# An optimization approach to finding roots of systems of nonlinear equations



IOP Conf. Series: Materials Science and Engineering **537** (2019) 042007 doi:10.1088/1757-899X/537/4/042007

# An optimization approach to finding roots of systems of nonlinear equations

**O Khamisov*** **and A Kolosnitsyn**

Melentiev Energy System Institute, 130, Lermontov street, Irkutsk, 664033, Russian Federation

* E-mail: khamisov@isem.irk.ru

**Abstract**. We describe and test an approach to finding roots of systems of nonlinear equations. This approach is based on a reduction to an auxiliary optimization problem. In spite of the fact that the reduced problem is, in general, a nonconvex optimization problem, we use only local search techniques and provide conditions for the obtained stationary point to be a root of the initial system of nonlinear equations. Results of computational experiments are given.

## 1. Introduction

We consider the problem of determining at least one root of a system of nonlinear equations

$$f_i(x) = 0, i = 1,...,n, x \in R^n, \tag{1.1}$$

where $f_i : R^n \to R, i = 1,...,n$ are continuously differentiable functions.

A huge number of papers and books are devoted to solving system (1.1). Let us only mention [1], one of the most well-known books. See also [2] for a recent review on iterative methods of solving systems of nonlinear equations. Among the modern effective methods, the LP-Newton method must be mentioned, see [3-5] and references therein. A comprehensive survey of the existing literature on the topic of the paper can be a subject of a separate paper. In this work, our goal consists in presenting a simple tool based on an optimization approach and some available optimization solvers. We consider four types of optimization problems tightly connected to finding roots of system (1.1). The suggested approach deserves wider applications in practice, first of all, due to its simplicity and effectiveness.

Assumption. It is assumed that system (1.1) has at least one root. The paper is organized in the following way. Section 2 describes the optimization reduction, gives theoretical justification and discusses some implementation questions concerning a starting point. Section 3 contains preliminary testing results.

## 2. Optimization reformulation

We consider two optimization problems associated with system (1.1). The first problem is well-known and has the following form

$$\min\{ F_1(x) = \sum_{i=1}^{n} f_i^2(x) : x \in R^n \}. \tag{2.1}$$

The second one is a constrained optimization problem

$$\min\{ F_2(x) : x \in X \}, \tag{2.2}$$

$$F_2(x) = \sum_{i=1}^{n} f_i(x),\ X = \{x \in R^n : f_i(x) \geq 0, i = 1,...,n\}. \tag{2.3}$$

For both optimization problems the following (obvious) statement is true. Let $x^*$ be a global minimum point. For k=1 or 2, if $F_k(x^*) > 0$, then system (1.1) is inconsistent. If $F_k(x^*) = 0$, then $x^*$ is a root of system (1.1). In order to prove inconsistency, we unavoidably have to globally solve problem (2.1) or (2.2)-(2.3). Without convexity assumptions, finding a global minimum point is computationally intractable problem except for some very particular cases. One of the main reasons for that is the absence of practical criteria of global optimality. In other words, if we have a point suspected to be globally optimal, we simply cannot check whether it is globally optimal or not by some non-exponentially enumerative (or non-NP hard) procedure. If we know that the investigated system is solvable, as in our case, then the situation is much easier. Equality $F_1(x^*) = 0$ or $F_2(x^*) = 0$ plays the role of a global optimality criterion. Without our assumption, we do not know whether the zero value of the objective functions is achievable. In this case, we can use local search algorithms [6] to find a stationary point $x^*$ and check the value of the objective.

Let us describe now a very important case when a stationary point is a global minimum point.

Theorem 1. *Let $x^*$ be a stationary point of problem (2.1) and let gradients $\nabla f_i(x^*), i = 1,...n$ be linearly independent. Then $x^*$ is a point of global minimum and $F_1(x^*) = 0$.*

Proof. Since $x^*$ is a stationary point,

$$\nabla F_1(x^*) = 2\sum_{i=1}^{n} f_i(x^*)\nabla f_i(x^*) = 0. \tag{2.4}$$

Gradients $\nabla f_i(x^*), i = 1,...n$ are linearly independent, therefore equality (2.4) is valid if and only if $f_i(x^*) = 0$, $i = 1,...,n$. The theorem is proved.

