

A visual and numerical comparative study of some parallel affine projection algorithms for solving the convex feasibility problem with application to scratch inpainting

IRINA MARIA ARTINESCU and COSTIN RADU BOLDEA

ABSTRACT. The paper compares four variants of algorithms that solve the problem of Convex Feasibility using affine combinations of projections, two classical variants of Parallel Projection Method (PPM) and two modified variants that involve variable weight, in terms of their effectiveness in inpainting a convex polygon, as well as in terms of their convergence in a finite number of steps. We also present a numerical study of the dependence of the efficiency and the execution speed of these algorithms on the shape of the inpainted convex set, as well as on the values of the relaxation parameter.

1. INTRODUCTION AND PROBLEM STATEMENT.

The problem of finding a point inside a convex polygon starting from a point outside it, equivalent to the problem of identifying a solution of a set of equations or linear inequalities has been known in the literature for a long time (see [4, 9, 17, 18] for its classical approaches). These first methods used iterated projections of polygon external points onto half-spaces defined by the polygon sides. This method becomes generalized to solve the so-called *convex feasibility problem - CFP* of finding a point in the intersection of a family of closed convex sets in a Hilbert space [14, 15, 16]. A detailed mathematical description can be found in [5, 10].

The projection methods that solve CFP knew numerous applications in reconstruction of tomography images, in data compression, in image filtering and pattern generations (see also [8] for an exhaustive description), but was first involved by the first author of this paper in reconstruction of an image by inpainting a small damaged area of it [2].

Although the mathematical analysis of the convergence of various parallel projection methods used to solve the CFP was extensively described in [5], the derived algorithms do not converge generally in finite time. Few studies have been conducted in the direction of determining which algorithm converges in finite time almost everywhere in the Euclidean space. For example, only in 2007 an algorithm in finite steps was proposed, based on alternative projections on two convex sets, to solve the linear conic optimization problem [1]. But in 2018 we proved [3] that the classical EMOPP algorithm [10] does not converge in finite steps for starting points belonging to large regions of the Euclidean plane, in the case of a simple rectangular convex set. Under these circumstances, a comparative study of the convergence of various affine projection methods is needed to verify their usefulness for graphical applications.

In this paper we study the convergence of four algorithms (PPM, EMOPP and two modified versions of them) for scratch inpainting based on affine projection methods, in order to determine the average number of steps for different shapes of the initial convex

Received: 13.09.2021. In revised form: 17.10.2021. Accepted: 25.10.2021

2010 *Mathematics Subject Classification.* 68U10, 68T10.

Key words and phrases. *convex feasibility problem, parallel projection method, convergence analysis.*

Corresponding author: Irina Maria Artinescu; irina.artinescu84@e-uvt.ro

set, to verify the dependence of the convergence on the relaxation parameters and to test the ability of the algorithms to "cover" the entire surface of the convex target.

2. THE CONVEX FEASIBILITY PROBLEM (CFP) AND THE PARALLEL PROJECTION METHOD (PPM)

The CFP can be formulated generally as [10]:

Considering m closed convex sets $D_1, D_2, \dots, D_m \subseteq \mathcal{R}^n$, defined by

$D_i = \{x \in \mathcal{R}^n \mid f_i(x) \leq 0\}$, where $f_i : \mathcal{R}^n \rightarrow \mathcal{R}$ are convex functions and having nonempty intersection, that is, $\cap D_i \neq \emptyset$, find a point $x \in D = \bigcap_{i=1}^m D_i$.

The CFP can be solved mathematically, starting from an arbitrary point exterior to D , by constructing a sequence of points obtained from affine combinations of projections of the point onto some subset of $\{D_i\}$. Under some appropriate conditions, this sequence of points converges to a point inside D [5].

