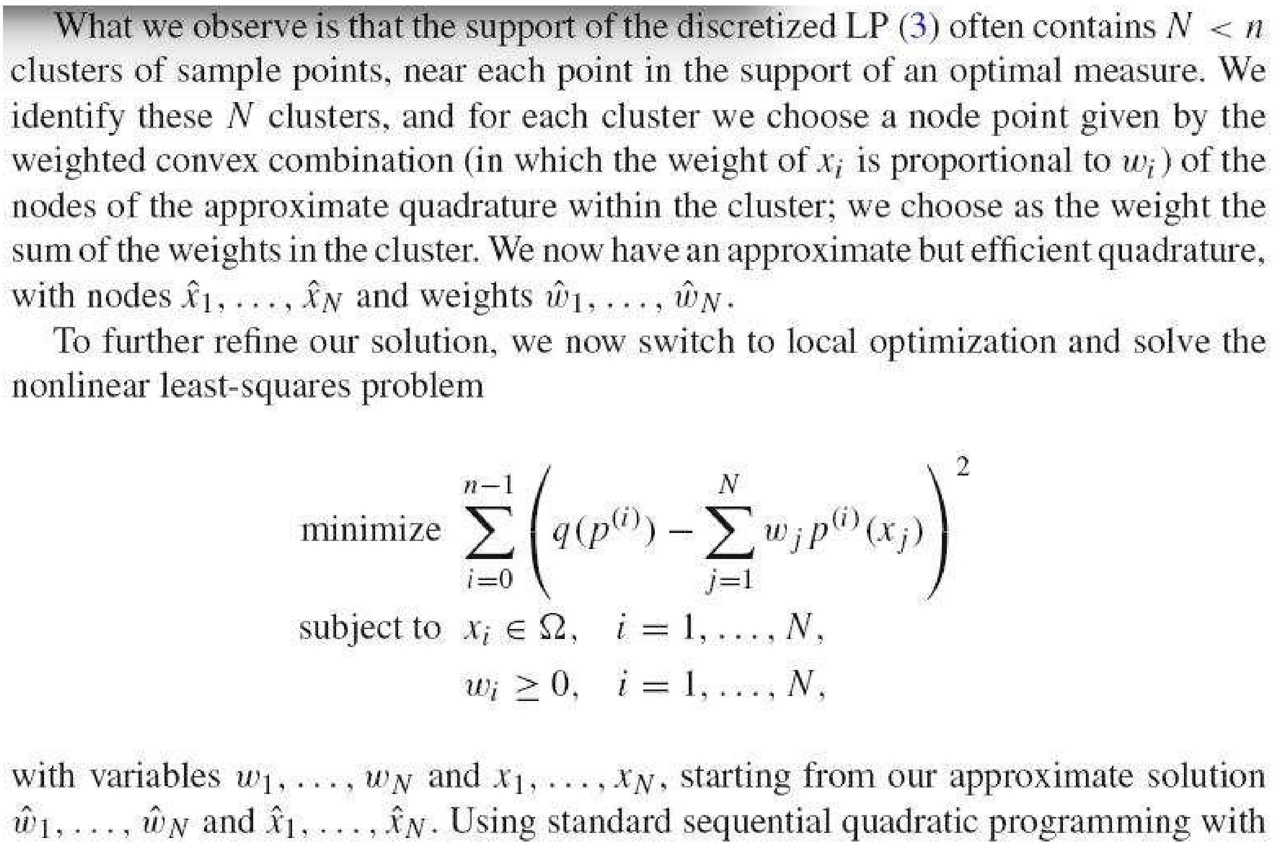
We choose a finite set of sample points $\mathcal{S} = \{s_1, \dots, s_M\} \subset \Omega$ (chosen to form a grid with small mesh size in Ω) and restrict μ to the finite-dimensional subspace of measures that are supported on $\mathcal{S}$ to obtain the problem

$$\begin{aligned} &\text{minimize } \mu(r) \\ &\text{subject to } \mu(p^{(i)}) = q(p^{(i)}), \quad i = 0, \dots, n-1, \\ &\qquad \mu \geq 0, \\ &\qquad \mathbf{supp}\, \mu \subseteq \mathcal{S}, \end{aligned} \tag{3}$$

with variable $\mu \in \mathcal{M}$. If we represent μ using $\mu = \sum_{i=1}^{M} \alpha_i \delta_{s_i}$, this problem reduces to an ordinary finite-dimensional LP for the (nonnegative) variables $\alpha_1, \dots, \alpha_M$, which is readily solved. The solution gives a quadrature using nodes contained in the sample set $\mathcal{S}$. Any basic feasible solution of the LP (say, the solution found using the simplex algorithm) has at most n nonzero coefficients. This gives us an order n quadrature, with at most $N = n$ nodes.

There is nothing obscure in the above excerpt. Everything is clear. When they represent the measure in a standard way, the problem boils down to an "ordinary finite-dimensional Linear Programming for the variables α_i". The nodes, of course, belong s to the sample set. This gives a quadrature with at most as many nodes as linearly independent functions. The only question here is: how to solve this LP program ? Via SIMPLEX ?