The linear independence property plays the same role in solving problem (2.2)-(2.3). Theorem 2 below is simply derived from a more general theory developed in [7,8].

Theorem 2. *Let $x^*$ be a stationary point of problem (2.2)-(2.3) and let gradients $\nabla f_i(x^*), i = 1,...n$ be linearly independent. Then $x^*$ is a point of global minimum and $F_2(x^*) = 0$.*

Proof. Write down the Lagrange function $L(x, \lambda) = F(x) - \sum_{i=1}^{n} \lambda_i f_i(x)$ and the necessary optimality conditions:

1. stationarity

$$\nabla_x L(x^*, \lambda^*) = \nabla F(x^*) - \sum_{i=1}^{n} \lambda_i^* \nabla f_i(x^*) = \sum_{i=1}^{n} (1 - \lambda_i^*)\nabla f_i(x^*) = 0, \tag{2.5}$$

2. complementarity

$$\lambda_i^* f_i(x^*) = 0, i = 1,...n, \tag{2.6}$$

3. feasibility

$$f_i(x^*) \geq 0, i = 1,...,n,$$

where $\lambda_i^* \geq 0, i = 1,...,n$ are the optimal Lagrange multipliers corresponding to $x^*$. Since $\nabla f_i(x^*), i = 1,...n$ are linear independent and due to stationarity conditions (2.5), we have $\lambda_i^* = 1, i = 1,...,n$. Substituting the optimal values of Lagrange multipliers in complementarity conditions (2.6), we obtain the desirable result $f_i(x^*) = 0, i = 1,...,n.$ The theorem is proved.

It should also be noted that the linear independence condition is essential for the Newton method as well. Otherwise, the auxiliary system of linear equations would not have a unique solution. The difference between the Newton method (and its modifications) and the optimization approaches based on problems (2.1) and (2.2)-(2.3) consists in the following. For the Newton method, we must have the linear independence property at every iteration $x^k$, while for the optimization approaches we need the linear independence only at the limit (stationary) point $x*$. In what follows, a stationary point that satisfies the linear independence condition will be called a regular stationary point.

The advantage of problem (2.1) is that it is an unconstrained problem. So, we do not have to find a feasible starting point. A local search algorithm can start from an arbitrary given point. Such a possibility is practically essential for the multistart procedure used below.

The advantage of problem (2.2)-(2.3) is that its objective does not contain squared terms. This property is important from the numerical point of view. However, problem (2.2)-(2.3) is a constrained problem with nonconvex and possibly disconnected feasible set. So, we have to provide a feasible starting point for a local search algorithm. It is well known that finding a feasible point of a nonconvex set is a global optimization problem itself. However, in our case, we take advantage of the description of $X$. Take an arbitrary point $x^0 \in R^n$ and calculate values $\gamma_i = sign(f_i(x^0)), i = 1,...,n,$ where $sign(z)$ is the sign function of scalar $z$, i.e. $sign(z) = -1$ when $z < 0$, $sign(z) = 0$ when $z = 0$ and $sign(z) = 1$ when $z > 0$. Introduce functions $\tilde{f}_i(x) = \gamma_i f_i(x)$, if $\gamma_i \neq 0$ and $\tilde{f}_i(x) = f_i(x)$, if $\gamma_i = 0,\ i = 1,...,n$. Then $\tilde{f}_i(x^0) \geq 0, i = 1,...,n.$ Hence, $x^0$ is a feasible point of set

$$\tilde{X} = \{x \in R^n : \tilde{f}_i(x) \geq 0, i = 1,...,n\}.$$

Define a new objective function

$$\tilde{F}(x) = \sum_{i=1}^{n} \tilde{f}_i(x).$$

Consider the problem of minimizing $\tilde{F}$ over $\tilde{X}$. It is of the same type as problem (2.2)-(2.3), so any regular stationary point $\tilde{x}$ of the latter problem satisfies equalities $\tilde{f}_i(\tilde{x}) = 0$, therefore $f_i(\tilde{x}) = 0, i = 1,...,n$.

