Manual of NSOCCP_ReSNA

— ReSNA Plugin for Nonlinear Second-Order Cone Complementarity Problems —

Shunsuke Hayashi*

September 3, 2013

1 Problem

NSOCCP_ReSNA.m solves (tries to solve) the Nonlinear Second-Order Cone Complementarity Problem (NSOCCP) expressed as follows:

$$\begin{array}{rl} \text{Find} & (x,y) \in \mathbb{R}^n \times \mathbb{R}^n \\ \text{such that} & x \in \mathcal{K},\ y \in \mathcal{K},\ x^\top y = 0, \\ & y = F(x), \end{array} \tag{1.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a given continuously differentiable function, and $\mathcal{K}$ is a Cartesian product of several second-order cones (SOCs), i.e.,

$$\mathcal{K} := \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_m}$$

with $n_1 + n_2 + \cdots + n_m = n$ and

$$\mathcal{K}^{n_i} := \begin{cases} \left\{ z \in \mathbb{R} \;\middle|\; z \geq 0 \right\} & (n_i = 1) \\ \left\{ z \in \mathbb{R}^{n_i} \;\middle|\; z_1 \geq \sqrt{z_2^2 + \cdots + z_{n_i}^2} \right\} & (n_i \geq 2). \end{cases}$$

2 How to use the plugin

Putting ReSNA.m in the same folder, you can use NSOCCP_ReSNA.m as follows.

Usage 1: `[x,y] = NSOCCP_ReSNA(FUNC,nabFUNC,K)`
Usage 2: `[x,y] = NSOCCP_ReSNA(FUNC,nabFUNC,K,x0,y0)`

- FUNC — implies the function $F : \mathbb{R}^n \to \mathbb{R}^n$ in problem (1.1). If function m-file F.m plays a role of function F, then put F.m in the same folder and let FUNC = @F. ("at mark" is required before the name of function m-file.)

- nabFUNC — implies $\nabla F : \mathbb{R}^n \to \mathbb{R}^{n \times n}$, i.e., the transposed Jacobian of function F. If function m-file nabF.m plays a role of function ∇F, then put nabF.m in the same folder and let nabFUNC = @nabF. If you do not have the closed form of $\nabla F(x)$, let nabFUNC = []. In this case, $\nabla F(x)$ is approximated by means of the finite difference method.

- K — implies the Cartesian structure of $\mathcal{K}$ in problem (1.1). K should be given as the row or column vector whose component corresponds to the dimension of each second-order cones. For example, when $\mathcal{K} = \mathcal{K}^3 \times \mathcal{K}^1 \times \mathcal{K}^2$, let K = [3,1,2]. When $\mathcal{K} = \mathbb{R}^{10}_+$, let K = ones(1,10).

*Graduate School of Information Sciences (GSIS), Tohoku University (s_hayashi@plan.civil.tohoku.ac.jp)

- `x0` — implies the initial point $x^{(0)}$ for the regularized smoothing Newton algorithm (Algorithm 4.1 in `manual_ReSNA.pdf`). `x0` should be given as a column vector whose length is equal to `sum(K)`. If you omit `x0` or let `x0 = []`, then ReSNA chooses a random vector from $[-1, 1]^n$ automatically.

- `y0` — implies the initial point $y^{(0)}$, which can be omitted similarly to `x0`.

Parameters in `ReSNA.m`

- `PROGRESS` — decides whether or not ReSNA displays the detailed progress of the iteration. The default value is `'Y'`.

- `tole` — is used for the termination criterion in Step 1 (Algorithm 4.1 in `manual_ReSNA.pdf`). When $\|H_{\rm NR}(w^{(k)})\| \leq$ `tole`, the algorithm terminates normally and the obtained output is guaranteed to be the solution of problem (1.1). The default value is `1e-8`.

- `tole_diff` — is used for approximating the Jacobian matrix by means of the finite difference method. The default value is `1e-8`.

- `eta, eta_bar, rho, sigma, kappa, kappa_bar, kappa_hat` — are the parameters indicated in Algorithm 4.1 in `manual_ReSNA.pdf`. Some default values are assigned automatically.