- What it bothers me is how intentionally obscure is the whole paper. They struggle to put the problem as abstract as possible, and then, the actual useful aspect, the implementation, is deliberately left incomplete. In the above excerpt, they admit the method per se does not yield an efficient rule. Because of this, they have to invoke next a local optimization problem:



What we observe is that the support of the discretized LP (3) often contains $N < n$ clusters of sample points, near each point in the support of an optimal measure. We identify these N clusters, and for each cluster we choose a node point given by the weighted convex combination (in which the weight of x_i is proportional to w_i) of the nodes of the approximate quadrature within the cluster; we choose as the weight the sum of the weights in the cluster. We now have an approximate but efficient quadrature, with nodes $\hat{x}_1, \ldots, \hat{x}_N$ and weights $\hat{w}_1, \ldots, \hat{w}_N$.

To further refine our solution, we now switch to local optimization and solve the nonlinear least-squares problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=0}^{n-1} \left(q(p^{(i)}) - \sum_{j=1}^{N} w_j p^{(i)}(x_j) \right)^2 \\ \text{subject to} & x_i \in \Omega, \quad i = 1, \ldots, N, \\ & w_i \geq 0, \quad i = 1, \ldots, N, \end{array}$$

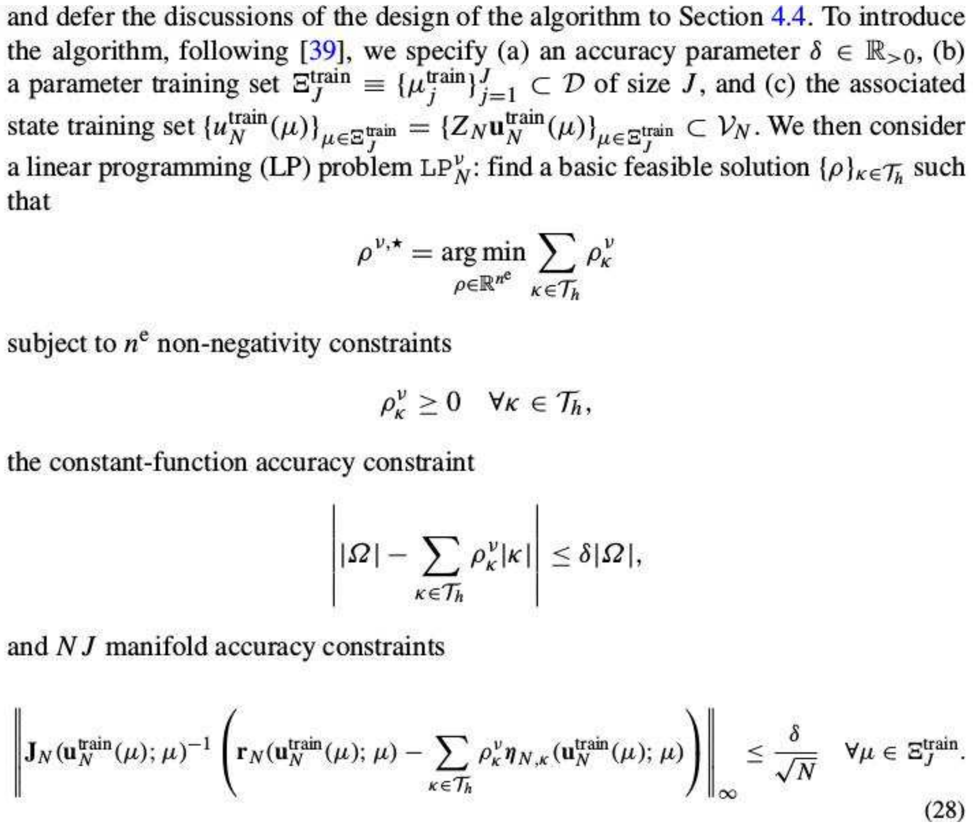
with variables $w_1, \ldots, w_N$ and $x_1, \ldots, x_N$, starting from our approximate solution $\hat{w}_1, \ldots, \hat{w}_N$ and $\hat{x}_1, \ldots, \hat{x}_N$. Using standard sequential quadratic programming with

an active set method [26], this typically converges quickly to a point with objective zero, and when it does (and if N ¡ n), we have an efficient quadrature.

- In summary, this paper only introduces a useful idea: posing the problem as a LP problem. But, eventually, the have to turn to heuristic to solve it. Hence, what are the gains?

2.4.4 Papers using Boyd's approach

- Let us analyze the approach is similar papers. For instance, in Yano (2019)



and defer the discussions of the design of the algorithm to Section 4.4. To introduce the algorithm, following [39], we specify (a) an accuracy parameter $\delta \in \mathbb{R}_{>0}$, (b) a parameter training set $\Xi_J^{\text{train}} \equiv \{\mu_j^{\text{train}}\}_{j=1}^J \subset \mathcal{D}$ of size J, and (c) the associated state training set $\{u_N^{\text{train}}(\mu)\}_{\mu\in\Xi_J^{\text{train}}} = \{Z_N\mathbf{u}_N^{\text{train}}(\mu)\}_{\mu\in\Xi_J^{\text{train}}} \subset \mathcal{V}_N$. We then consider a linear programming (LP) problem LP_N^ν: find a basic feasible solution $\{\rho\}_{\kappa\in\mathcal{T}_h}$ such that

$$\rho^{\nu,\star} = \underset{\rho\in\mathbb{R}^{n^e}}{\arg\min} \sum_{\kappa\in\mathcal{T}_h} \rho_\kappa^\nu$$

