Salud, Ciencia y Tecnología - Serie de Conferencias. 2023; 2:525
doi: 10.56294/sctconf2023525

Categoría: Congreso Científico de la Fundación Salud, Ciencia y Tecnología 2023

ORIGINAL

Conceptualizing the Iterative method of Newton-Raphson for systems with two equations

Conceptualización del método iterativo de Newton-Raphson para sistemas con dos ecuaciones

Serafeim A. Triantafyllou[1]

[1]Aristotle University, 54124, Thessaloniki, Greece.

Citar como: Triantafyllou SA. Conceptualizing the Iterative method of Newton-Raphson for systems with two equations. Salud, Ciencia y Tecnología - Serie de Conferencias 2023; 2:525. https://doi.org/10.56294/sctconf2023525

Recibido: 15-06-2023 **Revisado:** 17-08-2023 **Aceptado:** 19-10-2023 **Publicado:** 20-10-2023

ABSTRACT

In numerical analysis, Newton-Raphson method is a root-finding algorithm which generates iterative approximations to the zeroes (or roots) of a real-valued function. This paper describes in a detailed way the mathematical background around the iterative method of Newton-Raphson for systems with two equations. Next, an algorithmic implementation of the iterative method of Newton-Raphson for systems with two equations is developed in Pascal Programming Language, to represent the steps of this method with a procedural programming language with special emphasis on the use of computing in the scientific area of mathematics.

Keywords: Mathematics; Computer science; Algorithms.

RESUMEN

En análisis numérico, el método de Newton-Raphson es un algoritmo de búsqueda de raíces que genera aproximaciones iterativas a los ceros (o raíces) de una función de valor real. Este artículo describe de forma detallada los antecedentes matemáticos en torno al método iterativo de Newton-Raphson para sistemas con dos ecuaciones. A continuación, se desarrolla una implementación algorítmica del método iterativo de Newton-Raphson para sistemas con dos ecuaciones en lenguaje de programación Pascal, para representar los pasos de este método con un lenguaje de programación procedimental con especial énfasis en el uso de la computación en el área científica de las matemáticas.

Palabras clave: Matemáticas; Informática; Algoritmos.

INTRODUCTION

In numerical analysis, Newton-Raphson method is a root-finding algorithm which generates iterative approximations to the zeroes (or roots) of a real-valued function.[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15] The roots of a function cannot be computed precisely in closed form, and root-finding algorithms provide approximations to zeros, represented either in the form of floating-point numbers or isolating intervals, or disks for complex roots.[1,2] Numerous numerical root-finding methods generate successively a sequence of numbers that hopefully converges towards the root as its limit. [12,13,14,15,16,17,18]

Mathematical Background

Let the constant functions: f(x,y) , g(x,y) = D = {a ≤ x ≤ b, c ≤ y ≤ d} that have second order constant derivatives. In the system: f(x,y) = 0, g(x,y) = 0 there is a solution (x,y) in D and furthermore it is true that: $\begin{vmatrix} f_x\,(x,y) & f_y\,(x,y) \\ g_x\,(x,y) & g_y\,(x,y) \end{vmatrix} \neq 0$ then for (x_0, y_0) starting from close to (x,y) through the iterative method of Newton-Raphson it is proven that the sequence (x_i, y_i) converges to the solution (x,y) of the system.

$$x_{i+1} = x_i + \frac{g(xi,yi)fy(xi,yi) - f(xi,yi)gy(xi,yi)}{fx(xi,yi)gy(xi,yi) - fy(xi,yi)gx(xi,yi)}$$

$$y_{i+1} = y_i + \frac{f(xi,yi)gx(xi,yi) - g(xi,yi)fx(xi,yi)}{fx(xi,yi)gy(xi,yi) - fy(xi,yi)gx(xi,yi)}$$

Solving a system using Pascal programming language

A program is needed to solve the system with the iterative method of Newton-Raphson for systems with two equations:

$f(x,y) = x - x^2 - y^2 = 0$
$g(x,y) = y - x^2 + y^2 = 0$

Symbolisms:

$(x_i, y_i) = (x_a, y_a)$
$(x_{i+1}, y_{i+1}) = (x_t, y_t)$
f_x = f x; g_x = g x
f_y = f y; g_y = g y

The functions f, f x, f y, g, g x, g y are given as function subroutines, and the iterative method Newton-Raphson as a procedure subroutine named as NEWTON2.

We consider the initial point (x0, y0) = (xa, ya) = (0,8, 0,4) and stop the iterative method when for ε > 0 and the natural number m ∈ N we have: $|x_t - x_a| + |y_t - y_a| < \varepsilon$ or $i \geq m$.

The following algorithm implemented in Pascal programming language shows the above-mentioned subroutines. The Pascal programming language was chosen because it is very helpful to teach procedural programming in computing and mathematics university courses conducted in the first semesters of a university curriculum.

