

Optimal Approximation of Non-linear Power Flow Problem*

Nina Vishnyakova

ninavishn@yandex.ru

Moscow Institute of Physics and Technology (State University)

This paper addresses Alternating Current Power Flow Problem. Due to the non-linear Kirchoff equations, the set of feasible values for the parameters of the power grid is non-convex. An algorithm for highlighting the boundaries of a non-convex set is proposed. Only one point inside the feasible set is required as an initial condition. The algorithm builds two sets of points of a given density, which lie at given proximity to the border inside and outside the set. The method not only highlights the boundaries of a non-convex feasible set but also finds an approximate solution of the initial problem. If using the proposed algorithm there is no need to manually set the estimated boundaries of the feasible set in multidimensional space. The advantage in speed compared with the grid-search algorithm is equivalent to reducing the dimension of the problem by one order. The algorithm can be used as a reference solution when developing other algorithms. The method is also intended for research purposes in order to visually assess how well and correctly other algorithms for solving a problem work, for example, the ones using the restriction idea.

Index terms: *optimal power flow, non-convex optimization problem, trust region approach, convex restriction, non-convex feasible set visualisation .*

1 Introduction

In this paper, we explore the problem of optimization of the workload in the energy network in order to reduce the cost of electricity production. This is a very important topic in the power industry (1). Even small improvements lead to an economic effect.

The problem is called the AC-Optimal Power Flow problem. This optimization problem with constraints, imposed by nonlinear Kirchhoff's equations and the needs of consumers, has a convex functional and non-convex constraints. Her particular case is the classic QCQP NP-hard optimization problem. A large number of articles offer various approximate solutions. The linear approximation method suggests linearising power flow constraints. Local optimization methods are about looking for a local optimum of the AC-OPF problem. And global optimization methods propose convexifying the constraints imposed by the Kirchhoff's laws. For a general summary see (2).

We keep in mind a particular approach to solve the AC-OPF problem. This approach is based on the selection of restrictions: the convex set inside the feasible set of variables, containing the current operation point as described in (3). The optimization problem of minimization of the cost function is solved within the boundaries of this region. The resulting point is used as the starting point for constructing a new restriction in the next iteration. Repetition of this procedure, according to the sequence, that will be described, leads to the solution of the AC-OPF problem.

The solution using this approach can be broken down into several separate tasks, see section 4. One of the tasks is the recovery of one group of the problem variables: state variables, from another group: the set of control variables, such that the point specified by both sets of these variables belongs to the feasibility set.

* Academic supervisor: Vadim V. Strijov. The problem author: Michael Chertkov. Consultant: Yury Maximov.

This paper proposes a method for constructing points near boundaries and visualizing a non-convex feasible set. The method is using the procedure for checking the existence of feasible state variables corresponding to given control variables. The method starts with one feasible point, which is determined by the current state of work of the power network. Decreasing the step and calculating current results the approximation of the border in build. Because of the weakly quadratic property of the cost function to minimize, the points lying close to the borders of feasible set provide us with an approximate solution to the AC-OPF problem.

We study the performance of the method compared to the simple grid-search solution. Direct computations confirm the fast rate of convergence for the proposed method and a reduction in computational resources.

The method is easy to generalize to multidimensional case. In power grid analysis it is often enough to find the solution to the problem when only around 6 generators can be managed by the power network operator.

2 Power network structure

We take the designations that were used in (4).

Examples of data given in the tables Figure 1, Figure 4, Figure 6, are taken from file <https://github.com/lanl-ansi/PowerModels.jl/blob/master/test/data/matpower/case14.m>.

Consider a power network modeled by a n -node directed graph $\mathbf{G}(\mathcal{N}, \mathcal{E})$ where buses are represented by nodes in \mathcal{N} and transmission lines by edges in $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$.

The nodes have different types, namely, are generators or loads. Every node may include a shunt element. A generator is modeled by a complex power injection at a bus. For generator i , the injection is:

$$s_g^i = p_g^i + jq_g^i. \quad (1)$$

A load is modeled by a generator with a negative injection.

Loads with constant power are modeled by a node with a specific quantity of real and reactive power consumed. For bus i , the load is

$$s_d^i = p_d^i + jq_d^i. \quad (2)$$

The fixed generation is represented by negative quantities in these values.

A shunt element, such as a capacitor or inductor, is modeled by a fixed impedance to ground at a bus. The admittance of the shunt element at bus i is given as

$$y_{sh}^i = g_{sh}^i + jb_{sh}^i.$$

Constant impedance portions are modeled by shunt elements.

The information about variables $p_g, q_g, p_d, q_d, g_{sh}, b_{sh}$ can be found in table Figure 1, and the corresponding fields are decoded in the table Figure 2.

```

%% bus data
% bus_i type Pd Qd Gs Bs areaVm Va baseKV zone Vmax Vmin
mpc.bus = [
    1 3 0 0 0 0 1 1.06 0 0 1 1.06 0.94;
    2 2 21.7 12.7 0 0 1 1.045 -4.98 0 1 1.06 0.94;
    3 2 94.2 19 0 0 1 1.01 -12.72 0 1 1.06 0.94;
    4 1 47.8 -3.9 0 0 1 1.019 -10.33 0 1 1.06 0.94;
    5 1 7.6 1.6 0 0 1 1.02 -8.78 0 1 1.06 0.94;
    6 2 11.2 7.5 0 0 1 1.07 -14.22 0 1 1.06 0.94;
    7 1 0 0 0 0 1 1.062 -13.37 0 1 1.06 0.94;
    8 1 0 0 0 0 1 1.09 -13.36 0 1 1.06 0.94;
    9 1 29.5 16.6 0 19 1 1.056 -14.94 0 1 1.06 0.94;
    10 1 9 5.8 0 0 1 1.051 -15.1 0 1 1.06 0.94;
    11 1 3.5 1.8 0 0 1 1.057 -14.79 0 1 1.06 0.94;
    12 1 6.1 1.6 0 0 1 1.055 -15.07 0 1 1.06 0.94;
    13 1 13.5 5.8 0 0 1 1.05 -15.16 0 1 1.06 0.94;
    14 1 14.9 5 0 0 1 1.036 -16.04 0 1 1.06 0.94;
];

```

Figure 1 Bus data case 14 from IEEETable B-1: Bus Data (`mpc.bus`)

name	column	description
BUS_I	1	bus number (positive integer)
BUS_TYPE	2	bus type (1 = PQ, 2 = PV, 3 = ref, 4 = isolated)
PD	3	real power demand (MW)
QD	4	reactive power demand (MVAr)
GS	5	shunt conductance (MW demanded at $V = 1.0$ p.u.)
BS	6	shunt susceptance (MVAr injected at $V = 1.0$ p.u.)
BUS_AREA	7	area number (positive integer)
VM	8	voltage magnitude (p.u.)
VA	9	voltage angle (degrees)
BASE_KV	10	base voltage (kV)
ZONE	11	loss zone (positive integer)
VMAX	12	maximum voltage magnitude (p.u.)
VMIN	13	minimum voltage magnitude (p.u.)
LAM_P [†]	14	Lagrange multiplier on real power mismatch (u/MW)
LAM_Q [†]	15	Lagrange multiplier on reactive power mismatch (u/MVAr)
MU_VMAX [†]	16	Kuhn-Tucker multiplier on upper voltage limit ($u/\text{p.u.}$)
MU_VMIN [†]	17	Kuhn-Tucker multiplier on lower voltage limit ($u/\text{p.u.}$)

[†] Included in OPF output, typically not included (or ignored) in input matrix. Here we assume the objective function has units u .

