Computer-Assisted Proofs in Solving Linear Parametric Problems

Evgenija D. Popova
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Acad. G. Bonchev str., block 8, 1113 Sofia, Bulgaria
epopova@bio.bas.bg

Abstract

Consider a linear system $A(p)x = b(p)$ whose input data depend on a number of uncertain parameters $p = (p_1, \ldots, p_k)$ varying within given intervals $[p]$. The objective is to verify by numerical computations monotonic (and convexity/concavity) dependence of a solution component $x_i(p)$ with respect to a parameter p_j over the interval box $[p]$, or more general, to prove if some boundary $\inf / \sup x_i(p)$ for all $p \in [p]$ is attained at the end-points of $[p]$. Such knowledge is useful in many applications in order to facilitate the solution of some underlying linear parametric problem involving uncertainties.

In this paper we present a technique, for proving the desired properties of the parametric solution, which is alternative to the approaches based on extreme point computations. The proposed computer-aided proof is based on guaranteed interval enclosures for the partial derivatives of the parametric solution for all $p \in [p]$. The availability of self-validated methods providing guaranteed enclosure of a parametric solution set by floating-point computations is a key for the efficiency and the expanded scope of applicability of the proposed approach. Linear systems involving nonlinear parameter dependencies, and dependencies between $A(p)$ and $b(p)$, as well as non-square linear parametric systems can be handled successfully. Presented are details of the algorithm design and Mathematica *tools implementing the proposed approach. Numerical examples from structural mechanics illustrate its application.*

1. Introduction

Consider the linear algebraic system

$$A(p)x \ = \ b(p), \tag{1}$$

where the elements of the $n \times n$ matrix $A(p)$ and the vector $b(p)$ depend on a k-tuple of parameters $p = (p_1, \ldots, p_k)$ which are uncertain and varying within given intervals

$$a_{ij}(p) = a_{ij}(p_1, \ldots, p_k), \quad b_i(p) = b_i(p_1, \ldots, p_k), \quad i, j = 1, \ldots, n, \tag{2}$$

$$p \in [p] = ([p_1], \ldots, [p_k]). \tag{3}$$

Such systems are common in many engineering analysis or design problems, models in operational research, linear prediction problems, etc., where there are complicated dependencies between the coefficients of the system [1]-[4], [7], [13]. The uncertainties in the model parameters could originate from an inexact knowledge of these parameters, measurement imprecision, or round-off errors.

In this paper a real compact interval is denoted by $[a] = [a^-, a^+] := \{a \in \mathbb{R} \mid a^- \leq a \leq a^+\}$. By $\mathbb{IR}^n, \mathbb{IR}^{n\times m}$ we denote the sets of interval n-vectors, resp. interval $n \times m$ matrices. For $[a] = [a^-, a^+]$, $0 \notin [a]$, define $\mathrm{sgn}([a]) := \{1 \text{ if } a^- > 0, \ -1 \text{ if } a^+ < 0\}$. The end-point functionals $(\cdot)^-, (\cdot)^+$ and the sgn functional are applied to interval vectors and matrices componentwise. We assume the reader is familiar with conventional interval arithmetic [6]. Let $U(k) := \{u \in \mathbb{R}^k \mid |u| = (1, ..., 1)^\top\}$, where the absolute value is understood componentwise, $|u| = (|u_1|, \ldots, |u_k|)^\top)$ for $u \in \mathbb{R}^k$. For $[a] \in \mathbb{IR}^n$ and $u \in U(n)$, a^u is defined by $a_i^u := \{a_i^- \text{ if } u_i = 1; a_i^+ \text{ if } u_i = -1\}$, $i = 1, \ldots, n$. For a set of indices $\mathcal{I} = \{i_1, \ldots, i_n\}$, the vector $(x_{i_1}, \ldots, x_{i_n})^\top$ will be denoted by $x_{\mathcal{I}}$.

The set of solutions to (1–3), called parametric solution set, is

$$\begin{aligned} \Sigma^p &= \Sigma\left(A(p), b(p), [p]\right) \\ &:= \{x \in \mathbb{R}^n \mid \exists\, p \in [p], A(p)x = b(p)\}\,. \end{aligned} \tag{4}$$

In many applications, when the solution components $x_i = x_i(p)$ of (1–3) are proven to be convex, concave or monotonic, the solution of some underlying problem of interest can be facilitated. Most often, for a non-empty bounded set (4), one needs to find its convex hull

$$\square\, \Sigma^p \ := \ [\inf \Sigma^p, \ \sup \Sigma^p] \ = \ \cap\{[x] \in \mathbb{IR}^n \mid \Sigma^p \subseteq [x]\},$$

or an interval vector $[y] \supseteq \square\Sigma^p$ which is most tight. The following theorem is well-known.

Theorem 1 *Let $A(p)$ be nonsingular for all $p \in [p]$ and let the components of $x(p) = A^{-1}(p)b(p)$ be monotonic in $[p]$ with respect to each p_ν, $\nu = 1, \ldots, k$. Then for $i = 1, \ldots, n$*

$$\{\square\Sigma^p\}_i = \left[\left\{A^{-1}(p^u)\right\}_i b(p^u),\ \left\{A^{-1}(p^{-u})\right\}_i b(p^{-u})\right],$$

where $u_\nu = \operatorname{sign}\frac{\partial x_i(p)}{\partial p_\nu}$, $\nu = 1, \ldots, k$.

Even when the solution components are not monotonic in $[p]$ monotonicity can also be used provided the behavior of the solution is such that the following property holds

$$\square\Sigma^p = [\min_{u \in U(k)} A^{-1}(p^u)b(p^u),\ \max_{u \in U(k)} A^{-1}(p^u)b(p^u)].$$

Such parametric solution sets will be called *possessing the combinatorial property*.

