Manual of MNCP_ReSNA

— ReSNA Plugin for Mixed Nonlinear Complementarity Problems —

Shunsuke Hayashi*

September 3, 2013

1 Problem

`MNCP_ReSNA.m` solves (tries to solve) the Mixed Nonlinear Complementarity Problem (MNCP) expressed as follows:

$$\begin{array}{rl} \text{Find} & (x, y, p) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^l \\ \text{such that} & x \geq 0,\ y \geq 0,\ x^\top y = 0, \\ & y = F_1(x, p),\ \ F_2(x, p) = 0, \end{array} \tag{1.1}$$

where $F_1 : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^n$ and $F_2 : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^l$ are given continuously differentiable functions.

2 How to use the plugin

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

Usage 1: `[x,y,p] = MNCP_ReSNA(FUNC,nabFUNC,n,el)`
Usage 2: `[x,y,p] = MNCP_ReSNA(FUNC,nabFUNC,n,el,x0,y0,p0)`

- `FUNC` — implies the function $F : \mathbb{R}^{n+l} \to \mathbb{R}^{n+l}$ such that

  $$F(z) = \begin{pmatrix} F_1(x, p) \\ F_2(x, p) \end{pmatrix} \text{ with } z = \begin{pmatrix} x \\ p \end{pmatrix}.$$

  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 $\nabla F : \mathbb{R}^{n+l} \to \mathbb{R}^{(n+l)\times(n+l)}$, i.e., the transposed Jacobian of function F. More precisely,

  $$\nabla F(z) = \begin{pmatrix} \nabla_x F_1(x, p) & \nabla_x F_2(x, p) \\ \nabla_p F_1(x, p) & \nabla_p F_2(x, p) \end{pmatrix} \text{ with } z = \begin{pmatrix} x \\ p \end{pmatrix}.$$

  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 $\nabla F(z)$, let `nabFUNC = []`. In this case, $\nabla F(z)$ is approximated by means of the finite difference method.

- `n` — implies the value of n, i.e., the dimension of x or $F_1(x, p)$ in problem (1.1). `n` should be given as a positive integer.

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

- `el` — implies the value of l, i.e., the dimension of p or $F_2(x,p)$ in problem (1.1). `el` should be given as a nonnegative integer. If p and $F_2(x,p)$ are absent (non-mixed case), let `el = 0`.

- `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 `n`. 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`.

- `p0` — implies the initial point $p^{(0)}$ for the regularized smoothing Newton algorithm. `p0` should be given as a column vector whose length is equal to `el`. If you omit `p0` or let `p0 = []`, then ReSNA chooses a random vector from $[-1,1]^l$ automatically.

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 $\|H_{\mathrm{NR}}(w^{(k)})\| \leq$ `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.