Figure 2 Bus data from Matlab MatPower

The nodes are connected with transmission lines. All transmission lines, transformers and phase shifters are modeled with standard Π transmission line model with series impedance $z_s =$

$= r_s + jx_s$ and total charging susceptance b_c , in series with an ideal phase shifting transformers, see Figure 3.

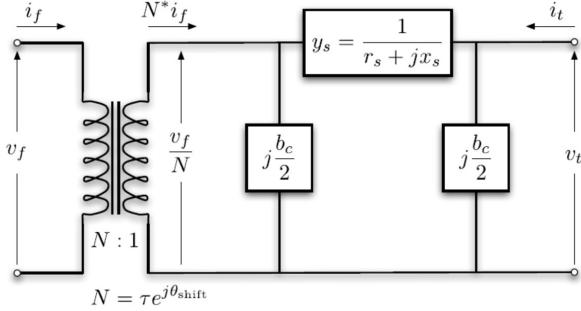


Figure 3 Branch model

The complex power injection i_f and i_t at the *from* and *to* ends of the branch respectively can be expressed in terms of the 2×2 branch admittance matrix Y_{br} and terminal voltages v_f and v_t :

$$\begin{bmatrix} i_f \\ i_t \end{bmatrix} = Y_{br} \begin{bmatrix} v_f \\ v_t \end{bmatrix}, \quad \text{где} \quad Y_{br} = \begin{bmatrix} (y_s + j\frac{b_c}{2})\frac{1}{r^2} & -y_s \frac{1}{\tau e^{j\theta_{shift}}} \\ -y_s \frac{1}{\tau e^{j\theta_{shift}}} & y_s + j\frac{b_c}{2} \end{bmatrix}.$$

The information about variables describing transmission lines Figure 4, and the transcript in table Figure 5.

```
%% branch data
%   fbus    tbus    r      x      b      rateA   rateB   rateC   ratio   angle   status   angmin   angmax
mpc.branch = [
    1     2     0.01938 0.05917 0.0528  0       0       0       0       0       1       -360     360;
    1     5     0.05403 0.22304 0.0492  0       0       0       0       0       1       -360     360;
    2     3     0.04699 0.19797 0.0438  0       0       0       0       0       1       -360     360;
    2     4     0.05811 0.17632 0.034   0       0       0       0       0       1       -360     360;
    2     5     0.05695 0.17388 0.0346  0       0       0       0       0       1       -360     360;
    3     4     0.06701 0.17103 0.0128  0       0       0       0       0       1       -360     360;
    4     5     0.01335 0.04211 0       0       0       0       0       0       1       -360     360;
    4     7     0       0.20912 0       0       0       0       0.978   0       1       -360     360;
    4     9     0       0.55618 0       0       0       0       0.969   0       1       -360     360;
    5     6     0       0.25202 0       0       0       0       0.932   0       1       -360     360;
    6     11    0.09498 0.1989  0       0       0       0       0       0       1       -360     360;
    6     12    0.12291 0.25581 0       0       0       0       0       0       1       -360     360;
    6     13    0.06615 0.13027 0       0       0       0       0       0       1       -360     360;
    7     8     0       0.17615 0       0       0       0       0       0       1       -360     360;
    7     9     0       0.11001 0       0       0       0       0       0       1       -360     360;
    9     10    0.03181 0.0845  0       0       0       0       0       0       1       -360     360;
    9     14    0.12711 0.27038 0       0       0       0       0       0       1       -360     360;
    10    11    0.08205 0.19207 0       0       0       0       0       0       1       -360     360;
    12    13    0.22092 0.19988 0       0       0       0       0       0       1       -360     360;
    13    14    0.17093 0.34802 0       0       0       0       0       0       1       -360     360;
];
```

Figure 4 Transmission line data Case 14 from IEEE

name	column	description
F_BUS	1	“from” bus number
T_BUS	2	“to” bus number
BR_R	3	resistance (p.u.)
BR_X	4	reactance (p.u.)
BR_B	5	total line charging susceptance (p.u.)
RATE_A	6	MVA rating A (long term rating), set to 0 for unlimited
RATE_B	7	MVA rating B (short term rating), set to 0 for unlimited
RATE_C	8	MVA rating C (emergency rating), set to 0 for unlimited
TAP	9	transformer off nominal turns ratio, if non-zero (taps at “from” bus, impedance at “to” bus, i.e. if $r = x = b = 0$, $\text{tap} = \frac{ V_f }{ V_t }$; $\text{tap} = 0$ used to indicate transmission line rather than transformer, i.e. mathematically equivalent to transformer with $\text{tap} = 1$)
SHIFT	10	transformer phase shift angle (degrees), positive \Rightarrow delay
BR_STATUS	11	initial branch status, 1 = in-service, 0 = out-of-service
ANGMIN*	12	minimum angle difference, $\theta_f - \theta_t$ (degrees)
ANGMAX*	13	maximum angle difference, $\theta_f - \theta_t$ (degrees)
PF†	14	real power injected at “from” bus end (MW)
QF†	15	reactive power injected at “from” bus end (MVar)
PT†	16	real power injected at “to” bus end (MW)
QT†	17	reactive power injected at “to” bus end (MVar)
MU_SF‡	18	Kuhn-Tucker multiplier on MVA limit at “from” bus (u/MVA)
MU_ST‡	19	Kuhn-Tucker multiplier on MVA limit at “to” bus (u/MVA)
MU_ANGMIN‡	20	Kuhn-Tucker multiplier lower angle difference limit (u/degree)
MU_ANGMAX‡	21	Kuhn-Tucker multiplier upper angle difference limit (u/degree)

* Not included in version 1 case format. The voltage angle difference is taken to be unbounded below if $\text{ANGMIN} \leq -360$ and unbounded above if $\text{ANGMAX} \geq 360$. If both parameters are zero, the voltage angle difference is unconstrained.

† Included in power flow and OPF output, ignored on input.

‡ Included in OPF output, typically not included (or ignored) in input matrix. Here we assume the objective function has units u .