A first iterative method described by the sequence, known as *algebraic reconstruction technique (ART)*, is controlled by the recursive construction:

$$(2.1) \quad (\forall n \in \mathcal{N}) Q_{n+1} = P_{n(\bmod m)+1}(Q_n),$$

where P_i represents the projection onto $\{D_i\}$ with $i \in \{1, \dots, m\}$. ART was used to recover the image of an object inside the intersection of a finite number of hyperplanes [15]. This method was generalized as the *Parallel Projection Method (PPM)*, which is defined by the iteration ([12]):

$$(2.2) \quad (\forall n \in \mathcal{N}) Q_{n+1} = Q_n + \lambda_n \left(\sum_{i \in 1..n} w_i P_i(Q_n) - Q_n \right),$$

where *relaxation* parameters λ_n verify $\epsilon \leq \lambda_n \leq 2 - \epsilon$ with $0 < \epsilon < 1$ and the weight $\sum_{i \in 1..n} w_i = 1$. Combettes [10] proposed a generalization of PPM under the name *Extrapolated Method of Parallel Projections (EMOPP)*, taking the affine combination of the projections only involving a subset of them controlled by a subset of indices $\{I_n\}$ (*control* sequence), which is variable from one iteration to another:

$$(2.3) \quad (\forall n \in \mathcal{N}) Q_{n+1} = Q_n + \lambda_n \left(\sum_{i \in I_n} w_i P_i(Q_n) - Q_n \right),$$

The classical variant of EMOPP uses only the projection of an intermediary point on the convex sets D_i that do not contained it [10].

Although EMOPP is generally faster than PPM, it does not always converge in finite number of steps [3]. To correct this deficiency we introduced in the next Section two variants of PPM and EMOPP that use variable weights of the affine combination of projections, weights that depend inverse proportionally to the distance of intermediary point relative to the convex set D_i .

3. THE PROPOSED VARIANTS OF PARALLEL PROJECTION METHOD

We propose in this section an implementation of PPM and EMOPP used to cover by inpainting the entire surface of the convex target. The two supplementary variants of these algorithms, proposed by us under the name modified Parallel Projection Method (mPPM) and modified Extrapolated Method of Parallel Projections (mEMOPP), respectively, use the same iterations as PPM and EMOPP with constant $\lambda_i \equiv 1$, but involve variable weights w_i for the affine combinations of projections at each iteration. We describe the inpainting methods based on mPPM and mEMOPP in the sequel.

mPPM AND mEMOPP ALGORITHMS

Input A convex polygon $\{\mathcal{P}\} = \{P_1, P_2, \dots, P_n\}$ that simulate a damaged area of an image.

Output The image of the inpainted polygon with $NPmax$ exterior points and the descriptive analysis of the number of iterations for all points.

- 1: Define $O \leftarrow \frac{1}{n} \sum_{i=1}^n P_i$; (*The center of the polygon*)
- 2: Define an outer region of the polygon from where the algorithm chooses randomly the starting points for the iterations:

$$\mathcal{RG} \leftarrow \{Q \in \mathcal{R}^2 | d(Q, O) < 2 * r\},$$

where $r \leftarrow \max_j \{d(O, P_j)\}$.

- 3: **for** $j \leftarrow 1$ **to** $NPmax$

Generate a random point $Q_{j,0} \in \mathcal{RG} \cap Ext(\mathcal{P})$

- 4: $j \leftarrow 1$ and $k \leftarrow 0$

- 5: **repeat**

- 5.1: Determine the projections on the polygon sides:

$$M_{j,k,i} = pr(Q_{j,k}, P_i P_{i+1}),$$

by convention ($P_{n+1} = P_1$) and $d_{i,k} = dist(Q_{j,k}, P_i P_{i+1})$

- 5.2: Compute the weights

$$w_{i,k} = \frac{1/(d_{i,k} + 1)}{\sum_{i \in I} 1/(d_{i,k} + 1)}$$

where $I = \{1, 2, \dots, n\}$.

- 5.3: The next point $Q_{j,k+1}$ is determined by

$$Q_{j,k+1} = Q_{j,k} + \lambda_k \left(\sum_{i \in I} w_i M_{j,k,i} - Q_{j,k} \right)$$

until ($Q_{j,k+1} \in Int(\mathcal{P})$) or ($k > NrMaxIterations$)

- 6: **if** $Q_{j,k+1} \in Int(\mathcal{P})$ **then** $color(Q_{j,k+1}) \leftarrow color(Q_{j,k})$

- 7: **repeat from Step 5 with** $j \leftarrow j + 1$ **and** $k \leftarrow 0$.