Practical examples exploiting the above properties are common in the design and modelling under deterministic uncertainties [1], [2], [13]. Similar examples can also be given in a stochastic context. For example, when it can be established that p enters a performance measure in a convex or concave manner, it becomes possible to carry out a so-called robust Monte Carlo simulation [2]. Also, when the components of p correspond to design parameters and $x_i(p)$ is some system quantity to be minimized, the convex dependence on p facilitates computation of some optimal setting $p = p^*$ for the system.

In order to be exploited, monotonicity, convexity/concavity, or combinatorial property of the parametric solution should be proven. Given a parametric system (1–3) with nonsingular matrix $A(p)$ for all $p \in [p]$, a solution component $x_i(p)$ is monotonic or convex/concave w.r.t. a parameter p_ν if $\frac{\partial x_i(p)}{\partial p_\nu}$, or $\frac{\partial^2 x_i(p)}{\partial p_\nu^2}$ respectively, have the same sign for all $p \in [p]$. For rank-one uncertainty structures[1], some extreme point results are given in [2]. Namely, it is proven that the desired same-sign conditions can be ascertain by checking the sign of certain multilinear affine functions at the (extreme) end-points of $[p]$. Another work [8] contains some verifiable sufficient conditions under which the parametric solution set (4) possesses the combinatorial property, or some bounds of the parametric solution set can be obtained by extreme point computations. The present work is motivated by the deficiencies of the approaches used in [2], [8]. First, proving solution set properties by extreme point computations is applicable only to linear systems with rank-one uncertainty structure. Second, point computations are guaranteed only if performed in exact arithmetic. Third, the computational efficiency of extreme point computations performed in exact arithmetic is too low.

[1] A pair $(A(p), b(p))$ has rank-one uncertainty structure if each parameter p_ν enters into either $A(p)$ or $b(p)$ but not both and the dependencies are affine-linear, cf. [2].

In view that proving the discussed properties of the parametric solutions is important also for other more general problem classes, e.g. such involving nonlinear parameter dependencies or non-square systems, in this work we present another approach for proving the desired properties. The computer-aided proof will be based on verifiable signs for the first partial derivatives (respectively the second partial derivatives for convexity/concavity property) of the parametric solution. The new line of attack consists in the way the proof will be done, namely, by self-validated solver of parametric interval linear systems which provides guaranteed solution enclosure in floating-point computations. In Section 2 the proposed alternative approach is presented in details together with its algorithmic design providing best computational efficiency. Some software tools implementing the proposed approach are presented in Section 3. This section demonstrates a much expanded scope of applicability of the proposed technique. Examples from structural mechanics illustrate its application. The computational efficiency of a proof based on self-validated parametric solver is compared to the approach based on extreme point computations.

2. Computer-aided proofs

The pioneering idea for computer-assisted monotonicity proofs related to parametric linear systems is due to Jiří Rohn [14]. He considered linear systems with rank-one uncertainty structure and employed the information about the monotonicity of the parametric solution with respect to the system parameters in order to obtain an enclosure of the parametric solution set. To prove the monotonicity properties Rohn's approach is based on enclosing the solution sets of corresponding linear systems for the partial derivatives of the original system. The deficiency of this first attempt in proving monotonicity properties of the parametric solution is that the linear systems for the partial derivatives are solved by "any classical method" for enclosing the solution of nonparametric interval linear systems. In other words, the idea of Rohn was to reduce the solution of a parametric linear system to the solution of several nonparametric interval linear systems and then to solve point linear systems corresponding to the monotonicity types. Since solving the derivative systems, which are also parametric, by nonparametric methods is practically useless due to the huge overestimation of the solutions (derivatives), Rohn's approach was improved in [9] by solving the derivative systems as parametric ones. In this section we further elaborate this approach ensuring its computational efficiency.

Consider the system of linear equations (1–3). Each parameter may enter both $A(p)$ and $b(p)$, and there is no restriction on the dependence between the parameters — $a_{ij}(p)$ and $b_i(p)$, $i, j = 1, \ldots, n$, could be either affine-linear or nonlinear functions of the parameters. The latter are considered to vary within given intervals $[p_1], \ldots, [p_k]$. In the sequel we assume that $A(p)$ is nonsingular for all $p \in [p]$ and there is a function implementing some method for guaranteed enclosure of the solution set (4) of (1–3). Such methods exist, e.g. a general-purpose self-validating method for parametric linear systems is proposed in [16] and generalized in [10]. For its implementation and application to a variety of practical problems involving either affine-linear or rational dependencies see [12] and the literature cited therein. A more efficient enclosure method is proposed in [7] for the special case of rank-one uncertainty structures. The concepts of self-validating methods and their mathematical background are discussed in many works, see [17] or [15], [16]. Mathematical rigor in the computer arithmetic using directed rounding, in algorithm design, and in program execution allow to guarantee that the hypotheses of suitable inclusion theorems are (or are not) satisfied and thus to guarantee that the stated problem has (or does not have) a solution in an enclosing interval. Note that self-validating methods for linear systems prove existence and uniqueness of the solution for $p \in [p]$ and therefore the non-singularity of $A(p)$.

