# Eigenvalues of Uncertain Symmetric Matrices

Jeroen De Vlieger  Karl Meerbergen  Department of Computer Science, K.U.Leuven

## Motivation

Mass-Stifness models

$$Kv = \lambda Mv$$

can be analysed using Frequency Respons Functions (FRF) or eigenvalue calculation.

The stiffness matrix $K$ and/or the mass matrix $M$ may however be subject to an 'uncertain' parameter $x$.

$$K(x)v(x) = \lambda(x)M(x)v(x)$$

In such a case one is interested in evaluating the effect of this uncertainty on the model. (i.e. calculating uncertain FRF's and uncertain eigenvalues)

## Uncertainty

There are 2 distict approaches to handle uncertainty:

- In the probabilistic case one models $x$ as a pdf and evaluates the probability of certain events.
- In the possibilistic case one is interested in the impact/sensitivity of the parameter $x$ on the system.

  To this end, several perturbation analysis techiniques are possible:
  - interval arithmetic
  - fuzzy eigenvalues
  - speudo spectral theory

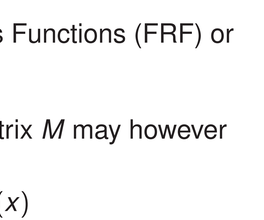
## Fuzzy number

A convex fuzzy number $\widetilde{x} \in \mathbb{F}_c$ is defined by its membership function

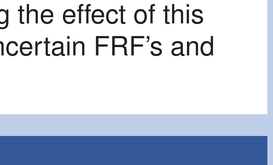
$$\mu_{\widetilde{x}}(x) : \mathbb{R} \rightarrow [0, 1]$$

Where an $\alpha$-cuts is the interval

$$[\widetilde{x}]_\alpha = \{x \mid \mu_{\widetilde{x}}(x) \geq \alpha\}$$



The $\alpha$-cut of a fuzzy vector $\widetilde{\boldsymbol{x}}$ of $N$ independent fuzzy numbers is a $N$-dimensional hypercube.



A fuzzy system f($\widetilde{\boldsymbol{x}}$) can than be evaluated using the extension principle:

$$[\widetilde{y}]_\alpha^- = \min_{\boldsymbol{x} \in [\widetilde{\boldsymbol{x}}]_\alpha} f(\boldsymbol{x})$$

$$[\widetilde{y}]_\alpha^+ = \max_{\boldsymbol{x} \in [\widetilde{\boldsymbol{x}}]_\alpha} f(\boldsymbol{x})$$

## Paremeter sensitivity of the largest eigenvalue

Given a symmetric linear matrix function
(for example a model with uncertain stiffness)

$$A + \sum_{i=1}^{n} x_i E_i$$

Then the largest eigenvalue $\lambda_1$ is a functions of $\boldsymbol{x}$.

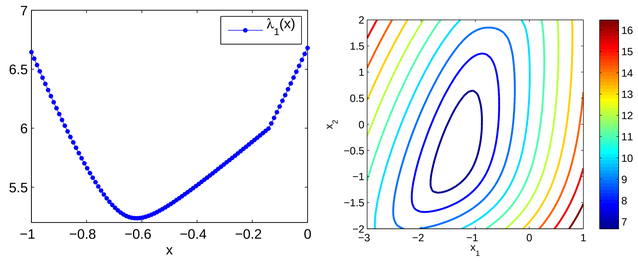



Figure: $\lambda_1(\boldsymbol{x})$ for a one and a two variate linear matrix function.

