



Combined Newton-Gradient Method for Constrained Root-Finding in Chemical Reaction Networks

Silvia Berra¹ · Alessandro La Torraca² · Federico Benvenuto¹ · Sara Sommariva¹ 

/ Accepted: 5 October 2023 / Published online: 16 November 2023
© The Author(s) 2023

Abstract

In this work, we present a fast, globally convergent, iterative algorithm for computing the asymptotically stable states of nonlinear large-scale systems of quadratic autonomous ordinary differential equations (ODE) modeling, e.g., the dynamic of complex chemical reaction networks. Toward this aim, we reformulate the problem as a box-constrained optimization problem where the roots of a set of nonlinear equations need to be determined. Then, we propose to use a projected Newton's approach combined with a gradient descent algorithm so that every limit point of the sequence generated by the overall algorithm is a stationary point. More importantly, we suggest replacing the standard orthogonal projector with a novel operator that ensures the final solution to satisfy the box constraints while lowering the probability that the intermediate points reached at each iteration belong to the boundary of the box where the Jacobian of the objective function may be singular. The effectiveness of the proposed approach is shown in a practical scenario concerning a chemical reaction network modeling the signaling network of colorectal cancer cells. Specifically, in this scenario the proposed algorithm is proved to be faster and more accurate than a classical dynamical approach where the asymptotically stable states are computed as the limit points of the flux of the Cauchy problem associated with the ODE system.

✉ Sara Sommariva
sommariva@dima.unige.it

Silvia Berra
silvia.berra@dima.unige.it

Alessandro La Torraca
alessandro.la_torraca@roche.com

Federico Benvenuto
benvenuto@dima.unige.it

¹ Dipartimento di Matematica, Università di Genova, Via Dodecaneso 35, 16146 Genoa, Italy

² Data and Analytics Chapter, Roche S.p.A., Monza, Italy

Keywords Box-constrained optimization · Nonnegative constraints · Chemical reaction network · Projected Newton's method · Projected gradient descent

Mathematics Subject Classification 65L05 · 65K10

1 Introduction

This paper is focused on the solution of root-finding problems in several variables where the system is composed by algebraic second-degree equations. This kind of problem is of interest in many application areas, including queuing problems, neutron transport theory, linear quadratic differential games (see, e.g., [29] and references therein). Our work is motivated by the study of nonnegative steady states of biological interaction networks which frequently arise in systems biology [10, 18, 19]. In particular, the problem of determining the steady states of complex chemical reaction networks (CRNs) in healthy and cancer cells is considered [23, 34]. In this respect, the G1/S transition of the cell cycle is a critical time when a cell becomes healthy or cancerous depending on the presence of certain predetermined genetic mutations. Several hundreds of proteins and as many reactions are involved in this transition, resulting in large ODE systems. A method for rapidly solving the system of equations is critical for modeling therapies that alter the rate constants of the system, such as current molecularly targeted therapies. The *in silico* study of these therapies is a primary perspective that this method aims to support. Indeed, by applying the law of mass action, the kinetics of the concentration of the proteins involved in the network can be modeled by a large first-order polynomial system of ordinary differential equations (ODEs). Finally, when no exogenous factors are considered, the equation system is quadratic and autonomous [8, 17, 44]. From an abstract point of view, steady states of large system of quadratic equations are far from being known, as a general theory exists up to the two-dimensional case [31]. In the case of an ODE representing a CRN, unknown concentrations cannot assume negative values, and then, the asymptotically steady states must fulfill the nonnegative constrained algebraic system of equations deriving from setting equal to zero the time derivatives of the ODE system. The number of involved unknown protein concentrations may scale up to several hundreds, and an efficient and accurate algorithm to solve the nonnegative steady-state problem is at the basis of tuning the kinetic parameters of the ODE system starting from experimental data, thus enabling the study of cell cancer behavior in real applications.

From a computational point of view, equilibrium can be found either in the direct way by taking the limit of the flux of the ODEs, or imposing the vanishing of the derivatives and solving the corresponding root-finding problem. The direct approach is computationally expensive, especially when the orbits of the dynamical system are bent around the equilibrium point as the time to run across the orbit may become arbitrarily large [15]. On the other hand, the main drawback of the second strategy is that these systems do not usually show any mathematical property which can ensure convergence of the root-finding algorithm. Indeed, pertinent good mathematical properties, such as matrix positive definiteness, depend on the form of the considered biological network and are not ensured in a general case. The typical structure of the ODE system

associated with a chemical reaction network (based on the mass action law) not only prevents us from exploiting recent methods to find the steady-state solutions by solving vector quadratic equations [29], but also makes it difficult to use classical methods, such as the Newton's or gradient descent methods. Indeed, as the nonnegative steady state normally belongs to the frontier of the positive cone, and therefore, it has many components equal to zero, classical nonnegative projected Newton-type methods are unstable as the Jacobian matrix—computed in a neighborhood of the solution—is strongly sparse and noninvertible. Moreover, classical projected gradient methods are known to be stable but with slow convergence, especially in cases of coupling them with a nonnegative projection.

In this work, we propose to overcome these limitations, by introducing a root-finding strategy based on combining steps of Newton's method and steps of gradient descent. While the Newton's method is applied to the algebraic equation system, the gradient descent is applied by scalarizing the system, i.e., minimizing the norm of the l.h.s. of the equation system [24]. To make the Newton's method more stable, instead of the standard orthogonal projection, we use a nonlinear projection operator onto the nonnegative orthant that is substantially an idempotent operator providing small positive entries rather than zero components. Doing so, it improves the condition number of the Jacobian matrix preventing the Newton's step to be unstable and hence making regularization unnecessary. Therefore, we combine the (nonlinearly projected) Newton's method with a gradient method to iteratively refine the starting point of the former until we get the convergence to a nonnegative stationary point. We prove the convergence of this combined technique provided that a proper backtracking rule on the gradient method is considered. Moreover, we test the efficiency of the proposed technique in the case of simulated CRN data, showing that, compared to standard ODE solvers, this method computes the steady states achieving greater accuracy in less time. The MATLAB[®] codes implementing the proposed approach are freely available at the GitHub repository https://github.com/theMIDAGroup/CRC_CRN.git.

The rest of the paper is organized as follows. In Sect. 2, we introduce the mathematical formulation of the problem and we describe the proposed algorithm whose convergence properties are studied in Sect. 3. In Sect. 4, we consider the problem of finding the asymptotically steady states of a CRN and we reformulate it as a nonnegative constrained root-finding problem. In Sect. 5, we show the results obtained by applying NLPC to a CRN designed for modeling cell signaling in colorectal cells and the most common mutations occurring in colorectal cancer. Finally our conclusions are offered in Sect. 6.

2 Mathematical Formulation

We consider the box-constrained set of nonlinear equations

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &= \mathbf{0} \\ \mathbf{x} &\in \Omega, \end{aligned} \tag{1}$$

where $\Omega = \times_{i=1}^n \Omega_i \subseteq \mathbb{R}^n$ is the Cartesian product of n closed intervals $\Omega_i \subseteq \mathbb{R}$, and $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a continuously differentiable function on Ω . In the considered problem of finding the nonnegative steady states of quadratic autonomous ODE systems, \mathbf{f} is composed by second-degree polynomials and Ω is the positive convex cone.

Several numerical approaches have been proposed to solve (1). Among these, a classical fast approach is the projected Newton’s method [2, 3] where the projector on the closed convex set Ω , $P : \mathbb{R}^n \rightarrow \Omega$ such that for all $\mathbf{z} \in \mathbb{R}^n$

$$P(\mathbf{z}) = \underset{\mathbf{y} \in \Omega}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{z}\|, \tag{2}$$

is applied at each iteration of a Newton’s scheme so that the final solution satisfies the box constraints in (1). However, in the general case convergence properties of the projected Newton’s methods strongly depend on the initial point, as no global convergence is guaranteed [26]. Additionally, the standard orthogonal projector P tends to provide iterative estimates on the boundary of Ω (e.g., when Ω is the positive cone P sets to zero all the negative components), and therefore, it may compromise the stability of the Newton’s method as the Jacobian of \mathbf{f} can be singular computed at these boundary estimates.

An alternative approach to Newton’s method consists in using the projected gradient descent method [21, 25] for solving the optimization problem

$$\mathbf{x} = \underset{\mathbf{x} \in \Omega}{\operatorname{argmin}} \Theta(\mathbf{x}), \tag{3}$$

where

$$\Theta(\mathbf{x}) = \frac{1}{2} \|\mathbf{f}(\mathbf{x})\|^2. \tag{4}$$

As opposite to Newton’s method, many convergence results may be proved for the projected gradient methods, see, e.g., [3, 42] and references therein. On the other hand, the projected gradient method only has a sub-linear convergence rate and thus results to be slower than the Newton’s algorithm also when properly designed strategies for selecting the step size are used [1, 11, 12, 33].

Motivated by this consideration, some recent works have proposed to combine the two approaches [9, 13, 22, 36]. Along this lines we present the nonlinearly projected combined (NLPC) method that is summarized in Algorithm 1. The main ideas behind NLPC are two. First of all, we replace the classical projector with a novel operator \mathcal{P} , introduced in the following definition, that ensures the constraint $\mathbf{x} \in \Omega$ to be respected while lowering the probability that the points defined at each iteration reach the boundary of Ω .

Definition 2.1 Given $\Omega = \times_{i=1}^n \Omega_i$, $\Omega_i \subseteq \mathbb{R}$ convex for all $i \in \{1, \dots, n\}$, and given $\mathbf{x} = (x_1, \dots, x_n)^\top \in \Omega$, we define the operator $\mathcal{P}(\cdot; \mathbf{x}) : \mathbb{R}^n \rightarrow \Omega$ so that, for all $\mathbf{z} \in \mathbb{R}^n$, $\mathcal{P}(\mathbf{z}; \mathbf{x}) = (p_1(z_1; x_1), \dots, p_n(z_n; x_n))^\top$ where

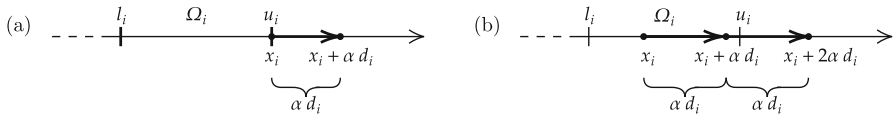


Fig. 1 **a** Example where $i \in \mathcal{B}(\mathbf{x}, \mathbf{d})$. **b** Example where $i \in \mathcal{M}_\alpha(\mathbf{x}, \mathbf{d})$ and $i \in \mathcal{N}_{2\alpha}(\mathbf{x}, \mathbf{d})$

$$p_i(v; w) = \begin{cases} v & \text{if } v \in \Omega_i \\ w & \text{if } v \notin \Omega_i \end{cases}$$

with $v \in \mathbb{R}$ and $w \in \Omega_i \subset \mathbb{R}$.