subject to n^e non-negativity constraints

$$\rho_\kappa^\nu \geq 0 \quad \forall\kappa \in \mathcal{T}_h,$$

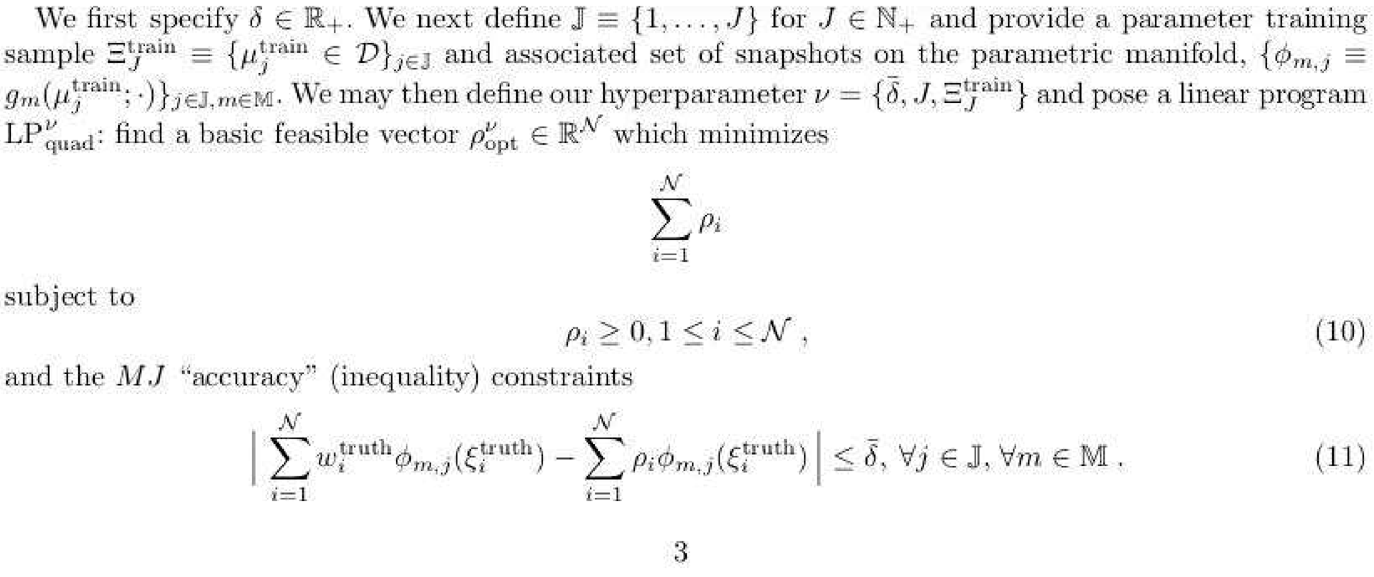
the constant-function accuracy constraint

$$\left| |\Omega| - \sum_{\kappa\in\mathcal{T}_h} \rho_\kappa^\nu |\kappa| \right| \leq \delta|\Omega|,$$

and NJ manifold accuracy constraints

$$\left\| \mathbf{J}_N(\mathbf{u}_N^{\text{train}}(\mu);\mu)^{-1} \left(\mathbf{r}_N(\mathbf{u}_N^{\text{train}}(\mu);\mu) - \sum_{\kappa\in\mathcal{T}_h} \rho_\kappa^\nu \eta_{N,\kappa}(\mathbf{u}_N^{\text{train}}(\mu);\mu) \right) \right\|_\infty \leq \frac{\delta}{\sqrt{N}} \quad \forall\mu \in \Xi_J^{\text{train}}. \tag{28}$$

In the above, ρ are the weights, the constraints are the expected ones, but what about the objective function ? What is the sense of this objective function ? Let's see what other paper state in this respect. In Patera and Yano (2017), we find



We first specify $\delta \in \mathbb{R}_+$. We next define $\mathbb{J} \equiv \{1,\ldots,J\}$ for $J \in \mathbb{N}_+$ and provide a parameter training sample $\Xi_J^{\text{train}} \equiv \{\mu_j^{\text{train}} \in \mathcal{D}\}_{j\in\mathbb{J}}$ and associated set of snapshots on the parametric manifold, $\{\phi_{m,j} \equiv g_m(\mu_j^{\text{train}};\cdot)\}_{j\in\mathbb{J},m\in\mathbb{M}}$. We may then define our hyperparameter $\nu = \{\bar{\delta}, J, \Xi_J^{\text{train}}\}$ and pose a linear program $\mathrm{LP}_{\text{quad}}^\nu$: find a basic feasible vector $\rho_{\text{opt}}^\nu \in \mathbb{R}^{\mathcal{N}}$ which minimizes