## Properties of $\lambda_1(\boldsymbol{x})$

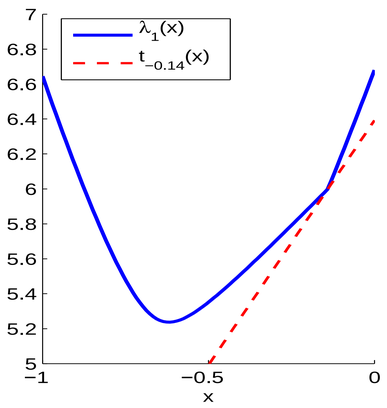
- Eigenvalues are the solution of the characteristic polynomial, thus the eigenvalue functions of a parameterized matrix are continuous, though not necessarily differentiable.

$$\lambda(\boldsymbol{x}) \in C^0$$

- Given an eigenvector $\boldsymbol{v}^*$ associated with $\lambda_1(\boldsymbol{x}^*)$, then the linear function

$$t_{\boldsymbol{x}^*} = \lambda_1(\boldsymbol{x}^*) + \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{x}_i^*) \frac{\boldsymbol{v}^{*T} E_i \boldsymbol{v}^*}{\boldsymbol{v}^{*T} \boldsymbol{v}^*} \quad \leq \quad \lambda_1(\boldsymbol{x})$$

is both a tangent line in this point $\boldsymbol{x}^*$ and a lower bound.




The largest eigenvalue function $\lambda_1(\boldsymbol{x})$ is thus a convex function.

## Calculation routine

Sampling of the system is an expensive but robust method to evaluate the extension principle.

It can however be done more cheaply by leveraging the problem specific knowledge that $\lambda_1(\boldsymbol{x})$ is a continuous, non-differential, convex function

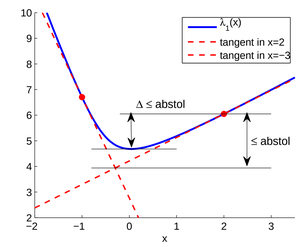
- $\left[\widetilde{y}\right]_\alpha^-$ is always located in one of the $2^N$ corner-points of the input $\alpha$-cut hypercube $[\widetilde{\boldsymbol{x}}]_\alpha$
- $\left[\widetilde{y}\right]_\alpha^+$ can be found using a *local* optimization routine.

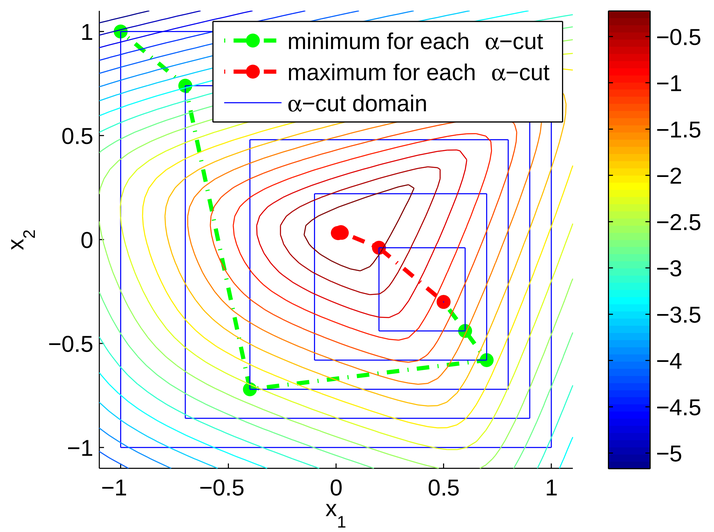
## Optimalization subproblem

One or more of these tangent lines define a piecewise linear lower-bound that can be solved using an LP-routine

$$[\overline{\lambda}_1, \overline{\boldsymbol{x}}] = \min \quad \lambda_1$$
$$st: \ \forall \boldsymbol{x}^i \ : \ \lambda_1 \geq t_{\boldsymbol{x}^i}(\boldsymbol{x})$$
$$\boldsymbol{x} \in [\widetilde{\boldsymbol{x}}]_\alpha$$

- $\overline{\boldsymbol{x}}$ can be used to construct a descending search direction.
- $\overline{\lambda}_1$ can be used to derive an upper bound for the error.




NATW - Department of Computer Science - Katholieke Universiteit Leuven - Leuven, Belgium

Mail: {jeroen.devlieger, karl.meerbergen}@cs.kuleuven.be