

## Método tipo-Newton proximal para o Problema de Equilíbrio

**Pedro Jorge Sousa dos Santos**

Universidade Federal do Piauí  
Campus Ministro Reis Velloso, Parnaíba - PI  
pedrojorge@ufpi.edu.br

**Susana Scheimberg**

Universidade Federal do Rio de Janeiro  
PESC/COPPE, Rio de Janeiro - RJ  
susana@cos.ufrj.br

**Paulo Sérgio Marques dos Santos**

Universidade Federal do Piauí  
Campus Ministro Reis Velloso, Parnaíba - PI  
psergio@ufpi.edu.br

### RESUMO

Neste trabalho apresentamos um algoritmo tipo-Newton proximal local não-suave para resolver o Problema de Equilíbrio que unifica problemas conhecidos, tais como: problemas de otimização, de desigualdade variacional e de equilíbrio de Nash. Provamos que o método é localmente convergente e estabelecemos a convergência superlinear do algoritmo para uma solução do problema sob hipóteses usuais. Em particular, consideramos a condição de qualificação de posto constante que é mais fraca que a condição de qualificação de independência linear. Alguns resultados numéricos são apresentados para ilustrar a performance do algoritmo. Além disso, nós comparamos nossos resultados numéricos com aqueles apresentados em [Santos e Scheimberg, 2011], [von Heusinger et al., 2012] e [Nasri et al., 2016] para os mesmos problemas teste.

**PALAVRAS CHAVE.** Problema de Equilíbrio, Método de Newton, Condição de Qualificação Posto Constante.

**Tópico (Programação Matemática)**

### ABSTRACT

In this work we present a nonsmooth local proximal Newton-type algorithm for solving the Equilibrium Problem that unifies known problems such as: optimization problems, variational inequalities and Nash games. We prove that the method is locally convergent and we establish the superlinear convergence of the algorithm to a solution of the problem under suitable assumptions. In particular, we consider the constant rank constraint qualification which is weaker than the linear independence constraint qualification. Some numerical results are presented to illustrate the performance of the algorithm. Moreover, we compare our numerical results with those presented in [Santos e Scheimberg, 2011], [von Heusinger et al., 2012] and [Nasri et al., 2016] for the same test problems.

**KEYWORDS.** Equilibrium Problem, Newton's Method, Constant Rank Constraint Qualification.

**Topic (Mathematical Programming)**

## 1. Introdução

Seja  $X \subset \mathbb{R}^n$  convexo, fechado e não vazio e  $f : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  uma bifunção de equilíbrio, i.e.,  $f(x, x) = 0$  para todo  $x \in \mathbb{R}^n$ . Consideraremos o seguinte problema de equilíbrio, denotado por  $EP(X, f)$ :

$$EP(X, f) := \begin{cases} \text{Encontrar } x^* \in X \text{ tal que} \\ f(x^*, y) \geq 0 \quad \forall y \in X. \end{cases}$$

Neste trabalho apresentaremos um método de Newton para resolver  $EP(X, f)$  baseado no trabalho de [von Heusinger et al., 2012] para encontrar um equilíbrio normalizado para o problema de equilíbrio de Nash generalizado. Ao longo deste trabalho assumiremos a seguinte hipótese:

### Hipótese 1.1.

(a) O conjunto  $X \subset \mathbb{R}^n$  possui a seguinte representação

$$X = \{x \in \mathbb{R}^n ; g(x) \leq 0\} \quad (1)$$

onde  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  é uma função duas vezes continuamente diferenciável e convexa.

(b) A bifunção  $f : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  é duas vezes continuamente diferenciável e, para cada  $x$  fixo, a função  $f(x, \cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$  é convexa.

A condição (a) da Hipótese 1.1 é clássica para problemas em que se deseja aplicar algum método tipo Newton.

A condição (b) da Hipótese 1.1 é uma generalização das condições solicitadas às funções ou operadores de problemas que podem ser reformulados como um problema de equilíbrio. Por exemplo:

Quando o problema de equilíbrio corresponde a um problema de otimização convexa, a bifunção  $f$  é definida por

$$f(x, y) = \varphi(y) - \varphi(x)$$

onde  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$  é a função convexa a ser minimizada sobre o conjunto  $X$  definido em (1). Neste caso, a condição (b) da Hipótese 1.1 se reduz a assumir que a função  $\varphi$  seja duas vezes continuamente diferenciável. [Blum e Oettli, 1994], [Bertsekas, 1996].

Quando o  $EP(X, f)$  corresponde a uma desigualdade variacional, isto é,

$$f(x, y) = \langle \Phi(x), y - x \rangle$$

onde  $\langle \cdot, \cdot \rangle$  corresponde à notação do produto interno em  $\mathbb{R}^n$ ,  $\langle x, y \rangle = x^T y$ , e  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  é o operador que define a desigualdade variacional, juntamente com o conjunto  $X$  das restrições dado por (1), verificando a condição (a) da Hipótese 1.1. A condição (b) desta hipótese se reduz a requerer que  $\Phi$  seja um operador contínuo e seu jacobiano também. Ver [Facchinei e Pang, 2003].

