

A Method for Solving Global Minimization Problems and Systems of Nonlinear Equations

Ivan Raykov

Department of Mathematics and Computer Science, University of Arkansas at Pine Bluff, AR, USA
Email: iraykov51@yahoo.edu

How to cite this paper: Raykov, I. (2026) A Method for Solving Global Minimization Problems and Systems of Nonlinear Equations. *Applied Mathematics*, 17, 693-713.
<https://doi.org/10.4236/am.2026.1710038>

Received: August 24, 2026

Accepted: October 27, 2026

Published: October 30, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this work, we introduce a method of finding the global minimum of bounded from below functions with continuous partial derivatives in an Euclidean space E^n with existing finite points of the global minimum. We create a function that finds the zeros of the sign function of the sum of the squares of the partial derivatives of the function to minimize. We can solve with this method also systems of simultaneous continuous nonlinear equations. We create, in that case, a function that finds the zeros of the sign function of the sum of the squares of the equations of the system to solve.

Keywords

Continuous Differentiable Functions, Global Minimum, Global Minimization Algorithm, Iterative Procedure, Composite Functions, Systems of Nonlinear Equations

1. Introduction

There are different approaches to find the Solutions of the Global Minimization Problems for continuous smooth functions.

Deterministic continuous global optimization methods (see [1], [2] and [3]).

Interval methods (see [4] and [5]).

There are also non-continuous optimization methods like:

Stochastic methods such as Direct Monte-Carlo sampling (for example traveling salesman problem) (see [6]);

Stochasting tunneling (STUN) based on the Monte Carlo method-sampling (see [7]).

Heuristics and metaheuristics, Ant colony optimization (ACO) (see [8]); Simulated annealing (see [9] and [10]); Tabu search (see [11]); and Evolutionary algorithms (see [12]).

Let $f(x_1, x_2, \dots, x_n)$ is bounded from below function with continuous partial derivatives in the set $D \subset E^n$, where E^n is the Euclidean space.

We consider next global minimization problem:

$$\min_{(x_1, x_2, \dots, x_n) \in D} f(x_1, x_2, \dots, x_n) = f_{\min}. \quad (1)$$

We assume that a finite solution point exists, meaning there is at least one finite point $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \in E^n$ where a solution to (1) exists.

We create the function: $f_{sq}(x_1, x_2, \dots, x_n) = (f_{x_1})^2 + (f_{x_2})^2 + \dots + (f_{x_n})^2$ of the sum of the squares of the partial derivatives of the given function to minimize. Then we create the composite function of the sign function with the created function of the sum of squares:

$$f_s(x_1, x_2, \dots, x_n) = \text{sign}(f_{sq}(x_1, x_2, \dots, x_n)) = \text{sign}\left((f_{x_1})^2 + (f_{x_2})^2 + \dots + (f_{x_n})^2\right)$$

The zeros of this composed function are the critical points of the given function. The global minimum of the given function is the minimum value of the given function at the found critical points.

This method can be used to solve also systems of simultaneous continuous nonlinear equations. We create in that case a function that finds the zeros of the sign of the sum of the squares of the equations of the system to solve:

$$\text{sign}\left((f_1)^2 + (f_2)^2 + \dots + (f_n)^2\right).$$

The numerical realisations need next steps:

We used MATLAB 2023b for solving numerical examples.

With num we denote the number of examples to solve.

To avoid skipping some critical points by the sign function f_{num-s} , we subtract a small number $0 < \alpha < st_{gr}$ from the function $f_{num-sq}(x_1, x_2, \dots, x_n)$, that is $f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha = \left((f_{x_1})^2 + (f_{x_2})^2 + \dots + (f_{x_n})^2\right) - \alpha$ (st_{gr} is the step of the meshgrid command in MATLAB. In our MATLAB 2023b $st_{gr} = 0.002$). We add 1 to the composite function: $(f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1)$. Therefore this function will have values 0 and 2. Therefore, finally, we multiply by 0.5 and receive: $f_{num-f}(x_1, x_2, \dots, x_n) = 0.5 * (f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1)$. By this way we find the critical points of the function to minimize by finding the points for which $f_{num-f}(x_1, x_2, \dots, x_n) = 0$, as well as the composite function $0.5 * (f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1) = 0.5 * \left(\text{sign}\left(\left((f_{x_1})^2 + (f_{x_2})^2 + \dots + (f_{x_n})^2\right) - \alpha\right) + 1\right) = 0$. The values of both functions can be only 0's and 1's.

It is not necessary for a system of simultaneous nonlinear equation that the number of nonlinear equations to be equal to the number of the independent variables.

In both cases, for minimization problems and for solving systems of nonlinear equations, we isolate the zeros of *the created composite function*

$f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n))$, respectively of *the function* $f_{num-f}(x_1, x_2, \dots, x_n)$. Therefore, the function sum of the partial derivatives is:

$$f_{num-sq}(x_1, x_2, \dots, x_n) = \left(\frac{\partial f}{\partial x_1}\right)^2 + \left(\frac{\partial f}{\partial x_2}\right)^2 + \dots + \left(\frac{\partial f}{\partial x_n}\right)^2 = \sum_{i=1}^n \left(\frac{\partial f}{\partial x_i}(x_1, x_2, \dots, x_n)\right)^2$$

Function f_{num-s} is the sign function. The values of composite function $f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha)$ at the zeros of the function f_{num-sq} will be -1 , or

$$\begin{aligned} & f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) \\ &= \text{sign}\left(\left(\frac{\partial f}{\partial x_1}\right)^2 + \left(\frac{\partial f}{\partial x_2}\right)^2 + \dots + \left(\frac{\partial f}{\partial x_n}\right)^2 - \alpha\right) \\ &= \text{sign}\left(\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i}(x_1, x_2, \dots, x_n)\right)^2 - \alpha\right) = -1. \end{aligned}$$

Therefore, we add 1 to the function $f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha)$, or $f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1$. The possible values of this function will be 0 and 2. Because of that if we multiply it finally by 0.5 the possible values of $f_{num-f}(x_1, x_2, \dots, x_n) = 0.5 * (f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1)$ will be 0 and 1, which are also the possible values of the composite function

$0.5 * (f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1)$. By this way we find the critical

point(s) of the function to minimize or the zeros (the solutions) of the system of simultaneous nonlinear equations. The critical point(s) for which the function to minimize has the minimum value is/are the solution point(s) of the minimization problem.

We can apply this method also for minimization of functions of more independent variables and to solve systems of more nonlinear equations with same or larger number of independent variables. Of course, visualizations are not possible for those cases. Therefore, most of our examples are for functions of two independent variables, because their presentations are more obvious and visualizations are possible.

We use programs in MATLAB (.m) files. We present one example at the end of the paper for solving a system of three simultaneous nonlinear equations with three independent variables. Of course, without visualizations. Our results are with precision of 10^{-6} .

This is not an iterative method.

The method is not for solving boundary optimization problems.

We present graphs of surfaces for $-5 \leq x \leq 5$ and $-5 \leq y \leq 5$ only because we cannot really present the graph of the surface on entire space E^3 .

We applied this method for many well known functions, partially listed at

the end of the manuscript.

There is not a stopping criterion, because we do not create iterative algorithms for finding the solution.

We suppose that the distance between two critical points is greater or equal $\sqrt{n} \times 0.001$. This restriction is from the software available, not from the algorithm.

There is not a convergence criteria, because no an iterative procedure is used.

For the values of exponential growth the function will not have minima for these values.

For the values of exponential decay the function will not have minima for finite points, according to the conditions to be satisfied.

There are no cases with boundary optima, because we do not consider boundary optimization problems.

We assume that a finite solution point exists, meaning there is at least one point with finite coordinates that solves the global minimization problem.

For the values of exponential decay the function will not have minima for finite points, according to the conditions to be satisfied.

We suppose also that a finite solution point exists for systems of nonlinear equations, meaning that there is at least one point with finite coordinates that solves the system of nonlinear equations.