$$\sum_{i=1}^{\mathcal{N}} \rho_i$$

subject to

$$\rho_i \geq 0, 1 \leq i \leq \mathcal{N}\ , \tag{10}$$

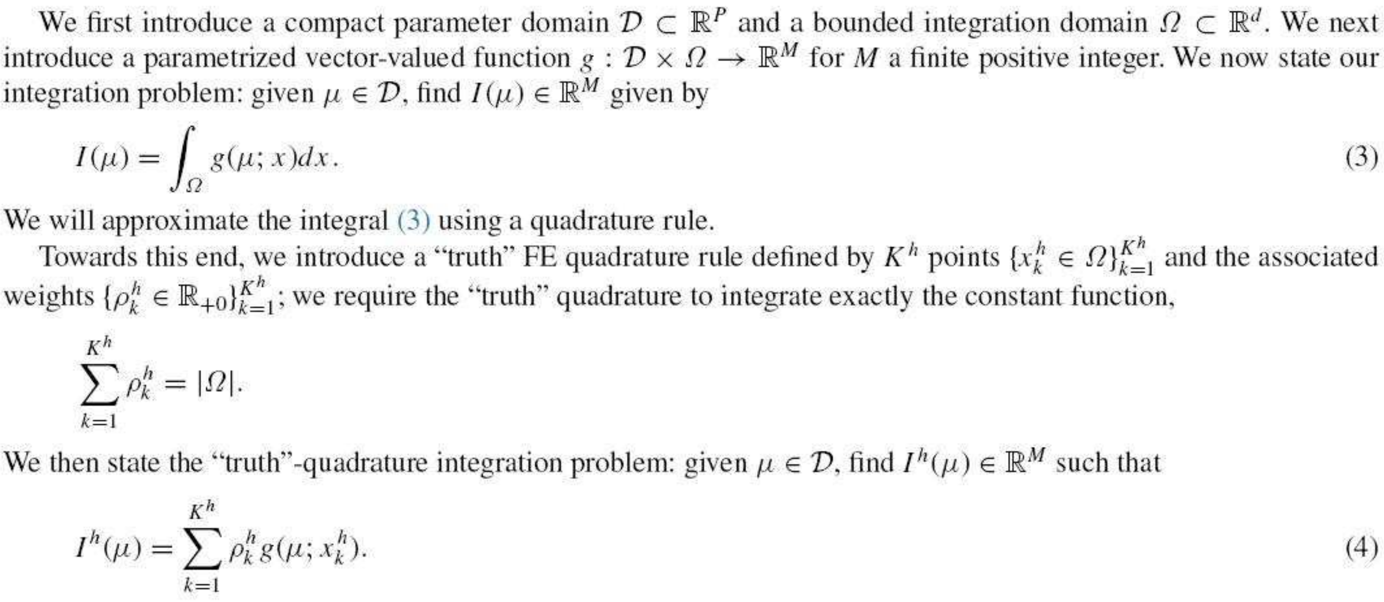
and the MJ "accuracy" (inequality) constraints

$$\left| \sum_{i=1}^{\mathcal{N}} w_i^{\text{truth}} \phi_{m,j}(\xi_i^{\text{truth}}) - \sum_{i=1}^{\mathcal{N}} \rho_i \phi_{m,j}(\xi_i^{\text{truth}}) \right| \leq \bar{\delta},\ \forall j \in \mathbb{J}, \forall m \in \mathbb{M}\ . \tag{11}$$

3

I don't get it. Why this objective function ? The sum of all weight should be minimum... Why ?

- Yano and Patera (2019) seems to be more accurate in the definitions. The formulation is quite clear



We first introduce a compact parameter domain $\mathcal{D} \subset \mathbb{R}^P$ and a bounded integration domain $\Omega \subset \mathbb{R}^d$. We next introduce a parametrized vector-valued function $g : \mathcal{D} \times \Omega \to \mathbb{R}^M$ for M a finite positive integer. We now state our integration problem: given $\mu \in \mathcal{D}$, find $I(\mu) \in \mathbb{R}^M$ given by

$$I(\mu) = \int_\Omega g(\mu; x)dx. \tag{3}$$

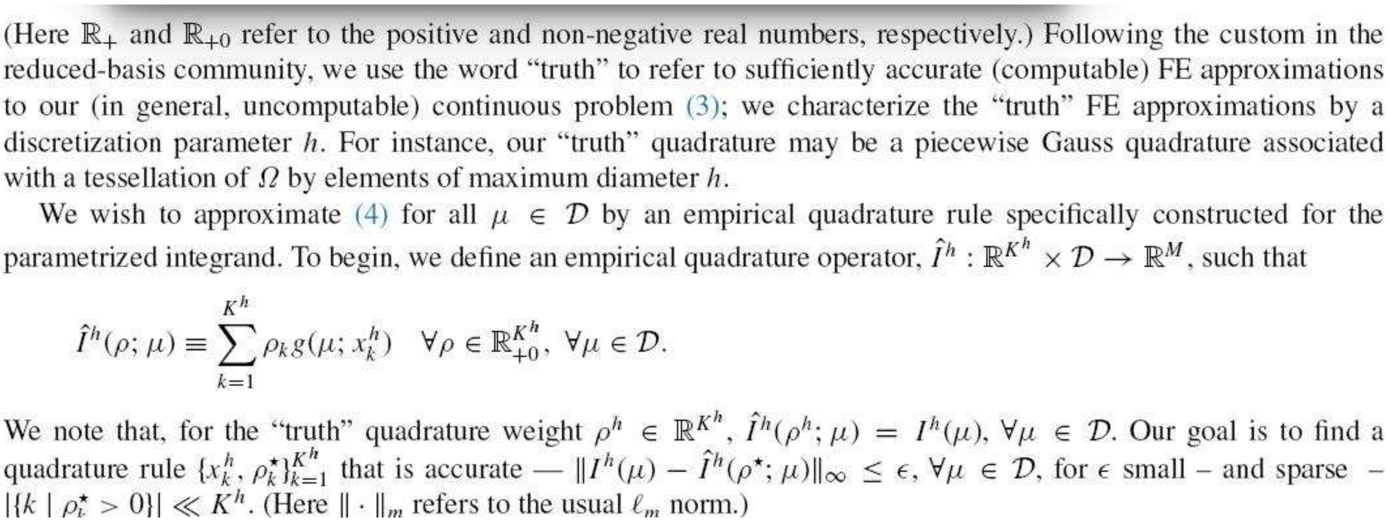
We will approximate the integral (3) using a quadrature rule.