2.1. Global monotonicity proof

Taking the partial derivative $\frac{\partial}{\partial p_\nu}$ on both sides of (1) we obtain

$$\frac{\partial A(p)}{\partial p_\nu} x(p) + A(p) \frac{\partial x(p)}{\partial p_\nu} \;=\; \frac{\partial b(p)}{\partial p_\nu}, \tag{5}$$

where the partial derivatives are applied to vectors and matrices componentwise. Denote $\Delta^\nu := \frac{\partial x(p)}{\partial p_\nu}$, $b^\nu(p) := \frac{\partial b(p)}{\partial p_\nu}$, $A^\nu(p) := \frac{\partial A(p)}{\partial p_\nu}$, where the superscripts do not mean power but corresponding ν-th derivative vector or matrix. Assuming that $[x^*] \in \mathbb{IR}^n$ is a solution enclosure generated by a self-verified parametric solver, $[x^*] \supseteq \square\Sigma^p$, from (5) we obtain the interval linear system

$$A(p)\Delta^\nu \;=\; b^\nu(p) - A^\nu(p)[x^*], \tag{6}$$

involving the original parametric matrix and a right-hand-side vector depending on the original parameters and the interval vector of the initial solution enclosures. If $\mathcal{X}^\nu$ denotes the solution set of (6), enclosing it by any self-validated numerical method will give us

$$[\Delta^\nu] \supseteq \square\mathcal{X}^\nu \supseteq \left\{ \frac{\partial x(p)}{\partial p_\nu} \mid p \in [p] \right\}. \tag{7}$$

Thus, if for some $i = 1, \ldots, n$, $0 \notin [\Delta^\nu]_i$ then the corresponding solution component $x_i(p)$ is monotonic with respect to p_ν with a monotonicity type sign $\left(\frac{\partial x_i(p)}{\partial p_\nu}\right) =$ sgn $([\Delta^\nu]_i)$. The success of this computer-aided proof will depend on the sharpness of the enclosure (7). To provide a best possible enclosure, the system (6) should be solved by a parametric method. Furthermore, if $A^\nu(p)x$ involves some $x_i, (1 \le i \le n)$ in more than one of the vector components, introducing additional parameters $x_i \in [x^*]_i, (1 \le i \le n)$ will reduce the solution overestimation due to the dependency problem. In order that no exceed parameters are introduced, we define an index set $J^\nu := \{j \mid \{A^\nu(p)\}_{\cdot j} \neq 0\}$, where $\{A^\nu(p)\}_{\cdot j} \neq 0$ means that the jth column of $A^\nu(p)$ is not equal to the zero vector. Denoting $r^\nu(p, x_{J^\nu}) := \frac{\partial b(p)}{\partial p_\nu} - A^\nu(p)x$ and solving the partial derivative system

$$\begin{aligned} A(p) \cdot \Delta^\nu(p, x_{J^\nu}) = \; & r^\nu(p, x_{J^\nu}) \\ & p \in [p], \quad x_{J^\nu} \in [x^*_{J^\nu}] \end{aligned} \tag{8}$$

by a self-validating parametric solver, will provide a best possible enclosure $[\Delta^\nu]$ for the set of ν-th partial derivatives. Since the quality of the enclosure $[\Delta^\nu]$ in (7) depends also on the sharpness of the initial enclosure $[x^*]$, it might be helpful to resolve the derivative parametric system with an improved enclosure $[x^*]$ of (1). This will be exploited in Section 2.2 and demonstrated in Section 3.

Let for fixed i, $1 \le i \le n$, there exist index sets

$$\begin{aligned} L_+ \;&:=\; \{\nu \mid \text{sgn}(\left[\frac{\partial x_i}{\partial p_\nu}\right]) = 1\}, \\ L_- \;&:=\; \{\nu \mid \text{sgn}(\left[\frac{\partial x_i}{\partial p_\nu}\right]) = -1\}. \end{aligned}$$

If $L_- \cup L_+ = \{1, \ldots, k\}$ then the exact bounds of $\{\Sigma^p\}_i$ can be obtained by solving two point linear systems, given below, in exact arithmetic or a very sharp floating-point enclosure can be delivered by a self-validated solver for the same two point systems

$$\begin{aligned} [\inf \Sigma^p, \sup \Sigma^p]_i = [\; & \{A^{-1}(p^-_{L_+}, p^+_{L_-}) b(p^-_{L_+}, p^+_{L_-})\}_i, \\ & \{A^{-1}(p^+_{L_+}, p^-_{L_-}) b(p^+_{L_+}, p^-_{L_-})\}_i]. \end{aligned}$$

However, for some i, $1 \le i \le n$, $x_i(p)$ may be not monotonic for p_ν in the domain $[p]$ or the enclosure $[\Delta^\nu]$ is so rough that $0 \in [\Delta^\nu]$, or just $x_i(p)$ may be not dependent of p_ν. This means that there exist index sets L_+, L_-, defined as above, and $L_0 := \{\nu \mid 0 \in [\Delta^\nu]_i\}$ so that

$$L_+ \cup L_- \cup L_0 = \{1, \ldots, k\}, \quad L_0 \neq \emptyset. \tag{9}$$

If (9) holds, considering two new parametric linear systems, involving a reduced number of parameters $p_{L_0} \in [p_{L_0}]$,

$$A^-(p_{L_0})y = b^-(p_{L_0}), \qquad A^+(p_{L_0})z = b^+(p_{L_0}), \tag{10}$$

wherein $a^-_{ij}(p_{L_0}) := a_{ij}(p^-_{L_+}, p^+_{L_-}, p_{L_0})$, $b^-_i(p_{L_0}) := b_i(p^-_{L_+}, p^+_{L_-}, p_{L_0})$, $a^+_{ij}(p_{L_0}) := a_{ij}(p^+_{L_+}, p^-_{L_-}, p_{L_0})$,

$b_i^+(p_{L_0}) := b_i(p_{L_+}^+, p_{L_-}^-, p_{L_0})$, a sharper enclosure of $\{\Sigma^p\}_i$ can be obtained as

$$[\inf \Sigma^p,\ \sup \Sigma^p]_i \subseteq [y^*]_i \cup [z^*]_i \subseteq [x^*]_i.$$

Next we present how in case of (9) it is still possible to proof monotonicity of $x_i(p)$ with respect to p_{L_0} or to proof combinatorial bounds for $\{\Sigma^p\}_i$.