2. Solving Global Minimization Problems

2.1. Theoretical Approach

Let $f(x_1, x_2, \dots, x_n)$ is bounded from below function with continuous partial derivatives in the set $D \subset E^n$, where E^n is the Euclidean space.

We consider next global minimization problem:

$$\min_{(x_1, x_2, \dots, x_n) \in D} f(x_1, x_2, \dots, x_n) = f_{\min}. \quad (1)$$

We assume that a finite solution point exists, meaning there is at least one finite point $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \in E^n$ where a solution to (1) exists.

Let us find the partial derivatives $\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n}$ and create the system of nonlinear equations:

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= 0 \\ \frac{\partial f}{\partial x_2} &= 0 \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= 0 \end{aligned} \quad (2)$$

As we wrote in introduction, we denote with the function $f_{\text{num-sq}}(x_1, x_2, \dots, x_n)$ the sum of the squares of the partial derivatives of the function being minimized,

or

$$f_{num-sq}(x_1, x_2, \dots, x_n) = \left(f_{num_{x_1}}\right)^2 + \left(f_{num_{x_2}}\right)^2 + \dots + \left(f_{num_{x_n}}\right)^2.$$

We denote with **num** the number of the function being minimized. We introduce four examples, therefore, **num** will have values from 1 to 4. For example, $f_{1-sq}(x_1, x_2, \dots, x_n)$ denotes the sum of the partial derivatives of the First function for minimization from the Numerical Problems (Example 1), or

$$f_{1-sq}(x_1, x_2, \dots, x_n) = \left(f_{1_{x_1}}\right)^2 + \left(f_{1_{x_2}}\right)^2 + \dots + \left(f_{1_{x_n}}\right)^2$$

$f_{2-sq}(x_1, x_2, \dots, x_n)$ denotes the sum of the partial derivatives of the Second function for minimization from the Numerical Problems (Example 2), or

$$f_{2-sq}(x_1, x_2, \dots, x_n) = \left(f_{2_{x_1}}\right)^2 + \left(f_{2_{x_2}}\right)^2 + \dots + \left(f_{2_{x_n}}\right)^2$$

and so on. The function $f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n))$ denotes the composition of the sign function with the function $f_{num-sq}(x_1, x_2, \dots, x_n)$, or

$$f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n)) = \text{sign}\left(\left(f_{num_{x_1}}\right)^2 + \left(f_{num_{x_2}}\right)^2 + \dots + \left(f_{num_{x_n}}\right)^2\right).$$

For example, the composition of the sign function with the sum function of the squares of the partial derivatives of the First function for minimization from the Numerical Problems is denoted with $f_{1-s}(f_{1-sq}(x_1, x_2, \dots, x_n))$, the sign function with the sum function of the partial derivatives of the Second function for minimization from the Numerical Problems is denoted with

$f_{2-s}(f_{2-sq}(x_1, x_2, \dots, x_n))$, and so on.

Function $f_{num-f}(x_1, x_2, \dots, x_n)$ in the general case will be:

$$\begin{aligned} f_{num-f}(x_1, x_2, \dots, x_n) &= f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n)) \\ &= \text{sign}(f_{num-sq}(x_1, x_2, \dots, x_n)) \\ &= \text{sign}\left(\sum_{i=1}^n \lambda_i \left(\frac{\partial f}{\partial x_i}(x_1, x_2, \dots, x_n)\right)^{m_k}\right) \end{aligned}$$

where $\lambda_i > 0$, and $m_k = 2 * k$, $k = 0, 1, 2, 3, \dots$.

We prove that function $f_{num-f}(x_1, x_2, \dots, x_n)$ is lower semicontinuous in the Lemma 2.1 below.

We use the property of the *sign function* $f_{num-s}(g(x_1, x_2, \dots, x_n))$ for any continuous function $g(x_1, x_2, \dots, x_n)$:

$$\begin{aligned} f_{num-s}(g(x_1, x_2, \dots, x_n)) \\ = \text{sign}(g(x_1, x_2, \dots, x_n)) \rightarrow \begin{cases} -1, & \text{if } g(x_1, x_2, \dots, x_n) < 0; \\ 0, & \text{if } g(x_1, x_2, \dots, x_n) = 0; \\ 1, & \text{if } g(x_1, x_2, \dots, x_n) > 0, \end{cases} \end{aligned}$$

Because $f_{num-sq}(x_1, x_2, \dots, x_n) \geq 0$

$$\begin{aligned}
& f_{num-s} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right) \\
&= \text{sign} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right) \rightarrow \begin{cases} 0, & \text{if } f_{num-sq} (x_1, x_2, \dots, x_n) = 0; \\ 1, & \text{if } f_{num-sq} (x_1, x_2, \dots, x_n) > 0. \end{cases}
\end{aligned}$$

Therefore, also

$$\begin{aligned}
& f_{num-f} (x_1, x_2, \dots, x_n) = f_{num-s} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right) \\
&= \text{sign} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right) \rightarrow \begin{cases} 0, & \text{if } f_{num-sq} (x_1, x_2, \dots, x_n) = 0; \\ 1, & \text{if } f_{num-sq} (x_1, x_2, \dots, x_n) > 0. \end{cases}
\end{aligned}$$

The surface of the function *sign of* f_{num-sq} , denoted with

$f_{num-s} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right)$ is the plane $f_{num-s} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right) = 1$, a.e. with exception the points in which $f_{num-sq} (x_1, x_2, \dots, x_n) = 0$ and therefore also with exception the points in which $f_{num-s} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right) = 0$. Also, the surface $f_{num-f} (x_1, x_2, \dots, x_n) = 1$ is the same plane a.e. with exception of the points in which $f_{num-f} (x_1, x_2, \dots, x_n) = 0$.

The composite function $f_{num-s} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right)$ **as well as the function** $f_{num-f} (x_1, x_2, \dots, x_n)$ **isolates i.e., finds the zeros of the function** $f_{num-sq} (x_1, x_2, \dots, x_n)$, **which are the critical points of the function to minimize** $f_{num} (x_1, x_2, \dots, x_n)$.

We adapt the Weierstrass Theorem to support our method.

See Proposition 1.3.4 from [13] and from [14] the Weierstrass Theorem (Theorem 1.29).

Before presenting the Theorem, we need next Lemma.

Lemma 2.1 *We denote with*

$$\begin{aligned}
f_{num-f} (x_1, x_2, \dots, x_n) &= f_{num-s} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right) \\
&= \text{sign} \left(f_{num-sq} (x_1, x_2, \dots, x_n) \right) \\
&= \text{sign} \left(\left(f_{num_{x_1}} \right)^2 + \left(f_{num_{x_2}} \right)^2 + \dots + \left(f_{num_{x_n}} \right)^2 \right).
\end{aligned}$$

Let D be a set of the Euclidean space E^n . The function $f_{num-f} : D \rightarrow R$ is lower semicontinuous if for any $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \in D$ and any sequence $\{(x_1^k, x_2^k, \dots, x_n^k)\}$ in D converging to the point $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \in D$, the limit of the function values at those points $f_{num-f} (x_1^k, x_2^k, \dots, x_n^k)$ the inequality

$$\liminf_{k \rightarrow \infty} f_{num-f} (x_1^k, x_2^k, \dots, x_n^k) \geq f_{num-s} (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n). \quad (3)$$

holds.

Proof. Consider a sequence $\{(x_1^k, x_2^k, \dots, x_n^k)\} = \{(\bar{x}_1 + 1/k, \bar{x}_2 + 1/k, \dots, \bar{x}_n + 1/k)\}$ converging to $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$:

Case 1: $f_{num-sq} (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) > 0$. Since, $f_{num-f} (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 1$.