Por sua vez, quando o problema de equilíbrio corresponde ao problema de Nash generalizado, com  $N$  jogadores  $v = 1, \dots, N$  e restrições compartilhadas, temos que

$$f(x, y) = \sum_{v=1}^N [\theta_v(y^v, x^{-v}) - \theta_v(x^v, x^{-v})]$$

onde  $n := n_1 + \dots + n_N$ ,  $x = (x^1, \dots, x^N) \in \mathbb{R}^n$ ,  $x^v \in \mathbb{R}^{n_v}$  e  $\theta_v : \mathbb{R}^n \rightarrow \mathbb{R}$  é a função custo do  $v$ -ésimo jogador que deve ser minimizada sujeito a  $(x^v, x^{-v}) \in X$ , com  $x^{-v} = (x^1, \dots, x^{v-1}, x^{v+1}, \dots, x^N)$  fixado, onde  $X$  é o conjunto comum de estratégias dado por (1). Neste caso, a Hipótese 1.1 (b) se reduz a pedir que a função custo  $\theta_v$ ,  $v = 1, \dots, N$ , seja duas vezes continuamente diferenciável e, como função de  $x^v$  apenas, seja convexa. Ver [von Heusinger et al., 2012].

Note que  $x^* \in X$  resolve EP(X,f) se, e somente se,  $x^*$  resolve o seguinte problema de programação convexa

$$\begin{aligned} \min_y \quad & f(x^*, y) \\ \text{s.a.} \quad & g(y) \leq 0. \end{aligned} \quad (2)$$

E, sob alguma condição de qualificação das restrições,  $x^*$  resolve (2) se, e somente se, satisfaz o seguinte sistema KKT do problema (2):

$$\begin{aligned} \nabla_y f(x, x) + \nabla g(x)\lambda &= 0 \\ \lambda \geq 0, \quad g(x) &\leq 0, \quad \langle \lambda, g(x) \rangle = 0. \end{aligned} \quad (3)$$

Os problemas (2) e (3) são equivalentes, por exemplo, se o conjunto  $X$  possui um ponto interior, isto é, se existe  $\hat{x} \in \mathbb{R}^n$  tal que  $g(\hat{x}) < 0$  [Jacinto e Scheimberg, 2008].

Temos, portanto, que sob a Hipótese 1.1 e alguma condição de qualificação das restrições como a mencionada acima,  $x^*$  resolve EP(X,f) se, e somente se, satisfaz (3).

Adotaremos a seguinte notação: O gradiente  $\nabla\varphi(x)$  de uma função diferenciável  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$  será sempre um vetor coluna. Além disso, denotaremos por  $\nabla\Phi(x)$  o jacobiano transposto da função diferenciável  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Para uma bifunção  $\varphi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  a derivada parcial com respeito à primeira e à segunda variável será denotada por  $\nabla_x\varphi(x, y) \in \mathbb{R}^n$  e  $\nabla_y\varphi(x, y) \in \mathbb{R}^m$ , respectivamente. A derivada parcial de segunda ordem com respeito primeiramente à segunda variável e depois com respeito à primeira variável será denotada por  $\nabla_{yx}^2\varphi(x, y) \in \mathbb{R}^{m \times n}$ .

## 2. Preliminares

Nesta seção estabeleceremos definições e resultados que serão utilizados nas próximas seções. Iniciamos com a definição de Jacobiano generalizado computável de uma função continuamente diferenciável por partes.

**Definição 2.1.** Uma função  $\varphi$  é continuamente diferenciável por partes numa vizinhança de um ponto  $x \in \mathbb{R}^n$ , se  $\varphi$  é contínua e existe uma vizinhança  $V(x)$  de  $x$  e um número finito de funções continuamente diferenciáveis  $\varphi_i : V(x) \rightarrow \mathbb{R}^n$ ,  $i = 1, \dots, l$ , tal que  $\varphi(y) \in \{\varphi_1(y), \dots, \varphi_l(y)\}$  para todo  $y \in V(x)$ .

**Definição 2.2.** Considere uma função  $\varphi$  continuamente diferenciável por partes numa vizinhança  $V(x)$  de um ponto  $x \in \mathbb{R}^n$  e seja  $\varphi_i : V(x) \rightarrow \mathbb{R}^n$ ,  $i = 1, \dots, l$ , tal que  $\varphi(y) \in \{\varphi_1(y), \dots, \varphi_l(y)\}$  para todo  $y \in V(x)$ . O Jacobiano generalizado computável de  $\varphi$  em  $x$  é dado por:

$$\partial_C\varphi(x) = \{\nabla\varphi_i(x); i = 1, \dots, l\}.$$

Considere a bifunção regularizada  $\Psi : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  dada por

$$\Psi(x, y) := f(x, y) + h(x, y)$$

onde  $h : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  satisfaz a seguinte hipótese

### Hipótese 2.1.

- (a)  $h : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  é duas vezes continuamente diferenciável;
- (b)  $h(x, \cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$  é fortemente convexa para todo  $x \in \mathbb{R}^n$ ;
- (c)  $\nabla_y h(x, x) = 0$  para todo  $x \in \mathbb{R}^n$ .

Note que nossa Hipótese 2.1 é semelhante à correspondente hipótese imposta sobre  $h$  em [Bigi e Passacantando, 2016] onde no entanto, embora  $h$  seja apenas continuamente diferenciável, se pede que  $h(x, y) \geq 0$  e  $\langle \nabla_x h(x, y) + \nabla_y h(x, y), y - x \rangle \geq 0 \forall x, y \in \mathbb{R}^n$  e  $h(z, z) = 0 \forall z \in \mathbb{R}^n$ .

A Hipótese 1.1 (b) juntamente com a Hipótese 2.1 (b) garantem que a função  $\Psi(x, \cdot)$  é fortemente convexa para cada  $x \in \mathbb{R}^n$  fixado. Isto, juntamente com a Hipótese 1.1 (a), implica que o problema de otimização

$$\begin{aligned} \min_y \quad & \Psi(x, y) \\ \text{s.a.} \quad & g(y) \leq 0. \end{aligned} \quad (4)$$

possui solução única (que depende de  $x$ ), a qual denotaremos por  $y(x)$ .

