Numerical Algorithms (2019) 81:1149–1155
https://doi.org/10.1007/s11075-018-0589-9

ORIGINAL PAPER

On the problem of starting points for iterative methods

Ştefan Măruşter[1] · Laura Măruşter[2]

Received: 11 January 2018 / Accepted: 21 August 2018 / Published online: 19 September 2018
© The Author(s) 2018

Abstract
We propose an algorithm to find a starting point for iterative methods. Numerical experiments show empirically that the algorithm provides starting points for different iterative methods (like Newton method and its variants) with low computational cost.

Keywords Iterative methods · Starting point · Numerical experiments

1 Introduction

Let $f : \mathbb{R}^m \to \mathbb{R}^m$ be a nonlinear mapping and Φ an iteration function with whose help we try to solve the nonlinear equation $f(x) = 0$, using the iterative scheme $x_{n+1} = \Phi(x_n)$. This means that the equation $f(x) = 0$ is equivalent to the fixed point problem $x = \Phi(x)$. The aim is to provide a good starting point x_0, usually sufficiently close to the solution. In this paper, we propose a new algorithm to find a starting point for the considered (Picard) iteration, which is experimentally verified.

The convergence of the iterative methods for various classes of operators was already investigated from early decades (see, for instance [1]). To find a suitable starting point for iterative methods, or to compute the convergence ball, numerous results were obtained especially for the Newton method and its variants. Several methods were developed to find the starting points or to localize the fixed points, such as the generalized bisection method [2, 3], cell exclusion [4], interval computing [5], homotopy continuation method [6], and random search [7]. This problem is still studied extensively and specialized methods are proposed. However "... effective, computable estimates for convergence radii are rarely available" [8] (1975), "... a priori knowledge about the radius of convergence of the local iterative procedure to be used is unknown, in general" [4] (1996). Similar remarks were made in more

Ştefan Măruşter is deceased.

✉ Laura Măruşter
l.maruster@rug.nl

1 West University of Timisoara, B-ul V. Parvan, No.4, 300223, Timisoara, Romania

2 University of Groningen, PO Box 800, 9700 AV, Groningen, The Netherlands

recent papers: "... no estimate for the size of an attraction ball is known" [9] (2009), "The location of starting approximations, from which the iterative methods converge to a solution of the equation, is a difficult problem to solve" [10] (2015).

Let Φ^k be the k iterate of Φ, $d\Phi^k(x)$ the derivative (Jacobian) of $\Phi^k(x)$ in the point x and $\varrho(d\Phi^k(x))$ the spectral radius of $d\Phi^k(x)$. The proposed algorithm is based on the following two facts:

1. Let p be a solution of the equation $f(x) = 0$ (or a fixed point of Φ) and $\mathcal{R}$ a region around p. If the numerical sequence $\{\varrho(df^k(p))\}$ is strictly decreasing, then the set sequence $C_k = \{x : x \in \mathcal{R},\ \varrho(df^k(x)) < 1\}$ is relatively increasing and tends to a limit set C (see Section 3, Experiment 1).
2. The limit set C is the convergence domain (attraction basin) of the Picard iteration for Φ.

Studying the sequence of sets C_k was inspired by [11] where the author formulated the following problem: In which conditions do we get that

$$\varrho(df(x)) < 1,\ \forall x \in \Omega\ \Rightarrow \varrho(df^k(x)) < 1,\ \forall x \in \Omega,\ k \in \mathbb{N}^*?$$

In the present study, we experimentally investigate the characteristics of the proposed algorithm from the computational effort point of view. As the computational effort is determined by the derivative of Φ^k (for large values of k the computation of $d\Phi^k(x)$ is exceedingly expensive, both symbolically and numerically) and by the number of net points n, we are concerned with the influence of k and n on the cost for computing the set C_k. In fact, we try to answer to the following question: How should the pair k, n be chosen such that $C_k \neq \emptyset$?