Algorithmic implementation in Pascal Programming Language

```
program PROBLEMp1(input,output);
uses Crt;
var m : integer; xa,ya : real; epsilon: real;
function f (var x,y: real):real;
 BEGIN
  f := x-x*x-y*y
END;
function fx(var x,y: real):real;
 BEGIN
```

https://doi.org/10.56294/sctconf2023525

```
  fx := 1-2*x
END;

function fy(var x,y: real):real;
 BEGIN
  fy := -2*y
END;

function g(var x,y: real):real;
 BEGIN
  g := y-x*x+y*y
END;
function gx(var x,y: real):real;
 BEGIN
  gx := -2*x
END;

function gy(var x,y: real):real;
 BEGIN
  gy := 1+2*y
END;

procedure NEWTON2 (xa,ya: real);
 var i: integer;
 xt, yt, norm: real;
af, afx, afy, ag, agx, agy, ax, ay, b: real;

BEGIN
 i:=0; xt:=xa; yt:=ya;
  REPEAT
    i:=i+1; xa:=xt; ya:=yt;
         af:=f(xa,ya); ag:=g(xa,ya);
         afx:=fx(xa,ya); agx:=gx(xa,ya);
         afy:=fy(xa,ya); agy:=gy(xa,ya);
         ax:=ag*afy-af*agy; ay:=af*agx-ag*afx;
         b:=afx*agy-afy*agx;
         xt:=xa+ax/b; yt:=ya+ay/b;
         norm:=ABS(xt – xa) + ABS(yt-ya)
UNTIL (norm < epsilon) OR (i >= m);
WRITELN;
WRITELN('   after', i:2, 'steps we have:');
WRITELN;
WRITELN(' x = ', xt:14:12, '  y=   ', yt:14:12);
END;
BEGIN
ClrScr;
WRITELN;
WRITE('    maximal number of steps :   '); READLN(m);
```

https://doi.org/10.56294/sctconf2023525

```
WRITE('                    epsilon   '); READLN(epsilon);
WRITELN;
WRITE('                        xa:  '); READLN(xa);
WRITE('                        ya:  '); READLN(ya);
WRITELN;
NEWTON2(xa,ya)
END.
```

RESULTS

The results after running the above-mentioned code in Pascal programming language are:

```
maximal number of steps : 20
                    epsilon : 1e – 10
                    xa : 0.8
                    ya : 0.4
```

after **4 steps** we have:

x = 0,771844506346   y = 0,419643377607

Importance of the Iterative method of Newton-Raphson for systems with two equations

The Newton-Raphson method is an iterative numerical technique used to find successively better approximations to the roots (or solutions) of a real-valued function. In the case of systems with two equations (a system of two nonlinear equations), the Newton-Raphson method can be extended for each variable in the system.

The Newton-Raphson method for systems with two equations has been a significant scientific contribution in the field of numerical analysis and computational mathematics. This iterative method is widely used for finding solutions to systems of nonlinear equations, and it has several notable contributions:

1. Efficiency in Convergence: The Newton-Raphson method converges rapidly when the initial guess is sufficiently close to the actual solution. This efficiency in convergence is a valuable trait in numerical methods, especially for solving complex systems of equations.
2. Local Convergence to Solutions: The method provides local convergence to solutions, meaning that if the initial guess is close enough to a solution, the method will converge to that specific solution. This local property makes it suitable for problems where an approximate solution is known to exist in a certain region.
3. Applicability to Systems of Equations: The extension of the Newton-Raphson method to systems of equations makes it versatile for solving interconnected problems where multiple variables are involved. This is particularly relevant in various scientific and engineering applications.
4. Sensitivity to Initial Conditions: The method's sensitivity to initial conditions allows researchers and practitioners to gain insights into the stability and behavior of solutions in the vicinity of a given starting point. This sensitivity can be useful in understanding the local landscape of solutions.
5. Derivative Information Utilization: The method uses derivative information through partial derivatives of the system of equations. This utilization of derivative information enhances the convergence speed, making it a powerful tool for solving systems where derivative information is readily available.
6. Applications in Engineering and Sciences: The Newton-Raphson method is applied extensively in

various scientific and engineering disciplines, such as physics, chemistry, biology, economics, and more. Its effectiveness in solving nonlinear systems contributes to advancements in these fields.

7. Numerical Analysis Foundation: The Newton-Raphson method is a foundational concept in numerical analysis. Its development and understanding have paved the way for the exploration of other iterative methods and numerical techniques for solving complex mathematical problems.
8. Influence on Optimization Algorithms: The Newton-Raphson method has influenced the development of optimization algorithms, where finding the roots of the gradient (or derivative) is a key step. Optimization plays a crucial role in many scientific and engineering applications.

CONCLUSIONS

The Newton Raphson method has an advantage that it allows us to guess the roots of an equation with a small degree very quickly and efficiently. The disadvantage of Newton Raphson method is that it tends to become very complex when the degree of the polynomial becomes very large.

Newton Raphson method can be used in real life in many cases. Specifically, this method can be used to analyze the flow of water in water distribution networks. Future studies will examine in detail many areas where the above-mentioned method can find practical application.