O sistema KKT associado ao problema (4) vem dado por

$$\begin{aligned} \nabla_y \Psi(x, y) + \nabla g(y)\lambda &= 0 \\ \lambda \geq 0, \quad g(y) \leq 0, \quad \langle \lambda, g(y) \rangle &= 0. \end{aligned} \quad (5)$$

E, sob alguma condição de qualificação das restrições,  $y(x)$  resolve o problema (4) se, e somente se, existe  $\lambda \in \mathbb{R}^m$  tal que  $(y(x), \lambda)$  resolve o sistema KKT (5).

**Proposição 2.1.** *Suponha válido uma certa condição de qualificação das restrições para o conjunto  $X$ . Temos:*

(a) *A função  $y : \mathbb{R}^n \rightarrow X \subset \mathbb{R}^n$  é contínua;*

(b) *Um vetor  $x \in X$  resolve  $EP(X, f)$  se, e somente se,  $x = y(x)$ .*

*Demonstração.* (a) Segue dos corolários 8.1 e 9.1 de [Hogan, 1973].

(b) Suponha que  $x$  resolve  $EP(X, f)$ . Então existe  $\lambda \in \mathbb{R}^m$  tal que  $(x, \lambda)$  satisfaz (3). Logo, pela Hipótese 2.1(c),  $(x, y, \lambda) = (x, x, \lambda)$  satisfaz (5). Isto significa que  $y(x) = x$ , uma vez que o vetor  $y$  satisfazendo (5) é único para cada  $x$  fixado.

Suponha, por outro lado, que  $x = y(x)$ . Então existe  $\lambda \in \mathbb{R}^m$  tal que  $(x, x, \lambda)$  satisfaz (5) o que implica que  $(x, \lambda)$  resolve (3), isto é,  $x$  é solução de  $EP(X, f)$ .  $\square$

Denotando  $I_m = \{1, \dots, m\}$ , seja  $I_0(x) := \{i \in I_m ; g_i(y(x)) = 0\}$  o conjunto de índices ativos em  $y = y(x)$ .

**Hipótese 2.2.** *Fixado  $x \in \mathbb{R}^n$ , a condição de qualificação posto constante (CRCQ) vale em  $y(x)$ , isto é, existe uma vizinhança  $N(y(x))$  de  $y(x)$  tal que para cada  $J \subseteq I_0(x)$ , o conjunto*

$$\{\nabla g_i(y) ; i \in J\}$$

*tem o mesmo posto (que depende de  $J$ ) para todo  $y \in N(y(x))$ .*

Vale ressaltar que a condição de qualificação de independência linear (LICQ) é mais forte que (CRCQ). Veja, por exemplo, [Janin, 1984]. Considere o conjunto dos multiplicadores de lagrange:

$$M(x) = \{\lambda \in \mathbb{R}^m ; (y(x), \lambda) \text{ satisfaz (5)}\}. \quad (6)$$

Sob a Hipótese 2.2,  $M(x)$  é sempre não vazio. Considere a seguinte família de subconjuntos do conjunto de índices ativos  $I_0(x)$  dada por

$$\begin{aligned} B(x) &= \{J \subseteq I_0(x) ; (\nabla g_i(y(x)))_{i \in J} \text{ é linearmente independente e} \\ &\quad \text{supp}(\lambda) \subseteq J \text{ para algum } \lambda \in M(x)\} \end{aligned} \quad (7)$$

onde, dado  $\lambda \in \mathbb{R}^m$ ,  $\text{supp}(\lambda) := \{i \in I_m ; \lambda_i > 0\}$ .

Com o objetivo de desenvolver um método do tipo Newton para  $EP(X, f)$ , considere a função  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  dada por  $F(x) := y(x) - x$ . O próximo lema está relacionado ao Jacobiano generalizado computável de  $F$ . Para mais detalhes sobre o referido Jacobiano e a demonstração do lema, veja [von Heusinger et al., 2012] (lemas 3.3, 3.4, 3.5 e 5.1).

**Lema 2.1.** (a)  $B(x) \neq \emptyset$  sempre que  $M(x) \neq \emptyset$ ;  
(b) Seja  $x \in \mathbb{R}^n$ , e suponha que a Hipótese 2.2 vale em  $y(x)$  então, fixado  $J \in B(x)$ , existe um único  $\lambda = \lambda^J(x) \in M(x)$  tal que  $g_J(y(x)) = 0$  e  $\lambda_j = 0$ , onde  $\hat{J} = \{1, \dots, m\} \setminus J$  e a partição  $(J, \hat{J})$  é usada para separar os vetores  $\lambda$  e  $g(\cdot)$  em  $\lambda = (\lambda_J, \lambda_{\hat{J}})$  e  $g(\cdot) = (g_J(\cdot), g_{\hat{J}}(\cdot))$ , respectivamente;  
(c) Seja  $\bar{x} \in \mathbb{R}^n$  dado, e assuma que a Hipótese 2.2 vale em  $\bar{y} = y(\bar{x})$ . Então existe uma vizinhança  $V(\bar{x})$  de  $\bar{x}$  tal que, para todo  $x \in V(\bar{x})$ , a Hipótese 2.2 vale em  $y(x)$  e  $B(x) \subseteq B(\bar{x})$ ;  
(d) Seja  $\bar{x} \in \mathbb{R}^n$  dado, e assuma que a Hipótese 2.2 vale em  $\bar{y} = y(\bar{x})$ . Então a função  $F(x) = y(x) - x$  é continuamente diferenciável por partes numa vizinhança de  $\bar{x}$ , e o Jacobiano generalizado computável de  $F$  em  $\bar{x}$  é dado por