Figure 5 Transmission line transcript from Matlab MatPower

All constant impedance elements of the model are incorporated into a complex $n \times n$ admittance matrix Y_{bus} that describe the relation of the complex nodal current injections I_{bus} to the complex node voltages V :

$$I_{bus} = Y_{bus} V.$$

Y_{bus} is a huge sparse matrix, which takes into account all transmission lines and shunts present in the energy network.

Vector V contains voltages on all the nodes. Vector I contains the corresponding currents.

The current injection I_{bus} is used to calculate the complex power injections as functions of the complex voltages V :

$$S_{bus}(V) = [V] I_{bus}^* = [V] Y_{bus}^* V^*.$$

The nodal bus injections are matched to the injections from loads and generators to form the AC nodal power balance equations. They are expressed as a function of the complex bus voltage and generator injections in complex matrix form as

$$g_S(V, S_g) = S_{bus}(V) + S_d - C_g S_g, \quad (3)$$

where S_d is vector, made up of s_d (2), S_g is vector, made up of s_g (1), and C_g - a sparse $n \times n$ generator connection matrix, where its (i, j)-th element is 1 if generator j is located as bus i and 0 otherwise. The binding of generators to nodes is determined by the table Figure 6.

```
% generator data
%      bus    Pg     Qg     Qmax   Qmin    Vg    mBase  status   Pmax   Pmin
mpc.gen = [
    1    232.4  -16.9   10      0       1.06   100     1        332.4   0
    2     40     42.4    50     -40      1.045  100     1        140     0
    3     0     23.4    40      0       1.01   100     1        100     0
    6     0     12.2    24     -6       1.07   100     1        100     0
    8     0     17.4    24     -6       1.09   100     1        100     0
];

```

Figure 6 Generators data Case 14 from IEEE

3 Problem Statement

Consider a power network modeled by a directed graph $\mathbf{G}(\mathcal{N}, \mathcal{E})$ where buses are represented by nodes in \mathcal{N} and transmission lines by edges in $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$. Let \mathcal{N}_{slack} denote the set of slack buses¹, $\mathcal{N}_{ns} = \mathcal{N} \setminus \mathcal{N}_{slack}$ - the set of non-slack buses. The sets of PV² and PQ³ buses are denoted by \mathcal{N}_{pv} and \mathcal{N}_{pq} respectively. Let $\mathcal{N}_G = \mathcal{N}_{pv} \cup \mathcal{N}_{slack}$ denote all generators.

Consider the following AC Power Flow equation (3) in polar form:

$$p_i^{inj} = \sum_{k \in \mathcal{N}} v_i v_k (G_{ik} \cos(\theta_{ik}) + B_{ik} \sin(\theta_{ik})) \quad (4)$$

$$q_i^{inj} = \sum_{k \in \mathcal{N}} v_i v_k (G_{ik} \sin(\theta_{ik}) - B_{ik} \cos(\theta_{ik})) \quad (5)$$

¹The slack bus provides for system losses. It emits or absorbs active and reactive power to and from the system.

²The name PV corresponds to letters p and v which are standard letters used to denote active power injection and voltage. PV-buses are the buses with active power injection and voltage magnitude fixed when reactive power injection and voltage phase angle are free.

³The name PQ corresponds to letters p and q which are standard letters used to denote active power injection and reactive power injection. PQ-buses are the buses with active power injection and reactive power injection fixed when voltage phase angle and magnitude are free.

Consider the following operational constraints:

$$p_i^{\min} \leq p_i^{\text{inj}} \leq p_i^{\max}, \quad i \in \mathcal{N}_G \quad (6)$$

$$q_i^{\min} \leq q_i^{\text{inj}} \leq q_i^{\max}, \quad i \in \mathcal{N}_G \quad (7)$$

$$v_i^{\min} \leq v_i^{\text{inj}} \leq v_i^{\max}, \quad i \in \mathcal{N}_{pq} \quad (8)$$

$$\varphi_l^{\min} \leq \theta_l^{\text{from}} - \theta_l^{\text{to}} \leq \varphi_l^{\max}, \quad l \in \mathcal{E}, \quad (9)$$

where $\theta_l^{\text{from}} - \theta_l^{\text{to}}$ denotes the voltage phase difference between *from* and *to* ends of the transmission lines. The operational constraints considered here are power limits and voltage magnitude limits at the generators and phase angle difference limits on transmission lines.

The power system operator has control over the generator powers, which are denoted by $u = p_{ns}^{\text{inj}}$. The corresponding internal states variables are $x = [\theta_{ns}^T \ v_{pq}^T]^T$. The system operator needs to determine the set of control variables u , subject to the power flow feasibility set in equations (4)-(9). To check the feasibility it is necessary to restore the state variables x corresponding to the control variables u — that is to solve general AC power flow problem. The equations (4) и (5) are nonlinear in relation to variables x .

An iteration of the iterative method for solving this problem is carried out according to the formula

$$x = -J_{f,0}^{-1}Mg(x, u), \quad (10)$$

where $J_{f,0}$ - jacobian is calculated at the starting point x_0, y_0 , for which all the feasibility conditions (4)-(9) are met. The equation(10) is written for the general system of nonlinear equations $f(x, u) = 0$, where $u \in \mathbb{R}^m$, $x \in \mathbb{R}^n$.

We denote basic functions $\psi : (\mathbb{R}^n, \mathbb{R}^m) \rightarrow \mathbb{R}^q$ so that

$$f(x, u) = M\psi(x, u) \quad (11)$$

where $M \in \mathbb{R}^{n \times q}$ — is constant matrix. It is convenient to use basic functions, since each basic function ψ_k depends on a small number of components of the vector x . We introduce a new function $g(x, u)$ as follows:

$$g(x, u) = \psi(x, u) - J_{\psi,0}x, \quad \text{where } J_{\psi,0} = \left. \frac{\partial \psi}{\partial x} \right|_{x_0, u_0}. \quad (12)$$

Note that $J_{f,0} = MJ_{\psi,0}$. That leads to the fact that the equation (10) is equivalent to the equations (4) - (5) when the following choice of the vector of functions ψ :

$$\psi(x, u) = \begin{bmatrix} p_i^{\text{inj}} \\ q_i^{\text{inj}} \\ v^f v^t \cos(\theta^f - \theta^t) \\ v^f v^t \sin(\theta^f - \theta^t) \\ (v^f)^2 \end{bmatrix} \quad (13)$$

The operator of the power grid changes the control variables not only to ensure the feasibility by satisfying the inequalities (6) - (9) but also to reach the minimum cost of energy production. To do this, there is a need to solve the AC - Optimal Power Flow Problem, for example, minimize the cost function:

$$\sum_{k \in \mathcal{N}_G} (c_{2k}((p_k^g)^2 + c_{1k}(p_k^g) + c_{0k}) \longrightarrow \min_{p^g} \quad (14)$$

where $c_{2k}, c_{1k}, c_{0k}, \forall k \in \mathcal{N}_G$ — generator cost components.