2 The algorithm

The algorithm has the following three main steps:

1. Compute the iterates of Φ for some k, Φ^k;
2. Compute the derivative (Jacobian) $d\Phi^k(x)$ of Φ^k on some net of points;
3. Find the set of points for which $\varrho(d\Phi^k(x)) < 1$;

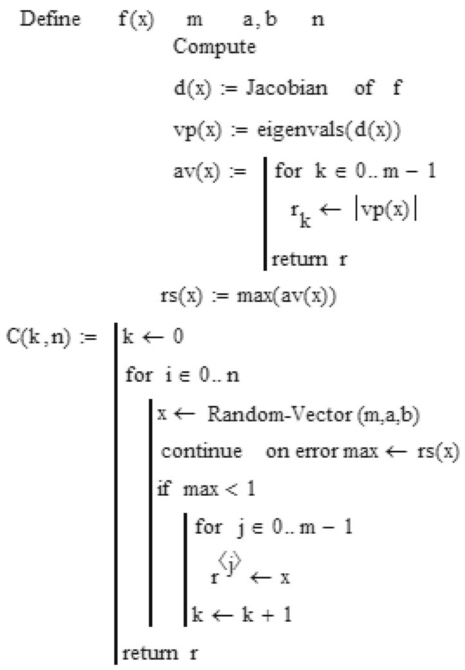
In Fig. 1 is shown the algorithm.

As, presumptively, the set C_k is close to the attraction basin, any point in this set can be a starting point for the considered iteration.

The net of points on which $\varrho(d\Phi^k(x))$ should be computed (to find those points which satisfies $\varrho(d\Phi^k(x)) < 1$) is usually inside of a given n-dimensional region. The attempt to use a brute force approach, that is going through all points of a uniform net, leads to a high computational cost. For mappings with moderate dimensions, this computation is not very computationally challenging, but for larger dimensions, it can become very expensive.

To implement the algorithm on a digital computer, a number of computational issues must be addressed. The following three seems to be the most important: the

Fig. 1 The algorithm to compute the set C_k



```
Define    f(x)    m    a,b    n
        Compute
        d(x) := Jacobian  of  f
        vp(x) := eigenvals(d(x))
        av(x) := | for k ∈ 0..m − 1
                 |    r_k ← |vp(x)|
                 | return r
        rs(x) := max(av(x))
C(k,n) := | k ← 0
          | for i ∈ 0..n
          |    | x ← Random-Vector (m,a,b)
          |    | continue  on error max ← rs(x)
          |    | if max < 1
          |    |    | for j ∈ 0..m − 1
          |    |    |    r^<j> ← x
          |    |    | k ← k + 1
          | return r
```

value of k, the region in which the starting points are searched, and the net of points. In Section 3, concerning the numerical examples, we use $k = 0, 1, 2$, a cube centered in a plausible point on the bissectrice of coordinate axes, and a net of points given by a function $Random-Vector(m, a, b)$, which generates a vector with m components between a and b. The routine $Random - Vector$ is called in the main loop of the algorithm (see Fig. 1), and a net with n points is obtained in the considered region. From this net of points we keep the points inside of C_k, which presumptively are in the attraction basin.

3 Numerical experiments

To emphasize the possibility of finding starting points with the proposed algorithm, we performed numerical experiments for different iterative methods and various mappings in several variables. The general conclusion is that the proposed algorithm gives starting points with relative low computational effort for mappings of medium dimension, $m \leq 100$. It is worth noticing that the algorithm works well for small values of k (even for $k = 0$), which reduces to a great extent the computational effort.

3.1 Experiment 1

This experiment is devoted to validate the properties of the sequence C_k. After accomplishing a significant number of numerical experiments, we conclude that

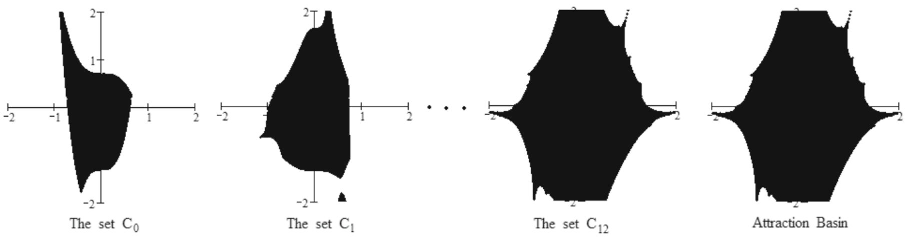


Fig. 2 The sets $C_0,\ C_1,\ C_{12}$ and the attraction basin for f_1

these properties hold if $\varrho(df^k(p)) < 1,\ k = 0, 1, ...$ Here are two examples of executed experiments.