$$\partial_C F(\bar{x}) = \{\nabla y^J(\bar{x})^T - I; J \in B(\bar{x})\}, \quad (8)$$

onde

$$\nabla y^J(\bar{x}) = A^T C^{-1} - A^T C^{-1} D (D^T C^{-1} D)^{-1} D^T C^{-1}, \quad (9)$$

com

$$\begin{aligned} A &= A(\bar{x}) = -\nabla_{yx}^2 \Psi(\bar{x}, \bar{y}), \\ C &= C^J(\bar{x}) = \nabla_{yy}^2 \Psi(\bar{x}, \bar{y}) + \sum_{i \in J} \lambda_i^J(\bar{x}) \nabla^2 g_i(\bar{y}), \\ D &= D^J(\bar{x}) = \nabla g_J(\bar{y}). \end{aligned} \quad (10)$$

onde  $\nabla g_J(\cdot)$  é a notação para a matriz com vetores colunas  $\{\nabla g_i(\cdot)\}_{i \in J}$  e  $I \in \mathbb{R}^{n \times n}$  denota a matriz identidade.

(e) Dado  $\bar{x} \in \mathbb{R}^n$ , suponha que alguma condição de qualificação das restrições vale em  $\bar{y} = y(\bar{x})$ . Seja  $\bar{\lambda}$  um vértice solução do seguinte problema de programação linear:

$$\begin{aligned} \min_{\lambda} \quad & \sum_{i \in I_0(\bar{x})} \lambda_i \\ \text{s.a.} \quad & \nabla g(\bar{y}) \lambda = -\nabla_y \Psi(\bar{x}, \bar{y}), \\ & \lambda_i \geq 0 \quad \forall i \in I_0(\bar{x}), \\ & \lambda_i = 0 \quad \forall i \in \{1, \dots, m\} \setminus I_0(\bar{x}), \end{aligned} \quad (11)$$

e defina  $J := \{i \in I_0(\bar{x}); \bar{\lambda}_i > 0\}$ . Então  $J$  pertence ao conjunto  $B(\bar{x})$  (e  $\bar{\lambda}$  é o  $\lambda$  único mencionado no item (b) acima).

Para estabelecer a convergência do método de Newton desenvolvido na próxima seção, precisaremos da seguinte hipótese.

**Hipótese 2.3.** Para cada  $J \in B(x)$  e  $\lambda \in M(x)$ , temos

$$d^T \left( M(x, y(x)) + \sum_{i \in J} \lambda_i \nabla^2 g_i(y(x)) \right) d \neq 0 \quad \forall d \in \mathcal{T}^J(x), d \neq 0$$

onde  $M(x, y) := \nabla_{yy}^2 \Psi(x, y) + \nabla_{yx}^2 \Psi(x, y)$  e  $\mathcal{T}^J(x) := \{d \in \mathbb{R}^n; \nabla g_i(y(x))^T d = 0 \forall i \in J\}$ .

**Lema 2.2.** Seja  $\bar{x} \in \mathbb{R}^n$  e  $\bar{y} = y(\bar{x})$ . Suponha válidas as Hipóteses 2.2 e 2.3 em  $\bar{y}$  e  $\bar{x}$ , respectivamente. Então a matriz  $\nabla y^J(\bar{x}) - I$  é não singular para qualquer  $J \in B(\bar{x})$ .

*Demonstração.* Observe, de (10), que  $C - A = M(\bar{x}, \bar{y}) + \sum_{i \in J} \lambda_i^J(\bar{x}) \nabla^2 g_i(\bar{x})$  e que  $C = C^T$ . A demonstração segue de lema 4.2 de [von Heusinger et al., 2012]. (Note que o referido lema está estabelecido para  $x \in X$ , no entanto o mesmo continua válido para todo  $x \in \mathbb{R}^n$ ).  $\square$

**Lema 2.3.** *Seja  $x^*$  uma solução de  $EP(X, f)$  satisfazendo as Hipóteses 2.2 e 2.3. Então existem  $\delta > 0$  e  $M > 0$  tais que*

$$\|(H(x))^{-1}\| \leq M \quad (12)$$

para todo  $x \in B(x^*, \delta)$  e para todo  $H(x) \in \partial_C F(x)$ .

*Demonstração.* Fixe  $J \in B(x^*)$  e considere  $H^J(x) = \nabla y^J(x) - I$ . Seja  $\sigma_J > 0$  o menor valor singular de  $H^J(x^*)$ . Temos que

$$\sigma_J \|d\| \leq \|H^J(x^*)d\| \quad \forall d \in \mathbb{R}^n. \quad (13)$$

Por outro lado, sendo  $H^J(\cdot)$  contínua em torno de  $x^*$  e dado  $c_J \in (0, 1)$ , existe  $\delta_J > 0$  tal que

$$\|H^J(x) - H^J(x^*)\| \leq c_J \sigma_J \quad \forall x \in B(x^*, \delta_J). \quad (14)$$

Pelas desigualdades (13) e (14), temos

$$\begin{aligned} \|H^J(x)d\| &= \|H^J(x^*)d + (H^J(x) - H^J(x^*))d\| \\ &\geq \|H^J(x^*)d\| - \|(H^J(x) - H^J(x^*))d\| \\ &\geq \sigma_J \|d\| - c_J \sigma_J \|d\| \\ &= (1 - c_J) \sigma_J \|d\|. \end{aligned}$$

Dado  $y \in \mathbb{R}^n$  com  $\|y\| = 1$  temos, pela desigualdade anterior para  $d = (H^J(x))^{-1}y$ , que

$$1 = \|y\| = \|H^J(x)(H^J(x))^{-1}y\| \geq (1 - c_J) \sigma_J \|(H^J(x))^{-1}y\|.$$

Assim, fazendo  $M_J = \frac{1}{(1 - c_J) \sigma_J}$ , ganhamos que  $\|(H^J(x))^{-1}\| \leq M_J$  para todo  $x \in B(x^*, \delta_J)$ .