We construct the sequence $\{(x_1^k, x_2^k, \dots, x_n^k)\} = \{(\bar{x}_1 + 1/k, \bar{x}_2 + 1/k, \dots, \bar{x}_n + 1/k)\}$ that converges to $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$. Let us suppose that for any $\varepsilon > 0$, there exists $\delta > 0$ such that $(x_1^k, x_2^k, \dots, x_n^k) \in B((\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n); \delta)$ and therefore exists also

an integer N , such that when $k > N$

$$f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) - \varepsilon < f_{num-f}(x_1^k, x_2^k, \dots, x_n^k).$$

It follows that

$$\begin{aligned} f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) - \varepsilon &\leq \liminf_{k \rightarrow \infty} f_{num-f}(x_1^k, x_2^k, \dots, x_n^k) \\ f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) - \varepsilon &\leq \liminf_{k \rightarrow \infty} f_{num-f}(\bar{x}_1 + 1/k, \bar{x}_2 + 1/k, \dots, \bar{x}_n + 1/k) = 1. \end{aligned}$$

Therefore, $f_{num-f}(x_1^k, x_2^k, \dots, x_n^k) = f_{num-f}(\bar{x}_1 + 1/k, \bar{x}_2 + 1/k, \dots, \bar{x}_n + 1/k) = 1$.

In other words, because $f_{num-f}(x_1, x_2, \dots, x_n) = 1$ is continuous for every points where $f_{num-sq}(x_1, x_2, \dots, x_n) \neq 0$, then for all these points $f_{num-f}(x_1, x_2, \dots, x_n)$ will be also lower semicontinuous.

Case 2: $f_{num-sq}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 0$, thus

$$f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = \text{sign}(f_{num-sq}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)) = 0.$$

The only potential “jump” occurs at $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ when $f_{num-sq}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 0$.

We construct again the sequence $\{(x_1^k, x_2^k, \dots, x_n^k)\} = \{(\bar{x}_1 + 1/k, \bar{x}_2 + 1/k, \dots, \bar{x}_n + 1/k)\}$ that converges to $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$. Let us suppose again that for any $\varepsilon > 0$, there exists $\delta > 0$ such that $(x_1^k, x_2^k, \dots, x_n^k) \in B((\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n); \delta)$ and therefore exists also an integer N , such that when $k > N$

$$f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) - \varepsilon < f_{num-f}(x_1^k, x_2^k, \dots, x_n^k).$$

It follows that

$$\begin{aligned} f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) - \varepsilon &\leq \liminf_{k \rightarrow \infty} f_{num-f}(x_1^k, x_2^k, \dots, x_n^k) \\ f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) - \varepsilon &\leq \liminf_{k \rightarrow \infty} f_{num-f}(\bar{x}_1 + 1/k, \bar{x}_2 + 1/k, \dots, \bar{x}_n + 1/k), \end{aligned}$$

or

$$\begin{aligned} \liminf_{k \rightarrow \infty} f_{num-f}(x_1^k, x_2^k, \dots, x_n^k) &= \liminf_{k \rightarrow \infty} f_{num-f}(\bar{x}_1 + 1/k, \bar{x}_2 + 1/k, \dots, \bar{x}_n + 1/k) \\ &\geq f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 0. \end{aligned}$$

Finally, because $f_{num-f}(x_1, x_2, \dots, x_n) = 1$ for every point in D , for which $f_{num-sq}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \neq 0$ then $\liminf_{k \rightarrow \infty} f_{num-f}(x_1^k, x_2^k, \dots, x_n^k) = 1$ when approaching a point for which $f_{num-sq}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 0$ from different directions and therefore

$$f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 0 \leq \liminf_{k \rightarrow \infty} f_{num-f}(x_1^k, x_2^k, \dots, x_n^k) = 1.$$

Consequently, $f_{num-f}(x_1, x_2, \dots, x_n)$ is lower semicontinuous also for every point $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ for which $f_{num-sq}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 0$, i.e.

$f_{num-s}(f_{num-sq}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)) = \text{sign}(f_{num-sq}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n))0$ and hence also $f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 0$, and thus for every point in D .

Let $f(x_1, x_2, \dots, x_n)$ is bounded from below function with continuous partial

derivatives now in the compact subset $D \subset E^n$, where E^n is the Euclidean space and the set D is defined as: $\{D \mid D \text{ is } [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]\}$, where a_k and b_k are finite and real numbers, and $a_k < b_k$, $\forall k = 1, 2, \dots, n$.

Theorem 2.1 Suppose D is a compact set of the Euclidean space E^n and $f_{num-f} : D \rightarrow R$ is lower semicontinuous. Then f_{num-f} has an absolute minimum on D . That means there exists $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \in D$ such that $f_{num-f}(x_1, x_2, \dots, x_n) \geq f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ for all $(x_1, x_2, \dots, x_n) \in D$.

Proof.

Since $\inf \{f_{num-f}(x_1, x_2, \dots, x_n) \mid (x_1, x_2, \dots, x_n) \in D\}$ is a finite number (due to compactness of D and lower semicontinuity of f_{num-f}), we can construct a sequence $\{(x_1^k, x_2^k, \dots, x_n^k)\}$ in D such that $f_{num-f}(x_1^k, x_2^k, \dots, x_n^k)$ approaches infimum as k goes to infinity.

In our case $f_{num-f}(x_1, x_2, \dots, x_n)$ is bounded below because it has only two possible values, 0 or 1. Therefore, its boundary from below is $f_{num-f}(x_1, x_2, \dots, x_n) = 0$.

Define

$$0 = \inf \{f_{num-f}(x_1, x_2, \dots, x_n) \mid (x_1, x_2, \dots, x_n) \in D\}$$

Therefore the set $\{f_{num-f}(x_1, x_2, \dots, x_n) \mid (x_1, x_2, \dots, x_n) \in D\}$ is nonempty, bounded below and its boundary from below is $0 \in R$.

For any $\varepsilon > 0$ there exists N such that for all $k > N$ we can construct a sequence $\{(x_1^k, x_2^k, \dots, x_n^k)\}$ in D such that

$f_{num-f}(x_1^k, x_2^k, \dots, x_n^k) < \inf \{f_{num-f}(x_1, x_2, \dots, x_n) \mid (x_1, x_2, \dots, x_n) \in D\} + \varepsilon$. By the compactness of D , the sequence $\{(x_1^k, x_2^k, \dots, x_n^k)\}$ has a convergent subsequence $\{(x_1^{k_l}, x_2^{k_l}, \dots, x_n^{k_l})\}$ that converges to some $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \in D$.

By the definition of lower semicontinuity, we have

$$f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \leq \liminf_{k \rightarrow \infty} f_{num-f}(x_1^{k_l}, x_2^{k_l}, \dots, x_n^{k_l})$$

Since $f_{num-f}(x_1^{k_l}, x_2^{k_l}, \dots, x_n^{k_l})$ approaches the infimum as k goes to infinity, we can conclude:

$$f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \leq \inf \{f_{num-f}(x_1, x_2, \dots, x_n) \mid (x_1, x_2, \dots, x_n) \in D\}$$

Because $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ is a point in D and $f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ is less than or equal to the infimum, it follows that $f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ must be the infimum, meaning the function attains its minimum value at $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$.

Therefore

$$\begin{aligned} \lim_{l \rightarrow \infty} f_{num-f}(x_1^{k_l}, x_2^{k_l}, \dots, x_n^{k_l}) &= \liminf_{l \rightarrow \infty} f_{num-f}(x_1^{k_l}, x_2^{k_l}, \dots, x_n^{k_l}) \\ &\geq f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \geq 0. \end{aligned}$$

This implies $0 = f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ and, hence,

$$f_{num-f}(x_1, x_2, \dots, x_n) \geq f_{num-f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \text{ for all } (x_1, x_2, \dots, x_n) \in D.$$

2.2. Numerical Algorithm

Our algorithm can be applied for any continuously differentiable functions of finite number independent variables.

The algorithm for finding the minimum of functions of n independent variables.

Let $f(x_1, x_2, \dots, x_n)$ is bounded from below function with continuous partial derivatives in the set $D \subset E^n$, where E^n is the Euclidean space.