Remark 3.1. The distances are measured in pixels and the weights $\{w_{i,k}\}$ are inversely proportional to $\{d_{i,k} + 1\}$ in order to eliminate the possible division by 0.

The control sequence λ_i is chosen to verify $1 \leq \lambda_i < 2 - \epsilon$ where $0 < \epsilon < 1$.

The modified version of EMOPP is obtained taking into consideration only the projections of $Q_{j,k}$ on the semi-planes that contain the polygon and do not contain the point $Q_{j,k}$. The weights are computed with the same formula, in the condition $\sum w_i = 1$ for $i \in I_k$ where I_k is the set of semi-planes taken into consideration at the k iteration.

Both algorithms are convergent, as a consequence of the next theorem:

Theorem 3.1. Consider a set of points recursively defined by the iteration $Q_{k+1} = f(Q_k)$ where $k \in \mathcal{N}$ and Q_0 is a fixed arbitrary point outside the polygon \mathcal{P} . Then for all points within the interior of the polygon $A \in int(\mathcal{P})$:

(1) We have

$$(3.4) \quad \|Q_{k+1} - A\|^2 \leq \|Q_k - A\|^2 - \lambda_k(2 - \lambda_k) \sum_{i \in I} w_{i,k} d_{i,k}^2$$

for (mPPM) and

$$(3.5) \quad \|Q_{k+1} - A\|^2 \leq \|Q_k - A\|^2 - \lambda_k(2 - \lambda_k) \sum_{i \in I_k} w_{i,k} d_{i,k}^2$$

for (mEMOPP), respectively, where I_k is the set of indexes i such that $P_i P_{i+1}$ separates Q_k from the rest of the polygon's \mathcal{P} vertices ($P_{n+1} = P_1$),

(2) For both algorithms:

$$(3.6) \quad \|Q_{k+1} - Q_k\|^2 \leq \frac{1}{\epsilon} (\|Q_k - A\|^2 - \|Q_{k+1} - A\|^2),$$

(3) We have $\sum_k \|Q_{k+1} - Q_k\|^2 \leq \frac{1}{\epsilon} \text{dist}(Q_0, \mathcal{P})$.

Then, for both (mPPM) and (mEMOPP) algorithms, the sequence $\{Q_k\}$ converges in norm to some interior point of \mathcal{P}

Proof. We note $M_{k,i} = pr(Q_k, P_i P_{i+1})$. The first point is obtained by the following relations (written here for mPPM algorithm):

$$(3.7) \quad \begin{aligned} & \|Q_{k+1} - A\|^2 = \\ & = \|Q_k - A + \lambda_k \cdot (\sum_{i \in I} w_{i,k} \cdot M_{k,i} - Q_k)\|^2 \\ & = \|Q_k - A\|^2 + \\ & + 2\lambda_k \left\langle Q_k - A, \sum_{i \in I} w_{i,k} \cdot M_{k,i} - Q_k \right\rangle \\ & + \lambda_k^2 \left\| \sum_{i \in I} w_{i,k} \cdot M_{k,i} - Q_k \right\|^2 \end{aligned}$$

But

$$(3.8) \quad \left\| \sum_{i \in I} w_{i,k} \cdot M_{k,i} - Q_k \right\|^2 \leq \sum_{i \in I} w_{i,k} \cdot \|M_{k,i} - Q_k\|^2$$

and we also have (see [11])

$$(3.9) \quad \langle M_{k,i} - A, M_{k,i} - Q_k \rangle \leq 0$$

Then

$$(3.10) \quad \begin{aligned} & \left\langle Q_k - A, \sum_{i \in I} w_{i,k} \cdot M_{k,i} - Q_k \right\rangle \leq \\ & - \sum_{i \in I} w_{i,k} \cdot \|M_{k,i} - Q_k\|^2 = - \sum_{i \in I} w_{i,k} d_{i,k}^2 \end{aligned}$$

The first point results from (3.7), (3.8) and (3.10). For the (mEMOPP) algorithm, the set of indices I is replaced by I_k .