The second idea behind NLPC method was inspired by [9] and consists in trying at each iteration a fixed number of step lengths α^j , $j \in \{0, \dots, J\}$, along the Newton’s direction \mathbf{d}_k , where \mathbf{d}_k is defined as the solution of the set of equations $\mathbf{J}_f(\mathbf{x}_k)\mathbf{d}_k = -\mathbf{f}(\mathbf{x}_k)$, being $\mathbf{J}_f(\mathbf{x}_k)$ the Jacobian matrix of \mathbf{f} evaluated in \mathbf{x}_k . If none of the tested step sizes satisfies the Armijo Rule

$$\|\mathbf{f}(\mathcal{P}(\mathbf{x}_k + \alpha^j \mathbf{d}_k; \mathbf{x}_k))\| \leq \sqrt{1 - \alpha^j \sigma_N} \|\mathbf{f}(\mathbf{x}_k)\|, \tag{5}$$

we then move along the gradient descent direction with a step size chosen so as to satisfy two conditions that, as we shall prove in Theorem 3.5, guarantee a convergence result for NLPC algorithm.

3 Convergence Properties of the NLPC Method

Now we present a convergence analysis of NLPC algorithm after describing the main tools exploited in the algorithm and the main properties of \mathcal{P} .

Definition 3.1 Given $\mathbf{x} \in \Omega$, $\mathbf{d} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ and $\alpha > 0$, we define

$$\mathcal{B}(\mathbf{x}, \mathbf{d}) := \{i \in \{1, \dots, n\} \text{ s.t. } x_i + \alpha d_i \notin \Omega_i \forall \alpha > 0\} \tag{6}$$

$$\mathcal{M}_\alpha(\mathbf{x}, \mathbf{d}) := \{i \in \{1, \dots, n\} \text{ s.t. } x_i + \alpha d_i \in \Omega_i\} \tag{7}$$

$$\mathcal{N}_\alpha(\mathbf{x}, \mathbf{d}) := \{1, \dots, n\} \setminus (\mathcal{B}(\mathbf{x}, \mathbf{d}) \cup \mathcal{M}_\alpha(\mathbf{x}, \mathbf{d})). \tag{8}$$

For the ease of notation, when $\mathbf{d} = -\nabla\Theta(\mathbf{x})$, the set defined in (6), (7) and (8) will be simply denoted as $\mathcal{B}(\mathbf{x})$, $\mathcal{M}_\alpha(\mathbf{x})$ and $\mathcal{N}_\alpha(\mathbf{x})$, respectively.

Remark 3.1 It can be easily shown that, for all $\alpha > 0$,

$$\mathcal{B}(\mathbf{x}, \mathbf{d}) \cup \mathcal{M}_\alpha(\mathbf{x}, \mathbf{d}) \cup \mathcal{N}_\alpha(\mathbf{x}, \mathbf{d}) = \{1, \dots, n\} \tag{9}$$

and the three sets are pairwise disjoint. More in detail, as illustratively depicted in Fig. 1a, the set $\mathcal{B}(\mathbf{x}, \mathbf{d})$ contains all the coordinates i that prevent \mathbf{d} from being a feasible

Algorithm 1: The NLPC algorithm

```

Input :  $\mathbf{x}_0 \in \Omega; \tau \in (0, +\infty); \alpha, \sigma_N, \sigma_G, \rho \in (0, 1); J \in \mathbb{N} \setminus \{0\}$ 
 $FLAG \leftarrow 0; k \leftarrow 0$ 
while  $\|\mathbf{f}(\mathbf{x}_k)\| > \tau$  do
  if  $FLAG=0$  then
    solve  $\mathbf{J}\mathbf{f}(\mathbf{x}_k)\mathbf{d}_k = -\mathbf{f}(\mathbf{x}_k)$ 
     $j \leftarrow 0$ 
    while  $j \leq J$  do
       $\mathbf{x}_{k+1} = \mathcal{P}(\mathbf{x}_k + \alpha^j \mathbf{d}_k; \mathbf{x}_k)$ 
      if  $\|\mathbf{f}(\mathbf{x}_{k+1})\| \leq \sqrt{1 - \alpha^j \sigma_N} \|\mathbf{f}(\mathbf{x}_k)\|$  then
         $j \leftarrow J + 1; FLAG \leftarrow 0; k \leftarrow k + 1$ 
      else
         $j \leftarrow j + 1; FLAG \leftarrow 1$ 
      end
    end
  end
  else
     $\mathbf{d}_k = -\nabla\Theta(\mathbf{x}_k)$ 
     $j \leftarrow 0$ 
    while  $FLAG = 1$  do
       $\mathbf{x}_{k+1} = \mathcal{P}(\mathbf{x}_k + \alpha^j \mathbf{d}_k; \mathbf{x}_k)$ 
      if
         $\Theta(\mathbf{x}_{k+1}) \leq \Theta(\mathbf{x}_k) + \sigma_G \nabla\Theta(\mathbf{x}_k)^T(\mathbf{x}_{k+1} - \mathbf{x}_k)$ 
        and

$$\sqrt{\sum_{i \in \mathcal{M}_{\alpha^j}(\mathbf{x}_k)} (P_i(x_{k,i} + d_{k,i}) - x_{k,i})^2} \geq \rho \sqrt{\sum_{i \in \mathcal{N}_{\alpha^j}(\mathbf{x}_k)} (P_i(x_{k,i} + d_{k,i}) - x_{k,i})^2}$$

        then
           $FLAG \leftarrow 0; k \leftarrow k + 1$ 
        else
           $j \leftarrow j + 1$ 
        end
      end
    end
  end
end

```

direction as moving along the corresponding component d_i violates the constraint in (1). As an example, when $\Omega = \times_{i=1}^n [\ell_i, u_i], \ell_i < u_i$,

$$\mathcal{B}(\mathbf{x}, \mathbf{d}) = \{i \in \{1, \dots, n\} \text{ s.t. } (x_i = \ell_i \wedge d_i < 0) \vee (x_i = u_i \wedge d_i > 0)\}.$$

Instead, fixed a step size $\alpha > 0$, $\mathcal{M}_\alpha(\mathbf{x}, \mathbf{d})$ contains all the components i for which $x_i + \alpha d_i$ still satisfies the constraint of the problem, while $\mathcal{N}_\alpha(\mathbf{x}, \mathbf{d})$ collects the components for which the step size α is too big, but a feasible vector may be found by lowering it, see Fig. 1b.

Proposition 3.1 *Given $\mathbf{x} \in \Omega$ and $\mathbf{d} \in \mathbb{R}^n$ it holds*

(a) *For all $\alpha > 0$*

$$(\mathbf{x} - \mathcal{P}(\mathbf{x} + \alpha \mathbf{d}; \mathbf{x}))^T (\mathbf{x} + \alpha \mathbf{d} - \mathcal{P}(\mathbf{x} + \alpha \mathbf{d}; \mathbf{x})) = 0 \tag{10}$$

and

$$\|\mathcal{P}(\mathbf{x} + \alpha \mathbf{d}; \mathbf{x}) - \mathbf{x}\| = \alpha \sqrt{\sum_{i \in \mathcal{M}_\alpha(\mathbf{x}, \mathbf{d})} d_i^2}; \quad (11)$$

(b) $\mathbf{g} : (0, \infty) \rightarrow \Omega$ s.t. $\mathbf{g}(\alpha) = \mathcal{P}(\mathbf{x} + \alpha \mathbf{d}; \mathbf{x})$ is continuous in 0;

(c) $\varphi : (0, \infty) \rightarrow \mathbb{R}$ s.t. $\varphi(\alpha) = \frac{\|\mathcal{P}(\mathbf{x} + \alpha \mathbf{d}; \mathbf{x}) - \mathbf{x}\|}{\alpha}$ is monotonically nonincreasing.

Proof (a) Equations (10) and (11) follow from Definition 2.1 which implies

$$\begin{aligned} & (\mathbf{x} - \mathcal{P}(\mathbf{x} + \alpha \mathbf{d}; \mathbf{x}))^T (\mathbf{x} + \alpha \mathbf{d} - \mathcal{P}(\mathbf{x} + \alpha \mathbf{d}; \mathbf{x})) \\ &= \sum_{i=1}^n (x_i - p_i(x_i + \alpha d_i; x_i)) (x_i + \alpha d_i - p_i(x_i + \alpha d_i; x_i)) = 0 \end{aligned}$$

and

$$\|\mathcal{P}(\mathbf{x} + \alpha \mathbf{d}; \mathbf{x}) - \mathbf{x}\| = \sqrt{\sum_{i=1}^n (p_i(x_i + \alpha d_i; x_i) - x_i)^2} = \alpha \sqrt{\sum_{i \in \mathcal{M}_\alpha(\mathbf{x}, \mathbf{d})} d_i^2}.$$

(b) The result directly follows from Eq. (11). Indeed

$$\|\mathbf{g}(\alpha) - \mathbf{g}(0)\| = \|\mathcal{P}(\mathbf{x} + \alpha \mathbf{d}; \mathbf{x}) - \mathbf{x}\| = \alpha \sqrt{\sum_{i \in \mathcal{M}_\alpha(\mathbf{x}, \mathbf{d})} d_i^2} \leq \alpha \|\mathbf{d}\| \xrightarrow{\alpha \rightarrow 0^+} 0.$$

(c) We observe that Eq. (11) implies $\varphi(\alpha) = \sqrt{\sum_{i \in \mathcal{M}_\alpha(\mathbf{x}, \mathbf{d})} d_i^2}$. Since Ω_i is a convex set, given $0 \leq \alpha_1 \leq \alpha_2$ it holds $\mathcal{M}_{\alpha_2}(\mathbf{x}, \mathbf{d}) \subseteq \mathcal{M}_{\alpha_1}(\mathbf{x}, \mathbf{d})$, and thus,

$$\varphi(\alpha_1) - \varphi(\alpha_2) = \sqrt{\sum_{i \in \mathcal{M}_{\alpha_1}(\mathbf{x}, \mathbf{d})} d_i^2} - \sqrt{\sum_{i \in \mathcal{M}_{\alpha_2}(\mathbf{x}, \mathbf{d})} d_i^2} \geq 0.$$

□

As we shall see in the next theorems, the results shown in Proposition 3.1 allow us to prove convergence properties of the proposed NLPC algorithm similar to those holding when the classical projector on the closed set Ω is employed instead of the operator \mathcal{P} [3, 9].