Seja  $M = \max\{M_J; J \in B(x^*)\}$  e  $\delta = \min\{\delta_J; J \in B(x^*)\}$ . Temos que  $\|(H^J(x))^{-1}\| \leq M$  para todo  $x \in B(x^*, \delta)$  e para todo  $J \in B(x^*)$ .

Pelo Lema 2.1 (c), e diminuindo  $\delta$  se necessário, temos  $B(x) \subseteq B(x^*)$  para todo  $x \in B(x^*, \delta)$ . Logo,

$$\|(H(x))^{-1}\| \leq M$$

para todo  $x \in B(x^*, \delta)$  e para todo  $H(x) \in \partial_C F(x)$ . □

### 3. O algoritmo

Utilizando os resultados e notações da seção anterior, estamos agora prontos para estabelecer o nosso algoritmo.

**Algoritmo 1.** *(Método de Newton para EP)*

(P.0) Escolha  $x^0 \in \mathbb{R}^n$  e faça  $k := 0$ .

(P.1) Resolva o problema de otimização (4) com  $x = x^k$ , encontrando  $y^k = y(x^k)$ .

Se  $y^k = x^k$ , Pare. ( $x^k$  é solução de  $EP(X, f)$ ).

(P.2) Resolva o problema linear (11) encontrando  $\lambda^k$  (um vértice solução).

Ponha  $J := \{i \in I_0(x^k); \lambda_i^k > 0\} \in B(x^k)$ ;

(P.3) Calcule  $H_k = \nabla y^J(x^k)^T - I$  de acordo com a equação (9) com  $\lambda^J(x^k) = \lambda^k$ , e encontre a solução  $d^k$  do sistema

$$H_k d = -F(x^k);$$

*Atualização do ponto:*

(P.4) Defina  $x^{k+1} = x^k + d^k$ ,  $k = k + 1$  e volte ao passo (P.1).

### 3.1. Boa definição local

Discutiremos nesta subseção a boa definição local do algoritmo tipo Newton apresentado acima.

Note que o passo (P.1) está bem definido, na realidade, para qualquer  $x^k \in \mathbb{R}^n$  uma vez que a função  $\Psi(x^k, \cdot)$  do problema de otimização (4) é fortemente convexa e  $X = \{y \in \mathbb{R}^n; g(y) \leq 0\}$  é convexo, fechado e não vazio. Por outro lado, pela Proposição 2.1 (b), se  $y^k = x^k$  então o ponto  $x^k$  resolve  $EP(X, f)$ .

Para o passo (P.2), temos a seguinte propriedade:

**Proposição 3.1.** *Seja  $x^* \in X$  uma solução do problema  $EP(X, f)$ . Suponha que a Hipótese 2.2 vale em  $y(x^*) = x^*$  então, existe uma vizinhança  $V(x^*)$  de  $x^*$  tal que o passo (P.2) está bem definido para todo  $x^k \in V(x^*)$ .*

*Demonstração.* De fato, pelo Lema 2.1 (c) existe uma vizinhança  $V(x^*)$  de  $x^*$  tal que a Hipótese 2.2 vale em  $y(x^k)$  para todo  $x^k \in V(x^*)$ . Segue portanto que  $M(x^k)$  é não vazio, logo (11) possui pelo menos um ponto viável. Por outro lado, a função objetivo em (11) é limitada inferiormente no conjunto viável. Portanto o problema de programação linear (11) possui solução (e, em particular, um vértice solução). A proposição segue agora do Lema 2.1 (e).  $\square$

**Observação 3.1.** *Note que no passo (P.2) podemos ter  $\lambda^k = 0$  implicando que  $J = \emptyset \in B(x^k)$ . Note também que  $0 \in M(\bar{x})$  se, e só se,  $\emptyset \in B(\bar{x})$  e que, para  $J = \emptyset \in B(\bar{x})$  a expressão (9) se resume a  $\nabla y^0(\bar{x}) = A^T C^{-1}$ .*

Para o passo (P.3), temos o seguinte:

**Proposição 3.2.** *Seja  $x^* \in X$  uma solução do problema  $EP(X, f)$ . Suponha que as Hipóteses 2.2 e 2.3 valem em  $y(x^*) = x^*$  então, existe uma vizinhança  $V(x^*)$  de  $x^*$  tal que as matrizes  $H_k \in \{\nabla y^J(x^k) - I; J \in B(x^k)\}$  são não singulares para todo  $x^k \in V(x^*)$  e, portanto, o passo (P.3) está bem definido (localmente).*

*Demonstração.* Ver Teorema 4.5 de [von Heusinger et al., 2012].  $\square$

Para concluir a boa definição, e também para realizarmos a análise de convergência local de nosso algoritmo, faremos uso do seguinte lema que é uma extensão para Problema de Equilíbrio do Lema 4.4 de [von Heusinger et al., 2012].