Example 1 The Picard iteration (successive approximation, $\Phi = f_1$) and the mapping:

$$f_1(x) = \left\{ \begin{array}{c} 0.3sin(x_1) + x_1x_2 \\ x_1^3 - 0.5x_2 \end{array} \right\}.$$

Note that $p = (0, 0)^T$ is the fixed point of $\Phi = f_1$. In Fig. 2 are depicted the sets $C_0,\ C_1,\ C_{12}$ and the attraction basin B. It can be seen that $C_0 \cap B \neq \emptyset$ and $C_1 \cap B \neq \emptyset$; furthermore, a significant subset of $C_0,\ C_1$ also belongs to the attraction basin B. Therefore, we can expect that in both sets C_0 and C_1, there exist starting points.

The numerical sequence $\{\varrho(df^k(p))\}$ is $0.5,\ 0.25, ..., 1.22 \times 10^{-4}, ...$

Example 2 The Picard iteration (successive approximation) and the mapping:

$$f_2(x) = \left\{ \begin{array}{c} 0.2x_1 + x_2^2 \\ x_1x_2 - cos(x_2) + 1 \end{array} \right\}$$

Note again that $p = (0, 0)^T$ is the fixed point of $\Phi = f_2$. In Fig. 3 are depicted the sets $C_0,\ C_1,\ C_{12}$ for function f_2.

The numerical sequence $\{\varrho(df^k(p))\}$ is $0.2,\ 0.04, ..., 8.19 \times 10^{-10}, ...$ For more details on the behavior of the sequence $\{\varrho(df^k(p))\}$, see the paper [12].

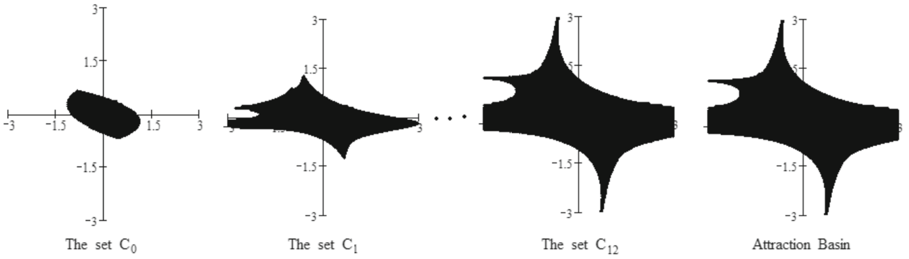


Fig. 3 The sets $C_0,\ C_1,\ C_{12}$ and the attraction basin for f_2

If the condition of the numerical sequence $\varrho(d\Phi^k(p))$ is not satisfied, then the two properties of C_k do not hold. For example, in the case of function $f(x, y) = (x - sin(x), x^2 + 0.5y)$, $\varrho(df^k(p)) = 1,\ k = 0, 1, ...$, the sequence $\{C_k\}$ is decreasing and tends to the empty set.

Remark 1 The proposed algorithm is based on the properties of the sequence of sets C_k; in our investigation these properties were observed and studied by numerical experiments, which can lead to misleading conclusions. But in this case, by taking into account the shape of the attraction basin, the identity between the limit set of C_k (or even the set C_k for low values of k) and the attraction basin, does not seem to be accidental.

3.2 Experiment 2

The classical Newton method ($\Phi(x) = x - df(x)^{-1} f(x)$) and the mapping:

$$f(x) = \left\{ \begin{array}{c} 0.9x_1 + 0.2x_2x_3^3 \\ x_1^3x_3 - 0.2x_2 \\ 0.2x_2^3 + 0.1x_3 \end{array} \right\},$$

This experiment has the purpose to show the influence of k and n on the number r_k of points in the set C_k.

In Fig. 4, the variation of r_k is presented for some particular values of k and n.

It can be seen that the number of points in C_k increases when k or n is augmented. This means that if $C_k = \emptyset$, then the values of k and n should be augmented, and the attempt repeated.

One remarkable fact is that $r_k \neq 0$ even for low values of k and n, and so, the points given by the algorithm can be taken as starting points. Note also that r_k has a moderate increase when k and n are getting larger. For example, in the case of considered mapping and for $k = 1,\ n = 5$, the algorithm gives the following three starting points (0.651, -0.030, -0.013), (0.037, 0.753, 0.425), (0.989, -0.480, 0.211).

