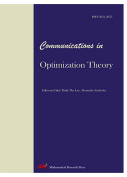
Commun. Optim. Theory 2023 (2023) 23 https://doi.org/10.23952/cot.2023.23

MATHRES

Communications in Optimization Theory

Available online at http://cot.mathres.org



NUMERICAL ASPECTS OF FINDING NONLINEAR PRODUCTION - CONSUMPTION EQUILIBRIUM

IGOR GRIVA[1,*], ROMAN POLYAK[1]

Department of Mathematical Sciences, George Mason University, Fairfax, VA USA

Dedicated to the memory of Professor Naum Z. Shor

Abstract. Recently introduced nonlinear production-consumption equilibrium (NPCE) is a combination and generalization of both linear programming (LP) and input-output (IO) models. Finding NPCE is equivalent to solving a variational inequality (VI) with a very simple feasible set, projection on which is a low cost operation. Therefore for finding NPCE we use Extra Pseudo-Gradient (EPG) method. We present and analyze numerical results obtained by using EPG method on a set of random generated NPCE. The obtained results show that the number of EPG steps required for finding NPCE grows linearly with the number of products of a given economy. The number of arithmetic operations or time required for finding NPCE grows as a cube of the number of products. The numerical results strongly corroborate the complexity bounds established under reasonable assumptions on the input data in [8].

Keywords. Extra projected gradient method; Production-consumption equilibrium; Variational inequality.

2020 Mathematics Subject Classification. 47J20, 49J40.

1. Introduction

The NPCE is a fundamental departure from both Linear Programming (LP) and Input–Output (IO) models.

First, the NPCE combines the production and the consumption sectors of an economy.

Second, in contrast to LP and IO models, the production cost per unit, the consumption vector and the factor (resources) vector are not fixed and not given *a priori*.

They are operators with values dependent on the production output, prices for goods and prices for factors.

As it turns out, finding NPCE is equivalent to solving a particular variational inequality (VI) with a simple feasible set. Projection on the feasible set is a low cost numerical operation. Therefore, for finding NPCE we used the EPG method introduced by Galina Korpelevich in

[*] Corresponding Author.

E-mail address: igriva@gmu.edu (I. Griva), rpolyak@gmu.edu (R. Polyak).

Received: January 23, 2022; Accepted: March 2, 2023.

the mid 1970s [1] and extensively studied in the last few decades (see [2]-[8] and references therein). The main computational cost at each step of the EPG method is computing the pseudo-gradient, which requires a few matrix by vector multiplications.

We apply the EPG method to a set of randomly generated NPCE problems with various production, consumption and factor operators. The obtained results confirm theoretical findings (see [8]): under reasonable assumptions on input data, the number of EPG steps required for finding NPCE grows linearly together with the number of products of a given economy. The number of operations, or solving time grows as a cube of the number of products.

The paper is organized as follows. In the second section, we describe the NPCE and equivalent variational inequality (VI). In the third section we describe the EPG method for finding NPCE, its convergence, complexity bounds and estimate the Lipschitz constant for the pseudo-gradient VI operator. We conclude the paper by discussing the numerical results.

2. Nonlinear Production-Consumption Equilibrium Problem

Consider an economy with n products. The components x_i, $1 \leq i \leq n$ of the production vector $x \in \mathbb{R}^n_+$ specify how many units of each product to be produced.

The components p_i, $1 \leq i \leq n$ of the cost vector $p \in \mathbb{R}^n_+$ define the production cost of a unit of the product $1 \leq i \leq n$.

Vector x is used to define the consumption vector $c = x - Ax = (I - A)x \in \mathbb{R}^n_+$, where elements a_{ij} of the balance matrix $A : \mathbb{R}^n \to \mathbb{R}^n$ define how much of a product $1 \leq i \leq n$ is required to produce a unit of a product $1 \leq j \leq n$.

We assume that A is a productive matrix, therefore for any $c \in \mathbb{R}^n_+$ the solution $x \in \mathbb{R}^n_+$ of the system $c = x - Ax$ exists.

The production uses additional m resources. The element b_{lj} of the technological matrix $B : \mathbb{R}^n \to \mathbb{R}^m$ describe how much of a resource $1 \leq l \leq m$ is required to produce an item of the product $1 \leq j \leq n$, $m \leq n$.

