Power flow feasibility problem

Anatoly Dymarsky, Elena Gryazina, Boris Polyak, Sergei Volodin, Yury Maximov

Skolkovo Insitute of Science and Technology

The problem

1. Large-scale power grids
2. Need to know if a regime of the grid is «normal», «safe»
3. Ohm's law $\Rightarrow$ quadratic equations:

$$y_i = f_i(x) = x^T A_i x + 2b_i^T x$$

   y (regime) known, x is not
4. Need to determine if $\exists x\colon y = f(x)$ (means «safe»)

This problem is known as Power Flow Feasibility problem.
To solve it, the image $f(\mathbb{R}^n)$ must be examined

State of the art

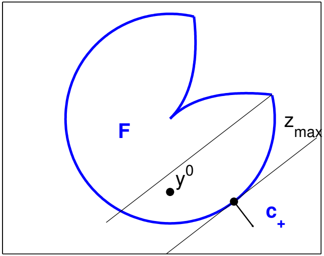
1. A. Dymarsky, Convexity of a Small Ball Under Quadratic Map, arXiv:1410.1553
2. A. Dymarsky, On the Convexity of Image of a Multidimensional Quadratic Map, arXiv:1410.2254

Given: the map $f : \mathbb{R}^n \to \mathbb{R}^m$, $f_i(x) = x^T A_i x + 2b_i^T x$, $A_i^T = A_i$

Proposed algorithm for examining $F = f(\mathbb{R}^n)$:

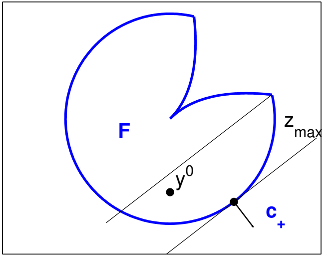
- Input: $y^0 \in F$, direction $c_+ : c_+ \cdot A > 0$
- Output: value $z_{\max}$ s.t. the cut $Q(c_+, z_{\max}, F)$ is convex

$$Q(c_+, z, F) = \{y \big| (y - y^*, c_+) \in [0, z]\} \cap F$$



The solution

1. Discovering boundary nonconvexities $\{F_i\}$ close to y^0
2. Projecting to c_+: (F_i, c_+)
3. Calculating $z_{\max} = \min\limits_{y \in F_i}(c_+, y)$



Boundary points of F on supporting hyperplane c:
$\partial F_c = \arg\min_{x \in \mathbb{R}^n} (c, f(x)) = \operatorname{Ker}(c \cdot A) - (c \cdot A)^g (c \cdot b)$ if

$$\begin{cases} c \cdot A \geqslant 0 \\ (c \cdot b)^T \operatorname{Ker}(c \cdot A) = 0 \end{cases}$$

or $\partial F_c = \emptyset$

- ∂F_c is nonconvex $\Rightarrow$ F has nonconvexity
- ∂F_c is nonconvex $\leftrightsquigarrow$ $\operatorname{Rg}(c \cdot A) < n$, $c \cdot A \geqslant 0$

Therefore, $z_{\max} = \inf_{c} \inf_{y \in \partial F_c} (c_+, \partial F_c)$
We assume $\operatorname{Rg}(c \cdot A) < n - 1$ *to be a rare case*

The solution

Linear change of basis s.t. $\begin{cases} c_+ \cdot A = I \\ c_+ \cdot b = 0 \end{cases} \Rightarrow$

- $\inf\limits_{y \in \partial F_c} (c_+, \partial F_c) = \|(c \cdot A)^g (c \cdot b)\|^2$ for c: $\mathrm{Rg}(c \cdot A) = n - 1$
- Define $z(c) = \|(c \cdot A - \lambda_{\mathsf{min}}(c \cdot A))^g (c \cdot b)\|^2$
- Define $c_{\mathrm{bad}} = \{c \big| \mathrm{Ker}(c \cdot A) \perp (c \cdot b)\}$
- Then $\boxed{z_{\mathsf{max}} = \inf\limits_{c \in c_{\mathrm{bad}}} z(c)}$.