We consider again the global minimization problem:

$$\min_{(x_1, x_2, \dots, x_n) \in D} f(x_1, x_2, \dots, x_n) = f_{num_{min}}. \quad (4)$$

We assume that a finite solution point exists, meaning there is at least one finite point $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \in E^n$ where a solution to (1) exists.

Let us denote again the partial derivatives $\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n}$ and create the system of nonlinear equations:

$$\begin{aligned} \frac{\partial f_{num}}{\partial x_1} &= 0 \\ \frac{\partial f_{num}}{\partial x_2} &= 0 \\ &\vdots \\ \frac{\partial f_{num}}{\partial x_n} &= 0 \end{aligned} \quad (5)$$

We denote again with num the number of the correspondent function to minimize (Example number). To solve the system (5), we create the function f_{num-sq} of the sum of the squares of the partial derivatives:

$$f_{num-sq}(x, y) = \left(\frac{\partial f_{num}}{\partial x_1} \right)^2 + \left(\frac{\partial f_{num}}{\partial x_2} \right)^2 + \dots + \left(\frac{\partial f_{num}}{\partial x_n} \right)^2 \quad (6)$$

To avoid skipping some critical points by the sign function we subtract a small number $0 < \alpha < st_{gr}$ from the function f_{num-sq} and then evaluate the composite function, $f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha)$. We denote with st_{gr} the step of the meshgrid command in MATLAB 2023b. In our MATLAB 2023b $st_{gr} = 0.002$. By this way we find the critical points of the function to minimize with finding the points for which $0.5 * (f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1) = 0$. We denoted $f_{num-f}(x_1, x_2, \dots, x_n) = 0.5 * (f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1)$, because $(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha)$ will be negative at the critical points. Therefore $f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) = \text{sign}(f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha))$ will take values of (-1) (negative 1) at the solution points and $(+1)$ (positive 1) for all other points. We will have that $0 < \alpha \leq \sqrt{n} \times 0.001$. Because of that $f_{num-f}(x_1, x_2, \dots, x_n) = 0.5 * (f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1)$ will take again values of zero at the solution points and of 1 for all other points (same values as in the theoretical approach).

The last function isolates the critical points of the function to minimize:

$$\begin{aligned}
& f_{num-f}(x_1, x_2, \dots, x_n) \\
&= 0.5 * (f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1) \\
&= 0.5 * (\text{sign}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1) \\
&= 0.5 * \left(\text{sign} \left(\left(\frac{\partial f_{num}}{\partial x_1} \right)^2 + \left(\frac{\partial f_{num}}{\partial x_2} \right)^2 + \dots + \left(\frac{\partial f_{num}}{\partial x_n} \right)^2 \right) - \alpha \right) + 1
\end{aligned} \tag{7}$$

We solve the equation of n variables:

$$\begin{aligned}
& f_{num-f}(x_1, x_2, \dots, x_n) \\
&= 0.5 * \left(\text{sign} \left(\left(\frac{\partial f_{num}}{\partial x_1} \right)^2 + \left(\frac{\partial f_{num}}{\partial x_2} \right)^2 + \dots + \left(\frac{\partial f_{num}}{\partial x_n} \right)^2 \right) - \alpha \right) + 1 = 0
\end{aligned} \tag{8}$$

For systems of simultaneous nonlinear equations,

$$\begin{aligned}
& f_1(x_1, x_2, \dots, x_n) = 0 \\
& f_2(x_1, x_2, \dots, x_n) = 0 \\
& \dots \\
& f_n(x_1, x_2, \dots, x_n) = 0
\end{aligned} \tag{9}$$

we create the function: $f_{num-sq}(x_1, x_2, \dots, x_n) = ((f_1)^2 + (f_2)^2 + \dots + (f_n)^2)$ and the function $f_{num-f}(x_1, x_2, \dots, x_n)$ for the general case:

$$\begin{aligned}
f_{num-f}(x_1, x_2, \dots, x_n) &= 0.5 * (f_{num-s}(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha) + 1) = \\
&= 0.5 * \text{sign}((f_1)^2 + (f_2)^2 + \dots + (f_n)^2 - \alpha + 1)
\end{aligned}$$

and solve by the same way.

We can apply this method also for minimization of functions of more independent variables and to solve systems of more nonlinear equations with same or more number of independent variables.

When we solve a minimization of a function of n variables $n > 2$ $f_{num}(x_1, x_2, \dots, x_n)$, we find again the function of the sum of squares of the partial derivatives, $f_{num-sq}(x_1, x_2, \dots, x_n)$ and then using the composite function of the *sign function*, f_{num-s} , with the function $f_{num-sq}(x_1, x_2, \dots, x_n)$, we create the function: $f_{num-f}(x_1, x_2, \dots, x_n)$. The values of this function will be only 0 or 1. The zeros will correspond to the critical numbers of the function to minimize. Therefore, the critical points of the given function to minimize $f_{num}(x_1, x_2, \dots, x_n)$ are found. To fix the isolated critical points of the function to minimize $f_{num}(x_1, x_2, \dots, x_n)$ we used MATLAB programs as m files. Of course, visualizations are not possible.

Visualizations are not possible also, when we solve systems of more than 2 equations and the functions are of more than 2 independent variables. In that case the function f_{num-sq} is equal to the sum of the equations of the system, or $f_{num-sq}(x_1, x_2, \dots, x_n) = ((f_1)^2 + (f_2)^2 + \dots + (f_n)^2)$.

Finally, our function is:

$$\begin{aligned}
 f_{num-f}(x_1, x_2, \dots, x_n) &= 0.5 * \left(f_{num-s} \left(f_{num-sq}(x_1, x_2, \dots, x_n) - \alpha \right) + 1 \right) \\
 &= 0.5 * \left(f_{num-s} \left((f_1)^2 + (f_2)^2 + \dots + (f_n)^2 - \alpha \right) + 1 \right).
 \end{aligned}$$

We present one example at the end of the paper of solving a system of three simultaneous nonlinear equations with three independent variables. We present mainly examples for functions of two independent variables, because the method is more clear with visualizations of the solutions.

The advantage of solving the cases in E^2 is that we can visualize the sign function.

These types of functions include many famous optimization problems, we solved already and we do not present in this manuscript, because the limit of pages. We present a list at the end of the additional problems solved.

At the end is introduced a solution of a system of three equations of functions of three independent variables, of course without visualization.

Because the restrictions of the software the distance between two critical points of functions with continuous partial derivatives should be less than $\sqrt{2} \times 0.001$ for functions of two variables and less than $\sqrt{n} \times 0.001$ for functions of n variables.

3. Numerical Problems

3.1. First Function

Let us find the Global Minimum of the function, known as Three-hump camel function. We are looking for the minimum in the region $-5 < x, y < 5$ of the next function:

Picture for Function 1.

$$f_1(x, y) = 2 * x^2 - 1.05 * x^4 + x^6 / 6 + x * y + y^2$$

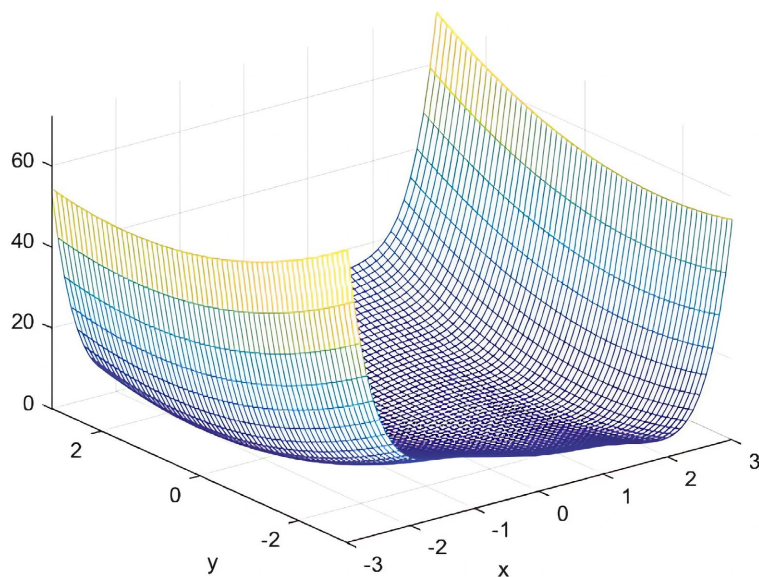


Figure 1. It shows the first surface for global minimization.