Towards this end, we introduce a "truth" FE quadrature rule defined by K^h points $\{x_k^h \in \Omega\}_{k=1}^{K^h}$ and the associated weights $\{\rho_k^h \in \mathbb{R}_{+0}\}_{k=1}^{K^h}$; we require the "truth" quadrature to integrate exactly the constant function,

$$\sum_{k=1}^{K^h} \rho_k^h = |\Omega|.$$

We then state the "truth"-quadrature integration problem: given $\mu \in \mathcal{D}$, find $I^h(\mu) \in \mathbb{R}^M$ such that

$$I^h(\mu) = \sum_{k=1}^{K^h} \rho_k^h g(\mu; x_k^h). \tag{4}$$



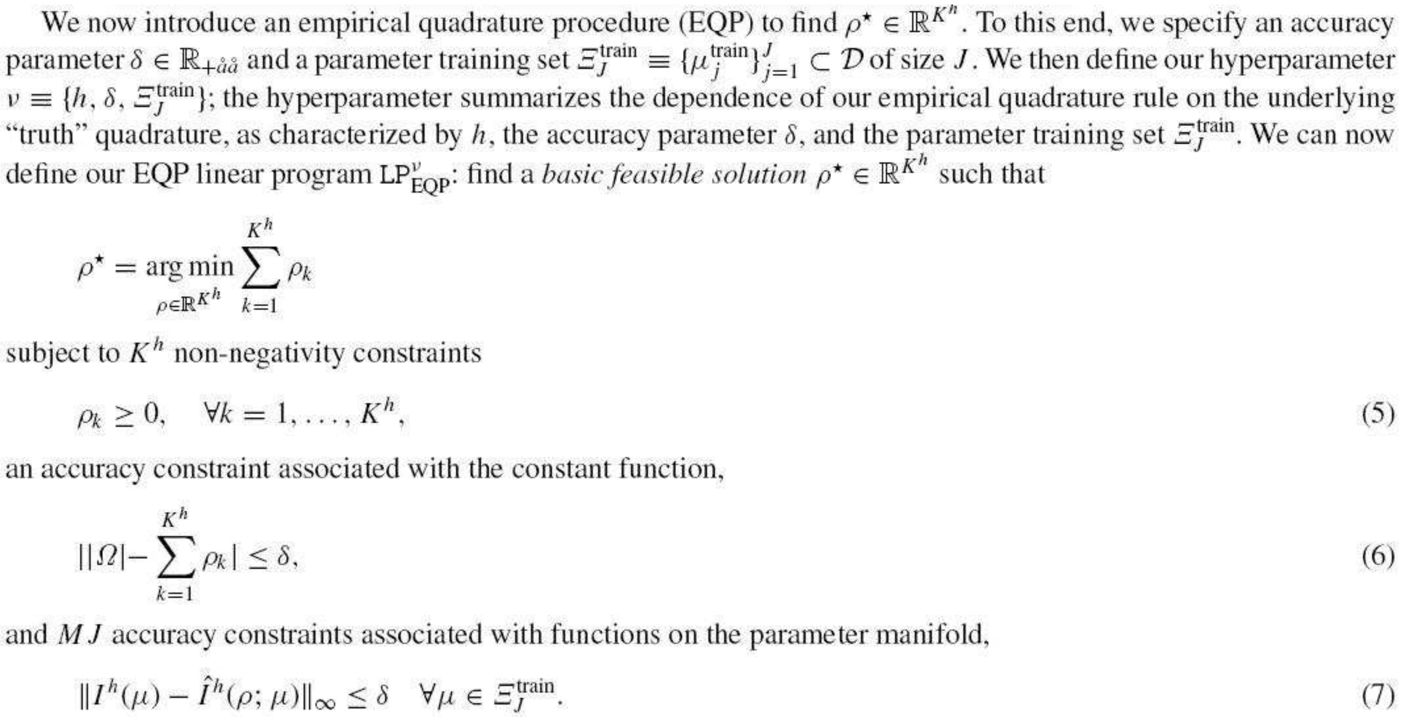
(Here $\mathbb{R}_+$ and $\mathbb{R}_{+0}$ refer to the positive and non-negative real numbers, respectively.) Following the custom in the reduced-basis community, we use the word "truth" to refer to sufficiently accurate (computable) FE approximations to our (in general, uncomputable) continuous problem (3); we characterize the "truth" FE approximations by a discretization parameter h. For instance, our "truth" quadrature may be a piecewise Gauss quadrature associated with a tessellation of Ω by elements of maximum diameter h.