**Lema 3.1.** *Seja  $x^*$  uma solução de  $EP(X, f)$ . Suponha que a Hipótese 2.2 vale em  $y(x^*) = x^*$ . Então, para qualquer  $H \in \partial_C F(x)$ , vale*

$$F(x) - F(x^*) - H(x - x^*) = o(\|x - x^*\|)$$

Para finalizarmos a boa definição local do algoritmo precisamos garantir a existência de uma vizinhança  $V(x^*)$  de  $x^*$  tal que  $x^{k+1} \in V(x^*)$  sempre que  $x^k \in V(x^*)$ . Pelo Lema 2.3 existem  $\delta_1 > 0$  e  $M > 0$  tais que, para todo  $x \in B(x^*, \delta_1)$ , tem-se

$$\|H^{-1}\| \leq M \tag{15}$$

para todo  $H \in \partial_C F(x)$ .

Por outro lado, pelo Lema 3.1, dado  $M > 0$ , existe  $\delta_2$  tal que

$$\|F(x) - F(x^*) - H(x - x^*)\| \leq \frac{1}{M+1} \|x - x^*\| \quad \forall x \in B(x^*, \delta_2).$$

Tomando  $\delta = \min\{\delta_1, \delta_2\}$  e considerando  $x^k \in B(x^*, \delta)$  e  $H_k \in \partial_C F(x^k)$ , temos

$$\begin{aligned} \|x^{k+1} - x^*\| &= \|x^k - H_k^{-1}F(x^k) - x^*\| \\ &= \|H_k^{-1}[H_k(x^k - x^*) - F(x^k)]\| \\ &= \|H_k^{-1}[F(x^*) - F(x^k) + H_k(x^k - x^*)]\| \\ &\leq \|H_k^{-1}\| \cdot \|F(x^k) - F(x^*) - H_k(x^k - x^*)\| \\ &\leq \frac{M}{M+1} \|x^k - x^*\| \\ &< \|x^k - x^*\| \end{aligned}$$

o que implica que  $x^{k+1} \in B(x^*, \delta)$ .

### 3.2. Análise de convergência local

Vimos anteriormente que se o algoritmo para em algum ponto, este ponto é solução do problema  $EP(X, f)$ . Suponha então que o algoritmo gera uma sequência. Temos o seguinte teorema:

**Teorema 3.1.** *Seja  $x^* \in X$  uma solução do problema  $EP(X, f)$ . Suponha que as Hipóteses 2.2 e 2.3 valem em  $y(x^*) = x^*$ . Então existe uma vizinhança  $V(x^*)$  de  $x^*$  tal que para qualquer ponto inicial  $x^0 \in V(x^*)$  a sequência  $\{x^k\}$  gerada pelo Algoritmo 1 converge superlinearmente para  $x^*$ .*

*Demonstração.* Vimos na subseção anterior que existe uma vizinhança  $V(x^*)$  de  $x^*$  tal que o algoritmo está bem definido para qualquer ponto inicial  $x^0 \in V(x^*)$  e que, pelo Lema 2.3, existe  $M > 0$  tal que  $\|H_k^{-1}\| \leq M \forall k \in \mathbb{N}$ .

Logo, do Lema 3.1, temos

$$\begin{aligned} \|x^{k+1} - x^*\| &= \|H_k^{-1}[F(x^k) - F(x^*) - H_k(x^k - x^*)]\| \\ &\leq M \|F(x^k) - F(x^*) - H_k(x^k - x^*)\| \\ &= o(\|x^k - x^*\|) \end{aligned}$$

o que prova a convergência (superlinear) da sequência  $\{x^k\}$ . □

## 4. Casos Particulares

Nesta seção consideramos a seguinte bifunção tipo-proximal  $h : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  satisfazendo a Hipótese 2.1:

$$h(x, y) = \frac{\alpha}{2} \|y - x\|^2, \quad \alpha > 0.$$

### 4.1. Otimização

Neste caso temos a bifunção  $f$  definida por

$$f(x, y) = \varphi(y) - \varphi(x)$$

onde  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$  é a função convexa a ser minimizada sobre o conjunto  $X$  definido em (1).

O passo (P.1) do algoritmo se reduz a uma iteração proximal do problema: fixado  $x^k$ , encontrar  $y^k = y(x^k)$  solução de

$$\begin{aligned} \min_y \quad & \varphi(y) + \frac{\alpha}{2} \|y - x^k\|^2 \\ \text{s.a.} \quad & g(y) \leq 0. \end{aligned}$$

No passo (P.3), que corresponde ao método de Newton aplicado a  $F(x)$ , a matriz  $\nabla y^J(x^k)$  está definida por

$$\nabla y^J(x^k) = \alpha C^{-1} [I - D(D^T C^{-1} D)^{-1} D^T C^{-1}]$$

sendo  $C = \nabla^2 \varphi(y^k) + \sum \lambda_i^J(x^k) \nabla^2 g_i(y^k) + \alpha I$  e  $D = \nabla g_J(y^k)$ .

## 4.2. Desigualdade Variacional

Neste caso a bifunção  $f$  é definida por

$$f(x, y) = \langle \Phi(x), y - x \rangle$$

onde  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  é o operador  $\mathcal{C}^1$  que define a desigualdade variacional, juntamente com o conjunto  $X$  das restrições dado por (1).