The second point is obtained by

$$\begin{aligned} \|Q_{k+1} - Q_k\|^2 & = \lambda_k \left\| \sum_{i \in I} w_{i,k} M_{k,i} - Q_k \right\|^2 \\ & \leq \lambda_k \left(\sum_{i \in I} w_{i,k} \|M_{k,i} - Q_k\|^2 \right) \\ & \leq \lambda_k \frac{\|Q_k - A\|^2 - \|Q_{k+1} - A\|^2}{\lambda_k (2 - \lambda_k)} \\ & \leq \frac{1}{\epsilon} (\|Q_k - A\|^2 - \|Q_{k+1} - A\|^2) \end{aligned}$$

for any point A interior to the polygon \mathcal{P}

The third statement derived from

$$\begin{aligned} \sum_{k=0}^n (\|Q_k - A\|^2 - \|Q_{k+1} - A\|^2) &= \|Q_0 - A\|^2 - \\ &- \|Q_n - A\|^2 \\ &\leq \|Q_0 - A\|^2 \end{aligned}$$

for any $C \in \text{int}(\mathcal{P})$, then

$$(3.11) \quad \sum_{k=0}^n \|Q_{k+1} - Q_k\|^2 \leq \frac{1}{\epsilon} \text{dist}(Q_0, \mathcal{P})^2$$

The convergence of Q_i is assured by the Corollary 3.3 from [5].

□

Note that the convergence of the algorithm depends on the initial distance from Q_0 to the convex polygon and also on the set $\{\lambda_k\}$, while the convergence of the algorithms is not assured for $\lambda_k \geq 2$.

In order to give a more accurate evaluation of the dependence of the algorithms convergence on the relaxation parameters, we presented a series of numerical experiments in the next Section.

4. ANALYSIS OF THE CONVERGENCE OF ALGORITHMS

The four methods described before (PPM, EMOPP, mPPM, mEMOPP) were implemented in C using a Borland Graphics Interface (BGI) emulator and tested to images that include various regions affected by an induced loss of information. We chose two different polygonal shapes: a long rectangle and a non-regular hexagon.

The number of vertices of the convex polygon affects directly the number of computations and the areas from where the points are transferred into the polygon. We imposed a fixed number of $NPM_{max} = 10,000$ starting points and a limit of $k > NrMaxIterations = 45$. The original image and the two polygonal shapes were presented in Figure 1 and 2.

The purpose of this implementation was to visualize where the four algorithms insert the exterior points into the predetermined convex polygon. If the PPM and the mPPM algorithms use a continuous function to determine the next iterated point $Q_{j,k} \rightarrow Q_{j,k+1}$, the EMOPP and mEMOPP involve a variable number of projections at each iteration, depending on the position of $Q_{j,k}$ related to the polygon, and then these algorithms do not use a continuous transformation at each iteration.

We tested the dependence of the convergence of the algorithms on the relaxation parameter Λ and on the capacity of each algorithm to "fill in" the polygonal shape, property that assures that the solutions of the convex feasibility problem, depending on the starting points, cover uniformly the interior of the given convex set.

The implemented algorithms produced a list of iterated positions ($Q_{j,k}$) and a short descriptive statistic of number of iterations executed for each starting points. The results for the four algorithms are presented in the next Tables 1 and 2, for the relaxation parameters $\lambda_i = \Lambda = 1.1$.

One observes that if the application of the weight factor depends on the position of the points, the number of iterations increases significantly for the case of mPPM algorithm, but not for the mEMOPP algorithm. The convergence of mEMOPP is comparable with the case when the weights are constants.

The same experiments were repeated for each value of $\Lambda \in \{1.1 + 0.2 \cdot i \mid i = 1, 5\}$. (Note that the convergence of the PPM and EMOPP method are not assured for $\Lambda \geq 2$ [10].)

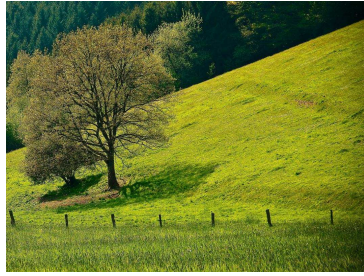


FIGURE 1. The initial image.