The vector $Bx \in \mathbb{R}^m$ defines how much of each resource is used for the production of $x \in \mathbb{R}^n_+$. Therefore, Bx should not exceed a given resources availability vector r.

The NPCE combines both production and consumption parts of the economy. Also, the basic vectors p, c and r are not fixed and not given *a priori*. They are replaced by operators with values dependent on production output $x \in \mathbb{R}^n_+$, consumption price vector $\lambda \in \mathbb{R}^n_+$, and factor price vector $v \in \mathbb{R}^m_+$.

In other words, for a given production vector x the production cost per unit is defined by the vector $p(x) = (p_1(x), \dots, p_n(x))$. For a given consumption price vector λ the consumption is defined by a vector $c(\lambda) = (c_1(\lambda), \dots, c_n(\lambda))$. For a given factor price vector v the availability of resources is defined by vector $r(v) = (r_1(v), \dots, r_m(v))$.

The NPCE is defined by a triple $y^* = (x^*, \lambda^*, v^*) \in \Omega = \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times \mathbb{R}^m_+$ that

$$x^* \in \mathrm{Agrmin}_x \left\{ p(x^*)^T x \,|\, (I - A)x \geq c(\lambda^*), Bx \leq r(v^*), x \geq 0 \right\} \tag{2.1}$$

and

$$\begin{aligned}(\lambda^*, v^*) \in \mathrm{Agrmax}_{\lambda, v} &\left\{ c(\lambda^*)^T \lambda - r(v^*)^T v \,|\right. \\ &\left.(I - A)^T \lambda - B^T v \leq p(x^*), \lambda \geq 0, v \geq 0 \right\}\end{aligned} \tag{2.2}$$

The existence and uniqueness of y^* follows from strong monotonicity of the operators p, c and r.

The convergence rate and the complexity bounds for the EPG method were established under strong monotonicity of the operators p, c and r only at NPCE y^* (see [8]).

The operator $p : \mathbb{R}^n_+ \to \mathbb{R}^n_+$ is strongly monotone increasing at x^* if there exists $\alpha > 0$ such that for any x from $\mathbb{R}^n_+$ the following inequality takes place

$$\langle p(x) - p(x^*), x - x^* \rangle \geq \alpha \|x - x^*\|^2, \tag{2.3}$$

where $\|x\| = \sqrt{\langle x, x \rangle}$. The operator $c : \mathbb{R}^n_+ \to \mathbb{R}^n_+$ is strongly monotone descreasing at λ^* if there exists $\beta > 0$ such that for any $\lambda \in \mathbb{R}^n_+$ the following inequality takes place

$$\langle c(\lambda) - c(\lambda^*), \lambda - \lambda^* \rangle \leq -\beta \|\lambda - \lambda^*\|^2. \tag{2.4}$$

The operator $r : \mathbb{R}^m_+ \to \mathbb{R}^m_+$ is strongly monotone increasing at v^* if there exists $\gamma > 0$ such that for any v from $\mathbb{R}^m_+$ the following inequality takes place

$$\langle r(v) - r(v^*), v - v^* \rangle \geq \gamma \|v - v^*\|^2. \tag{2.5}$$

The conditions (2.3)-(2.5) mean that production, consumption price, and factor price operators are sensitive to production, consumption and factor price violation at the equilibrium.

In addition to the strong monotonicity (2.3)-(2.5), we assume that p, c, and r are Lipschitz continuous, i.e. there exist constants L_p, L_c, and L_r such that for any x_1 and x_2 from $\mathbb{R}^n$, for any λ_1 and λ_2 from $\mathbb{R}^n$, and any v_1 and v_2 from $\mathbb{R}^m$, the following bounds take place

$$\|p(x_1) - p(x_2)\| \leq L_p \|x_1 - x_2\|, \tag{2.6}$$

$$\|c(\lambda_1) - c(\lambda_2)\| \leq L_c \|\lambda_1 - \lambda_2\|, \tag{2.7}$$