2.2. Local monotonicity proof

Let $[y^*], [z^*]$ be enclosures of the solution sets of the corresponding parametric systems (10). These enclosures usually give an overestimated enclosure of $\Sigma(A(p), b(p), [p])$ but they can be used in an attempt to prove monotonicity properties of $x(p)$ in the local domains $(p_{L_+}^-, p_{L_-}^+, [p_{L_0}])$, $(p_{L_+}^+, p_{L_-}^-, [p_{L_0}])$, respectively. To this end, let fix $q \in L_0$. Taking the partial derivative $\frac{\partial}{\partial p_q}$ on the equations (10), we solve the following derivative systems corresponding to (10) and the parameter $p_q \in p_{L_0}$

$$\begin{aligned} A^-(p_{L_0})\Delta^{y,q} &= r^{y,q}(p_{L_0}, y_{J^{y,q}}) \\ A^+(p_{L_0})\Delta^{z,q} &= r^{z,q}(p_{L_0}, z_{J^{z,q}}), \end{aligned}$$

where $\Delta^{y,q} := \frac{\partial y(p_{L_0})}{\partial p_q}$, $r^{y,q}(p_{L_0}, y_{J^{y,q}}) := \frac{\partial b^-(p_{L_0})}{\partial p_q} - A^-(p_{L_0}) y_{J^{y,q}}$, $J^{y,q} := \{j \mid \{\frac{A^-(p_{L_0})}{\partial p_q}\}_{\cdot j} \neq 0\}$; $\Delta^{z,q}$ and $r^{z,q}(p_{L_0}, y_{J^{z,q}})$ are defined analogously with respect to z. Now, if $[\Delta^{y,q}]$, $[\Delta^{z,q}]$ are the corresponding solution enclosures and $0 \notin [\Delta^{y,q}]_i$, $0 \notin [\Delta^{z,q}]_i$, define $\lambda := \mathrm{sgn}([\Delta^{y,q}]_i)$, $\mu := \mathrm{sgn}([\Delta^{z,q}]_i)$. Thus, $\{\inf \Sigma(A(p), b(p), [p])\}_i$ will be determined by the endpoint p_q^λ of $[p_q] := [p_q^-, p_q^+]$ and $\{\sup \Sigma(A(p), b(p), [p])\}_i$ will be determined by p_q^μ. Note that $\lambda = \mu$ means that $x_i(p)$ is monotone w.r.t. p_q while $\lambda \neq \mu$ means that the corresponding boundaries are combinatorial. In the latter case it may happen that $0 \notin [\Delta^{y,q}]_i$ but $0 \in [\Delta^{z,q}]_i$ and vice-versa.

To be more clear, below we give an algorithm for the local monotonicity procedure, where ParametricSSolve is a function providing guaranteed enclosure of a parametric linear system and globalMonotoneType is a function performing global monotonicity proof and delivering an $n \times k$ matrix monType with elements $-1, 1$, or 0.

localMonotoneType(Ap, bp, $[p]$, monType)
For $i = 1, \ldots, n$
  If $0 \notin \{\text{monType}\}_i$ then $\{\text{mT}\}_i = \{\text{monType}\}_i$
  else begin
    decompose $\{\text{monType}\}_i$ into L_+, L_-, L_0
    generate $A = (a_{ij})$ by $a_{ij} := \{Ap\}_{ij}(p_{L_+}^-, p_{L_-}^+, p_{L_0})$
        $b = (b_i)$ by $b_i := \{bp\}_i(p_{L_+}^-, p_{L_-}^+, p_{L_0})$
    $[s^*] = \text{ParametricSSolve}(A, b, [p_{L_0}])$
    $y = \{\text{globalMonotoneType}(A, b, [p_{L_0}], [s^*])\}_i$
    generate $A = (a_{ij})$ by $a_{ij} := \{Ap\}_{ij}(p_{L_+}^+, p_{L_-}^-, p_{L_0})$
        $b = (b_i)$ by $b_i := \{bp\}_i(p_{L_+}^+, p_{L_-}^-, p_{L_0})$
    $[s^*] = \text{ParametricSSolve}(A, b, [p_{L_0}])$
    $z = \{\text{globalMonotoneType}(A, b, [p_{L_0}], [s^*])\}_i$
    For $\nu = 1, \ldots, k$
      t = position of ν in L_0
      $\{\text{mT}\}_{i\nu} := \begin{cases} \{y_t, z_t\} & \text{if } \nu \in L_0 \\ \{\text{monType}\}_{i\nu} & \text{otherwise} \end{cases}$
    end
  end
end
Return mT

For the sake of an efficient implementation of the proposed approach it should be noted that the original parametric system and the parametric systems of the partial derivatives for all parameters have same matrix and differ only in their right hand side vectors. This allows inverting of only one point matrix and using the same iteration matrix when finding an initial solution enclosure and enclosing the solutions to all partial derivative systems.

2.3. Convexity/concavity proof

Following the considerations along the above lines, one can proof convexity or concavity by the same technique enclosing the solutions of the parametric systems for the second partial derivatives of the original system.

3. Illustrative examples

This section provides some examples illustrating the application of the approach proposed in Section 2. Furthermore, presented are examples demonstrating the advantages and an expanded scope of applications of the approach based on self-validated parametric solvers. In comparison to the approaches based on extreme point computations [1], [2], [8], dealing only with rank-one uncertainty structures, the proposed approach can handle dependencies between $A(p)$ and $b(p)$ as well as nonlinear dependencies. Combinatorial bounds of the parametric solution set can be proven as well as the respective properties of non-square linear parametric problems. The computational efficiency is also discussed.