Theorem 3.1 Given $\Theta : \mathbb{R}^n \rightarrow \mathbb{R}$ a continuously differentiable function on Ω and $\mathbf{x} \in \Omega$, then \mathbf{x} is a stationary point of Θ in Ω iff

$$\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}) = \mathbf{x} \quad \forall \alpha > 0. \quad (12)$$

Proof Let us consider the projector P on the closed convex set Ω , defined in Eq. (2). The following properties hold: (i) $P(\mathbf{z}) = (P_1(z_1), \dots, P_n(z_n))$, being

$$P_i(z_i) = \begin{cases} \ell_i & \text{if } z_i \leq \ell_i \\ z_i & \text{if } \ell_i < z_i < u_i \\ u_i & \text{if } z_i \geq u_i \end{cases},$$

where we denoted $\Omega_i = [\ell_i, u_i]$ with $\ell_i, u_i \in \mathbb{R} \cup \{\pm\infty\}$; and (ii) \mathbf{x} is a stationary point iff $P(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x})) = \mathbf{x} \forall \alpha > 0$ [3].

We now assume that condition (12) holds, and thus, $\forall i \in \{1, \dots, n\}$,

$$p_i(x_i - \alpha \partial_i \Theta(\mathbf{x}); x_i) = x_i \quad \forall \alpha > 0.$$

For each $i \in \{1, \dots, n\}$, we then have only three possibilities:

- $x_i \in (\ell_i, u_i)$ and $\partial_i \Theta(\mathbf{x}) = 0$. Then $P_i(x_i - \alpha \partial_i \Theta(\mathbf{x})) = P(x_i) = x_i \forall \alpha > 0$;
- $x_i = \ell_i$ and $\partial_i \Theta(\mathbf{x}) \geq 0$. In this case, $P_i(x_i - \alpha \partial_i \Theta(\mathbf{x})) = \ell_i = x_i \forall \alpha > 0$;
- $x_i = u_i$ and $\partial_i \Theta(\mathbf{x}) \leq 0$. In this case, $P_i(x_i - \alpha \partial_i \Theta(\mathbf{x})) = u_i = x_i \forall \alpha > 0$.

In all three cases, we obtained $P_i(x_i - \alpha \partial_i \Theta(\mathbf{x})) = x_i \forall \alpha > 0$ which implies that $\mathbf{x} \in \Omega$ is a stationary point on Ω .

Conversely, consider a stationary point $\mathbf{x} \in \Omega$ and let us assume it exists $\alpha > 0$ such that $\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}) \neq \mathbf{x}$. From Proposition 3.1 (a) it follows

$$\begin{aligned} 0 &= (\mathbf{x} - \mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}))^T (\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}) - \mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x})) \\ &= \|\mathbf{x} - \mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x})\|^2 - \alpha \nabla \Theta(\mathbf{x})^T (\mathbf{x} - \mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x})), \end{aligned}$$

and thus,

$$\nabla \Theta(\mathbf{x})^T (\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}) - \mathbf{x}) = -\frac{\|\mathbf{x} - \mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x})\|^2}{\alpha} < 0. \tag{13}$$

Equation (13) contradicts the assumption of \mathbf{x} being a stationary point that would imply $\nabla \Theta(\mathbf{x})^T (\mathbf{z} - \mathbf{x}) \geq 0 \forall \mathbf{z} \in \Omega$ [3]. □

Theorem 3.2 Given $\Theta : \mathbb{R}^n \rightarrow \mathbb{R}$ a continuously differentiable function on Ω and $\mathbf{x} \in \Omega$ that is not a stationary point of Θ , then it exists $\alpha^* > 0$ such that $\forall \alpha \in (0, \alpha^*]$ $(\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}) - \mathbf{x})$ is a descent direction for Θ .

Proof Since \mathbf{x} is not a stationary point, according to Theorem 3.1 it exists $\alpha^* > 0$ such that $\mathcal{P}(\mathbf{x} - \alpha^* \nabla \Theta(\mathbf{x}); \mathbf{x}) \neq \mathbf{x}$, and thus, $\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}) \neq \mathbf{x} \forall \alpha \in (0, \alpha^*]$ because Ω_i is a convex set $\forall i \in \{1, \dots, n\}$. Therefore, from (13) it follows $\nabla \Theta(\mathbf{x})^T (\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}) - \mathbf{x}) < 0$. □

Theorem 3.3 Given $\Theta : \mathbb{R}^n \rightarrow \mathbb{R}$ a continuously differentiable function on Ω , $\mathbf{x} \in \Omega$ and $\sigma_G \in (0, 1)$, it exists $\bar{\alpha} > 0$ so that for all $\alpha \in (0, \bar{\alpha}]$

$$\Theta(\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x})) \leq \Theta(\mathbf{x}) + \sigma_G \nabla \Theta(\mathbf{x})^T (\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}) - \mathbf{x}). \tag{14}$$

Proof If $\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}) = \mathbf{x}$ for all $\alpha > 0$, then the thesis holds for any $\bar{\alpha} > 0$. Therefore, we can assume it exists $\tilde{\alpha} \in (0, 1)$ such that $\mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x}) \neq \mathbf{x}$ for all $\alpha \in (0, \tilde{\alpha}]$. In the following, we shall denote $\mathbf{x}(\alpha) := \mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x})$. By the mean value theorem, it exists ξ_α on the segment between \mathbf{x} and $\mathbf{x}(\alpha)$ so that

$$\begin{aligned} \Theta(\mathbf{x}(\alpha)) - \Theta(\mathbf{x}) &= \nabla \Theta(\xi_\alpha)^\top (\mathbf{x}(\alpha) - \mathbf{x}) \\ &= \sigma_G \nabla \Theta(\mathbf{x})^\top (\mathbf{x}(\alpha) - \mathbf{x}) - (\sigma_G - 1) \nabla \Theta(\mathbf{x})^\top (\mathbf{x}(\alpha) - \mathbf{x}) \\ &\quad + (\nabla \Theta(\xi_\alpha) - \nabla \Theta(\mathbf{x}))^\top (\mathbf{x}(\alpha) - \mathbf{x}), \end{aligned}$$

and thus, the inequality (14) can be rewritten as

$$(\nabla \Theta(\xi_\alpha) - \nabla \Theta(\mathbf{x}))^\top (\mathbf{x}(\alpha) - \mathbf{x}) \leq (\sigma_G - 1) \nabla \Theta(\mathbf{x})^\top (\mathbf{x}(\alpha) - \mathbf{x}).$$

Since $\sigma_G < 1$, from Proposition 3.1 (c) it follows that

$$\begin{aligned} (\sigma_G - 1) \nabla \Theta(\mathbf{x})^\top (\mathbf{x}(\alpha) - \mathbf{x}) &= (1 - \sigma_G) \frac{\|\mathbf{x}(\alpha) - \mathbf{x}\|^2}{\alpha} \\ &\geq (1 - \sigma_G) \frac{\|\mathbf{x}(\tilde{\alpha}) - \mathbf{x}\|}{\tilde{\alpha}} \|\mathbf{x}(\alpha) - \mathbf{x}\| > 0. \end{aligned}$$

The theorem is proved if we show that it exists $\bar{\alpha} \in (0, \tilde{\alpha}]$ such that

$$(\nabla \Theta(\xi_\alpha) - \nabla \Theta(\mathbf{x}))^\top (\mathbf{x}(\alpha) - \mathbf{x}) \leq (1 - \sigma_G) \frac{\|\mathbf{x}(\tilde{\alpha}) - \mathbf{x}\|}{\tilde{\alpha}} \|\mathbf{x}(\alpha) - \mathbf{x}\|,$$

for all $\alpha \in [0, \bar{\alpha}]$. This follows from the fact that

$$\lim_{\alpha \rightarrow 0} \left| \frac{(\nabla \Theta(\xi_\alpha) - \nabla \Theta(\mathbf{x}))^\top (\mathbf{x} - \mathbf{x}(\alpha))}{\|\mathbf{x} - \mathbf{x}(\alpha)\|} \right| \leq \lim_{\alpha \rightarrow 0} \|\nabla \Theta(\xi_\alpha) - \nabla \Theta(\mathbf{x})\| = 0,$$

where the last equality is a consequence of Proposition 3.1 (b) and of the regularity assumptions on Θ . □

Theorem 3.4 *Given $\Theta : \mathbb{R}^n \rightarrow \mathbb{R}$, a continuously differentiable function on Ω , $\mathbf{x} \in \Omega$, and $\rho \in (0, 1]$, it exists $\bar{\alpha} > 0$ so that, for all $\alpha \in (0, \bar{\alpha}]$, $\mathcal{N}_\alpha(\mathbf{x}) = \emptyset$, and thus,*

$$\sqrt{\sum_{i \in \mathcal{M}_\alpha(\mathbf{x})} (P_i(x_i - \partial_i \Theta(\mathbf{x})) - x_i)^2} \geq \rho \sqrt{\sum_{i \in \mathcal{N}_\alpha(\mathbf{x})} (P_i(x_i - \partial_i \Theta(\mathbf{x})) - x_i)^2}. \tag{15}$$

Proof For all $i \in \{1, \dots, n\} \setminus \mathcal{B}(\mathbf{x})$, we only have three possibilities:

- $x_i \in \mathring{\Omega}_i$, where $\mathring{\Omega}_i$ denotes the interior of Ω_i . Then, since $\mathring{\Omega}_i$ is an open set, $\exists \bar{\alpha}_i > 0$ such that $x_i - \alpha \partial_i \Theta(\mathbf{x}) \in \mathring{\Omega}_i \subseteq \Omega_i \forall \alpha \leq \bar{\alpha}_i$;
- $x_i = \ell_i$ and $\partial_i \Theta(\mathbf{x}) < 0$. Then $x_i - \alpha \partial_i \Theta(\mathbf{x}) \in \Omega_i \forall \alpha \leq \bar{\alpha}_i := -\frac{u_i - \ell_i}{\partial_i \Theta(\mathbf{x})}$;
- $x_i = u_i$ and $\partial_i \Theta(\mathbf{x}) > 0$. Then $x_i - \alpha \partial_i \Theta(\mathbf{x}) \in \Omega_i \forall \alpha \leq \bar{\alpha}_i := \frac{u_i - \ell_i}{\partial_i \Theta(\mathbf{x})}$.