Let us describe now a simple local search procedure based on the ideas given above.

I. Take a point $x^0 \in R^n$.

II. Calculate scalars $\gamma_i = sign(f_i(x^0)), i = 1,...n$.

III. Apply a local search algorithm for the problem of minimizing function $\tilde{F}$ over set $\tilde{X}$ using $x^0$ as a starting point.

IV. If a regular stationary point is found, then it is a root of the system (1.1).

Several local search solvers can be used at step III. The question of choosing a solver is discussed in Section 3.

Let us now describe important, from the computational point of view, reformulations of problems (2.1) and (2.2)-(2.3). Introduce auxiliary variables $s_i$, $i = 1,...n$ and reformulate problem (2.1) in the following way

$$\min_{x,s} \{\sum_{i=1}^{n} s_i^2 : s_i = f_i(x), i = 1,...n, x \in R^n\}. \quad (2.7)$$

In problem (2.7) only variables $s_i$, $i = 1,...,n$ are squared, while functions $f_i$, $i = 1,...,n$ remain unchanged. From the mathematical point of view, problems (2.1) and (2.7) are equivalent. Providing a feasible starting point for (2.7) is not difficult: just take any $x^0 \in R^n$, calculate $s_i^0 = f_i(x^0)$, $i = 1,...,n$, and pair $(x^0, s^0)$ is a feasible point for (2.7).

The same can be done for problem (2.2)-(2.3). The result is as follows

$$\min_{x,s} \{\sum_{i=1}^{n} s_i : s_i = f_i(x), s_i \geq 0, i = 1,...,n\}. \quad (2.8)$$

Again, from the mathematical point of view, problems (2.2)-(2.3) and (2.8) are equivalent.

Nevertheless, the computational results for problems (2.1) and (2.7) are different, and so are the computational results for problems (2.2)-(2.3) and (2.8).

## 3. Comparative testing

The goal of the comparative testing was to assess effectiveness of finding roots of system (1.1) by means of local search algorithms only, without involving any global optimization technique. We compared computational effectiveness of the Newton method (Newton), damped Newton method (dNewton), and five local search solvers based on problems (2.1), (2.2)-(2.3), (2.7), and (2.8) correspondingly.

Let us briefly review the Newton method and the damped Newton method used in the testing. Their iterative descriptions are as follows.

The Newton method: $x^{k+1} = x^k - J(x^k)^{-1} f(x^k), k = 0,1,....$, where $J(x^k)$ is the Jacobian of functions $f_i$, $i = 1,...,n$ calculated at $x^k$.

The damped Newton method: $x^{k+1} = x^k - \alpha_k J(x^k)^{-1} f(x^k), k = 0,1,....$ Parameters $\alpha_k$ are determined the following way. If $\|f(x^{k+1})\| < \|f(x^k)\|$, then $\alpha_k = 1$, where $\|w\|$ is the Euclidean norm of vector $w \in R^n$ and $f(x) = (f_1(x),..., f_n(x))^T$. If $\|f(x^{k+1})\| \geq \|f(x^k)\|$, then $\alpha_k$ is sequentially decreased by the rule $\alpha_k = \delta\alpha_k$ until condition $\|f(x^{k+1})\| < \|f(x^k)\|$ is met, $\delta \in (0,1)$. In our testing, $\delta = 0.75$ and $\alpha_0 = 1$.

For all the approaches, the same, randomly chosen initial point $x^0$ is used.

The testing was performed for randomly generated systems of quadratic functions

$$f_i(x) = x^T Q_i x + c_i^T x + r_i, i = 1,...,n, \quad (2.9)$$

where $Q_i$ are arbitrary $n \times n$ matrices, $c_i \in R^n$ are arbitrary vectors, and scalars $r_i$ are selected to be such that the generated system (1.1) with functions (2.9) has at least one root.