In general case $|c_{\mathrm{bad}}|$ is continuum

The solution

We use gradient projection method to find

$$z_{\text{max}} = \inf_{c \in c_{\mathrm{bad}}} z(c)$$

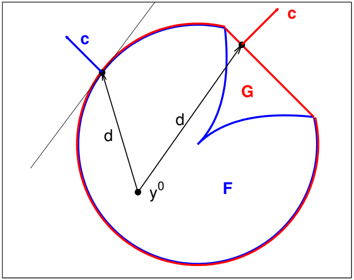
1. Start point: obtained via *nonconvexity certificate*
2. Gradient $\frac{dz}{dc}$ is calculated explicitly
3. Normal vector n for c_{bad}
4. Projection onto c_{bad} is done by adjusting λ in $c' + \lambda n$

The solution

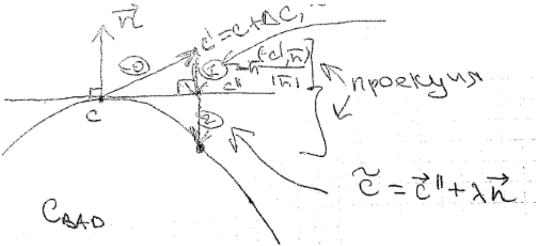
Nonconvexity certificate:

1. Input: y^0
2. Generating random directions d
3. Find $t: y^0 + td \in \partial \operatorname{conv} F$
4. $y^0 + td \in F$?
5. If not, obtain c via dual problem
6. Output: «nonconvex» c

$\Rightarrow$ Obtained start point for gradient descent



The solution



Gradient step + projection

1. Input: current point $c = c^k \in c_{\rm bad}$
2. Calculate $n(c)$, $\frac{dz}{dc}$, $c' = c + (\hat{1} - \frac{(\cdot, n)}{(n,n)} n) \frac{dz}{dc}$
3. Project $c' \notin c_{\rm bad}$ onto $c_{\rm bad}$
4. $c^{k+1} \leftarrow$ result of projection

The solution

Projection

Given: point $c' \notin c_{\mathrm{bad}}$, normal vector n

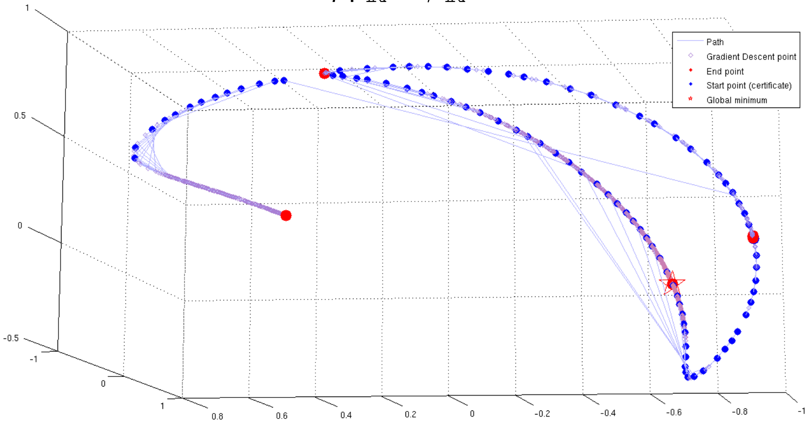
Find: point $c \in c_{\mathrm{bad}}$ close to c

1. $c(\lambda) = c' + \lambda n$
2. For some λ, $c(\lambda) \in c_{\mathrm{bad}}$
3. $c(\lambda) \in c_{\mathrm{bad}} \Leftrightarrow \mathrm{Ker}(c(\lambda) \cdot A) \perp c(\lambda) \cdot b$
4. $m(\lambda) = (c(\lambda) \cdot b)^T x_0(\lambda)$, $x_0(\lambda) \in \mathrm{Ker}(c(\lambda) \cdot A)$

Projection is done by finding root of $m(\lambda)$

Numerical experiment

$$f : \mathbb{R}^4 \to \mathbb{R}^4$$



Results

1. Algorithm for examining F was proposed
2. Cases when $|c_{\text{bad}}| = \infty$ and $|c_{\text{bad}}| < \infty$ were considered
3. Algorithm for $|c_{\text{bad}}| = \infty$ was implemented
4. Algorithm was tested on a number of maps f

Thank you!
Questions?