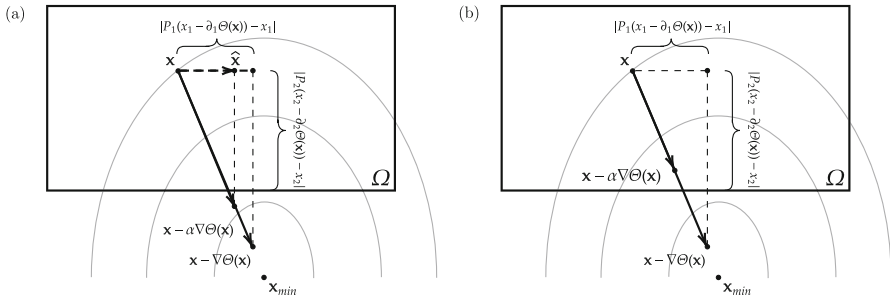


Fig. 2 Illustration of the benefit of the additional condition (15). In **a**, only the first component of \mathbf{x} is updated as $1 \in \mathcal{M}_\alpha(\mathbf{x})$ and $2 \in \mathcal{N}_\alpha(\mathbf{x})$. In this scenario, NLPC may get stuck in a point which is not stationary because the chosen step size is too big and the second component never updated. As shown in **b**, inequality (15) prevents this issue by promoting the choice of a smaller step size so that a higher number of components is updated. Here, $\hat{\mathbf{x}} = \mathcal{P}(\mathbf{x} - \alpha \nabla \Theta(\mathbf{x}); \mathbf{x})$, \mathbf{x}_{\min} is a stationary point of Θ , and $\rho = 1$

Therefore, for all $i \in \{1, \dots, n\} \setminus \mathcal{B}(\mathbf{x})$ it exists $\bar{\alpha}_i > 0$ such that $i \in \mathcal{M}_\alpha(\mathbf{x}) \forall \alpha \in (0, \bar{\alpha}_i]$. By choosing $\bar{\alpha} = \min_{i \in \{1, \dots, n\} \setminus \mathcal{B}(\mathbf{x})} \bar{\alpha}_i$, it follows that, for all $\alpha \leq \bar{\alpha}$, $\mathcal{N}_\alpha(\mathbf{x}) = \emptyset$, and thus,

$$\sqrt{\sum_{i \in \mathcal{M}_\alpha(\mathbf{x})} (P_i(x_i - \partial_i \Theta(\mathbf{x})) - x_i)^2} \geq 0 = \rho \sqrt{\sum_{i \in \mathcal{N}_\alpha(\mathbf{x})} (P_i(x_i - \partial_i \Theta(\mathbf{x})) - x_i)^2}. \tag{16}$$

Hence, the theorem is proved. □

Remark 3.2 The previous theorems hold in particular if Θ is defined as in equation (4). Specifically, inequality (14) is the classical Armijo rule along the projection arc where we employed the operator introduced in Definition 2.1. Inequality (15) is an additional condition that prevents NLPC from choosing a too large step size that would result in an actual update of only few components. An illustrative example can be seen in Fig. 2.

Theorems 3.3 and 3.4 together guarantee that the step size within the gradient descent step of the NLPC algorithm is well defined.

Henceforth, $\{\mathbf{x}_k\}_{k \in \mathbb{N}} \subseteq \Omega$ and $\{\alpha^{j_k}\}_{k \in \mathbb{N}}$ shall denote a sequence of points generated with the NLPC algorithm described in Algorithm 1, and the corresponding step sizes, respectively. In particular, $\alpha \in (0, 1)$, while j_k is a suitable exponent whose value belongs to a different range depending on whether the Newton’s or the gradient descent approach has been used at the k th iteration.

Lemma 3.1 *Let $\{\mathbf{x}_k\}_{k \in \mathbb{N}}$ be a sequence generated with the NLPC algorithm. For each $k \in \mathbb{N}$*

- (a) *if \mathbf{x}_{k+1} has been obtained with a projected gradient descent step, then $\Theta(\mathbf{x}_{k+1}) \leq \Theta(\mathbf{x}_k)$;*

(b) if \mathbf{x}_{k+1} has been obtained with a projected Newton's step, then

$$\Theta(\mathbf{x}_{k+1}) \leq \left(1 - \alpha^J \sigma_N\right)^{n_k+1} \Theta(\mathbf{x}_0),$$

being n_k the number of projected Newton's steps performed until iteration k ;

(c) $0 \leq \Theta(\mathbf{x}_{k+1}) \leq \Theta(\mathbf{x}_k)$, that is, $\{\Theta(\mathbf{x}_k)\}_{k \in \mathbb{N}}$ is bounded below by zero and not increasing.

Proof (a) By the Armijo rule along the projected arc in Eq. (14), we have

$$\Theta(\mathbf{x}_{k+1}) \leq \Theta(\mathbf{x}_k) + \sigma_G \nabla \Theta(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k) \leq \Theta(\mathbf{x}_k),$$

where the last inequality follows from Theorem 3.2. Thus, we have the thesis.

(b) If \mathbf{x}_{k+1} is defined with the projected Newton's method, then it exists $j \in \{0, \dots, J\}$ such that

$$\Theta(\mathbf{x}_{k+1}) \leq (1 - \alpha^j \sigma_N) \Theta(\mathbf{x}_k) \leq (1 - \alpha^J \sigma_N) \Theta(\mathbf{x}_k), \quad (17)$$

where in last inequality we exploited the fact that $\alpha < 1$.

The thesis follows by iteratively applying (17) for each projected Newton's step and the result in point (a) for each projected gradient descent step.

Point (c) follows straightforwardly from (a) and Eq. (17), since both α^J and σ_N belong to the interval $(0, 1)$.

□

Theorem 3.5 Let $\{\mathbf{x}_k\}_{k \in \mathbb{N}}$ be a sequence generated with the NLPC algorithm, and let \mathbf{x}^* be an accumulation point of $\{\mathbf{x}_k\}_{k \in \mathbb{N}}$; then, \mathbf{x}^* is a stationary point of Θ in Ω . Additionally if the projected Newton's method has been used for infinitely many k , then \mathbf{x}^* is a solution of (1).

Proof Since \mathbf{x}^* is an accumulation point of $\{\mathbf{x}_k\}_{k \in \mathbb{N}}$, there exists a subsequence $\{\mathbf{x}_k\}_{k \in K} \subseteq \{\mathbf{x}_k\}_{k \in \mathbb{N}}$, $K \subseteq \mathbb{N}$, such that $\lim_{k \in K \rightarrow \infty} \mathbf{x}_k = \mathbf{x}^*$, and thus, $\lim_{k \in K \rightarrow \infty} \Theta(\mathbf{x}_k) = \Theta(\mathbf{x}^*)$. For all $k \in K$, from Lemma 3.1 it follows

$$\Theta(\mathbf{x}_k) \leq \left(1 - \alpha^J \sigma_N\right)^{n_k} \Theta(\mathbf{x}_0).$$

If the projected Newton's method has been used for infinitely many k , then $\lim_{k \in K \rightarrow \infty} n_k = +\infty$; therefore,

$$\Theta(\mathbf{x}^*) = \lim_{k \in K \rightarrow \infty} \Theta(\mathbf{x}_k) \leq \lim_{k \in K \rightarrow \infty} \left(1 - \alpha^J \sigma_N\right)^{n_k} \Theta(\mathbf{x}_0) = 0.$$

Hence, $\Theta(\mathbf{x}^*) = 0$, that is, \mathbf{x}^* solves (1) and is a stationary point of Θ .

Instead, if the projected gradient direction has been used for all but finitely many iterations, then it exists $\bar{k} \in \mathbb{N}$ so that \mathbf{x}_{k+1} has been obtained through a gradient descent

step $\forall k \geq \bar{k}$. From Lemma 3.1 (c), it follows that $\{\Theta(\mathbf{x}_k)\}_{k \geq \bar{k}}$ is not increasing and bounded below by zero. Hence, it converges and

$$\lim_{k \rightarrow \infty} (\Theta(\mathbf{x}_{k+1}) - \Theta(\mathbf{x}_k)) = 0.$$

Henceforth, we shall denote with $\{\alpha^{j_k}\}_{k \in \mathbb{N}}$ the sequence of step sizes used within NLPC. From (14), (13) and (11), it holds

$$\begin{aligned} \Theta(\mathbf{x}_{k+1}) - \Theta(\mathbf{x}_k) &\leq \sigma_G \nabla \Theta(\mathbf{x}_k)^\top \left(\mathcal{P}(\mathbf{x}_k - \alpha^{j_k} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) - \mathbf{x}_k \right) \\ &= -\sigma_G \frac{\|\mathcal{P}(\mathbf{x}_k - \alpha^{j_k} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) - \mathbf{x}_k\|^2}{\alpha^{j_k}} \\ &= -\sigma_G \alpha^{j_k} \sum_{i \in \mathcal{M}_{\alpha^{j_k}}(\mathbf{x}_k)} (\partial_i \Theta(\mathbf{x}_k))^2 \leq 0, \end{aligned}$$

and thus,

$$\lim_{k(\in K) \rightarrow \infty} \alpha^{j_k} \sum_{i \in \mathcal{M}_{\alpha^{j_k}}(\mathbf{x}_k)} (\partial_i \Theta(\mathbf{x}_k))^2 = 0. \tag{18}$$

Two cases exist: $\liminf_{k(\in K) \rightarrow \infty} \alpha^{j_k} > 0$ (case 1) and $\liminf_{k(\in K) \rightarrow \infty} \alpha^{j_k} = 0$ (case 2).