and

$$\|r(v_1) - r(v_2)\| \leq L_r \|v_1 - v_2\|. \tag{2.8}$$

Consider the Lagrangian

$$L(y^*, X, \Lambda, V) = \langle p(x^*), X \rangle - \langle \Lambda, (I - A)X - c(\lambda^*) \rangle - \langle V, -BX + r(v^*) \rangle$$

for LP (2.1). Then

$$y^* \in \operatorname{Arg} \min_{X \in \mathbb{R}^n_+} \max_{\substack{\Lambda \in \mathbb{R}^n_+ \\ V \in \mathbb{R}^m_+}} L(y^*, X, \Lambda, V),$$

which means that finding NPCE is equivalent to solving a two person game with payoff functions

$$\varphi_1(y; X, \lambda, v) = -L(y, X, \lambda, v)$$

and

$$\varphi_2(y; x, \Lambda, V) = L(y, x, \Lambda, V)$$

and the corresponding strategies $X \in \mathbb{R}^n_+$ and $(\Lambda, V) \in \mathbb{R}^n_+ \times \mathbb{R}^m_+$. The normalized payoff function is

$$\Phi(y, Y) = \varphi_1(y; X, \lambda, v) + \varphi_2(y; x, \Lambda, V).$$

Therefore,

$$y^* \in \operatorname{Argmax}\{\Phi(y^*, Y) \,|\, Y \in \Omega\}.$$

The pseudo-gradient of $\Phi(y, Y)$:

$$\nabla_Y \Phi(y, Y)\,|_{Y=y} = g(y) = \left((I - A)^T \lambda - p(x) - B^T v;\, c(\lambda) - (I - A)x;\, Bx - r(v)\right)$$

is our main tool.

It has been established in [8] that finding $y^* = (x^*, \lambda^*, v^*)$ from (2.1)-(2.2) is equivalent to solving the following variational inequality (VI)

$$\langle g(y^*), y - y^* \rangle \leq 0, \forall y \in \Omega. \tag{2.9}$$

From (2.3)-(2.5) for $\delta = \min\{\alpha, \beta, \gamma\}$ follows

$$\langle g(y) - g(y^*), y - y^* \rangle \leq -\delta \|y - y^*\|^2.$$

From (2.6)-(2.8) follows existence of $L > 0$ that

$$\|g(y_1) - g(y_2)\| \leq L\|y_1 - y_2\|, \quad \forall y_1, y_2 \in \Omega.$$

For a triple $(\bar{x}, \bar{\lambda}, \bar{v}) \in \Omega$ to be the NPCE it is necessary and sufficient that the feasibility

$$(I - A)^T \bar{\lambda} - p(\bar{x}) - B^T \bar{v} \leq 0, \tag{2.10}$$

$$c(\bar{\lambda}) - (I - A)\bar{x} \leq 0, \tag{2.11}$$

$$B\bar{x} - r(\bar{v}) \leq 0, \tag{2.12}$$

and complementarity conditions

$$\left\langle (I - A)^T \bar{\lambda} - p(\bar{x}) - B^T \bar{v}, \bar{x} \right\rangle = 0, \tag{2.13}$$

$$\left\langle c(\bar{\lambda}) - (I - A)\bar{x}, \bar{\lambda} \right\rangle = 0, \tag{2.14}$$

$$\langle B\bar{x} - r(\bar{v}), \bar{v} \rangle = 0 \tag{2.15}$$

are satisfied.

Let us define the complementarity violation function $C : \Omega \to \mathbb{R}_+$ by formula

$$C(y) = C(x, \lambda, v) = \max\{\left|\left\langle (I - A)^T \lambda - p(x) - B^T v, x \right\rangle\right|;$$

$$|\langle c(\lambda) - (I - A)x, \lambda \rangle|; |\langle Bx - r(v), v \rangle|\}$$

and the optimality violation function $\mu : \Omega \to \mathbb{R}_+$ by formula

$$\mu(x, \lambda, v) = \max\left\{\|[(I - A)^T \lambda - p(x) - B^T v]_+\|_\infty,\right.$$

$$\left.\|[c(\lambda) - (I - A)x]_+\|_\infty, \|[Bx - r(v)]_+\|_\infty, C(x, \lambda, v)\right\},$$

where

$$[z]_+ = ([z_1]_+, \ldots, [z_q]_+) \in \mathbb{R}^q, \quad [z_i]_+ = \begin{cases} z_i, & \text{if}\, z_i \geq 0 \\ 0, & \text{if}\, z_i < 0. \end{cases}$$