We wish to approximate (4) for all $\mu \in \mathcal{D}$ by an empirical quadrature rule specifically constructed for the parametrized integrand. To begin, we define an empirical quadrature operator, $\hat{I}^h : \mathbb{R}^{K^h} \times \mathcal{D} \to \mathbb{R}^M$, such that

$$\hat{I}^h(\rho; \mu) \equiv \sum_{k=1}^{K^h} \rho_k g(\mu; x_k^h) \quad \forall \rho \in \mathbb{R}_{+0}^{K^h}, \ \forall \mu \in \mathcal{D}.$$

We note that, for the "truth" quadrature weight $\rho^h \in \mathbb{R}^{K^h}$, $\hat{I}^h(\rho^h; \mu) = I^h(\mu)$, $\forall \mu \in \mathcal{D}$. Our goal is to find a quadrature rule $\{x_k^h, \rho_k^\star\}_{k=1}^{K^h}$ that is accurate — $\|I^h(\mu) - \hat{I}^h(\rho^\star; \mu)\|_\infty \leq \epsilon$, $\forall \mu \in \mathcal{D}$, for ϵ small – and sparse – $|\{k \mid \rho_k^\star > 0\}| \ll K^h$. (Here $\|\cdot\|_m$ refers to the usual ℓ_m norm.)

- First consideration. As in Farhat's approaches, the solution of the problem should be a sparse vector. Here it goes the definite statement



We now introduce an empirical quadrature procedure (EQP) to find $\rho^\star \in \mathbb{R}^{K^h}$. To this end, we specify an accuracy parameter $\delta \in \mathbb{R}_{+\text{åä}}$ and a parameter training set $\Xi_J^{\mathrm{train}} \equiv \{\mu_j^{\mathrm{train}}\}_{j=1}^J \subset \mathcal{D}$ of size J. We then define our hyperparameter $\nu \equiv \{h, \delta, \Xi_J^{\mathrm{train}}\}$; the hyperparameter summarizes the dependence of our empirical quadrature rule on the underlying "truth" quadrature, as characterized by h, the accuracy parameter δ, and the parameter training set Ξ_J^{train}. We can now define our EQP linear program $\mathrm{LP}_{\mathrm{EQP}}^{\nu}$: find a *basic feasible solution* $\rho^\star \in \mathbb{R}^{K^h}$ such that

$$\rho^\star = \arg\min_{\rho \in \mathbb{R}^{K^h}} \sum_{k=1}^{K^h} \rho_k$$

subject to K^h non-negativity constraints

$$\rho_k \geq 0, \quad \forall k = 1, \ldots, K^h, \tag{5}$$

an accuracy constraint associated with the constant function,

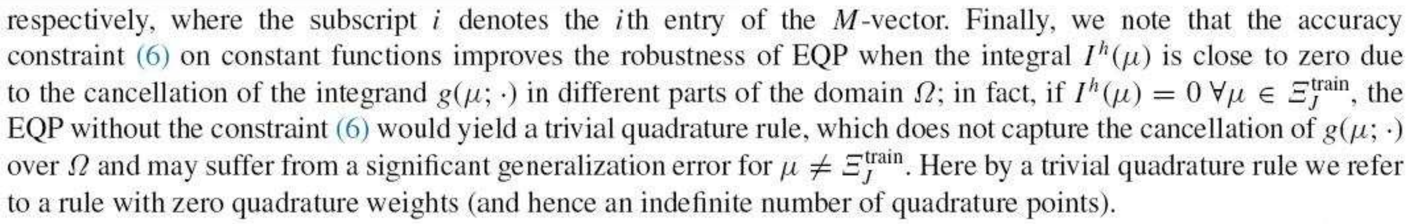
$$||\Omega| - \sum_{k=1}^{K^h} \rho_k| \leq \delta, \tag{6}$$

and MJ accuracy constraints associated with functions on the parameter manifold,

$$\|I^h(\mu) - \hat{I}^h(\rho; \mu)\|_\infty \leq \delta \quad \forall \mu \in \Xi_J^{\mathrm{train}}. \tag{7}$$

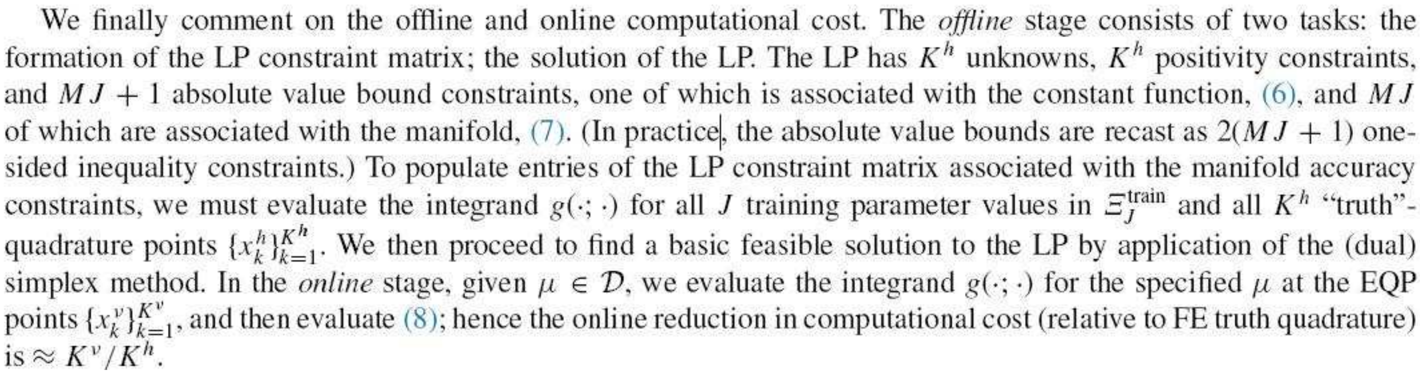
Why the objective function ?? I know that the sum of the weights must be equal to the volume. But why to minimize it ? Furthermore, the constraints are not linear. Why therefore do they call it "linear programming" ? Infinity norm ? Is not the maximum ?