If case 1 holds, then Eq. (18) implies

$$\begin{aligned} 0 &= \lim_{k(\in K) \rightarrow \infty} \sum_{i \in \mathcal{M}_{\alpha^{j_k}}(\mathbf{x}_k)} (\partial_i \Theta(\mathbf{x}_k))^2 \\ &\geq \lim_{k(\in K) \rightarrow \infty} \sum_{i \in \mathcal{M}_{\alpha^{j_k}}(\mathbf{x}_k)} (P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i})^2 \\ &\geq \rho^2 \lim_{k(\in K) \rightarrow \infty} \sum_{i \in \mathcal{N}_{\alpha^{j_k}}(\mathbf{x}_k)} (P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i})^2, \end{aligned}$$

where the last inequality comes from the constraint described by (15). Hence, in particular

$$\begin{aligned} \lim_{k(\in K) \rightarrow \infty} \sum_{i \in \mathcal{M}_{\alpha^{j_k}}(\mathbf{x}_k)} (P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i})^2 &= \\ \lim_{k(\in K) \rightarrow \infty} \sum_{i \in \mathcal{N}_{\alpha^{j_k}}(\mathbf{x}_k)} (P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i})^2 &= 0. \end{aligned}$$

Since additionally $(P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i}) = 0$ for all $i \in \mathcal{B}(\mathbf{x}_k)$, from Eq. (9), from the continuity of $\nabla \Theta$ and of the classical projector, and being \mathbf{x}^* the limit point

of $\{\mathbf{x}_k\}_{k \in K}$, it follows

$$\|P(\mathbf{x}^* - \nabla \Theta(\mathbf{x}^*)) - \mathbf{x}^*\|^2 = \lim_{k(\in K') \rightarrow \infty} \sum_{i=1}^n (P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i})^2 = 0.$$

Hence, \mathbf{x}^* is a stationary point.

On the other hand, *case 2* implies it exists an infinite set $K' \subset K$ such that $\lim_{k(\in K') \rightarrow \infty} \alpha^{jk} = 0$, and thus, $\lim_{k(\in K') \rightarrow \infty} \alpha^{jk-1} = 0$. Therefore, by defining $J = \{i \in \{1, \dots, n\} \text{ s.t. } i \notin \mathcal{B}(\mathbf{x}^*) \wedge |\partial_i \Theta(\mathbf{x}^*)| > 0\}$, the following holds.

(i) \mathbf{x}^* is a stationary point iff $J = \emptyset$. More specifically, $|P_i(x_i^* - \partial_i \Theta(\mathbf{x}^*)) - x_i^*| = 0$ iff $i \notin J$.

(ii) It exists \bar{k} such that $\forall k \in K', k \geq \bar{k}, J \subseteq \mathcal{M}_{\alpha^{jk-1}}(\mathbf{x}_k)$.

Indeed, let us consider $i \in J$. Since in particular $i \notin \mathcal{B}(\mathbf{x}^*)$ and $\alpha < 1$, from Theorem 3.4 it follows that it exists $\bar{j} \in \mathbb{N}$ such that $\mathcal{N}_{\alpha^j}(\mathbf{x}^*) = \emptyset$ and $i \in \mathcal{M}_{\alpha^j}(\mathbf{x}^*) \forall j \geq \bar{j}$. It can be easily shown that, being \mathbf{x}^* the limit point of $\{\mathbf{x}_k\}_{k \in K}$, this implies it exists k_i such that $\forall k \in K'$ with $k \geq k_i, i \in \mathcal{M}_{\alpha^j}(\mathbf{x}_k) \forall j > \bar{j}$. Additionally, since $\lim_{k(\in K') \rightarrow \infty} \alpha^{jk-1} = 0$, it exists $k'_i \geq k_i$ such that $\forall k \in K', k \geq k'_i, \alpha^{jk-1} <$

$\alpha^{\bar{j}}$. Hence, the thesis follows by considering $\bar{k} = \max_{i \in \{1, \dots, n\}} \{k'_i\}$.

To prove that \mathbf{x}^* is a stationary point, we proceed by contradiction and we assume that it exists $i \in J$. Then, by the results in (i) and (ii), it follows

$$\begin{aligned} & \lim_{k(\in K') \rightarrow \infty} \sqrt{\sum_{i \in \mathcal{M}_{\alpha^{jk-1}}(\mathbf{x}_k)} |P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i}|^2} \\ & \geq \lim_{k(\in K') \rightarrow \infty} \sqrt{\sum_{i \in J} |P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i}|^2} \\ & = \sqrt{\sum_{i \in J} |P_i(x_i^* - \partial_i \Theta(\mathbf{x}^*)) - x_i^*|^2} > 0, \end{aligned}$$

while, denoted $J^C = \{1, \dots, n\} \setminus J$,

$$\begin{aligned} & \lim_{k(\in K') \rightarrow \infty} \rho \sqrt{\sum_{i \in \mathcal{N}_{\alpha^{jk-1}}(\mathbf{x}_k)} |P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i}|^2} \\ & \leq \rho \lim_{k(\in K') \rightarrow \infty} \sqrt{\sum_{i \in J^C} |P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i}|^2} \\ & = \rho \sqrt{\sum_{i \in J^C} |P_i(x_i^* - \partial_i \Theta(\mathbf{x}^*)) - x_i^*|^2} = 0. \end{aligned}$$

Therefore, for sufficiently large $k \in K'$

$$\begin{aligned} & \sqrt{\sum_{i \in \mathcal{M}_{\alpha^{j_k-1}}(\mathbf{x}_k)} |P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i}|^2} \\ & \geq \rho \sqrt{\sum_{i \in \mathcal{N}_{\alpha^{j_k-1}}(\mathbf{x}_k)} |P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i}|^2}, \end{aligned}$$

i.e., condition (15) is satisfied by the step size α^{j_k-1} which is the last step size tried by NLPC before the chosen one. As a consequence, such a step size cannot satisfy condition (14), i.e.,

$$\begin{aligned} & \Theta(\mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k)) - \Theta(\mathbf{x}_k) \\ & > \sigma_G \nabla \Theta(\mathbf{x}_k)^\top \left(\mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) - \mathbf{x}_k \right). \end{aligned}$$

By the mean value theorem, it exists $\tau \in (0, 1)$ such that, defined $\xi_k = \tau \mathbf{x}_k + (1 - \tau) \mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k)$, then

$$\begin{aligned} & \Theta(\mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k)) - \Theta(\mathbf{x}_k) \\ & = \nabla \Theta(\xi_k)^\top \left(\mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) - \mathbf{x}_k \right) \\ & = (\nabla \Theta(\xi_k) - \nabla \Theta(\mathbf{x}_k))^\top \left(\mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) - \mathbf{x}_k \right) \\ & \quad + \nabla \Theta(\mathbf{x}_k)^\top \left(\mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) - \mathbf{x}_k \right). \end{aligned}$$

Together with the previous result, this implies

$$\begin{aligned} & (1 - \sigma_G) \nabla \Theta(\mathbf{x}_k)^\top \left(\mathbf{x}_k - \mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) \right) \\ & < (\nabla \Theta(\xi_k) - \nabla \Theta(\mathbf{x}_k))^\top \left(\mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) - \mathbf{x}_k \right) \\ & \leq \|\nabla \Theta(\xi_k) - \nabla \Theta(\mathbf{x}_k)\| \cdot \|\mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) - \mathbf{x}_k\|; \end{aligned}$$

hence,

$$\begin{aligned} \frac{1}{1 - \sigma_G} \|\nabla \Theta(\xi_k) - \nabla \Theta(\mathbf{x}_k)\| & > \frac{\nabla \Theta(\mathbf{x}_k)^\top \left(\mathbf{x}_k - \mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) \right)}{\|\mathcal{P}(\mathbf{x}_k - \alpha^{j_k-1} \nabla \Theta(\mathbf{x}_k); \mathbf{x}_k) - \mathbf{x}_k\|} \\ & = \sqrt{\sum_{i \in \mathcal{M}_{\alpha^{j_k-1}}(\mathbf{x}_k)} (\partial_i \Theta(\mathbf{x}_k))^2}, \end{aligned}$$

where the last equality comes from Eqs. (13) and (11). Therefore, from the properties of the classical projector and from the result previously shown in (ii), it follows

$$\begin{aligned} 0 &= \lim_{k \in K' \rightarrow \infty} \sqrt{\sum_{i \in \mathcal{M}_{\alpha}^{j_k-1}(\mathbf{x}_k)} (\partial_i \Theta(\mathbf{x}_k))^2} \\ &\geq \lim_{k \in K' \rightarrow \infty} \sqrt{\sum_{i \in \mathcal{M}_{\alpha}^{j_k-1}(\mathbf{x}_k)} |P_i(x_{k,i} - \partial_i \Theta(\mathbf{x}_k)) - x_{k,i}|^2} \\ &\geq \sqrt{\sum_{i \in J} |P_i(x_i^* - \partial_i \Theta(\mathbf{x}^*)) - x_i^*|^2}, \end{aligned}$$

which is possible only if $J = \emptyset$ and this contradicts the hypothesis. \square

Finally, we remark that all the results proved in this section can be easily extended to the case where the gradient direction is scaled for example by normalizing it [43] or by exploiting Barzilai–Borwein’s step lengths [1].

4 Application to Chemical Reaction Networks

Let us consider a chemical reaction network (CRN) composed of r chemical reactions involving n well-mixed proteins. Specifically, in this work we will focus on the CRN devised for modeling cell signaling during the G1/S transition phase in colorectal cells described in [39, 40] and henceforth denoted as CR-CRN. In this case, $n = 419$ and $r = 851$.