The testing was done in GAMS system [9], demo-version 24.7.4. The computations were performed on a 4-core laptop Intel Core i7/2.3 GHz/ 8GB. The following stopping criterion was used: $\left\| f(x^k) \right\| \le \varepsilon$ with $\varepsilon = 10^{-4}$. The starting point was the same in all the trials.

First, we selected the best solver for local optimization of problems (2.1), (2.2)-(2.3), (2.7), and (2.8). The results are given in table 1. Five solvers used were conopt, minos, ipopt, snopt, and knitro [9]. The number of randomly generated systems (2.9) was 30, the number of variables $n = 30$.

The numbers in the body of table 1 show how many test problems of each type were solved by each solver. For example, solver conopt solved 15 problems of type (2.2)-(2.3) out of 30. This means that 30 random systems (2.9) were created and converted to problems of type (2.2)-(2.3), then 15 of them were solved to global optimality, i.e. a root was found in 15 systems out of 30. From table 1, we can make the following conclusions: solver conopt was the most successful for problems (2.1) and (2.2)-(2.3), while solver ipopt was the best for problems (2.7) and (2.8).

**Table 1.** Number of problems globally solved.

| Problem/Solvers | conopt | minos | ipopt | snopt | knitro |
|---|---|---|---|---|---|
| (2.1) | 16 | 16 | 12 | 0 | 12 |
| (2.2)-(2.3) | 15 | 11 | 0 | 10 | 9 |
| (2.7) | 16 | 16 | 22 | 0 | 18 |
| (2.8) | 1 | 11 | 16 | 10 | 12 |

The overall best result was shown by solver ipopt for type (2.7) problems. It has found a root in 22 cases out of 30.

Next, a testing was performed for different dimensions. The results are presented in table 2.

**Table 2.** Testing results for different dimensions.

| Approach/Dimension | n=5 | n=10 | n=20 | n=30 |
|---|---|---|---|---|
| (1.1), Newton | 0 | 1 | 1 | 1 |
| (1.1), dNewton | 0 | 2 | 2 | 2 |
| (2.1), conopt | 23 | 15 | 21 | 14 |
| (2.2)-(2.3), conopt | 23 | 12 | 11 | 7 |
| (2.7), ipopt | 26 | 20 | 26 | 21 |
| (2.8), ipopt | 25 | 21 | 15 | 14 |

The approaches are listed in the first column of table 2. Notation “(1.1), Newton” means that the Newton method was used to solve a system (1.1); “(1.1), dNewton” means using the damped Newton method for a system (1.1); “(2.1), conopt” means applying solver conopt to a problem of type (2.1); “(2.2)-(2.3), conopt” means applying solver conopt to a problem (2.2)-(2.3); “(2.7), ipopt” means applying solver ipopt to a problem (2.7); “(2.8), ipopt” means applying solver ipopt to a problem (2.8). For each of the four chosen dimension, 30 problems were randomly generated. The body of the table consists of the numbers of successful runs of the corresponding algorithms on problems of each of the dimensions. Consider, for example, $n = 30$.

For this dimension, the Newton method solved 1 system out of 30, the damped Newton method solved 2 systems, solver conopt solved 14 problems out of 30 of type (2.1) to global optimality, and so on. Solving problem (2.7) by solver ipopt turned out to be the most successful. All the solvers were used with the default options.

A natural generalization of the local methodology consists in combining a local search with the multistart technique and parallelization. A local search was performed by solver ipopt for problems of type (2.7). Table 3 shows the results of the testing.