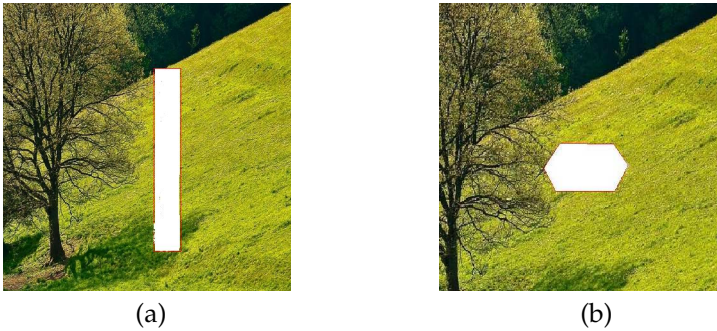


FIGURE 2. The damaged images for a long rectangle (a) and a hexagonal convex (b).

TABLE 1. The distribution of the starting points so that the algorithms stop in k steps for a long rectangle.

Number of iterations	Starting points(%) for PPM	Starting points(%) for mPPM	Starting points(%) for EMOPP	Starting points(%) for mEMOPP
1	13.35	6.32	48.07	48.78
2	20.28	10.38	2.62	2.18
3	46.22	13.47	1.33	7.47
4	16.24	13.30	1.64	12.51
5	0	11.65	1.50	10.89
6	0	12.07	1.44	3.91
7	0	9.21	1.27	0.93
8	0	6.78	1.15	0.15
9	0	4.74	0.93	0.09
≥ 10	3.91	12.08	40.05	13.08

The dependence of the number of iterations on the value of Λ is represented in the next figure.

From the numerical data displayed in Figure 3, one can notice a direct dependence of the convergence rate on the relaxation parameter Λ , more accentuated for the mPPM and mEMOPP algorithms, phenomenon that was not predicted by the superior limitation from the Theorem 3.1.

In order to analyze the final solution image (the region inside the polygon where the solution's points are distributed), we tested all the algorithms for each Λ (see Figure 4).

TABLE 2. The distribution of the starting points so that the algorithms stop in k steps for a hexagonal convex set.

Number of iterations	Starting points(%) for PPM	Starting points(%) for mPPM	Starting points(%) for EMOPP	Starting points(%) for mEMOPP
1	46.64	4.67	14.74	15.67
2	41.29	9.27	32.34	31.36
3	0	14.86	31.65	25.08
4	0	16.68	6.56	9.35
5	0	12.71	1.62	3.60
6	0	8.08	0.49	1.65
7	0	4.80	0.13	0.65
8	0	5.93	0.03	0.14
9	0	6.23	0	0.07
≥ 10	12.07	16.77	12.44	12.44

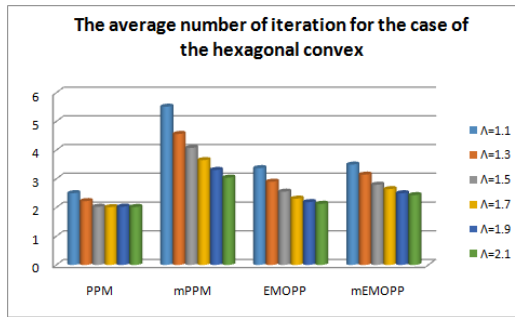


FIGURE 3. The average number of iterations for the case of the hexagonal convex set.

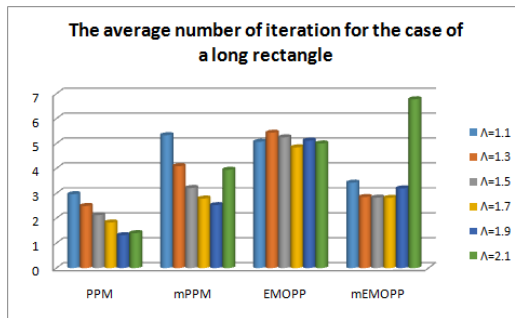


FIGURE 4. The average number of iterations for the case of a long rectangle.

The significant differences between the average number of iterations of the EMOPP algorithm in the case of a rectangular versus the case of hexagonal shape of the convex set is the result of the fact that for starting point near the prolongation of the diagonals of the rectangle, the EMOPP do not converge in finite number of steps. (Our implementation

limited the number of steps to 45.) The mEMOPP possesses larger convergence regions than EMOPP for $\Lambda < 2$, but does not always converge for $\Lambda > 2$.