By assuming that the law of mass action holds [38, 44], the dynamics of the CRN gives rise to a set of n ordinary differential equations (ODEs)

$$\dot{\mathbf{x}} = \mathbf{S}\mathbf{v}(\mathbf{x}, \mathbf{k}), \quad (19)$$

where the state vector $\mathbf{x} \in \mathbb{R}_+^n$ contains the protein molecular concentrations (nM); the superposed dot denotes the time derivative; \mathbf{S} is the constant stoichiometric matrix of size $n \times r$; $\mathbf{k} \in \mathbb{R}_+^r$ is the vector of the rate constants of the reactions; and $\mathbf{v}(\mathbf{x}, \mathbf{k}) \in \mathbb{R}_+^r$ is the time-variant vector of the reaction fluxes. Specifically, from the law of mass action it follows [28]

$$\mathbf{v}(\mathbf{x}, \mathbf{k}) = \text{diag}(\mathbf{k})\mathbf{z}(\mathbf{x}), \quad (20)$$

where the elements of $\mathbf{z}(\mathbf{x})$ are monomials of the form $z_j(\mathbf{x}) = \prod_{i=1}^n x_i^{p_{ij}}$, $\forall j = 1, \dots, r$. In the CR-CRN, $p_{ij} \in \{0, 1, 2\}$, because all the reactions involve up to two reactants.

Given a solution $\mathbf{x}(t)$ of system (19), a semi-positive conservation vector is a constant vector $\boldsymbol{\gamma} \in \mathbb{N}^n \setminus \{\mathbf{0}\}$ for which it exists $c \in \mathbb{R}_+$ so that $\boldsymbol{\gamma}^T \mathbf{x}(t) = c \forall t > 0$ [37, 38]. Conservation vectors can be determined by studying the kernel of \mathbf{S}^T [32].

In the remaining of the paper, we shall assume that the considered CRN satisfies the following properties in terms of its conservation vectors.

- (i) The CRN is *weakly elemented* [38], i.e., it exists a set of independent generators $\{\gamma_1, \dots, \gamma_p\} \subset \mathbb{N}^n \setminus \{\mathbf{0}\}$ of the semi-positive conservation vectors such that $p = n - \text{rank}(\mathbf{S})$ and, up to a change of the proteins order,

$$\mathbf{N} := \begin{bmatrix} \gamma_1^T \\ \vdots \\ \gamma_p^T \end{bmatrix} = [\mathbf{I}_p, \mathbf{N}_2] \tag{21}$$

being \mathbf{I}_p the identity matrix of size $p \times p$.

- (ii) The CRN satisfies the *global stability condition* [38], i.e., for each $\mathbf{c} \in \mathbb{R}_+^p$ it exists a unique asymptotically stable state on the stoichiometric compatibility class (SCC) $\{\mathbf{x} \in \mathbb{R}_+^n \text{ s.t. } \mathbf{N}\mathbf{x} = \mathbf{c}\}$. Fixed a SCC, the corresponding asymptotically stable state $\mathbf{x}_e \in \mathbb{R}_+^n$ solves the system

$$\begin{aligned} \mathbf{S}\mathbf{v}(\mathbf{x}, \mathbf{k}) &= 0 \\ \mathbf{N}\mathbf{x} - \mathbf{c} &= 0. \end{aligned} \tag{22}$$

Lemma 4.1 *For a weakly elemented CRN satisfying the global stability condition, the system in (22) is equivalent to the square system*

$$\begin{aligned} \mathbf{S}_2\mathbf{v}(\mathbf{x}, \mathbf{k}) &= 0 \\ \mathbf{N}\mathbf{x} - \mathbf{c} &= 0, \end{aligned} \tag{23}$$

where \mathbf{S}_2 is a matrix of size $(n - p) \times r$ defined by the last $n - p$ rows of \mathbf{S} .

Proof Obviously, a solution of (22) also solves (23). On the other hand, let \mathbf{x}_e be a solution of (23). The theorem is proved by showing that

$$\mathbf{S}_1\mathbf{v}(\mathbf{x}_e, \mathbf{k}) = 0$$

being \mathbf{S}_1 the matrix of size $p \times r$ defined by the first p rows of \mathbf{S} . To this end, we observe that, since any conservation vector belongs to the kernel of \mathbf{S}^T , it holds

$$\mathbf{0} = \mathbf{N}\mathbf{S} = \mathbf{S}_1 + \mathbf{N}_2\mathbf{S}_2.$$

Therefore,

$$\mathbf{S}_1\mathbf{v}(\mathbf{x}_e, \mathbf{k}) = -\mathbf{N}_2\mathbf{S}_2\mathbf{v}(\mathbf{x}_e, \mathbf{k}) = 0.$$

□

According to Lemma 4.1, in a weakly elemented CRN satisfying the global stability condition, the equilibrium point on a fixed SCC can be computed by solving a box-constrained system as in Eq. (1), being $\Omega = \mathbb{R}_+^n$ and

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \mathbf{S}_2 \mathbf{v}(\mathbf{x}, \mathbf{k}) \\ \mathbf{N} \mathbf{x} - \mathbf{c} \end{bmatrix}. \quad (24)$$

Lemma 4.2 Consider the function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ defined as in Eq. (24). \mathbf{f} is continuously differentiable on \mathbb{R}_+^n and

$$\mathbf{J}_f(\mathbf{x}) = \begin{bmatrix} \mathbf{S}_2 \text{diag}(\mathbf{k}) \mathbf{J}_z(\mathbf{x}) \\ \mathbf{N} \end{bmatrix}, \quad (25)$$

where $[\mathbf{J}_z(\mathbf{x})]_{ji} = p_{ij} x_i^{p_{ij}-1} \prod_{\ell=1, \ell \neq i}^n x_\ell^{p_{\ell j}} \forall i \in \{1, \dots, n\}$ and $j = 1, \dots, r$.

Proof The thesis follows from the definition of the reaction fluxes in Eq. (20). \square

5 Numerical Results on the CR-CRN

5.1 General Consideration

To show the advantages of using NLPC for computing the asymptotically stable states of a CRN, we applied it to the CR-CRN. The parameters describing the network in a physiological state have been extensively described in previous works [38–40] and can be downloaded from the GitHub repository https://github.com/theMIDAGroup/CRC_CRN.git as MATLAB[®] structure. This includes the list of proteins and reactions involved in the network, as well as the values of the rate constants \mathbf{k} and of the total conserved moieties \mathbf{c} . The corresponding stoichiometric matrix \mathbf{S} and reaction fluxes $\mathbf{v}(\mathbf{x}, \mathbf{k})$ can be derived as described in the previous section. The aforementioned repository also contains the MATLAB[®] codes implementing the NLPC algorithm and the analysis shown in this paper.

We exploited the model introduced by Sommariva and colleagues [38, 39] to test the proposed approach under different biologically plausible conditions. Specifically, we modified the values of the parameters \mathbf{k} and \mathbf{c} as described in [39] to simulate the effect of some of the mutations that most commonly arise in colorectal cancer. A total of 9 different mutations was considered (loss of function of APC, AKT, SMAD4, PTEN, p53 and gain of function of k-Ras, Raf, PI3K, Betacatenin) which give rise to as many different mutated networks.

From a practical point of view, if not otherwise specified, the parameters required in input by Algorithm 1 were set as follows. The threshold within the stopping criterion was $\tau = 10^{-12}$, while $\sigma_N = \sigma_G = 10^{-4}$ and $\rho = 10^{-2}$. The initial step size was $\alpha = 0.79$ and a maximum of $J = 20$ step sizes was tested within each iteration of the Newton's method. NLPC is initialized with a point \mathbf{x}_0 randomly drawn from the SCC $\{\mathbf{x} \in \mathbb{R}_+^n \text{ s.t. } \mathbf{N} \mathbf{x} = \mathbf{c}\}$ by exploiting the procedure presented in [38]. Additionally, we

only retained points such that the condition number of the Jacobian matrix $\mathbf{J}_f(\mathbf{x}_0)$ was lower than 10^{17} . To avoid the algorithm getting stuck in a stationary point that is not a zero of \mathbf{f} , we also set a maximum number of allowed iterations: If the stopping criterion was not reached after 250 iterations, then a new initial point was drawn from the SCC and NLPC was restarted. Finally, to speed up NLPC (i) we fixed a maximum number of tested step sizes also within the gradient descent method: if conditions (14) and (15) were not met after 40 possible values of the step length, we chose the last tested value, and at the following NLPC iteration, we performed again a gradient descent step; (ii) at each iteration, we scaled the gradient direction: two scaling procedures were tested, namely we normalized the gradient and we implemented an approach inspired by the Barzilai–Borwein’s (BB) rule thoroughly described in Online Resource 1.

5.2 Comparison with a Classical Dynamic Approach

A classical approach [38, 39] for computing the stationary state of system (19) on a given SCC consists in simulating the whole concentration dynamics $\mathbf{x}(t)$ by solving the Cauchy problem

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Sv}(\mathbf{x}, \mathbf{k}) \\ \mathbf{x}(0) &= \mathbf{x}_0,\end{aligned}\tag{26}$$

where \mathbf{x}_0 is a point on the SCC, and then computing the asymptotic value

$$\mathbf{x}_{\text{dyn}} = \lim_{t \rightarrow +\infty} \mathbf{x}(t).\tag{27}$$

In this section, we compare the results obtained through this approach with those from NLPC algorithm.

To this end, we started from the CR-CRN and we built 10 different experiments, by varying the values of the kinetic parameters \mathbf{k} and of the total conserved moieties \mathbf{c} that define the SCC, so as to mimic a colorectal cell either healthy or affected by one of the 9 mutations listed in Sect. 5.1. For each experiment, we sampled 50 initial points $\mathbf{x}_0^{(j)}$ on the corresponding SCC. For each initial point, i.e., for $j \in \{1, \dots, 50\}$, we computed the solutions $\mathbf{x}_{\text{nlpc}}^{(j)}$ and $\mathbf{x}_{\text{nlpc_bb}}^{(j)}$ provided by the NLPC algorithm when the gradient direction is scaled by its norm or through the BB-inspired scaling rule, respectively. We compared these results with the asymptotically stable state $\mathbf{x}_{\text{dyn}}^{(j)}$ computed through the dynamic approach previously described.