There are variations when functional take into account the cost of reactive power and other parameters. Also sometimes it is written in the form of a piecewise-continuous function.

4 Implementation overview

The approximate solution of the AC-OPF problem using the idea of restrictions and the "trust region approach" can be divided into the following steps. Each stage implies the solution of a separate task. These problems will be solved several times at different stages and iterations of the algorithm. Some tasks, such as 3, 5, 6, and 7, are of independent scientific interest and have already been implemented in a number of software packages. See (4), (5).

1. The task of adaptation and verification of input data. It includes checking data from various sources with different formats for correctness, making a summary of a convenient format for further work. The need for this is due to the fact that the algorithm has features, such as the separation of vertices by type (PQ, PV, slack). The cost function for optimization also needs to be verified, since it should only be in variables u and use all variables u .
2. The task of determining the starting point. In the trivial case, it is specified by the operator. The point must be verified according to the plan described in 7.
3. The task of constructing a convex contraction according to the article (3).
4. The task of analyzing the area of restriction. Includes visualization of two-dimensional slices of a non-convex feasible set and convex restriction in independent linear combinations of the variables u for research purposes. It includes drawing a sequence of any points obtained by some iterative algorithm.
5. Optimization of the function over a convex region. Solved in u variables. There can be several methods for searching for the optimum.
6. The task of obtaining all the variables describing the system from control variables (u). The paper (3) provides an algorithm that describes the iterations of the Newton-Raphson method.
7. The task of building the best projection of a point on the region with conditions of the problem. Used when the point obtained at 6 is outside the feasibility set.
8. A software product that implements managing programs that solve the above problems. The trust region approach is used here. It performs in turn: 1, 2, 7, 3, 5, 6, 7, then check the quality of the solution, and, if it is not of sufficient quality, it returns to 3.

Of particular interest is the task of restoring state variables (x) from given control variables (u). Indeed, having an algorithm for solving this problem and a sufficient computational resource, any problem associated with Power Flow can be solved by simply iterating over the grid. It is clear that this is not done because of the huge dimension of the problem. But in some cases, a brute force solution can be useful, for example, to construct a slice of a feasible region. This is important in the study of new algorithms for visual comparison of results and validation of the final result of the full algorithm for solving the AC-OPF problem.

5 An algorithm for visualizing boundaries of feasible set and finding AC-OPF Problem approximate solution

In this section we explicitly describe the operation of the proposed algorithm and give computational results on a two-dimensional test.

5.1 Description of the algorithm

To simplify the formulation of the algorithm, we consider the two-dimensional case.

Let one point A be given inside the feasible set. For example, the current state of the working system. Let us construct the second one, which lies outside the feasible set. We select any fixed step and any direction on a 2D plane. Consider the point which lies at a distance that equals the chosen step to the chosen direction from point A, and check whether it is feasible. See scheme at the left upper corner of Figure 7. Here we need to use the procedure, that was mentioned in chapter 4, which checks the existence of corresponding state variables to form a feasible point together with control variables. We won't mention this further, but imply it when talk about checking the feasibility. Sequentially doubling the shoes step the point B outside the feasible set will be obtained.

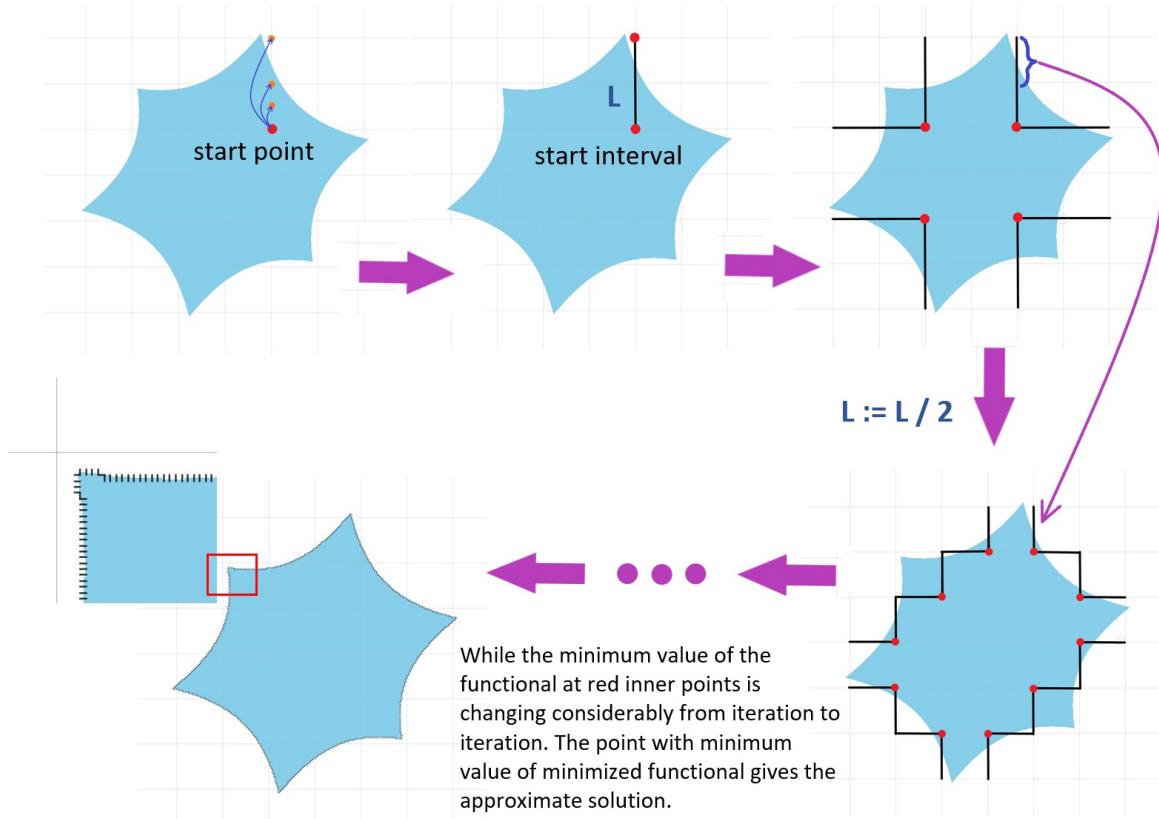


Figure 7

Let two points A and B be given, the first inside the feasible set, the second outside. They form the interval.

Having an interval, one of the ends inside the feasible set, and another outside, we begin the iteration of the main algorithm, which builds other intervals of the same length. They will

be located tightly because have the ends on the grid with the step which equals to the interval length. And all they will be intersecting the boundary.

Use the queue data structure. Place the first interval in it.

At each step, we extract the interval from the head of the queue. Check whether 4 points obtained by plotting, see Figure 8, are inside or outside the feasible set. The formed 6 intervals that have one end inside and the other outside the feasible set are placed in a queue. Continue.