From (2.10)-(2.15) follows that for $y = (x, \lambda, v)$ to be NPCE it is necessary and sufficient that $\nu(y) = 0$. We use the value of $\nu(y)$ to measure the violation of $y \in \Omega$ from y^*.

```
1. Select $0 < t < (\sqrt{2}L)^{-1}$.
2. Select $x_0 \in \mathbb{R}^n_{++}$, $\lambda_0 \in \mathbb{R}^n_{++}$, and $v_0 \in \mathbb{R}^m_{++}$, Set $s = 1$.
3. Set $\hat{x}_s = [x_{s-1} + t((I-A)^T\lambda_s - p(x_{s-1}) - B^T v_{s-1})]_+$.
4. Set $\hat{\lambda}_s = [\lambda_{s-1} + t(c(\lambda_{s-1}) - (I-A)x_{s-1})]_+$
5. Set $\hat{v}_s = [v_{s-1} + t(Bx_{s-1} - r(v_{s-1}))]_+$
6. Set $x_s = [x_{s-1} + t((I-A)^T\hat{\lambda}_s - p(\hat{x}_s) - B^T\hat{v}_s)]_+$.
7. Set $\lambda_s = [\lambda_{s-1} + t(c(\hat{\lambda}_s) - (I-A)\hat{x}_s)]_+$
8. Set $v_s = [v_{s-1} + t(B\hat{x}_s - r(\hat{v}_s))]_+$
9. If $\mu(x_s, \lambda_s, v_s) \geq \varepsilon$, Set $s = s+1$, Goto Step 3.
10. Output $(x_s, \lambda_s, v_s)$.
```

FIGURE 1. Extra Pseudo Gradient algorithm

3. NUMERICAL EXPERIMENTS

Application of the EPG method for solving VI (2.9) leads to the following two stage method, which is nothing but a pricing mechanism for finding the NPCE. First, one predicts the triple $\hat{y}_s = (\hat{x}_s, \hat{\lambda}_s, \hat{v}_s)$ by formula

$$\hat{y}_s = P_\Omega\left(y_{s-1} + tg(y_{s-1})\right) = [y_{s-1} + tg(y_{s-1})]_+\,, \tag{3.1}$$

then one finds a new approximation

$$y_s = P_\Omega\left(y_{s-1} + tg(\hat{y}_s)\right) = [y_{s-1} + tg(\hat{y}_s)]_+\,. \tag{3.2}$$

The EPG algorithm is summarized in Figure 1.

Under assumptions (2.3)-(2.8) the algorithm converges to NPCE $y^* = (x^*, \lambda^*, v^*)$ with a linear rate (see Theorem 5 in [8]).

Our main purpose is to analyze numerical performance of the EPG method and compare the numerical results with the complexity bounds established in Theorem 5 [8]. The number of iterations required to solve the problem with accuracy $\varepsilon > 0$ is

$$N = O(\delta^{-1}L\ln\varepsilon^{-1}). \tag{3.3}$$

From Lipschitz continuity (2.6)-(2.8) of the operators p, c, and r follows directly that the operator g is also Lipschitz continuous, i.e. there exists $L > 0$ such that for any y_1 and y_2 from Ω the following bound holds:

$$\|g(y_1) - g(y_2)\| \leq L\|y_1 - y_2\|.$$

To find an upper bound for the Lipschitz constant L let us consider the Jacobian of g :

$$J = \nabla g(x,\lambda,v) = \begin{bmatrix} -\nabla p(x) & (I-A)^T & -B^T \\ -(I-A) & \nabla c(\lambda) & 0_{(n,m)} \\ B & 0_{(m,n)} & -\nabla r(v) \end{bmatrix}.$$

Then

$$L \leq \|J\|_F = \sqrt{\sum_{i=1}^{2n+m}\sum_{j=1}^{2n+m} J_{ij}^2} \leq \sqrt{\sum_{i=1}^{2n+m}\sum_{j=1}^{2n+m} M^2} = (2n+m)M,$$

where $M = \max\{J_{kl} \,|\, 1 \leq k \leq 2n+m,\ 1 \leq l \leq 2n+m\}$. Therefore, keeping in mind $m \leq n$, we obtain that $L = O(n)$. Hence, for a fixed $\delta > 0$, and a fixed small enough $0 < \varepsilon \ll \delta$, from (3.3) follows $N = O(n)$. So it takes $O(n)$ EPG steps to find an approximation $y \in \Omega$ for NPCE y^* with optimality violation given by $\varepsilon > 0$.