Specifically, as in [38], we used the MATLAB[®] tool `ode15s` [35] to integrate the ODE system in (26) on the interval $[0, 2.5 \cdot 10^7]$ and we defined $\mathbf{x}_{\text{dyn}}^{(j)}$ as the value of the computed solution at the last time point of the interval.

All the other options of the `ode15s` routine were kept fixed to MATLAB[®] default values. The choice of this parameter setting was inspired by previous works [38, 39] where the obtained results were extensively validated. In particular, one of the parameters affecting the quality of the final solution is the span of the time interval. However, some tests summarized in Online Resource 1 suggested that larger intervals

do not noticeably improve the accuracy of the solution while they may introduce numerical errors.

As shown in Figs. 3 and 4, NLPC outperforms the dynamic approach in terms of both accuracy of the obtained results and computational cost. Indeed, Fig. 3 shows that, in all the 10 considered experiments, when the gradient direction is normalized, the elapsed time for the NLPC algorithm averaged across 50 runs obtained by varying the initial points, ranges from about 23 s (mutated network with gain of function of k-Ras) to 59 s (mutated network with loss of function of PTEN). Such values decrease even more when the BB rule is chosen for scaling, going down to a range between 5 and 37 s. Despite this, it is important to underline that in this second case sometimes (in our simulations, with about 5.4% of probability) numerical issues occur compromising the stability of the method and preventing it from returning the desired equilibrium point. On the contrary, the results of the dynamic approach show a higher variability across the different CRNs and the averaged elapsed time scales up to about 8 min in the network incorporating a gain of function mutation of PI3K. It is worth noticing that, for each of the 10 experiments, few runs of NLPC required a higher elapsed time (higher than the third quartile of the corresponding distributions). These runs needed a large number of restarts of the NLPC algorithm due to the fact that the maximum number of 250 iterations was reached without meeting the stopping criterion on the norm of \mathbf{f} , probably because the gradient method tended to stationary points that were not roots of \mathbf{f} . Future work will be devoted to refining the stopping criterion so that, when needed, NLPC is restarted before reaching 250 iterations.

As a final test, we verified whether the parameter setting in the two approaches coincide in terms of solution accuracy. Since we are looking for the roots of \mathbf{f} , the accuracy of the obtained results was evaluated by computing the ℓ_2 -norm of \mathbf{f} in the solutions provided by the two algorithms, namely $\mathbf{x}_{\text{nlpc}}^{(j)}$ and $\mathbf{x}_{\text{dyn}}^{(j)}$, $j \in \{1, \dots, 50\}$. As shown in Fig. 4, for all 10 considered experiments the norm of \mathbf{f} in the NLPC solutions, $\mathbf{x}_{\text{nlpc}}^{(j)}$, was always below 10^{-12} as imposed by the stopping criterion of the algorithm. Instead the value of $\|\mathbf{f}(\mathbf{x}_{\text{dyn}}^{(j)})\|$ ranged between 10^{-2} and 10^1 , regardless of the time employed to compute the solution $\mathbf{x}_{\text{dyn}}^{(j)}$.

5.3 Benefits of the Operator \mathcal{P} over the Classical Projector

The goal of this section is to quantify the benefit of using the operator \mathcal{P} instead of the classical projector P on the closed convex set Ω defined in Eq. (2). To this end, for each of the 10 experiments defined in the previous section, and for each of the 50 initial points $\mathbf{x}_0^{(j)}$, $j \in \{1, \dots, 50\}$, drawn on the corresponding SCCs, we computed the solution of NLPC (with normalized gradient) by replacing in Algorithm 1 the proposed operator \mathcal{P} with the classical projector P . We denoted with $\mathbf{x}_{\text{ort}}^{(j)}$ the corresponding solution.

As shown in Fig. 5, if combined with the classical projector, NLPC algorithm requires a higher number of restarts and thus a higher elapsed time than those required when the proposed operator is used. Specifically, the ratio between the number of restarts required by the projector P and the one required by the operator \mathcal{P} , averaged

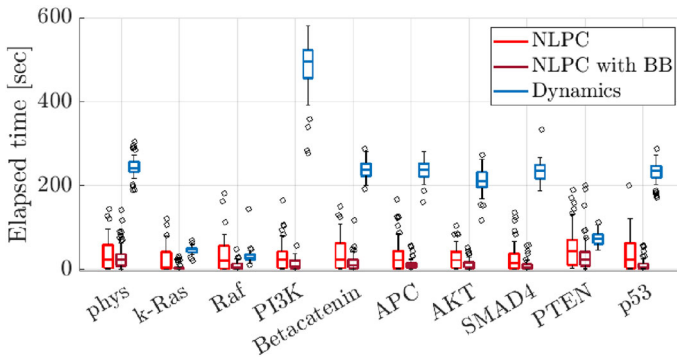


Fig. 3 Elapsed time for the NLPC algorithm to converge, when the gradient direction is normalized (NLPC) or is scaled through a Barzilai–Borwein-inspired rule (NLPC with BB), compared to the time required to compute the equilibrium point by solving the dynamical system in (26). Boxplots summarize the values obtained across 50 different runs for 10 distinct networks mimicking either a physiological state (phys) or a mutation affecting the protein shown in the axis labels

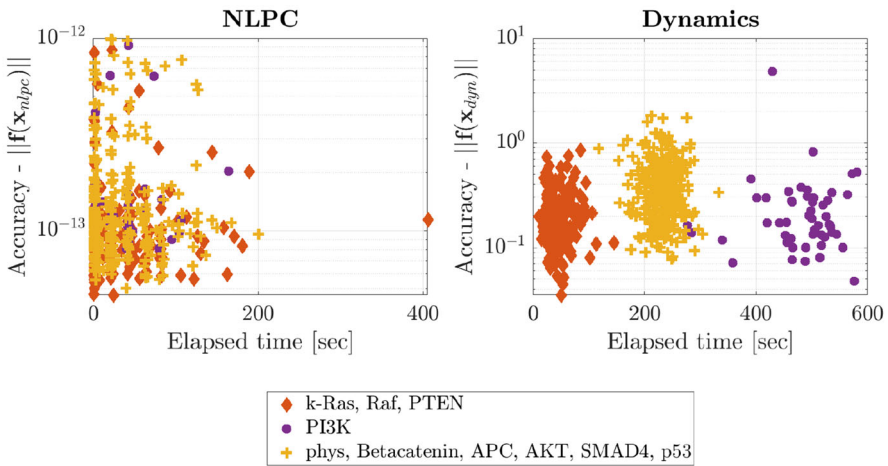


Fig. 4 Accuracy as a function of the elapsed time for the NLPC algorithm (left) and the dynamic approach (right). Accuracy is quantified as the norm of \mathbf{f} evaluated in the results provided by the two algorithms, \mathbf{x}_{nlpc} and \mathbf{x}_{dyn} , respectively. In each panel, 50 different results are shown for each of the considered CRNs that mimic mutations of k-Ras, Raf and PTEN (orange diamonds), physiological state and mutations of Betacatenin, APC, AKT, SMAD4, PTEN, p53 (yellow crosses), and mutation of PI3K (purple dots). This color code has been chosen so as to cluster together results for which the times required for computing \mathbf{x}_{dyn} were similar, as depicted in Fig. 3. Notice the different scale on the y-axis. The analogous plot for the NLPC with BB approach can be found in Online Resource 1

over all the 10 considered experiments and all the sampled initial points, is around 4.87.

The bad performances of the projector P are caused by the fact that at any given iteration k all the negative components of the novel proposed point \mathbf{x}_{k+1} are set equal to zero. As a consequence, the percentage of proteins estimated as having a null concentration increases sharply and this results in a high condition number of the

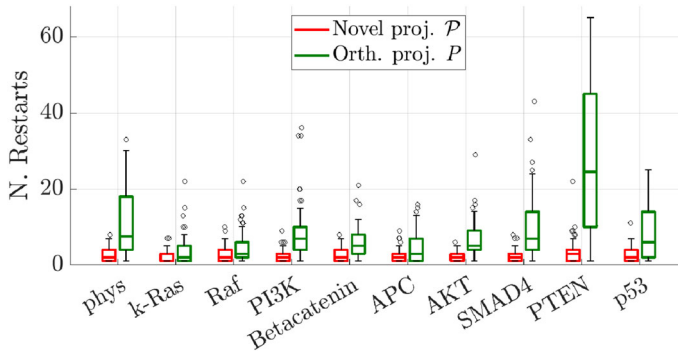


Fig. 5 Number of restarts required by NLPC in order to satisfy the stopping criterion within the fixed maximum number of iterations. The boxplots describe the values obtained across 50 different runs for 10 distinct networks when, within NLPC, we employed the proposed nonlinear projector \mathcal{P} (red) or the classical orthogonal projector P (green)

Table 1 Average and standard deviation over 50 initial points of the maximum number of null components (first row) and the maximum condition number of the Jacobian matrix \mathbf{J}_f (second row) reached across the iterations performed by NLPC

	Novel proj. \mathcal{P}	Orth. proj. P
Num. null components (%)	0.59 ± 0.14	36.99 ± 8.42
Cond. number \mathbf{J}_f (log. scale)	14 ± 2	17 ± 3

Results obtained by using the novel nonlinear projector \mathcal{P} (first column) and the classical orthogonal projector P (second column) are compared. Since results across the 10 considered experiments were similar, only those concerning the original physiological CR-CRN are shown

corresponding Jacobian matrix \mathbf{J}_f defined as in (25). In turn, the ill-conditioning of \mathbf{J}_f compromises the stability of Newton's method, and thus, NLPC algorithm tends to spend most of the allowed iterations by performing gradient descent steps. As shown in Table 1, the use of the operator \mathcal{P} helps preventing this issue.