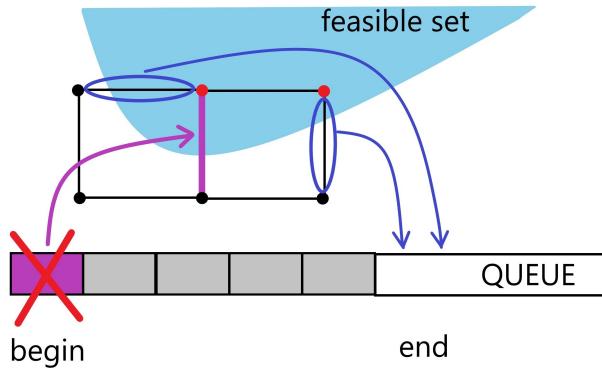


Figure 8

It is important not to put an interval in the queue if it was once put there.

Terminate when the queue is empty. Since the ends of the intervals lie on the grid with the selected step, the algorithm, having bypassed the boundary, will complete the work.

From any of the obtained intervals, we obtain an interval with a halved length that satisfies the required conditions and apply the algorithm to it, see steps at Figure 7.

Note that it is required to never check the feasibility of the point twice to get the performance advantage, but reuse the results of the previous check.

Continue these iterations until the minimum of the function on the selected interior points ceases to change significantly.

If the length of the interval is small enough, then the image of the border is acquired - the intervals merge into a solid line on a picture.

The internal point with the smallest value of the cost function which was obtained before the termination of the algorithm is the sought-for approximation for the AC-OPF problem.

This algorithm is generalized for a multidimensional case by considering multidimensional cubes instead of intervals.

5.2 Computational results

Let us consider the results of the operation of the algorithm using the example of a non-convex set shown in Figure 9 in the upper left corner. This area, in the context of this work, models the feasible set. This area is obtained by intersecting areas bounded by three hyperboles. The starting interior point has coordinates (5, 5). The starting external point has coordinates (5, 15). Thus, on the first iteration, the interval length is 10 units. At the next iteration, the interval length is halved compared to the previous one. Figure 10 shows the recovered bounds for higher iterations. Each segment crosses the boundaries of the region.

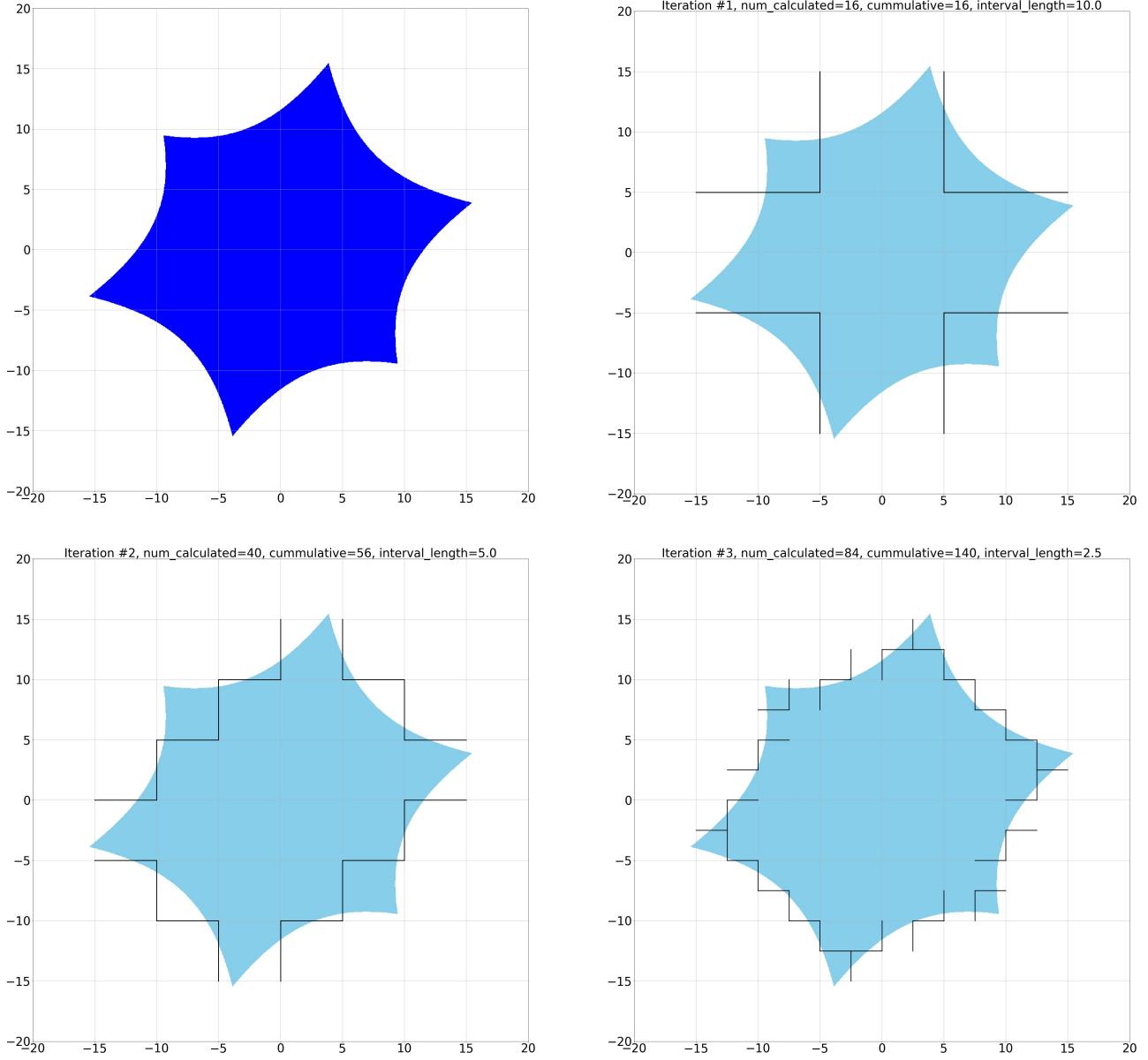


Figure 9 The first iterations of the boundaries highlighting algorithm. The blue area is the area defined by the constraints. Segments are shown by black segments. Above the graphs are written the iteration number, the number of feasibility checks at this iteration, the cumulative sum of the number of feasibility checks, taking into account the previous iterations. The coordinates are in arbitrary units.

Consider the efficiency of this algorithm by comparing the computational complexity of the proposed method with a simple grid search with the step equal to the interval length. Computational complexity is measured with the number of points for which the feasibility is checked.

Note that checking the feasibility of the point includes an iterative procedure for restoring state variables having only control variables. So the computational resource to be spent on these checks significantly prevails over other operations. The computational complexity of the grid search is inversely proportional to the square of the grid step. Therefore, if we reduce the grid step twice, the computational complexity increases by 4 times.