**Table 3.** Testing results for a combination of a local search with multistart and parallelization.

| n | s | ss | dss | t |
|---|---|---|---|---|
| 5 | 30 | 23 | 7 | 0:00:01.126 |
| 5 | 50 | 39 | 8 | 0:00:1.916 |
| 5 | 70 | 53 | 8 | 0:00:03.101 |
| 5 | 100 | 82 | 8 | 0:00:05:134 |
| 5 | 150 | 120 | 8 | 0:00:06.807 |
| 10 | 30 | 23 | 12 | 0:00:01.297 |
| 10 | 50 | 23 | 18 | 0:00:02.148 |
| 10 | 70 | 49 | 21 | 0:00:03.642 |
| 10 | 100 | 66 | 26 | 0:00:05.205 |
| 10 | 150 | 88 | 28 | 0:00:07.926 |
| 15 | 30 | 13 | 12 | 0:00:01.583 |
| 15 | 50 | 23 | 21 | 0:00:02.807 |
| 15 | 70 | 32 | 27 | 0:00:03.304 |
| 15 | 100 | 44 | 33 | 0:00:05.880 |
| 15 | 150 | 68 | 44 | 0:00:08.895 |
| 20 | 30 | 11 | 10 | 0:00:06.633 |
| 20 | 50 | 21 | 19 | 0:00:04.823 |
| 20 | 70 | 29 | 26 | 0:00:10.364 |
| 20 | 100 | 39 | 33 | 0:00:11.903 |
| 20 | 150 | 58 | 58 | 0:00:15:787 |
| 25 | 30 | 11 | 11 | 0:00:07.535 |
| 25 | 50 | 20 | 20 | 0:00:13:176 |
| 25 | 70 | 27 | 27 | 0:00:17.154 |
| 25 | 100 | 35 | 35 | 0:00:17:447 |
| 25 | 150 | 55 | 55 | 0:00:34.528 |
| 30 | 30 | 10 | 10 | 0:00:10.452 |
| 30 | 50 | 19 | 19 | 0:00:18.552 |
| 30 | 70 | 26 | 26 | 0:00:30.910 |
| 30 | 100 | 33 | 33 | 0:00:36.476 |
| 30 | 150 | 49 | 49 | 0:00:49.930 |

In table 3, $n$ denotes a dimension, $s$ - the number of multistarts, $ss$ - the number of successful multistarts, $dss$ - the number of different roots among all roots found, $t$ - time, hours:minutes:seconds. Successful multistarts are those that resulted in a root of the system. Otherwise, a multistart is unsuccessful, what means that the local search procedure has stopped at a stationary point that was not a root of the system. Starting points were chosen randomly, all $s$ optimization problems were solved in parallel. All computations were performed in GAMS.

## 4. Conclusion

Our preliminary testing shows that the suggested optimization approach can be successfully used for solving system (1.1). The advantage of the approach consists in using only local search methods. If the Assumption made in section 1 is not guaranteed, then the problem of verifying consistency of system (1.1) arises. The most difficult task then is to prove that system (1.1) does not have solutions. In this case, global optimization methods must be used. Nevertheless, the suggested approach can be used as an offhand and easy tool to find solutions of (1.1). Further investigations and experiments will be performed for a wider range of tests and for higher dimensions.

## Acknowledgments

The investigation was supported by project III.17.3.1 of the Program of fundamental investigations RAS, No. AAAA-A17-117030310444-2.

## References

[1] Dennis J and Schnabel R 1996 Numerical methods for unconstrained optimization and nonlinear equations *SIAM, Philadelphia, PA*
[2] Amat S and Busquier S (eds.) 2016 Advances in iterative methods for nonlinear equations *SEMA SIMAI Springer Series* **10**
[3] Facchinei F, Fischer A and Herrich M 2011 An LP-Newton method: nonsmooth equations, KKT systems, and nonisolated solutions Technical Report Available from http://www.optimization-online.org/DB_HTML/2011/09/3177.html
[4] Facchinei F, Fischer A and Herrich M 2013 A family of Nrewton methods for nonsmooth constrained systems with nonisolated solutions *Math. Meth. Oper. Res.* **77** 433-43
[5] Izmailov A, Kurennoy A and Solodov M 2018 Critical solutions of nonlinear equations: local attraction for Newton type-methods *Math. Program.* Ser. A doi 10.1007/s10107-017-1128-5
[6] Bulatov V 1977 Imbedding methods in optimization problems *Novosibirsk, Nauka* (in Russian)
[7] Bulatov V 2000 Numerical methods for finding all real roots of systems of nonlinear equations *Computational Mathematics and Mathematical Physics* **40(3)** 348-55
[8] Nocedal J and Wright S 2006 Numerical optimization *Springer*
[9] Electronic resource Available from https://www.gams.com