6 Conclusions

In this paper an iterative algorithm for solving root-finding box-constrained problems is presented. It combines Newton's and gradient descent methods and exploits the operator \mathcal{P} in Definition (2.1) for assuring the required constraints at each iteration (and preventing numerical instability issues that would occur if the projector P was applied). Together with a suitable backtracking rule we prove that the method converges to a stationary point of the objective function in Eq. (4). Despite outperforming the dynamic approach both in accuracy and speed, in CRNs' framework the NLPC algorithm provides less information than simulating the whole concentration dynamics. However, in many contexts such as tuning kinetic parameters starting from experimental data or for topics described in [38, 39] the comprehension of the whole dynamic is not required, but only knowing equilibrium points of the system is of inter-

est. The present work could be extended in different aspects. On the one hand, defining and implementing a stopping criterion in case the algorithm converged to stationary points which do not coincide with roots of \mathbf{f} would be interesting. On the other hand, general properties of the NLPC algorithm should be investigated in extreme scenarios such those of problems where the solution is not unique or does not exist. In our case, this means that the function \mathbf{f} has multiple nonnegative roots or does not have any. Some preliminary results shown in Online Resource 1 suggest that in the former case NLPC is able to find the different roots by changing the initial point, while in the latter case the stopping criterion $\|\mathbf{f}(\mathbf{x}_k)\| > \tau$ is not met and the algorithm may approach local minima of Θ possibly on the boundary of Ω . In this respect, a different future development of this research could go in the direction of coupling the homotopy continuation methods with the proposed numerical tools. Finally, an interesting study would concern a thorough benchmarking with other state-of-the-art methods also in terms of scalability [7, 14, 16, 20, 30]. Some preliminary tests on the comparison with the interior point optimizer (IPOPT, [27, 41]) and the scaled gradient projection (SGP [4–6]) method are presented in Online Resource 1.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s10957-023-02323-z>.

Acknowledgements S.B. was granted a Ph.D. scholarship by Roche S.p.A., Italy. F.B. and S.S. have been partially supported by Gruppo Nazionale per il Calcolo Scientifico (GNCS-INdAM). Alessandro Cartatone is kindly acknowledged for his support with the installation of some of the toolboxes used as benchmark.

Funding Open access funding provided by Università degli Studi di Genova within the CRUI-CARE Agreement. The work by S.B., F.B., and S.S. was carried out within the framework of the project HubLife Science – Digital Health (LSH-DH) PNC-E3-2022-23683267 - Progetto DHEAL-COM – CUP:D33C22001980001, founded by Ministero della Salute within “Piano Nazionale Complementare PNRR Ecosistema Innovativo della Salute - Codice univoco investimento: PNC-E.3”.

Data Availability The datasets and codes generated and analyzed during the current study are available in the GitHub repository, https://github.com/theMIDAGroup/CRC_CRN.git.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. *IMA J. Numer. Anal.* **8**(1), 141–148 (1988)
2. Bertsekas, D.P.: Projected newton methods for optimization problems with simple constraints. *SIAM J. Control Optim.* **20**(2), 221–246 (1982)
3. Bertsekas, D.P.: *Nonlinear Programming*. Athena Scientific, Belmont (1997)
4. Bonettini, S., Loris, I., Porta, F., Prato, M., Rebegoldi, S.: On the convergence of a linesearch based proximal-gradient method for nonconvex optimization. *Inverse Probl.* **33**(5), 055005 (2017)

5. Bonettini, S., Prato, M.: New convergence results for the scaled gradient projection method. *Inverse Probl.* **31**(9), 095008 (2015)
6. Bonettini, S., Zanella, R., Zanni, L.: A scaled gradient projection method for constrained image deblurring. *Inverse Probl.* **25**(1), 015002 (2008)
7. Byrd, R.H., Nocedal, J., Waltz, R.A.: KNITRO: an integrated package for nonlinear optimization. In: *Large-Scale Nonlinear Optimization*, pp. 35–59. Springer (2006)
8. Chellaboina, V., Bhat, S.P., Haddad, W.M., Bernstein, D.S.: Modeling and analysis of mass-action kinetics. *IEEE Control Syst. Mag.* **29**(4), 60–78 (2009)
9. Chen, J., Vuik, C.: Globalization technique for projected Newton–Krylov methods. *Int. J. Numer. Methods Eng.* **110**(7), 661–674 (2017)
10. Conradi, C., Saez-Rodriguez, J., Gilles, E.D., Raisch, J.: Using chemical reaction network theory to discard a kinetic mechanism hypothesis. *IEE Proc. Syst. Biol.* **152**(4), 243–248 (2005)
11. Crisci, S., Ruggiero, V., Zanni, L.: Steplength selection in gradient projection methods for box-constrained quadratic programs. *Appl. Math. Comput.* **356**, 312–327 (2019)
12. Dai, Y.H., Fletcher, R.: Projected Barzilai–Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.* **100**(1), 21–47 (2005)
13. Di Serafino, D., Toraldo, G., Viola, M.: Using gradient directions to get global convergence of Newton-type methods. *Appl. Math. Comput.* **409**(125), 612 (2021)
14. Di Serafino, D., Toraldo, G., Viola, M., Barlow, J.: A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables. *SIAM J. Optim.* **28**(4), 2809–2838 (2018)
15. Dormand, J., Prince, P.: A family of embedded Runge–Kutta formulae. *J. Comput. Appl. Math.* **6**(1), 19–26 (1980)
16. Drud, A.: CONOPT: a GRG code for large sparse dynamic nonlinear optimization problems. *Math. Program.* **31**, 153–191 (1985)
17. Feinberg, M.: Chemical reaction networks structure and the stability of complex isothermal reactors-I. *Chem. Eng. Sci.* **42**(10), 2229–2268 (1987)
18. Feinberg, M.: The existence and uniqueness of steady states for a class of chemical reaction networks. *Arch. Ration. Mech. Anal.* **132**(4), 311–370 (1995)
19. Gábor, A., Hangos, K.M., Banga, J.R., Szederkényi, G.: Reaction network realizations of rational biochemical systems and their structural properties. *J. Math. Chem.* **53**(8), 1657–1686 (2015)
20. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Rev.* **47**(1), 99–131 (2005)
21. Goldstein, A.A.: Convex programming in Hilbert space. *Bull. Am. Math. Soc.* **70**(5), 709–710 (1964)
22. Han, L., Neumann, M.: Combining quasi-Newton and steepest descent directions. *Int. J. Appl. Math.* **12**, 167–171 (2003)
23. Jordan, J.D., Landau, E.M., Iyengar, R.: Signaling networks: the origins of cellular multitasking. *Cell* **103**(2), 193–200 (2000)
24. Khanh, P.N.Q.: Optimality conditions via norm scalarization in vector optimization. *SIAM J. Control Optim.* **31**, 646–658 (1993)
25. Levitin, E.S., Polyak, B.T.: Constrained minimization methods. *USSR Comput. Math. Math. Phys.* **6**(5), 1–50 (1966)
26. Nesterov, Y., Polyak, B.T.: Cubic regularization of newton method and its global performance. *Math. Program.* **108**(1), 177–205 (2006)
27. Nocedal, J., Wächter, A., Waltz, R.A.: Adaptive barrier update strategies for nonlinear interior methods. *SIAM J. Optim.* **19**(4), 1674–1693 (2009)
28. Otero-Muras, I., Yordanov, P., Stelling, J.: Chemical reaction network theory elucidates sources of multistability in interferon signaling. *PLoS Comput. Biol.* **13**(4), e1005454 (2017)
29. Poloni, F.: Quadratic vector equations. *Linear Algebra Appl.* **438**(4), 1627–1644 (2013)
30. Pougkakiotis, S., Gondzio, J.: An interior point-proximal method of multipliers for convex quadratic programming. *Comput. Optim. Appl.* **78**, 307–351 (2021)
31. Reyn, J.W.: *Phase Portraits of Planar Quadratic Systems* (2007)
32. Schuster, S., Höfer, T.: Determining all extreme semi-positive conservation relations in chemical reaction systems: a test criterion for conservativity. *J. Chem. Soc. Faraday Trans.* **87**(16), 2561–2566 (1991)
33. Serafini, T., Zanghirati, G., Zanni, L.: Gradient projection methods for quadratic programs and applications in training support vector machines. *Optim. Methods Softw.* **20**(2–3), 353–378 (2005)

34. Sever, R., Brugge, J.S.: Signal transduction in cancer. Cold Spring Harb. Perspect. Med. **5**(4), a006098 (2015)
35. Shampine, L.F., Reichelt, M.W.: The MATLAB ODE suite. SIAM J. Sci. Comput. **18**(1), 1–22 (1997)
36. Shi, Y.: A globalization procedure for solving nonlinear systems of equations. Numer. Algorithms **12**(2), 273–286 (1996)
37. Shinar, G., Alon, U., Feinberg, M.: Sensitivity and robustness in chemical reaction networks. SIAM J. Appl. Math. **69**(4), 977–998 (2009)
38. Sommariva, S., Caviglia, G., Piana, M.: Gain and loss of function mutations in biological chemical reaction networks: a mathematical model with application to colorectal cancer cells. J. Math. Biol. **82**(6), 1–25 (2021)
39. Sommariva, S., Caviglia, G., Ravera, S., Frassoni, F., Benvenuto, F., Tortolina, L., Castagnino, N., Parodi, S., Piana, M.: Computational quantification of global effects induced by mutations and drugs in signaling networks of colorectal cancer cells. Sci. Rep. **11**(1), 1–13 (2021)
40. Tortolina, L., Duffy, D.J., Maffei, M., Castagnino, N., Carmody, A.M., Kolch, W., Kholodenko, B.N., De Ambrosi, C., Barla, A., Biganzoli, E.M., et al.: Advances in dynamic modeling of colorectal cancer signaling-network regions, a path toward targeted therapies. Oncotarget **6**(7), 5041–5058 (2015)
41. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: motivation and global convergence. SIAM J. Optim. **16**(1), 1–31 (2005)
42. Wang, C., Xiu, N.: Convergence of the gradient projection method for generalized convex minimization. Comput. Optim. Appl. **16**(2), 111–120 (2000)
43. Watt, J., Borhani, R., Katsaggelos, A.K.: Machine Learning Refined: Foundations, Algorithms, and Applications. Cambridge University Press, Cambridge (2020)
44. Yu, P.Y., Craciun, G.: Mathematical analysis of chemical reaction systems. Isr. J. Chem. **58**(6–7), 733–741 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.