O passo (P.1) do algoritmo se reduz a uma iteração do problema quadrático: fixado  $x^k$ , encontrar  $y^k = y(x^k)$  solução de

$$\begin{aligned} \min_y \quad & \langle \Phi(x^k), y - x^k \rangle + \frac{\alpha}{2} \|y - x^k\|^2 \\ \text{s.a.} \quad & g(y) \leq 0. \end{aligned}$$

No passo (P.3), que corresponde ao método de Newton aplicado a  $F(x)$ , a matriz  $\nabla y^J(x^k)$  está definida por

$$\nabla y^J(x^k) = [\alpha I - \nabla \Phi(x^k)] C^{-1} [I - D(D^T C^{-1} D)^{-1} D^T C^{-1}]$$

sendo  $C = \sum \lambda_i^J(x^k) \nabla^2 g_i(y^k) + \alpha I$  e  $D = \nabla g_J(y^k)$ .

## 4.3. Problema de Nash Generalizado

Neste caso nossa bifunção  $f$  é definida por

$$f(x, y) = \sum_{v=1}^N [\theta_v(y^v, x^{-v}) - \theta_v(x^v, x^{-v})]$$

onde  $\theta_v : \mathbb{R}^n \rightarrow \mathbb{R}$  é a função custo do  $v$ -ésimo jogador que deve ser minimizada sujeito a  $(x^v, x^{-v}) \in X$  dado por (1), sendo  $\theta_v \in \mathcal{C}^2$ ,  $v = 1, \dots, N$  e convexa como função de  $x^v$ .

Nosso algoritmo, neste caso, reencontra aquele apresentado em [von Heusinger et al., 2012].

## 5. Resultados Numéricos

Nesta seção nós avaliamos o desempenho de nosso algoritmo apresentando alguns resultados numéricos. Algumas comparações são também realizadas. Nós utilizamos o SCILAB em nosso computador pessoal para obter nossos resultados. Em todos os exemplos a bifunção  $h$  escolhida foi  $h(x, y) = \frac{\alpha}{2} \|y - x\|^2$  com  $\alpha = 10^{-4}$  e o algoritmo para se  $\|y^k - x^k\| < \varepsilon$ . Em todos os exemplos faremos  $\varepsilon = 10^{-12}$  o que significa que nós requeremos uma precisão muito alta para a solução o que nem sempre é alcançado por outros métodos. O primeiro exemplo consiste de um problema de Nash que se encontra em [von Heusinger et al., 2012], os outros dois são problemas de equilíbrio quadráticos que se encontram em [Santos e Scheimberg, 2011] e [Nasri et al., 2016].

*Exemplo 5.1* Considere o problema de equilíbrio de Nash com dois jogadores dados em [von Heusinger et al., 2012] e sua reformulação como um problema de equilíbrio.

A bifunção  $f$  é dada por

$$f(x, y) = \sum_{v=1}^2 [\theta_v(y^v, x^{-v}) - \theta_v(x^v, x^{-v})],$$

com

$$\theta_1(x_1, x_2) = \frac{1}{2} x_1^2 - x_1 x_2 \text{ e } \theta_2(x_1, x_2) = x_2^2 + x_1 x_2,$$

e o conjunto comum de restrições é dado por  $X = \{x \in \mathbb{R}^2; x_1 \geq 1, x_2 \geq 0, x_1 + x_2 \geq 1\}$ . O ponto ótimo é  $x^* = (1, 0)^T$  e o ponto inicial  $x_0 = (1, 1)^T$ .

Tabela 1: Resultado numérico para o Exemplo 5.1.

$k$	$x_1^k$	$x_2^k$	$\ y^k - x^k\ $
0	1,000000	1,000000	1,000000
1	1,000000	0,000000	0,000000

Uma vez que nosso algoritmo reencontra o algoritmo de [von Heusinger et al., 2012] era de se esperar resultados análogos àqueles. Este fato é comprovado pela Tabela 1.

*Exemplo 5.2* Considere os dois problemas de equilíbrio que se encontram em [Santos e Scheimberg, 2011] e [Nasri et al., 2016], com  $X = \{x \in \mathbb{R}^5; g_i(x) \leq 0, i = 1, \dots, 6\}$ , onde

$$g_i(x) = x_i^2 - 5, i = 1, \dots, 5$$

$$g_6(x) = -1 - \sum_{i=1}^5 x_i,$$

e a bifunção  $f$  é da forma

$$f(x, y) = \langle Px + Qy + q, y - x \rangle.$$

As matrizes  $P$ ,  $Q$  e o vetor  $q$  são dados por

$$P_1 = \begin{bmatrix} 3,1 & 2 & 0 & 0 & 0 \\ 2 & 3,6 & 0 & 0 & 0 \\ 0 & 0 & 3,5 & 2 & 0 \\ 0 & 0 & 2 & 3,3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}, P_2 = \begin{bmatrix} 3,1 & 2 & 0 & 0 & 0 \\ 2 & 3,6 & 0 & 0 & 0 \\ 0 & 0 & 3,5 & 2 & 0 \\ 0 & 0 & 2 & 3,3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix},$$

$$Q = \begin{bmatrix} 1,6 & 1 & 0 & 0 & 0 \\ 1 & 1,6 & 0 & 0 & 0 \\ 0 & 0 & 1,5 & 1 & 0 \\ 0 & 0 & 1 & 1,5 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \text{ e } q = \begin{bmatrix} 1 \\ -2 \\ -1 \\ 2 \\ -1 \end{bmatrix},$$

onde o  $i$ -ésimo problema considera  $P = P_i, i = 1, 2$ .