The global minimum will be when:

$$f_{1_x}(x, y) = 4 * x - 4.2 * x^3 + x^5 + y = 0 \quad \text{and} \quad f_{1_y}(x, y) = x + 2 * y = 0$$

We create the function:

$$f_{1-sq}(x, y) = (f_{1_x}(x, y))^2 + (f_{1_y}(x, y))^2 = (4 * x - 4.2 * x^3 + x^5 + y)^2 + (x + 2 * y)^2$$

and graph the function $f_{1-f}(x, y) = 0.5 * (f_{1-sq}(x, y) - \alpha) + 1$:

$$\begin{aligned} f_{1-f}(x, y) &= 0.5 * (f_{1-sq}(x, y) - \alpha) + 1 \\ &= 0.5 * (\text{sign}(f_{1-sq}(x, y) - \alpha) + 1) \\ &= 0.5 * \left(\text{sign} \left((4 * x - 4.2 * x^3 + x^5 + y)^2 + (x + 2 * y)^2 - \alpha \right) + 1 \right) \end{aligned}$$

Picture of the Function f_{1-f} .

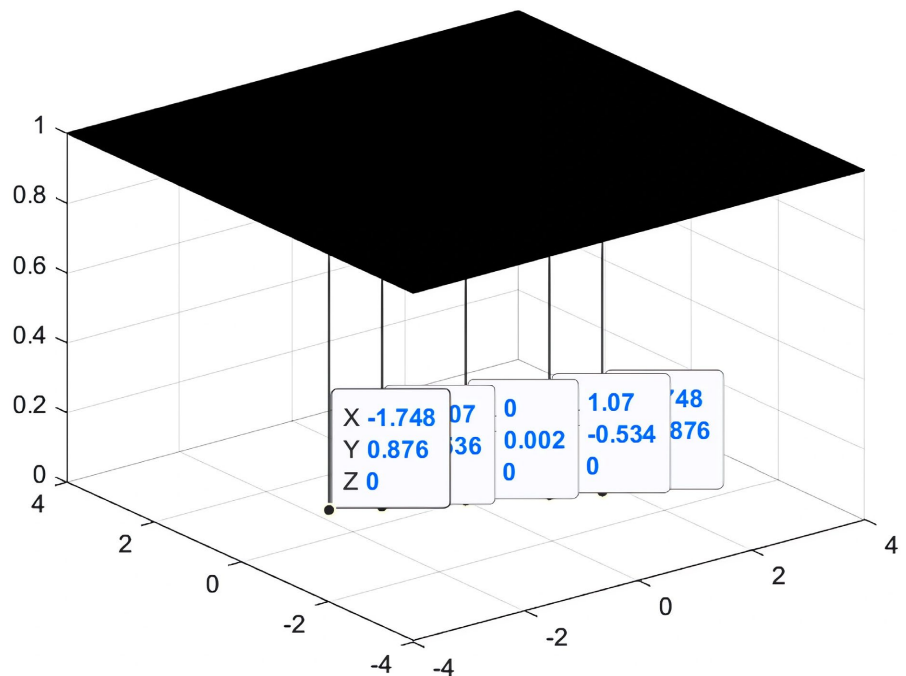


Figure 2. It shows the critical points of the Function 1 to minimize.

Figure 1 and **Figure 2** provide an illustration of the given global minimization problem and of finding of its critical points.

We received five zeros of the function $f_{1-f}(x, y)$, which is the critical points of the given differentiable function to minimize $f_1(x, y)$. The coordinates of these critical points are:

$$(x = 1.748, y = -0.876, f_{1-f}(1.748, -0.876) = 0);$$

$$(x = 1.07, y = -0.536, f_{1-f}(1.07, -0.536) = 0);$$

$$(x = 0, y = 0.002, f_{1-f}(0, 0.002) = 0);$$

$$(x = -1.07, y = 0.536, f_{1-f}(-1.07, 0.536) = 0);$$

$$(x = -1.748, y = 0.876, f_{1-f}(-1.748, 0.876) = 0).$$

We evaluate the function to minimize by substituting the correspondent coordinates x and y , we receive the values of $f_6(x, y)$ as follows:

$$(x = 1.748, y = -0.876, f_1(1.748, -0.876) = 0.2986436108);$$

$$(x = 1.07, y = -0.536, f_1(1.07, -0.536) = 0.8773619148);$$

$$(x = 0, y = 0.002, f_1(0, 0.002) = 0.000004);$$

$$(x = -1.07, y = 0.536, f_1(-1.07, 0.536) = 0.8773619148);$$

$$(x = -1.748, y = 0.876, f_1(-1.748, 0.876) = 0.2986436108).$$

Therefore, the global minimum is at $(x = 0, y = 0.002, f_1(0, 0.002) = 0.000004)$

We present a second solution, using the Substitution Method for comparison:

We received with Substitution Method the global minimum to be $f_1(0, 0) = 0$ with 78 iterations, precision = $8.2705E-9$, and time = 0.14 seconds.

These results are summarized in the next **Table 1**.

Table 1. Comparative study of the solutions of the Global Minimization Problem 1.

Algorithm	x -results	y -results	$f(x, y)$ -results	time
New Algorithm	0.000000	0.002000	0.000004	0 sec.
Subst. Algorithm	0.000000	0.000000	0.000000	0.14 sec.
GlobOpt $(x_0, y_0) = (5, 5)$	0.000011	0.000009	4.22×10^{-10}	0.687 sec.
GlobOpt $(x_0, y_0) = (-5, -5)$	-0.000021	-0.000013	1.324×10^{-9}	0.58 sec.
GlobOpt $(x_0, y_0) = (-5, 5)$	-0.000012	0.000010	2.68×10^{-10}	0.635 sec.
GlobOpt $(x_0, y_0) = (5, -5)$	0.000013	-0.000015	3.68×10^{-10}	0.593 sec.

Table 1 shows that our method achieves higher precision with less computational time than existing alternatives for global optimization.

3.2. Second Function

Let us find the Global Minimum of the function:

$$f_2(x, y) = x * e^{-(x^2 + y^2)} + (x^2 + y^2) / 20.$$

Picture of Function 2.

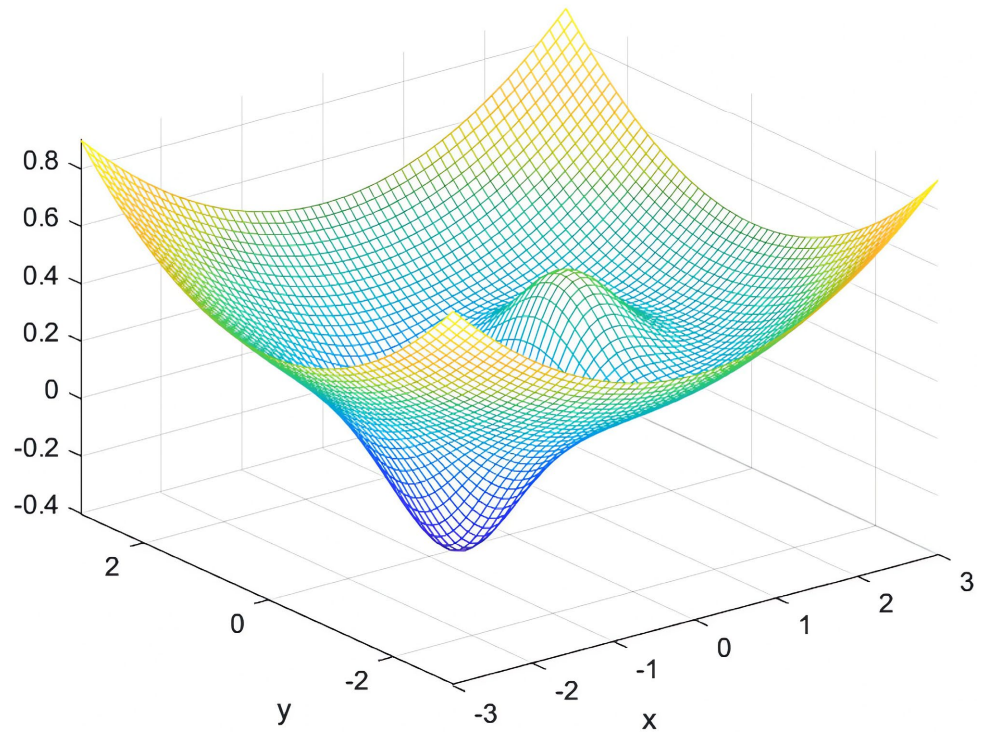


Figure 3. It shows the second surface for global minimization.