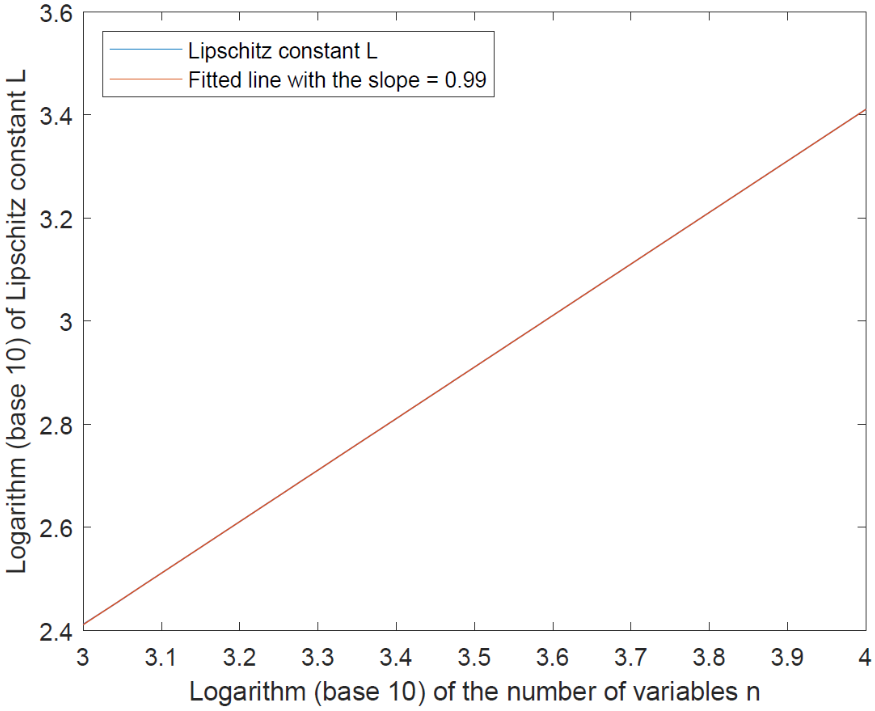


FIGURE 2. The upper bound for the Lipschitz constant as a function of n. The blue and the red lines almost coincide

We generate NPCE problems with $n = 1000, 1100, 1200, \ldots, 10000$ with $\delta = 1$ and study empirically the trends of the growth of both the total iteration count and the solving time. The accuracy in the stopping criteria is selected $\varepsilon = 10^{-2}$.

We use the following vector functions

$$\hat{p}(x) = Px + \bar{p},\ \hat{c}(\lambda) = -C\lambda + \bar{c},\ \hat{r}(v) = Rv + \bar{r}, \tag{3.4}$$

where $P, C : \mathbb{R}^n \to \mathbb{R}^n$, $R : \mathbb{R}^m \to \mathbb{R}^m$, $\bar{p}, \bar{c} \in \mathbb{R}^n$, $\bar{r} \in \mathbb{R}^m$, are randomly generated positive definite matrices with the smallest eigenvalue equal 1, and the positive entries not exceeding 2. Matrices A and B has randomly generated entries uniformly distributed on $(0,1)$. Matrix A is ensured to be productive.