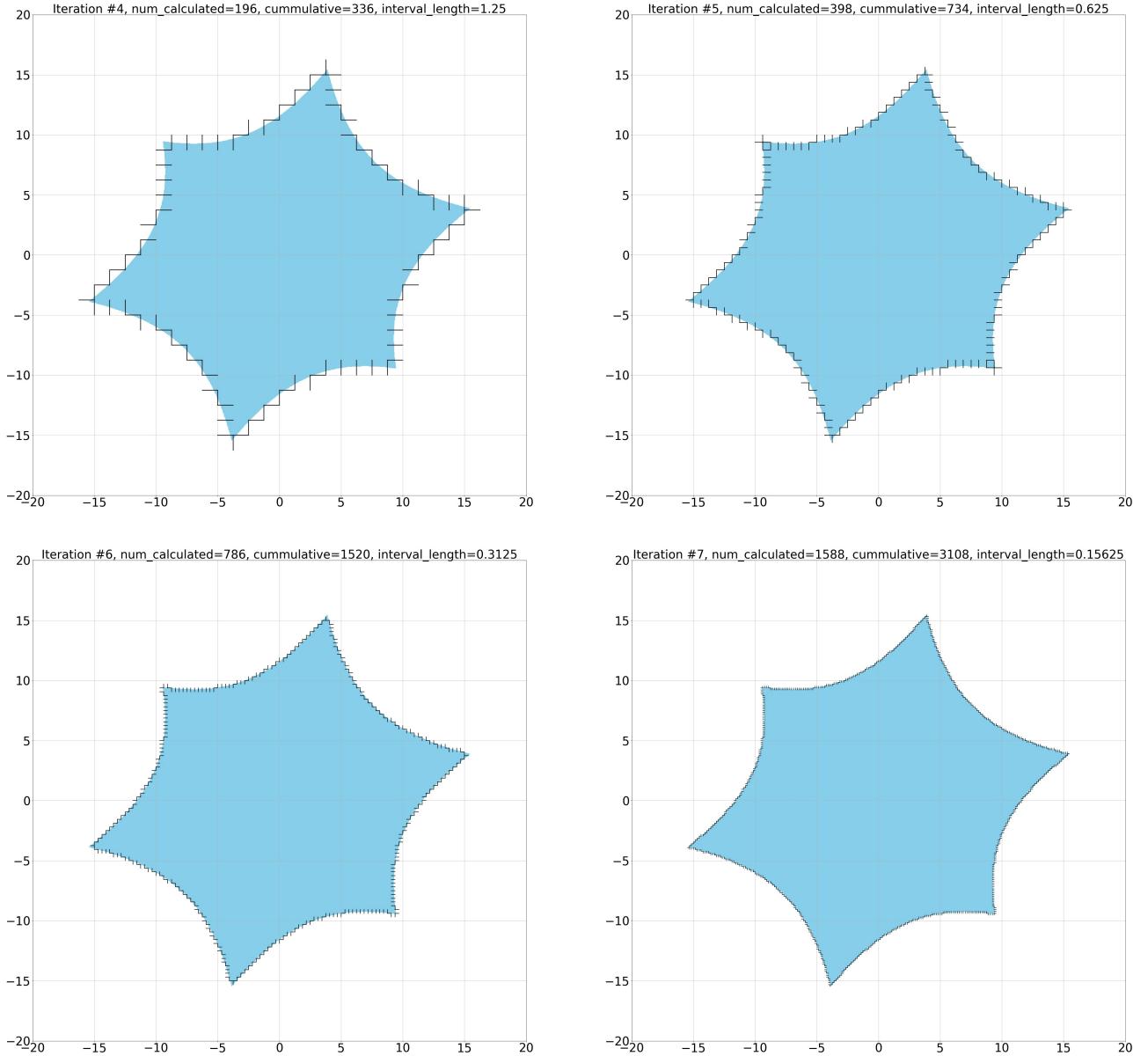


Figure 10 Further iterations of the boundaries highlighting algorithm. The interval size is halved with each iteration. At the 8th iteration, the interval length is 0.15625 arbitrary units. The set of segments is visually indistinguishable from the boundaries.

For the considered example the graph of the logarithm of computational complexity depending on the iteration number is shown in Figure 11 with a blue line. A similar graph for a proposed algorithm is shown with a red line. The relative efficiency of the algorithm at each iteration is shown by the difference between the two lines. At the 4th iteration with an interval length of 1.25, the efficiency of the proposed algorithm is $2^{3.607} \simeq 12.2$ times the efficiency of a simple iteration over the grid. For the 8th iteration - 166.3 times, and for the 12th iteration - 2619.9 times.

The advantage in speed compared with the grid-search algorithm is equivalent to reducing the dimension of the problem by one order. In reality the actual advantage over the grid-search is even more. For grid-search algorithm there is the necessity to manually specify the region,

knowingly containing the feasible set. So area for grid-search has a margin and the number of steps to get the same results, as a proposed method, will be more.

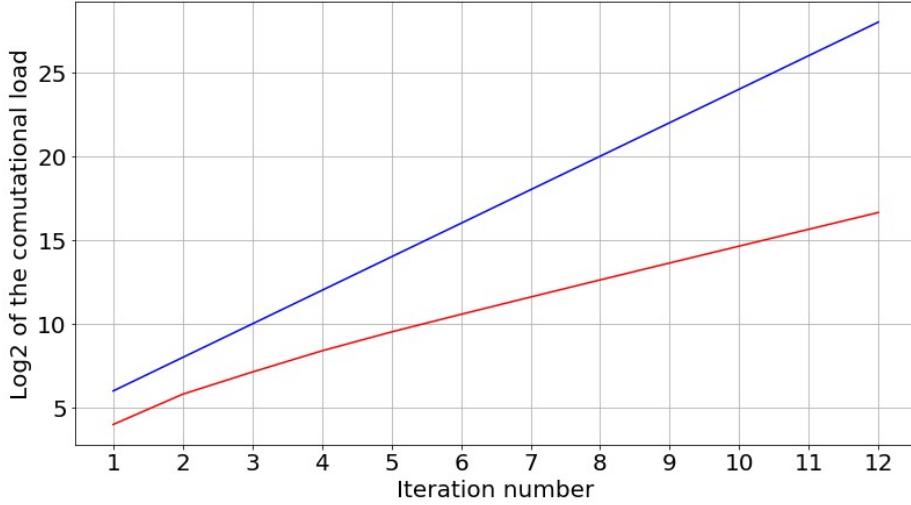


Figure 11 Graph comparing the efficiency of the economical algorithm (red line) and the brute force algorithm on the grid-search (blue line). The ratio of the efficiency of the algorithms at each iteration is 2^δ , where δ is the difference between the graphs.

6 Using the algorithm for finding the boundaries and solving AC-OPF Problem

This section provides the visualization and analysis of the approximate solution of the AC-OPF Problem for the IEEE 39 bus system, see Figure 12⁴, using the method to restore control variables from state variables and the method introduced at the previous chapter. Consider 2 dimensional slice of space denoted by control variables p_{33}^{inj} and p_{34}^{inj} . The blue and green area represents the feasibility set, orange lines are the boundaries specifying individual constraints.