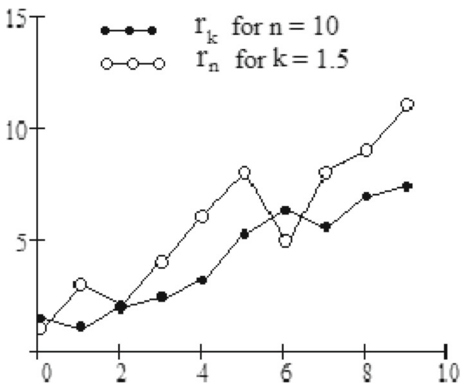


Fig. 4 The variation of r_k

Remark 2 The algorithm searches for a starting point in an n-dimensional cube with side size $2a$. If the algorithm does not find starting points, it means that inside this cube, the equation $f(x) = 0$ has no solutions.

3.3 Experiment 3

The parallel lines method and the function from [13], $f(x) = (f_1(x), ..., f_m(x))$,

$$f_i(x) = x_i - \frac{1}{2m}\left(\sum_{j=1}^{m} x_j^3 + i\right), \ i = 1, ..., m.$$

Recall that the parallel lines method is an iterative process of the form

$$x_{n+1} = x_n - \alpha f(x_n),$$

where α is a positive real constant (in the term of iteration function, $\Phi(x) = x - \alpha f(x)$). The sequence $\{x_n\}$ generated by this method converges to a solution p of the equation $f(x) = 0$ if $\langle f'(p)x, x\rangle \geq \|x\|^2, \ \forall x \in D$ and α satisfies $\alpha < \{1, \|f'(p) - I\|^{-2}$.

We search for the set $C_0 = \{x : \varrho(f'(x) - I) < 1\}$ inside the cube $D = [0, a]^m, \ 0 < a < 1$. Presumptively, the points in C_0 are starting points for the parallel lines method. The results are satisfactory. For example, if $m = 5, \ a = 0.6, \ n = 10$ the algorithm found 10 points in C_0, which are all starting points; if $m = 5, \ a = 1, \ n = 10$, the algorithm found three points in C_0; if $m = 30, \ a = 0.6, \ n = 200$, then C_0 contains a few number of points, 1–3. It is worth noticing that the proposed algorithm is not very computationally expensive, even for medium large m.

4 Conclusions

In this paper we propose an experimentally verified algorithm that finds the starting point for the considered (Picard) iteration. It is shown that the algorithm gives the starting points for various iterative methods with low computational effort. Presumptively, if the algorithm does not give any starting point it means that the considered n-dimensional region does not contain solutions of the equation $f(x) = 0$. If repeatedly $C_k = \emptyset$ for relatively high values of k and n, it can mean that equation $F(x) = 0$ does not have solutions in the considered region.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Maruster, S.: The solution by iteration of nonlinear equations in Hilbert spaces. Proc. Amer. Math. Soc. **63**, 69–73 (1977)
2. Baker Kearfott, R.: Abstract generalized bisection and cost bound. Math. Comput. **49**(179), 187–202 (1987)

3. Adler, A.: On the bisection method for triangles. Math. Comput. **40**(162), 571–574 (1983)
4. Xu, Z.-B., Zhang, J.-S., Wang, W.: A cell exclusion algorithm for determining all the solutions of a nonliniar system of equations. Appl. Math. and Comput. **80**, 181–208 (1996)
5. Hansen, E.R., Greenberg, R.I.: An interval Newton method. Appl. Math. Comput. **12**(2-3), 89–98 (1983)
6. Allgower, E.L., Georg, K.: Simplicial and continuation methods for approximating fixed points and solutions to systems of equations. SIAM Rev. **22**, 28–85 (1980)
7. Karra, C.L., Weckb, B., Freemanc, L.M.: Solutions to systems of nonlinear equations via a genetic algorithm. Eng. Appl. Artif. Intel. **11**(3), 369–376 (1998)
8. Rheinboldt, W.C.: An adaptive continuation process for solving systems of nonlinear equations. Polish Acad. Sci. Banach Center Publ. **3**, 129–14 (1975)
9. Catinas, E.: Estimating the radius of an attraction ball. Appl. Math. Lett. **22**, 712–714 (2009)
10. Hernández-Veron, M.A., Romero, N.: On the local convergence of a third order family of iterative processes. Algorithms **8**, 1121–1128 (2015)
11. Rus, I.A.: A conjecture on global asymptotic stability, workshop iterative approximation of fixed points. In: 19Th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara (2017)
12. Berinde, V., Maruşter, St., Rus, I.A.: On an open problem regarding the spectral radius of the derivatives of a function and of its iterates (in preparation)
13. Miyajima, S., Kashiwagi, M.: Existence test for solution of nonlinear systems applying affine arithmetic. J. Comput. Appl. Math. **199**, 304–309 (2007)