In summary, the iterative method of Newton-Raphson for systems with two equations has made significant contributions to scientific and computational fields, providing a powerful tool for solving nonlinear systems efficiently and accurately. Its widespread use and impact continue to influence numerical analysis and contribute to the advancement of various scientific disciplines.

REFERENCES

1. J.M. McNamee: "Numerical Methods for Roots of Polynomials - Part I", Elsevier (2007).

2. J.M. McNamee and Victor Pan: "Numerical Methods for Roots of Polynomials - Part II", Elsevier (2013).

3. Triantafyllou, S. A. (2022). Constructivist Learning Environments. Proceedings of The 5th International Conference on Advanced Research in Teaching and Education, 2022. https://www.doi.org/10.33422/5th.icate.2022.04.10

4. Triantafyllou, S.A. (2023). A Quantitative Research About MOOCs and EdTech Tools for Distance Learning. In: Auer, M.E., El-Seoud, S.A., Karam, O.H. (eds) Artificial Intelligence and Online Engineering. REV 2022. Lecture Notes in Networks and Systems, vol 524. Springer, Cham. https://doi.org/10.1007/978-3-031-17091-1_52

5. Triantafyllou, S.A. (2022). TPACK and Toondoo Digital Storytelling Tool Transform Teaching and Learning. In: Florez, H., Gomez, H. (eds) Applied Informatics. ICAI 2022. Communications in Computer and Information Science, vol 1643. Springer, Cham. https://doi.org/10.1007/978-3-031-19647-8_24

6. Farhaoui, Y.and All, Big Data Mining and Analytics, 2023, 6(3), pp. I–II, DOI: 10.26599/BDMA.2022.9020045

7. Triantafyllou, S. A., & Sapounidis, T. (2023). Game-based Learning approach and Serious Games to learn while you play. 2023 IEEE World Engineering Education Conference (EDUNINE). https://doi.org/10.1109/EDUNINE57531.2023.10102872

8. Triantafyllou, S. A., & Georgiadis, C. K. (2022). Gamification Design Patterns for user engagement. Informatics in Education, 655–674. https://doi.org/10.15388/infedu.2022.27

9. Triantafyllou, S.A. (2023). A Detailed Study on the 8 Queens Problem Based on Algorithmic Approaches Implemented in PASCAL Programming Language. In: Silhavy, R., Silhavy, P. (eds) Software Engineering Research in System Science. CSOC 2023. Lecture Notes in Networks and Systems, vol 722. Springer, Cham. https://doi.org/10.1007/978-3-031-35311-6_18

10. Triantafyllou, S. A. (2022). Magic squares in order 4K+2. 2022 30th National Conference with International Participation (TELECOM). https://doi.org/10.1109/TELECOM56127.2022.10017312

11. Triantafyllou, S. A. (2023). Aristotle's Rhetoric and Semiotics. iNTERCULTURAL tRANSLATION iNTERSEMIOTIC, 12(2). https://doi.org/10.26262/iti.v12i2.9874

12. Triantafyllou, S. Gamification and new approaches in Engineering Education. Preprints 2023, 2023120198. https://doi.org/10.20944/preprints202312.0198.v1

13. Ben-Israel, A. (1966). A Newton-Raphson method for the solution of systems of equations. Journal of Mathematical analysis and applications, 15(2), 243-252.

14. Farhaoui, Y. and All, Big Data Mining and Analytics, 2022, 5(4), pp. I IIDOI: 10.26599/BDMA.2022.9020004

15. Bruck, H. A., McNeill, S. R., Sutton, M. A., & Peters, W. H. (1989). Digital image correlation using Newton-Raphson method of partial differential correction. Experimental mechanics, 29, 261-267.

16. Hartmann, S. (2005). A remark on the application of the Newton-Raphson method in non-linear finite element analysis. Computational Mechanics, 36, 100-116.

17. Pho, K. H. (2022). Improvements of the Newton–Raphson method. Journal of Computational and Applied Mathematics, 408, 114106.

18. Triantafyllou, S.A.: The Effects of Constructivism Theory in the Environment of E-learning. GRIN Verlag, Munich (2013)

FINANCING
The authors did not receive funding for the development of this research.

CONFLICT OF INTEREST
The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION
Conceptualization: Serafeim A. Triantafyllou.
Data curation: Serafeim A. Triantafyllou.
Formal analysis: Serafeim A. Triantafyllou.
Research: Serafeim A. Triantafyllou.
Methodology: Serafeim A. Triantafyllou.

Project administration: Serafeim A. Triantafyllou.
Software: Serafeim A. Triantafyllou.
Supervision: Serafeim A. Triantafyllou.
Validation: Serafeim A. Triantafyllou.
Visualization: Serafeim A. Triantafyllou.
Writing - original draft: Serafeim A. Triantafyllou.
Writing - proofreading and editing: Serafeim A. Triantafyllou.

https://doi.org/10.56294/sctconf2023525