Then, we add nonlinear terms $\tilde{p}$, $\tilde{c}$ and $\tilde{r}$:

$$p(x) = \hat{p}(x) + \tilde{p}(x),$$

with $\tilde{p}(x) = (\tilde{p}_1(x), \ldots, \tilde{p}_n(x))$, $\tilde{p}_i(x) = \xi_i \ln(1 + x_i)$ and ξ_i being random numbers uniformly distributed on $(0,1)$. Similarly, we define

$$c(\lambda) = \hat{c}(\lambda) - \tilde{c}(\lambda),\ r(v) = \hat{r}(v) + \tilde{r}(v).$$

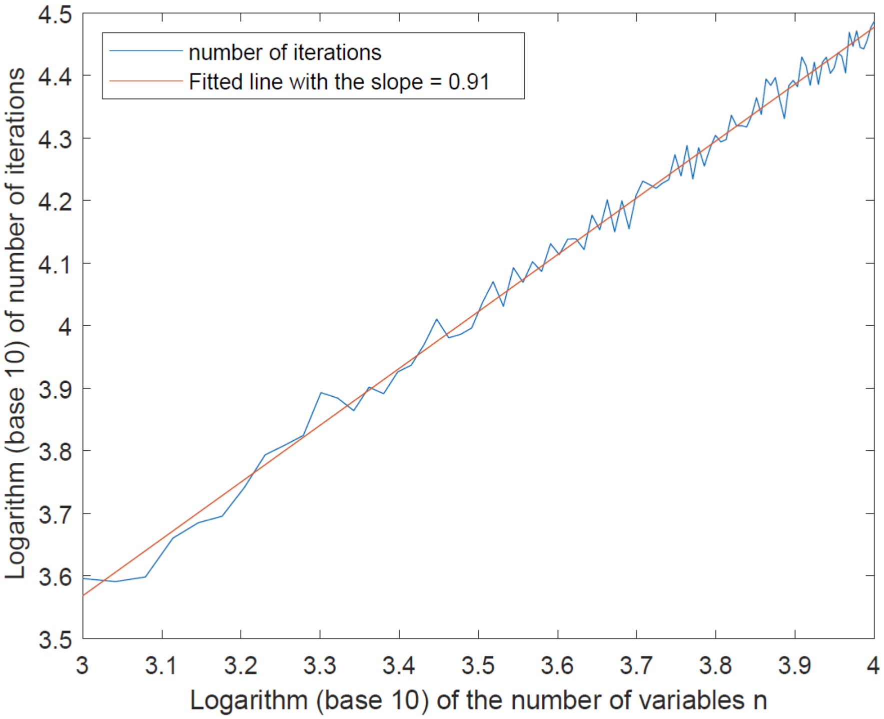


FIGURE 3. The number of iterations it takes for EPG to solve NPCE problems as a function of n (blue), a fitted line with the slope 0.91 (red).

Note that adding the nonlinear terms $\tilde{p}$, $\tilde{c}$ and $\tilde{r}$ do not increase the elements J_{ij} of the Jacobian J by more than 1. Therefore due to the boundedness of the elements J_{ij} of the Jacobian J the Lipschitz constant L grows as $L = O(n)$ for the generated problems.

Every iteration requires $O(n^2)$ arithmetic operations, therefore the complexity of the EPG method in Figure 1 is:

$$COMP(EPG) = O(n^2\delta^{-1}L\ln\varepsilon^{-1}). \tag{3.5}$$

We used a Windows laptop with Intel Xeon CPU E3-1535M v6 to run computational experiments. While solving NPCE problems we record three parameters for each solved problem: the upper bound for L, the number of iterations and the solving time. Then we generate a logarithmic (base 10) plot. The advantage of using the logarithmic plot is twofold. First, if the logarithms of the plotted parameter follow a linear pattern (a line), that implies a polynomial growth. Second, the slope of that line approximates the power of the polynomial. Therefore, to find the slope we fit a line to the logarithmic data.

Figure 2 shows that the Lipschitz constant scales up as $O(n)$. The line that shows the Lipschitz constants is covered by a perfectly fitted red line with the slope close to 1. We expect that the number of iterations scale up similarly. Indeed, as Figure 3 suggests, the fitted line has a slope of 0.91.

Each iterations requires $O(n^2)$ arithmetic operations. Therefore the total number of operations is expected to be $O(n^3)$, which is proportional to the time it takes to solve a NPCE problem. Figure 4 suggests that the slope of the fitted line is 2.90 consistent with the bound (3.5) established in [8].