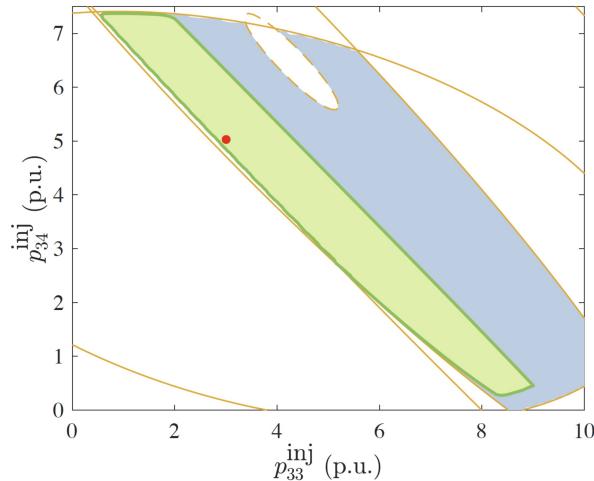


Figure 12 The example of feasible set slice for IEEE 39 bus system

⁴The picture is taken from (3).

The results are shown and described in Figure 13 and Figure 14. The convergence of the approximate solution is shown in Figure 15.

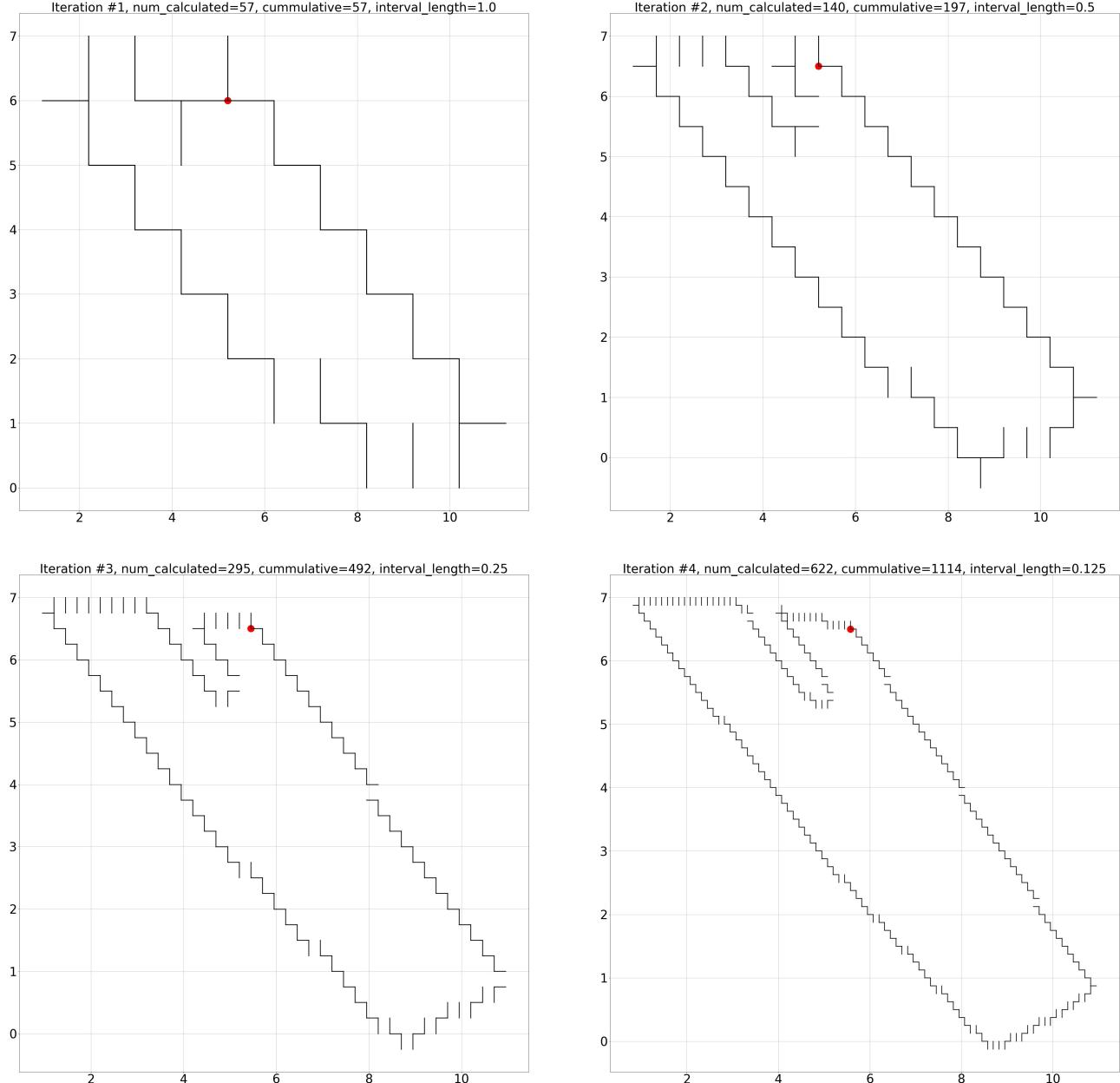


Figure 13 The result of the first iterations of feasible set boundaries highlighting algorithm (black segments) and points (red color), corresponding to the minimum of the cost function on the internal points of the obtained intervals.