First we present two problems from structural mechanics which are chosen to be small so that the structure of the parameter dependencies in the corresponding systems is visible. In [13] the presented approach is applied to larger systems coming from finite element approximation of mechanical problems. Namely, combinatorial property of the parametric solution set hull is proven for parametric systems with nonlinear dependencies modeling a one-bay steel frame and a two-bay two-story frame, the latter involving 18 equations and 13 uncertain parameters. The third example, considered in Section 3.3, involves 81 equations and 101 uncertain parameters. It should be noted that when the concerned parametric linear system is a result of discretized equation of a continued problem, e.g. the finite element discretization of a partial differential equation, proving a certain property of the linear system would not mean the same property for the original infinite dimensional problem. It is beyond the scope of this method to account for the discretization error of the mathematical model in addition to the uncertainty in the parameters, although there are some recent investigations in this direction.

The numerical computations are performed in the environment of *Mathematica* [18] where numerous functions are developed particularly for solving linear parametric problems with uncertain data. The initial version of a *Mathematica* package `IntervalComputations`LinearSystems`` [9] supported functions for verified solving of non-parametric interval linear systems and parametric linear systems involving affine-linear dependencies. Latter on the package was extended by a function for verified solving of linear systems whose input data are rational functions of interval parameters [12], [13] and functions allowing efficient handling of non-square (over- or underdetermined) linear parametric systems [11]. All self-verified solvers optionally deliver an inner estimation of the computed outer enclosure of the solution set hull so that the quality of the latter can be estimated. Expanding further the above functionality, several functions, supporting the approach presented in Section 2, are implemented: For a given parametric system (1–3) and an enclosure `xp` of the parametric solution set, `globalMonotoneType(Ap,bp,pars,xp)` is a function that solves k parametric systems (8) and delivers an $n \times k$ matrix with elements $-1, 1$, or 0 denoting the corresponding partial derivative signs, 0 means that monotonicity cannot be proven. For a given parametric system (1–3) and an $n \times k$ matrix `monType`, containing information about the global monotonicity properties of the solution and involving zero elements, `localMonotoneType(Ap,bp,pars,monType)` verifies local monotonicity properties of the parametric solution and delivers an $n \times k$ table whose elements can be $0, 1, -1$, or $\{\lambda, \mu\}$, where $\lambda, \mu \in \{0, 1, -1\}$, corresponding to global monotonicity, or combinatorial, or both parametric solution properties. For a given parametric system (1–3) and an $n \times k$ table `monType`, containing information about the monotonicity and/or combinatorial properties of the solution, `monotoneParametricSolve(Ap,bp,pars,monType)` is a function which based on the information in `monType` delivers a guaranteed enclosure of the parametric solution set. The elements of `monType` can be $0, 1, -1$, or $\{\lambda, \mu\}$ where $\lambda, \mu \in \{0, 1, -1\}$.

The implementation of all parametric solvers is based on a general purpose self-validated parametric method proposed by S. Rump [16] and generalized in [10]. Although the computations presented below in this section use these solvers (the function `ParametricSSolve`), the proposed approach can be applied by using other self-validated parametric solvers, e.g. one based on method proposed in [7].

3.1. Planar frame

Consider a simple planar frame with three types of support and an external load distributed uniformly along the beam as shown in Figure 1. The frame is modelled by using

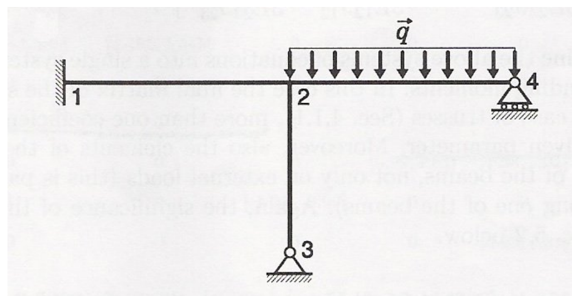

Figure 1. Planar frame after [4].

the method of forces and assuming small displacements and linear elastic material law (for more details see [4], [12]). It is assumed that all beams have the same Young modulus E but momentum of inertia J of the beam cross-sections are related by the formula $J_{12} = J_{23} = 1.5J_{24}$. The lengths of the beams and the load are considered to be uncertain with $l_{12}, l_{24} \in [0.95, 1.05]$, $l_{23} \in [0.7125, 0.7875]$, $q \in [7.5, 12.5]$.

Bounds for the moments and the reactions in the planar frame model should be obtained by solving the following linear system

$$\begin{pmatrix} 2l_{12} & l_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ l_{12} & 2l_{12}+2l_{23} & -2l_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & -2l_{23} & 2l_{23}+3l_{24} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & l_{12} & l_{12}+l_{24} & 0 & l_{23} \\ -1 & 1 & 0 & -l_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & l_{24} & 0 & 0 \end{pmatrix} x = \begin{pmatrix} 0 \\ 0 \\ -\frac{3}{8}ql_{24}^3 \\ 0 \\ ql_{24} \\ ql_{24}l_{12} + \frac{1}{2}l_{24} \\ 0 \\ \frac{1}{2}ql_{24}^2 \end{pmatrix}.$$