The global minimum of this function will be for $x < 0$ and the partial derivatives should be equal to zero:

$$f_{2_x}(x, y) = (1 - 2 * x^2) * e^{-(x^2 + y^2)} + x/10 = 0$$

$$f_{2_y}(x, y) = y * \left(0.1 - 2 * x * e^{-(x^2 + y^2)} \right) = 0$$

We create the function:

$$\begin{aligned} f_{2-sq}(x, y) &= \left(f_{2_x}(x, y) \right)^2 + \left(f_{2_y}(x, y) \right)^2 \\ &= \left((1 - 2 * x^2) * e^{-(x^2 + y^2)} + x/10 \right)^2 + \left(y * \left(0.1 - 2 * x * e^{-(x^2 + y^2)} \right) \right)^2 \end{aligned}$$

and graph the function:

$$\begin{aligned} f_{2-f}(x, y) &= 0.5 * \left(\text{sign} \left(\left((1 - 2 * x^2) * e^{-(x^2 + y^2)} + x/10 \right)^2 \right. \right. \\ &\quad \left. \left. + \left(y * \left(0.1 - 2 * x * e^{-(x^2 + y^2)} \right) \right)^2 - \alpha \right) + 1 \right) \end{aligned}$$

Picture of the Function f_{2-f} .

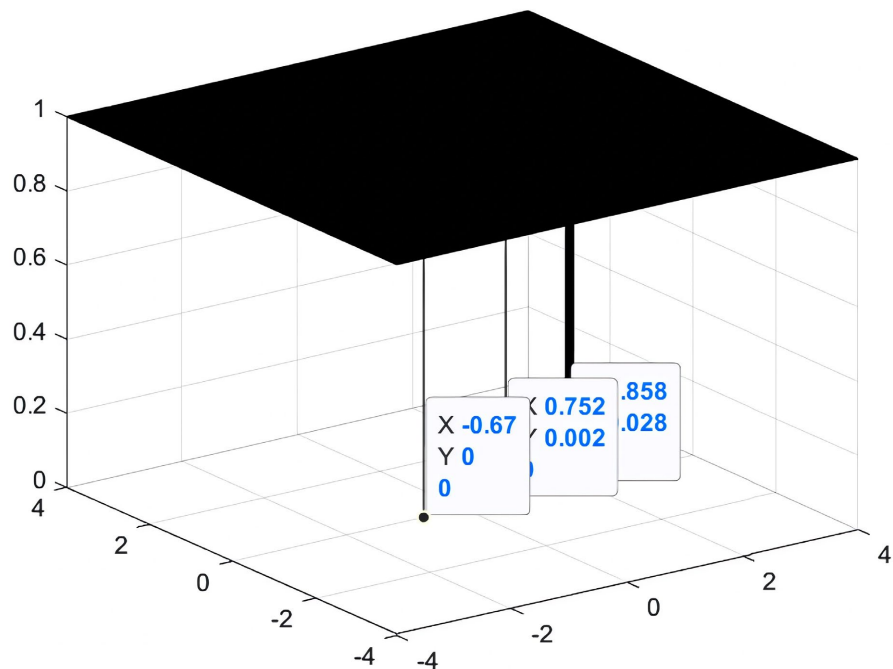


Figure 4. It shows the critical points of the Function 2 to minimize.

Figure 3 and **Figure 4** provide an illustration of the given global minimization problem and of finding of its critical points.

We found three solutions: $(x = -0.67, y = 0.0, f_{2-f}(-0.67, 0.0) = 0.0)$; $(x = 0.752, y = 0.002, f_{2-f}(0.752, 0.002) = 0.0)$ and $(x = 1.858, y = 0.028, f_{2-f}(1.858, 0.028) = 0.0)$. We evaluate now the given function to minimize $f_2(x, y)$ by substituting the correspondent coordinates of x and y and we receive the values of $f_2(x, y)$ as follows:

$$f_2(0.752, 0.002) = 0.4554651628, \quad f_2(-0.67, 0.0) = -0.4052360521 \quad \text{and} \\ f_2(1.858, 0.028) = 0.2314571654.$$

Hence, the global minimum as we predicted is for $x < 0$. It is with precision $= 10^{-6}$ at $(-0.67, 0.0)$ and it is $f_2(-0.67, 0.0) = -0.4052360521$.

We present our second approach for finding the global minimum of the given function, using the Substitution Method for comparison.

We received with Substitution Method solution: $x = -0.6691$ and $f_2(-0.6691, 0) = -0.4052$, with precision $9.3132\text{E}-9$ for 28 iterations and time less than or equal to 0.000001 sec.

These results are summarized in the next **Table 2**.

Table 2. Comparative study of the solutions of the Global Minimization Problem 2.

Algorithm	x -results	y -results	$f(x, y)$ -results	time
New Algorithm	-0.670000	0.000000	-0.4052360521	0 sec.
Subst. Algorithm	-0.669100	0.000000	-0.405200	$t \leq 0.000001$ sec.

Continued

GlobOpt $(x_0, y_0) = (-5, 5)$	-0.670200	0.000110	-0.4052356558	1.71 sec.
GlobOpt $(x_0, y_0) = (5, -5)$	-0.667500	-0.000300	-0.4052344789	1.843 sec.
GlobOpt $(x_0, y_0) = (5, 5)$	-0.668200	0.000230	-0.4052361227	1.865 sec.
GlobOpt $(x_0, y_0) = (-5, -5)$	-0.670220	-0.000210	-0.4052355973	1.7324 sec.

Table 2 shows that our method achieves higher precision with less computational time than existing alternatives for global optimization.

3.3. Third Function

Let us find the Global Minimum of the function:

$$f_3(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$$

This function is known as Himmelblau's function (see [15]; https://en.wikipedia.org/wiki/Test_functions_for_optimization).

Picture of Function 3.

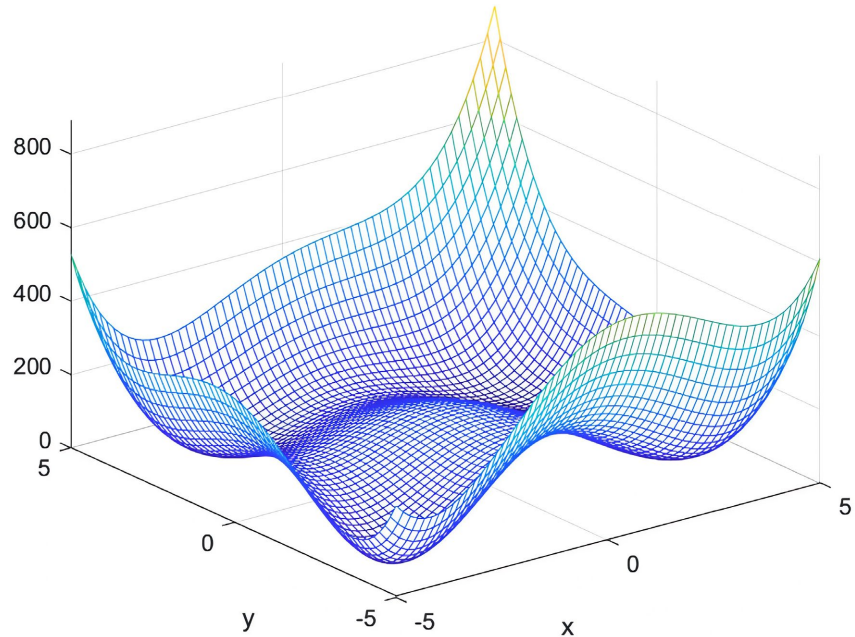


Figure 5. It shows the third surface for global minimization.