In the second part, we have analysed the dependence of the execution time on the relaxation parameter Λ for the two cases of a convex target. The results are centralized in Table 3 and Table 4.

One can notice that, if the execution time do not depend significantly on the value of relaxation parameter Λ for the case of PPM and mPPM algorithms, the EMOPP and mEMOPP algorithms are faster for $\Lambda = 1.7, 1.9$. The increased values of execution time for $\Lambda = 2.1$ is the result of multiplication of the starting points for which the algorithms do not converge in finite steps, in the case of mEMOPP. Finally, there are significant differences between the first two algorithms, on the one hand, and the last ones, on other hand, in what concern the speed of convergence.

In order to test the "efficiency" of the algorithms to "cover" the entire surface of the convex target, we systematized the spatial distribution of the solutions obtained for all the considered algorithms, only for the case of the relaxation parameter $\Lambda = 1.1, \Lambda = 1.7$ and $\Lambda = 2.1$ in the Table 6 and Table 7 from the Annexe. For small values of Λ , the solutions of the mPPM, EMOPP and mEMOPP are concentrated near the borders of the initial considered convex polygon. For increased Λ , the filled region becomes larger in the case of these three algorithms. Only for EMOPP and mEMOPP the entire surface of the convex set will be covered, for $\Lambda \geq 1.9$, more efficiently in the case of mEMOPP.

In the end of our work, we investigated the shape and the dimension of the finite steps convergence regions. Table 9 presents comparatively the regions of initial points, exterior to the polygon, from where the studied algorithms converge in one step (in red), two steps (in blue), respectively three steps (in green), in the case of the hexagonal initial convex. If for PPM, EMOPP and mEMOPP algorithms the borders of these convergence regions seem to be defined by parallels to the polygon sides and diagonals, for mPPM algorithm the convergence regions are similar to a fractal structure. A future study will be conducted to investigate the existence of this fractal shape of the convergence region.

TABLE 3. The comparative execution times for different values of Λ in the case of a long rectangle.

Λ	PPM	mPPM	EMOPP	mEMOPP
1.1	1069	1041	907	594
1.3	1034	1031	786	754
1.5	1026	1001	851	498
1.7	1018	993	757	540
1.9	997	981	656	529
2.1	1065	991	844	749

TABLE 4. The comparative execution times for different values of Λ in the case of a hexagon.

Λ	PPM	mPPM	EMOPP	mEMOPP
1.1	941	943	455	448
1.3	959	931	766	703
1.5	934	927	742	457
1.7	1020	920	460	454
1.9	926	899	486	459
2.1	953	901	463	628

5. CONCLUSIONS

The comparative analysis of the graphical implementations for the PPM, EMOPP and their modified versions proposed in Section 2 allows to identify the way in which the variable weight and the relaxation factor Λ influence the number of iterations calculated until reaching the solution of the convex feasibility problem. The results of our experiments are consistent with the analysis of the effectiveness of projection methods for convex feasibility problems presented in [8] for the PPM and EMOPP.

The graphical comparison allows also to verify the spatial distribution of the solutions, depending on the shape of the convex polygon and the relaxation factor. Even if the methods are not efficient to recover a damaged image by inpainting, in the case of $\lambda \simeq 1.7, 2.1$ the polygon is uniformly covered by the solution points, for EMOPP and mEMOPP. The distribution of the solutions depends strongly on the number of edges and the shape of the convex polygon.

All previous specialized studies analyze the existence of a single solution to the Convex Feasibility Problem. The article proposes for the first time an analysis of the uniform coverage of a convex (understood as intersections of closed half-planes) using iterative methods of solving CFP, which can lead to determining its shape and size using PPM or methods derived from it.

Particular emphasis was placed on the dependence of the execution speed on the relaxation coefficient used in iterations, a parameter that seems to be of great importance for the correct coverage of the regions affected by information loss. The relaxation parameter influences more the convergence of the algorithms than the weights and also modifies the dimension of the convergence regions. The convergence regions (defined by the starting points for which the algorithms converge in a given number of steps) presented a very interesting fragmentation for the EMOPP and mEMOPP, and a fractal shape for mPPM.