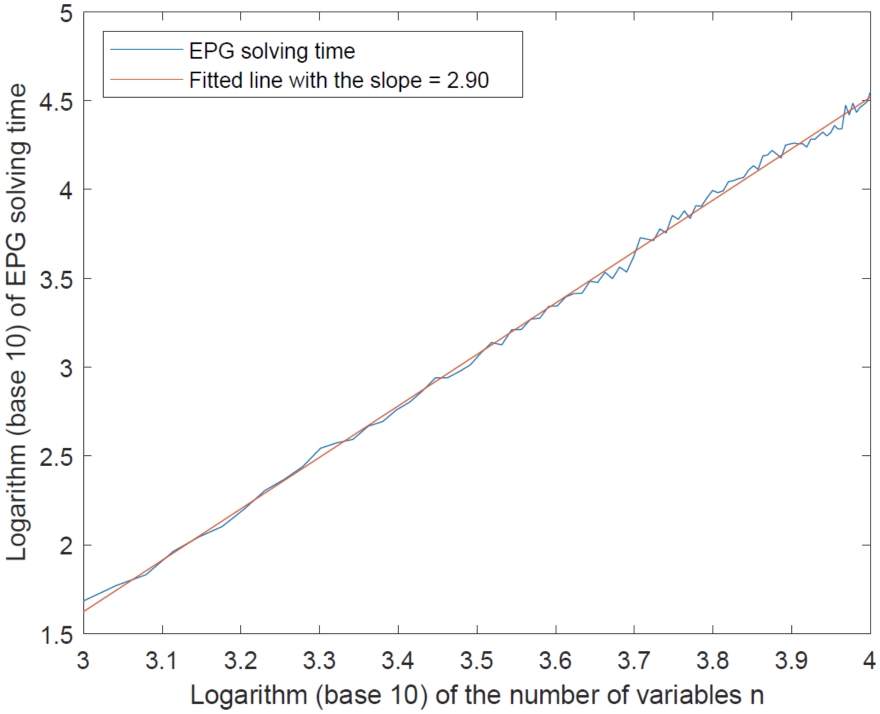


FIGURE 4. The time in seconds it took for the EPG to solve NPCE problems as a function of n (blue), a fitted line with the slope 2.90 (red).

4. DISCUSSION AND CONCLUDING REMARKS

The EPG method (3.1)-(3.2) not only leads to a pricing mechanism for establishing equilibrium, it also fully decompose the problem of finding NPCE. Such a decomposition allows computing the primal and dual variables at each step simultaneously without solving any optimization problem or system of linear equations. Instead, each step requires a few matrix by vector multiplications, which can be done in parallel. That allows solving large scale NPCE problems.

From the obtained numerical results follows a few conclusions.

First, under a fixed $\delta > 0$ and $0 < \varepsilon \ll \delta$ the number of iterations grows linearly with n. Second, the time required for finding NPCE grows as $O(n^3)$. The conclusions corroborate the bounds established in [8].

Finally, it is rather remarkable that finding NPCE has similar complexity as solving a linear system of equations of the same size.

REFERENCES

[1] G. Korpelevich, An Extragradient Method for Finding Saddle Points and for Other Problems, Matecon, 12 (1976) 747–756.

[2] A. Antipin, The gradient and extragradient approaches in bilinear equilibrium programming, Dorodnizin Computing Center RAS (in Russian), 2002.

[3] Y. Censor, A. Gibali, and S. Reich, Extensions of Korpelevich's extragradient method for the variational inequality problem in Euclidean space, Optimization 61 (2012) 1119–1132.

[4] A. Iusem, and M. Nasri, Korpelevich's method for variational inequality problems in Banach space, J. Global Optim. 50 (2010) 2086-2099.

[5] I. Konnov, Equilibrium models and Variational inequalities, Elsevier, Amsterdam, 210, 2001.

[6] J. Mashreghi, and M. Nasri, Forcing strong convergence of Korpelevich's method in Banach spaces with its applications in game theory, Nonlinear Anal, 72 (2010) 2086-2099.

[7] R. A. Polyak, Introduction to Continuous Optimization, Springer, Switzerland, 2021.

[8] R. A. Polyak, Finding Nonlinear Production - Consumption Equilibrium, to appear in a volume entitled "David Gale: Mathematical Economist: Essays in Appreciation on his 100th Birthday" in the series Monographs in Mathematical Economics (Springer), 2023.