- What bothers me the most is that they have taken concepts from Hernández et al. (2017) without properly acknowledging it:



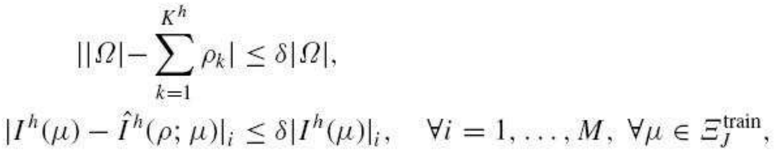
respectively, where the subscript i denotes the ith entry of the M-vector. Finally, we note that the accuracy constraint (6) on constant functions improves the robustness of EQP when the integral $I^h(\mu)$ is close to zero due to the cancellation of the integrand $g(\mu; \cdot)$ in different parts of the domain Ω; in fact, if $I^h(\mu) = 0 \; \forall \mu \in \Xi_J^{\mathrm{train}}$, the EQP without the constraint (6) would yield a trivial quadrature rule, which does not capture the cancellation of $g(\mu; \cdot)$ over Ω and may suffer from a significant generalization error for $\mu \neq \Xi_J^{\mathrm{train}}$. Here by a trivial quadrature rule we refer to a rule with zero quadrature weights (and hence an indefinite number of quadrature points).

- The ensuing excerpt contains very useful information, for it throws some light on the actual method employed for solving the problem



We finally comment on the offline and online computational cost. The *offline* stage consists of two tasks: the formation of the LP constraint matrix; the solution of the LP. The LP has K^h unknowns, K^h positivity constraints, and $MJ + 1$ absolute value bound constraints, one of which is associated with the constant function, (6), and MJ of which are associated with the manifold, (7). (In practice, the absolute value bounds are recast as $2(MJ + 1)$ one-sided inequality constraints.) To populate entries of the LP constraint matrix associated with the manifold accuracy constraints, we must evaluate the integrand $g(\cdot; \cdot)$ for all J training parameter values in Ξ_J^{train} and all K^h "truth"-quadrature points $\{x_k^h\}_{k=1}^{K^h}$. We then proceed to find a basic feasible solution to the LP by application of the (dual) simplex method. In the *online* stage, given $\mu \in \mathcal{D}$, we evaluate the integrand $g(\cdot; \cdot)$ for the specified μ at the EQP points $\{x_k^\nu\}_{k=1}^{K^\nu}$, and then evaluate (8); hence the online reduction in computational cost (relative to FE truth quadrature) is $\approx K^\nu / K^h$.

In practice, the absolute value bounds are recast as 2(MJ + 1) one-sided inequality constraints. Furthermore, they put the problem in a way in which it is clear that the constraints are in the l1-norm:



$$||\Omega| - \sum_{k=1}^{K^h} \rho_k| \leq \delta |\Omega|,$$
$$|I^h(\mu) - \hat{I}^h(\rho;\mu)|_i \leq \delta |I^h(\mu)|_i, \quad \forall i = 1, \ldots, M, \ \forall \mu \in \Xi_J^{\mathrm{train}},$$

And what is the “dual simplex method” ???? How can this method be optimal ?

sec:tenta

2.4.4.1 Summary

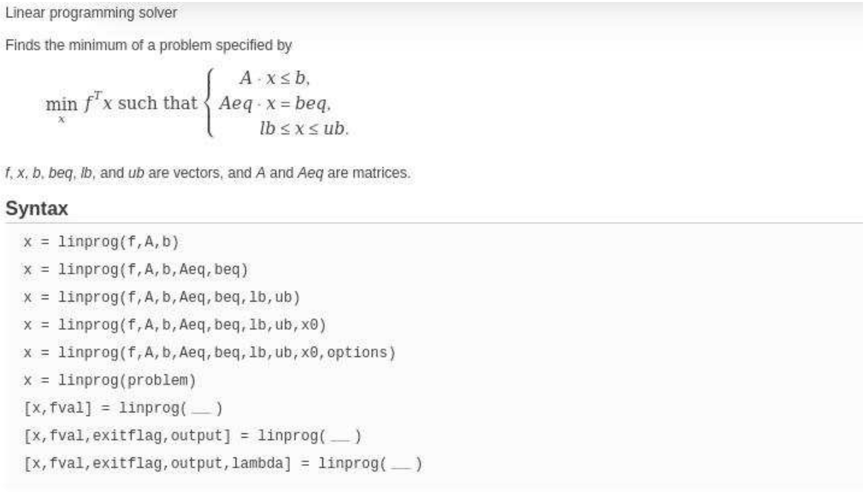
- Paper Yano and Patera (2019) is a good starting point for exploring the lp-quadrature method.
- It only remains to ascertain how a constraint of the type $|Ax - b| \geq c$ can be put in a way in which the absolute value operator disappears. Why not to introduce a “slack” variable ? There is actually information about this in wikipedia: (see here). So it is indeed feasible to solve this problem. It would be of utmost importance to evaluate the performance of this method against the ECM. I think this is relatively trivial using matlab function “linprog”. Yano and Patera (2019) suggests to use the dual simplex method

```
options = optimoptions('linprog','Algorithm','dual-simplex');
```