Em cada um destes dois exemplos realizamos dois testes, um com a tolerância  $\varepsilon = 10^{-3}$  e outro com a tolerância  $\varepsilon = 10^{-12}$ . Nosso objetivo foi primeiramente comparar nossos resultados com aqueles que se encontram em [Santos e Scheimberg, 2011] e [Nasri et al., 2016] onde os mesmos utilizam  $\varepsilon = 10^{-3}$ . Em segundo lugar realizamos nossos testes com  $\varepsilon = 10^{-12}$  para exibir a alta precisão de nosso algoritmo e sua taxa de convergência superlinear. Em todos os casos utilizamos o mesmo ponto inicial,  $x^0 = (1, 3, 1, 1, 2)^T$ , utilizados nos referidos artigos citados.

No caso em que  $\varepsilon = 10^{-3}$  vemos, de acordo com a Tabela 2 e Tabela 3, que o nosso algoritmo resolve ambos os problemas em apenas duas iterações. Por outro lado, em [Nasri et al., 2016] os problemas são resolvidos em três iterações e em [Santos e Scheimberg, 2011] eles são resolvidos em 10 iterações. Vale ressaltar que o algoritmo de [Nasri et al., 2016] usa técnica de lagrangeana aumentada e incorpora em sua rotina características do método de Newton enquanto o algoritmo de [Santos e Scheimberg, 2011] é do tipo subgradiente projetado e possui, portanto, baixo custo computacional comparado com os demais.

Nas Tabelas 4 e 5 encontram-se os resultados numéricos para os dois problemas com  $\varepsilon = 10^{-12}$ . Verificamos que mesmo para essa precisão elevada nosso algoritmo resolve o problema em uma quantidade relativamente pequena de iterações. O primeiro problema ( $P = P_1$ ) é resolvido em cinco iterações enquanto o segundo ( $P = P_2$ ) é resolvido em oito iterações apenas.

Tabela 2: Resultado numérico para o Exemplo 5.2 usando  $\epsilon = 10^{-3}$  e  $P = P_1$ .

$k$	$x_1^k$	$x_2^k$	$x_3^k$	$x_4^k$	$x_5^k$	$\ y^k - x^k\ $
0	1	3	1	1	2	-
2	-0,7253883	0,8031085	0,7199996	-0,8666663	0,2500000	0,0000013

Tabela 3: Resultado numérico para o Exemplo 5.2 usando  $\epsilon = 10^{-3}$  e  $P = P_2$ .

$k$	$x_1^k$	$x_2^k$	$x_3^k$	$x_4^k$	$x_5^k$	$\ y^k - x^k\ $
0	1	3	1	1	2	-
2	-0,7253882	0,8031084	0,7199996	-0,8666662	0,1999997	0,0000015

Tabela 4: Resultado numérico para o Exemplo 5.2 usando  $\epsilon = 10^{-12}$  e  $P = P_1$ .

$k$	0	5
$x_1^k$	1	-0.72538860058418
$x_2^k$	3	0.80310880804453
$x_3^k$	1	0.71999999986217
$x_4^k$	1	-0.86666666621553
$x_5^k$	2	0.24999999996298
$\ y^k - x^k\ $	-	$0,291 \cdot 10^{-13}$

Tabela 5: Resultado numérico para o Exemplo 5.2 usando  $\epsilon = 10^{-12}$  e  $P = P_2$ .

$k$	0	8
$x_1^k$	1	-0,72538860091503
$x_2^k$	3	0,80310880942015
$x_3^k$	1	0,72000000044800
$x_4^k$	1	-0,86666666664117
$x_5^k$	2	0,20000000071062
$\ y^k - x^k\ $	-	$3,7404 \cdot 10^{-13}$

## 6. Conclusão

Neste trabalho apresentamos um método tipo-Newton proximal para o problema de equilíbrio cujo conjunto viável é definido por desigualdades convexas. Provamos a convergência superlinear local do método sob hipóteses razoáveis. Também implementamos nosso algoritmo e comparamos os resultados com outros encontrados na literatura. Observamos que o nosso algoritmo possui uma boa eficiência frente àqueles apresentados em [Santos e Scheimberg, 2011] e [Nasri et al., 2016]. Os resultados numéricos indicam que o método é confiável e é capaz de encontrar uma solução com uma precisão muito elevada. Acreditamos que este método é muito promissor.

## Referências

Bertsekas, D. P. (1996). *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific.

- Bigi, G. e Passacantando, M. (2016). Gap functions for quasi-equilibria. Technical report, Università de Pisa.
- Blum, E. e Oettli, W. (1994). From optimization and variational inequalities to equilibrium problems. *The Mathematics Student*, 63(1–4):123–145.
- Facchinei, F. e Pang, J. S. (2003). *Finite-Dimensional Variational Inequalities and Complementary Problems*, volume 1. Springer-Verlag, New York.
- Hogan, W. W. (1973). Point-to-set maps in mathematical programming. *SIAM Review*, 15:591–603.
- Jacinto, F. M. O. e Scheimberg, S. (2008). Duality for generalized equilibrium problem. *Optimization*, 57(6):795–805.
- Janin, R. (1984). Directional derivative of the marginal function in nonlinear programming. *Mathematical Programming Study*, 21:110–126.
- Nasri, M., Matioli, L. C., Ferreira, E. M. S., e Silveira, A. (2016). Implementation of augmented lagrangian methods for equilibrium problems. *Optimization Theory and Applications*, 168(3): 971–991.
- Santos, P. S. M. e Scheimberg, S. (2011). An inexact subgradient algorithm for equilibrium problems. *Computational & Applied Mathematics*, 30(1):91–107.
- von Heusinger, A., Kanzow, C., e Fukushima, M. (2012). Newton’s method for computing a normalized equilibrium in the generalized nash game through fixed point formulation. *Mathematical Programming*, 132:99–123.