For this particular function to minimize we do not need to find the sum of the squares of the partial derivatives, because the function is a sum of two squared expressions.

We denote now with f_{3-sq} the square of the function $f_3(x, y)$:

$$f_{3-sq}(x, y) = (f_3(x, y))^2 = \left((x^2 + y - 11)^2 + (x + y^2 - 7)^2 \right)^2$$

Let us graph the function:

$$\begin{aligned} f_{3-f}(x, y) &= 0.5 * (f_{3-sq}(x, y) - \alpha) + 1 \\ &= 0.5 * (\text{sign}(f_{3-sq}(x, y) - \alpha) + 1) \\ &= 0.5 * \left(\text{sign} \left(\left((x^2 + y - 11)^2 + (x + y^2 - 7)^2 \right)^2 - \alpha \right) + 1 \right) \end{aligned}$$

Picture of the Function f_{3-f} .

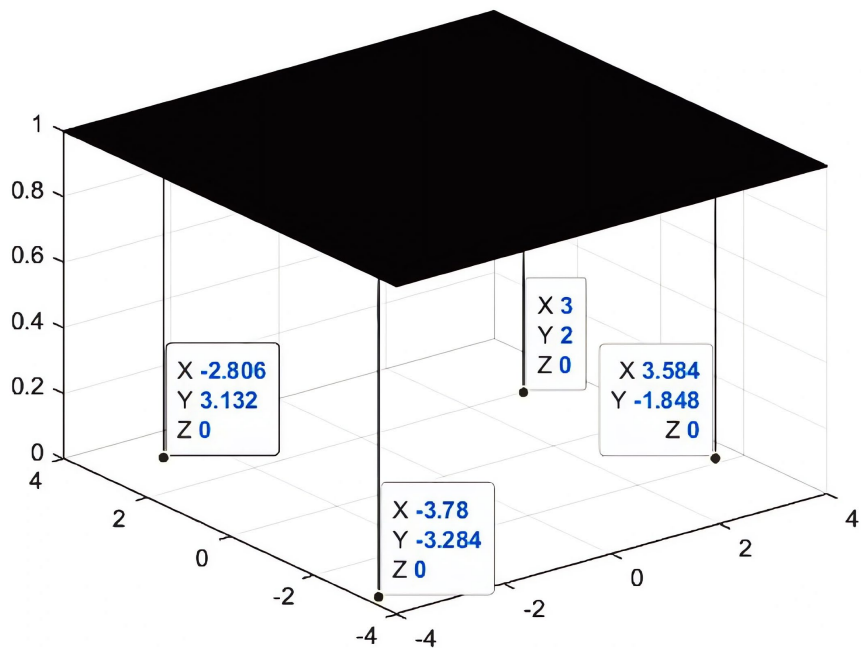


Figure 6. It shows the critical points of the Function 3 to minimize.

Figure 5 and **Figure 6** provide an illustration of the given global minimization problem and of finding of its critical points.

We received four zeros of the function of $f_{3-f}(x, y)$.

$(x = -3.78, y = -3.284, f_{3-f}(-3.78, -3.284) = 0)$;
 $(x = 3.584, y = -1.848, f_{3-f}(3.584, -1.848) = 0)$; $(x = 3, y = 2, f_{3-f}(3, 2) = 0)$
 and $(x = -2.806, y = 3.132, f_{3-f}(-2.806, 3.132) = 0)$.

We evaluate the function to minimize $f_3(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$ for the correspondent x and y coordinates of each of these points and receive:
 $(x = -3.78, y = -3.284)$ and receive $f_3(-3.78, -3.284) = 4.1038336 \times 10^{-05}$;
 $(x = 3.584, y = -1.848)$ and receive $f_3(3.584, -1.848) = 9.469952 \times 10^{-06}$;
 $(x = 3, y = 2)$ and receive $f_3(3, 2) = 0$; and $(x = -2.806, y = 3.131)$ and receive $f_3(-2.806, 3.132) = 4.3488272 \times 10^{-05}$.

Therefore, we received 4 (four) global minima with value $f_2(x, y) = 0$ with precision = 10^{-6} .

We present a second solution by decreasing the number of the independent variables, using the Substitution Method for comparison.

We received by Substitution Method with precision = 0.000001 for 857 iterations and time = 0.19 seconds:

$$\begin{aligned} f_3(-3.77930, -3.283186) &= 3.79786109 \times 10^{-12}; \\ f_3(3.5844266, -1.848127) &= 1.67682157 \times 10^{-10}; \\ f_3(3, 2) &= 0; \text{ and } f_3(-2.805118, 3.1313125) = 2.56860596 \times 10^{-13}. \end{aligned}$$

The results are summarized in the next **Table 3**.

Table 3. Comparative study of the solutions of the Global Minimization Problem 3.

Algorithm	x-results	y-results	$f(x, y)$ -results	time
New algorithm-1	3.000000	2.000000	0.000000	0 sec. for sol. 1,2,3, and 4
New algorithm-2	-3.780000	-3.284000	4.1038336×10^{-5}	0 sec. for sol. 1,2,3, and 4
New algorithm-3	3.584000	-1.848000	9.469952×10^{-6}	0 sec. for sol. 1,2,3, and 4
New algorithm-4	-2.806000	3.132000	4.3488272×10^{-5}	0 sec. for sol. 1,2,3, and 4
Subst. Algorithm-1	3.000000	2.000000	0.000000	0.19 sec. for sol. 1,2,3, and 4
Subst. Algorithm-2	-3.779300	-3.283186	$3.79786109 \times 10^{-12}$	0.19 sec. for sol. 1,2,3, and 4
Subst. Algorithm-3	3.5844266	-1.848127	$1.67682157 \times 10^{-10}$	0.19 sec. for sol. 1,2,3, and 4
Subst. Algorithm-4	-2.805118	3.1313125	$2.56860596 \times 10^{-12}$	0.19 sec. for sol. 1,2,3, and 4
GlobOpt $(x_0, y_0) = (5, 5)$	3.0000011	2.0000009	$5.62501188 \times 10^{-11}$	0.483 sec.
GlobOpt $(x_0, y_0) = (-5, -5)$	-3.779341	-3.283223	$8.362830238 \times 10^{-8}$	0.3508 sec.
GlobOpt $(x_0, y_0) = (5, -5)$	3.5844283	-1.848132	4.30291×10^{-10}	0.442 sec.
GlobOpt $(x_0, y_0) = (-5, 5)$	-2.805126	3.1313141	$2.11775266 \times 10^{-9}$	0.4075 sec.

Table 3 shows that our method achieves higher precision with less computational time than existing alternatives for global optimization.

3.4. Solving a System of 3 Nonlinear Equations

We can apply this method also to solve systems of more nonlinear equations with same or more number of independent variables. Of course, visualizations are not possible.