We choose this type of implementation in order to verify if the analyzed methods for solving the convex feasibility problem can be used to recover a damaged image by inpainting points from the exterior in the affected area. In conclusion, the EMOPP and mEMOPP method can be better used for this purpose, but only for small regions affected and a good choice of relaxation parameter Λ ; moreover, the image of the covered area does not reproduce the neighborhood of the convex polygon. All the results of this paper are obtained by numerical experiments, therefore more theoretical results must be obtained in order to sustain the presented analysis of the convergence of the four algorithms.

REFERENCES

- [1] Ait Rami, M.; Helmke, U.; Moore, J. B. A finite steps algorithm for solving convex feasibility problems. *J. Glob. Optim.* **38** (2007), no. 1 143–160.
- [2] Artinescu, I. M.; Maftciu-Scai, L. O. A Scratch Covering Algorithm using Affine Projection Method. *Math. Comp. Scien.*, **12** (2018a), 235–246.
- [3] Artinescu, I. M. A comparative analysis of the convergence regions for different parallel affine projection algorithms. *Stud. Univ. Babeş-Bolyai Math.* **63** (2018b), 401–411.
- [4] Agmon, S. The relaxation method for linear inequalities. *Can. J. Math* **6** (1954), 382–392.
- [5] Bauschke, H. H.; Borwein, J. M. On Projection Algorithms for Solving Convex Feasibility Problems. *SIAM Review* **38** (1996), 367–426.
- [6] Bertalmio, M.; Sapiro, G.; Caselles, V.; Ballester, C. Image inpainting. *ACM Comput. Graph. (SIGGRAPH 2002) SIGGRAPH: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, New Orleans, ACM Press/Addison-Wesley, New York, (2000), 417–424.
- [7] Bertalmio, M.; Vese, L.; Sapiro, G.; Osher, S. Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.*, **12** (2003), no. 8, 882–889.
- [8] Censor, Y.; Chen, W.; Combettes, P.; David, R.; Herman, G. On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints. *Comput. Optimiz. Applic.*, Kluwer Academic Publishers Norwell, USA, April, **51** (2012), 1065–1088.

- [9] Cimmino, G. Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari. *La Ricerca Scientif.* (Roma), **1** (1938), 326–333.
- [10] Combettes, P. L. The Convex Feasibility Problem in Image Recovery. *Advan. Imag. Elec. Physics* **95** (1996), 155–270.
- [11] Combettes, P. L. Hilbertian Convex Feasibility Problem: Convergence of Projection Methods. *Appl. Math. Optim.* **35** (1997a), 311–330.
- [12] Combettes, P. L. Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections. *IEEE Transact. Im. Proc.* **6** (1997b), 493–506.
- [13] Criminisi, A.; Perez, P.; Toyama, K. Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Transact. Im. Proc.*, **13** (2004), 1200–1212.
- [14] Gilbert, P. Iterative methods for the three-dimensional reconstruction of an object from projections. *J. Theor. Biol.* **36** (1972), 105–117.
- [15] Gordon, R.; Bender, R.; Herman, G. T. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. Theor. Biol.* **29** (1970), 471–481.
- [16] Gubin, L. G.; Polyak, B. T.; Raik, E. V. The method of projections for finding the common point of convex sets. *USSR Comput. Mathe. Math. Physics*, **7** (1967), 1–24.
- [17] Merzlyakov, Y. I. On a relaxation method of solving systems of linear inequalities. *USSR Comput. Math. Math. Physics* **2** (1963), 504–510.
- [18] Motzkin, T. S.; Schoenberg, I. J. The relaxation method for linear inequalities. *Canad. J. Math.* **6** (1954), 393–404.