Note that the right-hand side depends on the beam lengths, not only on the external load (this is partly due to the presence of distributed load along one of the beams), and the dependence is nonlinear. Our goal is to prove monotonicity properties of the system solution or some combinatorial properties of the solution which will allow a sharp enclosure of the system response. We assume that the required *Mathematica* functions are loaded in the memory and that the variables `mat, vec, pars` contain the parametric matrix, the r.h. side vector and the 4-tuple of parameters $(l_{12}, l_{23}, l_{24}, q)$ with their interval values, respectively. Note that the computer algebra environment of *Mathematica* allows the same mathematical (symbolic) notations, as in the above definition of the system, to be used for entering the input data. Our approach requires that first we find an initial enclosure of the parametric solution set and then execute the function `globalMonotoneType`.

```
In[2]:=xp=ParametricSSolve[mat, vec, pars];
    mT1 = globalMonotoneType[mat,vec,pars,xp]
Out[3]={{0, 1,0, 1},{1,-1,0,-1},
        {1,-1,0,-1},{1,-1,0,-1},{-1,1,0,1},
        {1, 0,0, 1},{0, 0,0,-1}, {0,0,0,1}}
```

Thus, for each solution component we managed to prove monotonic dependence with respect to some of the parameters. The information about that dependence (`mT1`) is used by the function `localMonotoneType` to compute sharper solution enclosures for the initial parametric system and to prove local monotonicity properties of the derivative systems corresponding to the lower and upper solution set bounds for each of the solution components.

```
In[4]:= mT2=localMonotoneType[mat, vec,
                 pars, mT1] /.  {{t_, t_}->t}
Out[4]={{-1,1,1,1}, {1,-1,-1,-1},
        {1,-1,1,-1},{1,-1,-1,-1},{-1,1,1,1},
        {1,1,1,1}, {0,-1,1,-1}, {0,1,-1,1}}
```

The proven local monotonicity is of the same type for both derivative systems which means corresponding global monotonicity properties. Unfortunately, the dependence of the last two solution components w.r.t. the first parameter is still not known. However, we can use again the improved knowledge (`mT2`) about the solution global monotonicity properties.

```
In[5]:= localMonotoneType[mat, vec, pars,
                        mT2] /.  {{t_, t_}->t}
Out[5]={{-1,1,1,1}, {1,-1,-1,-1},
       {1,-1,1,-1},{ 1,-1,-1,-1},{-1,1,1,1},
       {1, 1,1, 1},{-1,-1, 1,-1},{1,1,-1,1}}
```

Thus global monotonicity properties of the parametric solution are rigorously proven which allows finding the exact bounds or a guaranteed very sharp enclosure for the parametric system response.

3.2. Cantilevar beam

Consider the cantilevar beam shown in Figure 2. Each

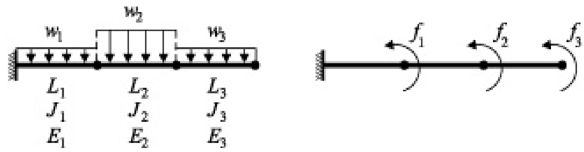

Figure 2. Cantilevar beam.

beam element is characterized by its length L_i, modulus of elasticity E_i, and momentum of inertia J_i, and is loaded with uniformly distributed load w_i, $i = 1, 2, 3$. It is assumed that the axial deformations are neglected. All the parameters are considered to be uncertain. The beam is modelled by the following linear system

$$\begin{pmatrix} \frac{4p_1}{L_1} + \frac{4p_2}{L_2} & \frac{2p_2}{L_2} & 0 \\ \frac{2p_2}{L_2} & \frac{4p_2}{L_2} + \frac{4p_3}{L_3} & \frac{2p_3}{L_3} \\ 0 & \frac{2p_3}{L_3} & \frac{4p_3}{L_3} \end{pmatrix} x = \begin{pmatrix} -\frac{w_1 L_1^2}{12} + \frac{w_2 L_2^2}{12} \\ -\frac{w_2 L_2^2}{12} + \frac{w_3 L_3^2}{12} \\ -\frac{w_3 L_3^2}{12} \end{pmatrix},$$

where $p_1, p_2, p_3 \in [0.018905, 0.021105]$, $p_i = E_i J_i$, $i = 1, 2, 3$, represent variations in the material properties, the element lengths are $L_1 \in [9.95, 10.05]$, $L_2 \in [7.96, 8.04]$, $L_3 \in [5.97, 6.03]$, and the distributed loads w_1, w_2, w_3 vary independently within $[0, 4]$. The goal is to prove solution properties allowing a sharp enclosure of the system response. We apply our approach as in the previous example but with new input data. The 9-tuple of parameters is $(p_1, p_2, p_3, L_1, L_2, L_3, w_1, w_2, w_3)$.

```
In[7]:=xp=ParametricSSolve[mat, vec, pars];
     mT1=globalMonotoneType[mat,vec,pars,xp]
Out[8]= {{0,0,0,0,0,0,-1, 1,-1},
         {0,0,0,0,0,0, 1,-1, 1},
         {0,0,0,0,0,0,-1, 1,-1}}
```

It is proven a monotonic dependence of the parametric solution with respect to the last three (load) parameters w_1, w_2, w_3. Due to this we can check the local monotonicity of the corresponding derivative system couples and we get the following result

```
In[9]:= localMonotoneType[mat,vec,pars,mT1]
Out[9]={
{{1,-1},{1,-1},{1,-1},{-1,1},{-1,1},{-1,1},-1,1,-1},
{{1,-1},{1,-1},{1,-1},{-1,1},{-1,1},{-1,1},1,-1,1},
{{1,-1},{1,-1},{1,-1},{-1,1},{-1,1},{-1,1},-1,1,-1}}
```

which shows the combinatorial property of the parametric solution set. Furthermore, to compute the solution set hull there is no need to solve $2^9 = 256$ point linear systems corresponding to all possible combinations of the interval end-points but only those 4 corresponding to the global and local monotonicity types, obtained in `Out[9]`.