2.4.5 Tentative implementation

- FOLDER: LPquadrature
- Application to polynomials functions. See IntegrationPoly1.m
- The problem we have to address is as follows
  - Design variable: $\boldsymbol{w} \in \mathbb{R}^M$, where M is the number of Gauss points (total)
  - Objective function
  $$\boldsymbol{\mathcal{I}}^T \boldsymbol{w} \tag{2.4.1}$$
  where $\boldsymbol{\mathcal{I}} = [1, 1, 1, 1...]^T$
  - M Positive constraints $\boldsymbol{I}\boldsymbol{w} \geq 0$, where $\boldsymbol{I}$ is the identity matrix, or equivalently
  $$-\boldsymbol{I}\boldsymbol{w} \leq \boldsymbol{0} \tag{2.4.2}$$
  - 2 “measure” constraints: $|\boldsymbol{\mathcal{I}}^T \boldsymbol{w} - V| \leq \epsilon_v V$, where $V = \boldsymbol{\mathcal{I}}^T \boldsymbol{W}$ and $0 \leq \epsilon_v \leq 1$. This in turn translates into
  $$(\boldsymbol{\mathcal{I}}^T \boldsymbol{w} - V) \leq \epsilon_v V \quad \Rightarrow \quad \boldsymbol{\mathcal{I}}^T \boldsymbol{w} \leq (1 + \epsilon_v) V \tag{2.4.3}$$
  $$-(\boldsymbol{\mathcal{I}}^T \boldsymbol{w} - V) \leq \epsilon_v V \quad \Rightarrow \quad -\boldsymbol{\mathcal{I}}^T \boldsymbol{w} \leq (-1 + \epsilon_v) V \tag{2.4.4}$$
  - $2n$ constraints of the functions themselves. Such functions are normalized so that $\boldsymbol{W}^T \boldsymbol{g} = \boldsymbol{0}$. Accordingly, we can write $|\boldsymbol{G}\boldsymbol{w}| \leq \epsilon_g$. This boils down to
  $$\boldsymbol{G}\boldsymbol{w} \leq \epsilon_g \tag{2.4.5}$$
  and
  $$-\boldsymbol{G}\boldsymbol{w} \leq \epsilon_g \tag{2.4.6}$$

- Matlab format



Linear programming solver

Finds the minimum of a problem specified by

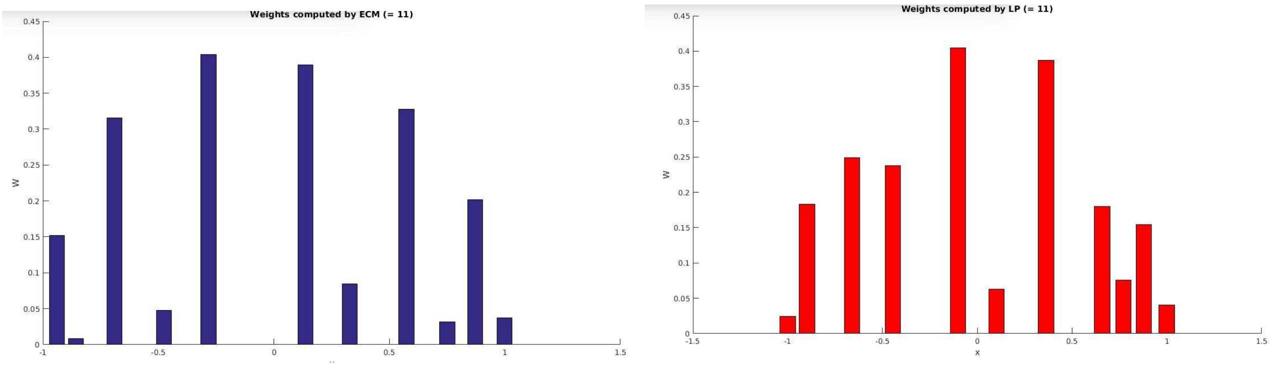
$$\min_x f^T x \text{ such that } \begin{cases} A \cdot x \le b, \\ Aeq \cdot x = beq, \\ lb \le x \le ub. \end{cases}$$

f, x, b, beq, lb, and *ub* are vectors, and *A* and *Aeq* are matrices.

Syntax

```
x = linprog(f,A,b)
x = linprog(f,A,b,Aeq,beq)
x = linprog(f,A,b,Aeq,beq,lb,ub)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)
x = linprog(problem)
[x,fval] = linprog(___)
[x,fval,exitflag,output] = linprog(___)
[x,fval,exitflag,output,lambda] = linprog(___)
```

- Code: see script IntegrationPoly1.m
- RESULTS:



(a) (b)

Figure 2.1 *a) Integration of a polynomial of degree* $p = 10$ *in [-1,1]. a) ECM. b) LP. Notice that both methods give the same number of points (*$p = 11$*).*

2.4.6 Conclusion (26-Feb-2020)

In this example, I see no apparent advantage in using the LP method. In fact, the ECM turns out to be faster. Nevertheless, it may prove interesting to test this approach in other scenarios, such as in ROM problems —lest the simplicity of the studied be masking any unknown virtue.

2.5 Towards and efficient Empirical Cubature Methods

2.5.1 Abstract

The present paper is concerned with the issue of dimensional hyperreduction. The goal of this paper is twofold. First, we aim at providing a benchmark problem for testing existing approaches for hyperreduction. The chosen ROM is the classical of modal analysis. We wish to explore how many integration points are required to integrate a ROM of n modes, using existing techniques. Secondly, we wish to know whether existing techniques provide a lower bound for the number of points —that is, what is the minimum number