Let us solve the system introduced by Burden and Faires (see [16])

$$\begin{aligned} 3x_1 - \cos(x_2 x_3) - \frac{1}{2} &= 0 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 1.06 &= 0 \\ e^{-(x_1 x_2)} + 20x_3 + \frac{10\pi - 3}{3} &= 0 \end{aligned} \quad (10)$$

We solve the system by creating the function:

$$\begin{aligned}
 f_{4-sq}(x_1, x_2, x_3) = & \left(3x_1 - \cos(x_2 * x_3) - \frac{1}{2} \right)^2 \\
 & + \left(x_1^2 - 81 * (x_2 + 0.1)^2 + \sin(x_3) + 1.06 \right)^2 \\
 & + \left(e^{-(x_1 * x_2)} + 20 * x_3 + \frac{10\pi - 3}{3} \right)^2
 \end{aligned} \quad (11)$$

and solve the equation:

$$\begin{aligned}
 f_{4-f}(x, y) = & 0.5 * \left(f_{4-s}(f_{4-sq}(x_1, x_2, x_3) - \alpha) + 1 \right) \\
 = & 0.5 * \left(\left(\text{sign}(f_{4-sq}(x_1, x_2, x_3)) - \alpha \right) + 1 \right) \\
 = & 0.5 * \left(\text{sign} \left(\left(3x_1 - \cos(x_2 * x_3) - \frac{1}{2} \right)^2 \right. \right. \\
 & \left. \left. + \left(x_1^2 - 81 * (x_2 + 0.1)^2 + \sin(x_3) + 1.06 \right)^2 \right. \right. \\
 & \left. \left. + \left(e^{-(x_1 * x_2)} + 20 * x_3 + \frac{10\pi - 3}{3} \right)^2 - \alpha \right) + 1 \right) \\
 = & 0
 \end{aligned} \quad (12)$$

with a program in MATLAB (.m) file, which locates the isolated zeros of the above function: $f_{4-f}(x, y) = 0.5 * (f_{4-s}(f_{4-sq}(x_1, x_2, x_3) - \alpha) + 1)$. The next results were received:

Number of iterations: 508452, ($x_1 = 0.5000$, $x_2 = -6.1756 \times 10^{-16}$, $x_3 = -0.5200$) with precision: 0.000001 and time: 0.0323 sec.

GlobOpt program solved the given system by finding the minimum of the function:

$$\begin{aligned}
 f_{4-sq}(x_1, x_2, x_3) = & \left(3x_1 - \cos(x_2 * x_3) - \frac{1}{2} \right)^2 \\
 & + \left(x_1^2 - 81 * (x_2 + 0.1)^2 + \sin(x_3) + 1.06 \right)^2 \\
 & + \left(e^{-(x_1 * x_2)} + 20 * x_3 + \frac{10\pi - 3}{3} \right)^2
 \end{aligned} \quad (13)$$

The results are summarized in the next **Table 4**.

Table 4. It presents a comparative study with the results of other methods for solving systems of nonlinear equations.

Algorithm	x_1	x_2	x_3	time
New Algorithm	0.5000	-6.1756×10^{-16}	-0.5200	0.0323 sec.
GlobOpt $(x_{i_0}, x_{2_0}, x_{3_0}) = (5, 5, 0)$	0.5004	4.8363×10^{-08}	-0.5198	0.972 sec.
GlobOpt $(x_{i_0}, x_{2_0}, x_{3_0}) = (0, -5, -5)$	0.5008	-5.67624×10^{-08}	-0.5207	0.931 sec.
GlobOpt $(x_{i_0}, x_{2_0}, x_{3_0}) = (5, 0, 5)$	0.5012	0	-0.5196	0.883 sec.

Continued

GlobOpt $(x_{1_0}, x_{2_0}, x_{3_0}) = (-5, -5, -5)$	0.4985	-6.2185×10^{-08}	-0.5205	1.205 sec.
GlobOpt $(x_{1_0}, x_{2_0}, x_{3_0}) = (5, 5, 5)$	0.5004	3.9847×10^{-08}	-0.5192	1.013 sec.

Table 4 shows that our method also achieves higher precision and requires less computational time than existing alternatives for solving systems of nonlinear equations.

We found, with the same method, the global minima of the next functions:

1. Rosenbrock Function (Banana Function):

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2.$$

2. Beale Function:

$$f(x, y) = (1.5 - x * (1 - y))^2 + (2.25 - x * (1 - y^2))^2 + (2.625 - x * (1 - y^3))^2$$

3. Booth Function:

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

4. Matyas Function:

$$f(x, y) = 0.26 * (x^2 + y^2) - 0.48 * x * y$$

5. Ackley Function:

$$f(x, y) = -20 * e^{-0.2 \sqrt{0.5(x^2 + y^2)}} - e^{0.5 * (\cos(2\pi x) + \cos(2\pi y))} + e + 20$$

6. McCormick Function:

$$f(x, y) = \sin(x + y) + (x - y)^2 - 1.5x + 2.5y + 1$$

7. Function:

$$f(x, y) = \sin(x + y) + (x - y)^2 - 1.5 * x + 2.5y + 1$$

8. Function:

$$f(x, y) = (x - 2) * e^{-(x-2)^2 + (y-3)^2} + ((x - 2)^2 + (y - 3)^2) / 10$$

9. We solved also next system of nonlinear equations:

$$x_1 x_2^2 + x_1 x_3^2 - 1.1x_1 + 1 = 0$$

$$x_2 x_1^2 + x_2 x_3^2 - 1.1x_2 + 1 = 0$$

$$x_3 x_1^2 + x_3 x_2^2 - 1.1x_3 + 1 = 0$$

Acknowledgements

We thank to Prof. Bruce Lundberg, CSU-Pueblo, Prof. Hans Mittelman, ASU, and Mr. Joshua Strong, UAPB, for their cooperation and support.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Neumaier, A. (2004) Complete Search in Continuous Global Optimization and Constraint Satisfaction. *Acta Numerica*, **13**, 271-369.
<https://doi.org/10.1017/s0962492904000194>
- [2] Misener, R. and Floudas, C.A. (2014) ANTIGONE: Algorithms for Continuous/Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization*, **59**, 503-526. <https://doi.org/10.1007/s10898-014-0166-2>
- [3] Horst, R., Pardalos, P.M. and Thoai, N.V. (2000) Introduction to Global Optimization. 2nd Edition, Kluwer Academic Publishers.
- [4] Jaulin, L., Kieffer, M., Didrit, O. and Walter, E. (2001) Applied Interval Analysis. Springer.
- [5] Hansen, E.R. (1992) Global Optimization Using Interval Analysis. Marcel Dekker.
- [6] Zhigljavsky, A. (1991) Theory of Global Random Search. Kluwer Academic Publishers.
- [7] Hamacher, K. (2006) Adaptation in Stochastic Tunneling Global Optimization of Complex Potential Energy Landscapes. *Europhysics Letters (EPL)*, **74**, 944-950.
<https://doi.org/10.1209/epl/i2006-10058-0>
- [8] Dorigo, M. and Stützle, T. (2004) Ant Colony Optimization. The MIT Press.
<https://doi.org/10.7551/mitpress/1290.001.0001>
- [9] Granville, V., Krivanek, M. and Rasson, J.-P. (1994) Simulated Annealing: A Proof of Convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**, 652-656. <https://doi.org/10.1109/34.295910>
- [10] Nolte, A. and Schrader, R. (1997) A Note on the Finite Time Behaviour of Simulated Annealing. In: Zimmermann, U., Derigs, U., Gaul, W., Möhring, R.H. and Schuster, K.-P., Eds., *Operations Research Proceedings*, Springer, 175-180.
https://doi.org/10.1007/978-3-642-60744-8_32
- [11] Glover, F. and Laguna, M. (1997) Tabu Search. Kluwer Academic Publishers.
- [12] Slowik, A. and Kwasnicka, H. (2020) Evolutionary Algorithms and Their Applications to Engineering Problems. *Neural Computing and Applications*, **32**, 12363-12379. <https://doi.org/10.1007/s00521-020-04832-8>
- [13] Aubin, J.P. and Ekeland, I. (1984) Applied Nonlinear Analysis. Wiley.
- [14] Bauschke, H.H. and Combettes, P.L. (2017) Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer.
- [15] Himmelblau, D. (1972) Applied Nonlinear Programming. McGraw-Hill.
- [16] Burden, R.L. and Faires, J.D. (2005) Numerical Solutions of Non-Linear Systems of Equations. In: Burden, R.L. and Faires, J.D., Eds., *Numerical Analysis*, Thomson Brooks/Cole, 597-640.