3.3. Truss Cantilevar

As a large example we consider the truss cantilevar shown in Figure 3 and proposed as a benchmark example by R. Muhanna [5]. A finite element model for a one-bay

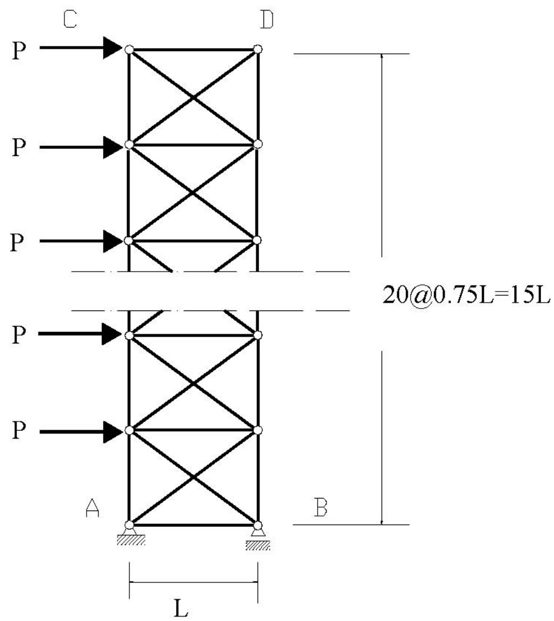

Figure 3. Truss cantilevar after [5].

20-floor truss cantilever consists of 42 nodes and 101 elements. The modulus of elasticity E_i of each element is considered to be uncertain. A complete description of the problem can be found in [5]. The model leads to a parametric linear system involving 81 variables and 101 uncertain parameters E_i connected by affine-linear dependencies. The computer-assisted approach based on a self-verified parametric solver is applied for proving the combinatorial property of the solution set hull for 2% parameter uncertainties, that is $E_i \in [0.99, 1.01], i = 1, \ldots, 101$.

The computer-aided proof was slow for such a problem involving large number of uncertain parameters. Some experiments showed that ignoring the rounding errors in the calculations not explicitly involving intervals (approach used by many authors for the sake of speed) while solving the parametric systems leads to some wrong monotonicity types. This example demonstrates the necessity of a fully rigorous implementation of the parametric solvers used for a computer-aided proof. Monotonicity or combinatorial properties of most solution components are proven. Here we present the properties of the last two solution components corresponding to the displacements in horizontal and vertical direction of the upper right corner (D, in Figure 3), which is the problem of interest. While the solution component 80 is monotone with respect to all parameters: (-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1), the last solution component possesses the combinatorial property w.r.t. E_{11}, that is (1, 1, 1, 1, -1, -1, 1, 1, 1, -1, {-1, 1}, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1).

Although it can be theoretically proven for some problems that the extremal values of any solution component are attained at vertices of the parameter box, which are these vertices is not known. Thus, finding the exact solution set hull by combinatorial extreme point calculations may be prohibitive for large number of parameters (truss cantilever system requires solving approx. 2.5353×10^{30} point systems), while the proof based on verified parametric solvers, although heavy and slow, is usually feasible providing in addition explicit information for the monotonicity types.

3.4. Properties of least squares solutions

It was mentioned in [2], as an interesting open problem, the extent to which least squares solutions

$$x^{\text{LS}}(p) = \left(A^\top(p)A(p)\right)^{-1} A^\top(p)b(p),$$

where $A(p)$ is an $n \times m$ matrix with rank $A(p) = m < n$ for all admissible $p \in [p]$, are monotonic, convex or concave with respect to the uncertain model parameters. Results along these lines would be important when studying the dependence of the least squares solution on either design variables or uncertain parameters.

Non-parametric interval least squares estimates are considered by several authors. In [3] a solution enclosure is sought by solving the normal equation $A^\top Ax = A^\top b$. Because the spectral radius of the matrix is greater than 1 for most problems, interval enclosures are obtained by a specially designed methodology based on QR decomposition. The overestimation of the obtained enclosures is assessed by computing the convex hull of the solution basing on the signs of the partial derivatives of the solution with respect to the elements of the matrix and the right hand side vector. The utilized monotonicity approach is almost the same as the exploited in this work with the important difference that the sign of the partial derivatives is evaluated at the interval mid points and not proven over the interval box. It is well known that such point estimates may be not correct in general.

To avoid the ill-conditioning of $A^\top(p)A(p)$ in enclosing the solutions of non-square over- or underdetermined parametric linear systems, we follow the proposal of S. Rump [15] for nonparametric systems, and consider a corresponding augmented square linear system. For the least squares problem, the augmented system is

$$\begin{pmatrix} A(p) & -I \\ 0 & A^\top(p) \end{pmatrix} \cdot \begin{pmatrix} x^{\mathrm{LS}} \\ y \end{pmatrix} \quad = \quad \begin{pmatrix} b(p) \\ 0 \end{pmatrix}. \qquad (11)$$

The augmented system, being parametric even for nonparametric nonsquare systems, helps also for reducing the dependencies in the parametric normal equation. How to handle efficiently the parameter dependence in the above augmented system is detailed in [11]. Thus, for a rigorous computer-assisted proof, the corresponding derivative systems should be constructed from the augmented system (11) and their solutions should be enclosed by the algorithms presented in the preceding sections.

3.5. Computational efficiency

Beside the expanded scope of applications of the presented approach, it is computationally cheaper compared to the extreme point computations [2], [8]. For rank-one uncertainty structures, checking monotonicity properties of the parametric solution, as proposed in [2], depends exponentially on the number of the parameters and requires about $\alpha 2^k$ solutions of point linear systems, where α is a factor depending on the implementation. Proving the monotonicity properties by the presented approach, based on self-validated parametric solvers, requires solving of $1+k$ parametric interval linear systems. In case that local monotonicity should be additionally proven, the number of parametric solvers increases by $2(n+1)$ for each execution of this procedure.