WEST UNIVERSITY OF TIMIȘOARA
DEPARTMENT COMPUTER SCIENCE
VASILE PARVAN 4, TIMIȘOARA, ROMANIA
Email address: irina.artinescu84@e-uvvt.ro

UNIVERSITY OF CRAIOVA
DEPARTMENT COMPUTER SCIENCE
ROMANIA
Email address: cboldea@inf.ucv.ro

TABLE 5. The filled regions defined by the solution of the convex feasibility problem, for different algorithms and different values of Λ (the case of the hexagonal convex set)

Λ	1.1	1.7	2.1
PPM			
mPPM			
EMOPP			
mEMOPP			

TABLE 6. The filled regions defined by the solution of the convex feasibility problem, for different algorithms and different values of Λ (the case of the hexagonal convex set)

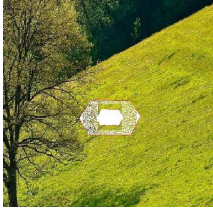

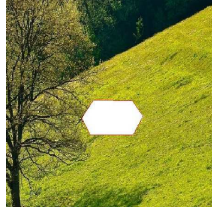
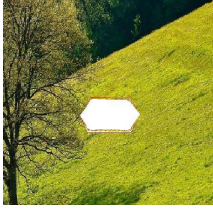
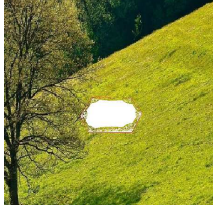


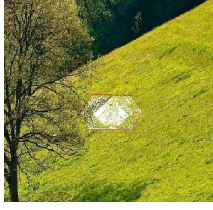




Λ	1.1	1.7	2.1
PPM			
mPPM			
EMOPP			
mEMOPP			

TABLE 7. The filled regions defined by the solution of the convex feasibility problem, for different algorithms and different values of Λ (the case of a long rectangle)



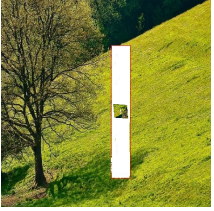
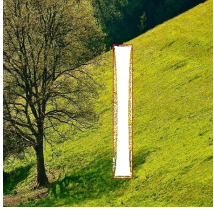


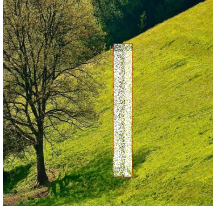





Λ	1.1	1.7	2.1
PPM			
mPPM			
EMOPP			
mEMOPP			

TABLE 8. Convergence zones for the case of the hexagonal convex set

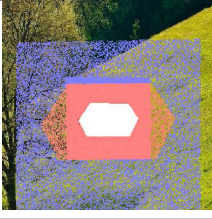
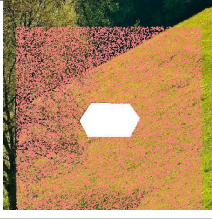
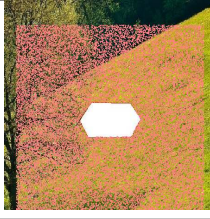
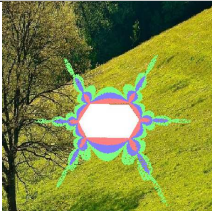
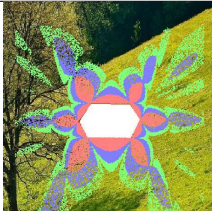
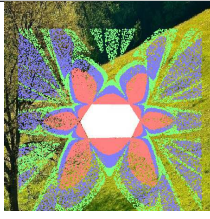
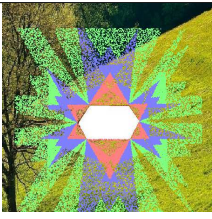
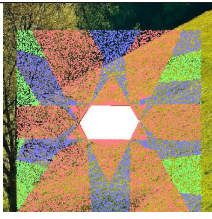
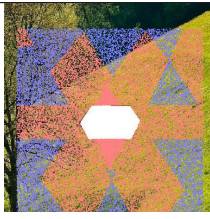
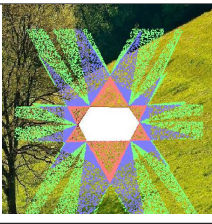
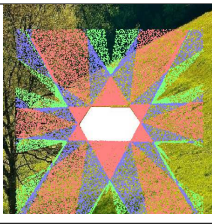
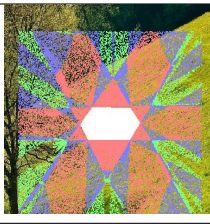
Λ	1.1	1.7	2.1
PPM			
mPPM			
EMOPP			
mEMOPP			

TABLE 9. Convergence zones for the case of the rectangle

Λ	1.1	1.7	2.1
PPM			
mPPM			
EMOPP			
mEMOPP			