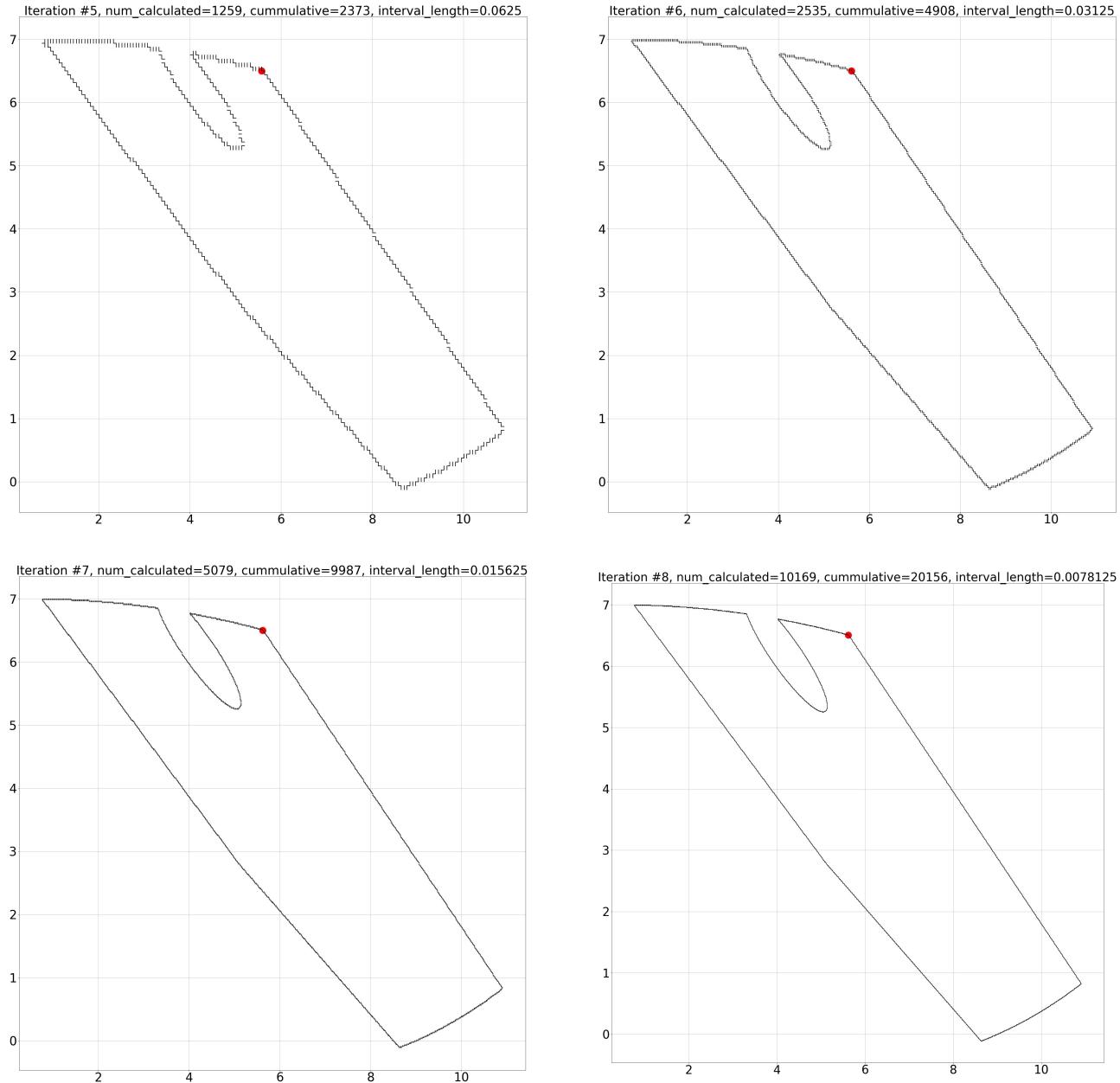


Figure 14 The results of the senior iterations for IEEE 39. The set of intervals visually form the border contour, which coincides with the border on Figure 12.

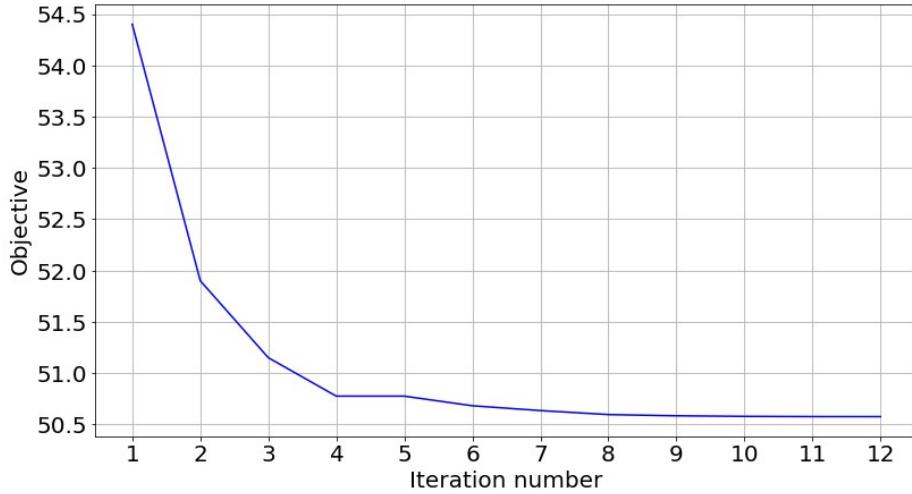


Figure 15 The number of iteration is plotted on the abscissa. The ordinate is the value of the functional at the best interior point of the set of interval ends.

7 Conclusion

The paper considers the AC - Power Flow Problem. This problem is of practical importance for optimizing the operation of power grids. The main goal of optimization is to reduce the cost of electricity generation.

The complexity of the AC-OPF problem is due to two things. Firstly, these constraint equations are non-linear. Secondly, that the problem is a high dimension one.

The structure and purpose of the data and the structure of the energy network is described. The mathematical formulation of the problem includes a set of constraints and a target function which is to minimize. Part of the restrictions are equations containing strict equalities, and the other part of the restrictions are inequalities.

The overview of the particular approach using convex restrictions idea and «trust regain approach» to solve AC-OPF is given. The particular step for reaching the global goal is considered.

A method for highlighting the boundaries of a non-convex feasible set is proposed. It uses the procedure for checking the existence of state variables for given control variables, such that the point specified by both sets of these variables belongs to the feasibility set. To start the algorithm, it suffices to specify one point inside the feasible area. The algorithm is described for a 2-dimensional case.

The set of intervals of a given length is built, which crosses the boundaries. With each subsequent iteration, the length of the interval crossing the border is halved. The procedure of forming an interval ensures that each of them is separated from each other by no more than the top of this segment and tightly covers the boundary of the area.

Note that to restore the boundary, it is not required to explicitly highlight significant constraints, and also it is not required to specify the region, where the entire feasibility set lies. Since each segment contains a point belonging to the set of feasibility, we get an approximate solution for the optimization problem.

The operation of the method in two-dimensions on the IEEE 39-bus power network is explored. In spite of the significant savings of the computational resource as compared to a simple «over the grid» iterative solution, the computational expenses for solving a problem

of large dimensionality will be extremely high. The method is intended for use in research purposes, to see how well the algorithm for constructing a restriction works or to validate the result obtained by some other solution.

8 *

References

- [1] Steven H Low. Convex relaxation of optimal power flow—part i: Formulations and equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.
- [2] Ian A. Hiskens Daniel K. Molzahn. A survey of relaxations and approximations of the power flow equations. foundations and trends. *Electric Energy Systems*, 4(1–2):1–221, 2019.
- [3] Krishnamurthy Dvijotham Konstantin Turitsyn Dongchan Lee, Hung D. Nguyen. Convex restriction of power flow feasibility sets. 2019.
- [4] Carlos E. Murillo-Sanchez Ray D.Zimmerman. Matpower user’s manual. <http://www.pserc.cornell.edu/matpower/manual.pdf>, 2018.
- [5] Carleton Coffrin, Russell Bent, Kaarthik Sundar, Yeesian Ng, and Miles Lubin. Powermodels.jl: An open-source framework for exploring power flow formulations. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–8, June 2018.