Based on the available self-validated parametric solver, the success of a proof depends on the quality of that solver, that is its ability to provide sharp solution enclosures. For example, the general-purpose parametric fixed-point iteration [16] is convergent only for strongly regular parametric matrices and may fail for some $A(p)$ which is regular but not strongly regular on $[p]$. It is also well-known that this interval parametric method produces quite overestimated solution enclosures, and even fail, for very large parameter intervals. On another side, a recently developed method [7] is extremely efficient for enclosing the solution set of large scale parametric systems involving rank-one uncertainty structures and wide parameter intervals.

4. Conclusion

We demonstrated the application of self-verified parametric solvers for computer-assisted numerical proof of some properties of the parametric solution. Namely, monotonicity and convexity/concavity dependence of the solution components with respect to the system parameters, as well as extreme point results for the convex hull of the parametric solution set can be proven by guaranteed floating-point computations. The methodological novelties of the proposed technique reside at two places of the paper: First, the algorithm for proving monotonicity of the solution components requires solving a parametric linear system (with the same matrix) involving additional parameters in the right-hand side, the parameters corresponding to the components of the initial solution enclosure. This provides sharper enclosure of the derivatives, and thus the success of the method (without necessity of interval subdivision). Second, the combinatorial hull of the parametric solution set is achieved by proving local monotonicity properties of the solution components.

The proposed technique, based on self-validated parametric solvers, does not have the limitations of the extreme point methods [2], [8] for rank-one uncertainty structures, allows handling of more general parameter dependencies and non-square parametric linear systems at a cheaper price providing in addition guaranteed results. The numerical proof of parametric solution properties is applicable to problems formulated in terms of different uncertainty theories which rely on interval arithmetic for computations, such as deterministic interval uncertainties, fuzzy set theory, random set theory, probability bounds theory, or mixed type uncertainties.

Acknowledgements. This work was partially supported by the Bulgarian National Science Fund under grant No. MM1301/03.

References

[1] A. Dreyer. *Interval Analysis of Analog Circuits with Component Tolerances*. PhD thesis, Kaiserslautern Univ. of Technology, Kaiserslautern, 2005.

[2] A. Ganesan, S. Ross, B. Ross Barmish. An Extreme Point Result for Convexity, Concavity and Monotonicity of Parametrized Linear Equation Solutions. *LAA*, 390:61–73, 2004.

[3] D. M. Gay. Interval Least Squares — a Diagnostic Tool. In R. E. Moore (Ed). Reliability in Computing, *Perspectives in Computing* 19, Academic Press, San Diego, 1988, 183–205.

[4] Z. Kulpa, A. Pownuk, I. Skalna. Analysis of Linear Mechanical Structures with Uncertainties by Means of Interval Methods. *CAMES* 5:443–477, 1998.

[5] R. L. Muhanna. Benchmarks for Interval Finite Element Computations. Web site, 2004. http://www.gtsav.gatech.edu/rec/ifem/benchmarks.html

[6] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.

[7] A. Neumaier, A. Pownuk. Linear Systems with Large Uncertainties, with Applications to Truss Structures, November 2004. http://www.mat.univie.ac.at/ neum/ms/linunc.pdf

[8] E. D. Popova. Quality of the Solution Sets of Parameter-Dependent Interval Linear Systems. *ZAMM*, 82(10):723–727, 2002.

[9] E. D. Popova. Parametric Interval Linear Solver. *Numerical Algorithms*, 37(1–4):345–356, 2004.

[10] E. D. Popova. Generalizing the Parametric Fixed-Point Iteration. *Proceedings in Applied Mathematics & Mechanics* (PAMM) 4(1):680–681, 2004.

[11] E. D. Popova. Improved Solution Enclosures for Over- and Underdetermined Interval Linear Systems. In I. Lirkov, S. Margenov, J. Wasniewski (Eds). Proceedings of LSSC 2005, *Lecture Notes in Computer Science* 3743, 2006, 305–312.

[12] E. D. Popova. Solving Linear Systems whose Input Data are Rational Functions of Interval Parameters. Preprint 3/2005, Institute of Mathematics and Informatics, BAS, Sofia, 2005. (http://www.math.bas.bg/˜epopova/papers/05PreprintEP.pdf)

[13] E. D. Popova, R. Iankov, Z. Bonev. Bounding the Response of Mechanical Structures with Uncertainties in All the Parameters. In: R. Muhannah, R. Mullen (Eds). Proc. of the NSF Workshop on Reliable Engineering Computing, Svannah, Georgia, USA, Feb. 22-24, 2006, 245–265.

[14] J. Rohn. Linear Interval Equations with Dependent Coefficients. Symposium "Interval Methods for Numerical Computations", Oberwolfach, 1990. In: J. Rohn. A Method for Handling Dependent Data in Interval Linear Systems. Technical Report No. 911, Institute of Computer Science, Academy of Science of the Czech Republic, July 2004.

[15] S. Rump. Solving Algebraic Problems with High Accuracy. In: U. Kulisch and W. Miranker (Eds). *A New Approach in Scientific Computation*. Academic Press, 1983, 51–120.

[16] S. Rump. Verification methods for dense and sparse systems of equations. In: J. Herzberger (Ed). *Topics in Validated Computations*. Elsevier Science B. V., 1994, 63–135.

[17] S. Rump. Computer-assisted Proofs and Self-validating Methods. In: Bo Einarsson (Ed). *Accuracy and Reliability in Scientific Computing*. SIAM, 2005, 195–240.

[18] S. Wolfram. *The Mathematica Book*. Wolfram Media/Cambridge U. Press, 4th ed., 1999.