SOLUTION OF POLYNOMIAL SYSTEM USING PHCPACK

[1]Mohammed Abdullahi, [2]Bello I. Mshelia and [2]Sabo Hamma
[1]Department of Mathematics and Statistics, University of Maiduguri, Borno State, Nigeria
[2]Mathematical Science Program, Abubakar Tafawa Balewa University, Bauchi
E-mail: abdull109@yahoo.com

ABSTRACT

In the year 1225, Leonardo of Pisa studied the equation $x^3 + 2x^2 + 10x - 20 = 0$ and produced x = 1.368 808 107. Nobody knows by what method Leonardo found this value. In this paper, we examined this Leonardo's equation using iterative, Newton's and phc (Polynomial Homotopy Continuation) and compare the results.
Keywords: Leonardo's equation, polynomial systems, Homotopy continuation, PHCpack.

INTRODUCTION

Continuation, embedding or homotopy methods have long served as useful theoretical tools in modern mathematics. The use of deformation to solve nonlinear systems of equation may be traced back at least to Lahaye (1934). The classical embedding methods were the first deformation methods to be numerically implemented and may be regarded as a forerunner of predictor-corrector method for path following. Introductions into aspects of the subject may be found in the book of Garcia and Zangwill(1981); Keller (1987), Rheinboldt (1977) and Allgower and George (2003). In the literature of numerical analysis, the term numerical continuation and path following are used interchangeably. For contemporary applications of this powerful method, see Sharma and Methi (2011), Barari *et al* (2008), Choobbasti *et al* (2008), Sazzad (2011), Mirgolbabaei and Ganji (2009), Fazeli *et al* (2008), Ganji *et al* (2008), Twinkle *et al* (2012). The necessity of solving systems of nonlinear equations often arises in simulating and designing a chemical plant or optimizing a process. When f(x) = 0 where f : $R^n \to R^n$ is a C^2 – map is a system of nonlinear equations, we want to find the solution of f(x). The traditional Newton's method applied to nonlinear equations of the form f(x) = 0 in general will only converge if the iteration is started near a root of the equation. Kuno and Seader (1988) said that, Newton's method is locally convergent. Tzong-Mou (2004) extended it to cover Kinematics Design. In addition, the method is designed to locate, at best, just one root even though multiple solutions may exist. This is the major disadvantage. However, Tzong-Mou (2005) used technique to avoid divergence for planar and spatial Newton's homotopy continuation method. Let f(x) = 0 be a system of n polynomial equations in n unknowns. Denoting f = (f_1, f_2, ···, f_n), we want to find all isolated solutions of

$$f_1(x_1, x_2, \cdots, x_n) = 0$$
$$\vdots \qquad\qquad\qquad (1.0)$$
$$f_n(x_1, x_2, \cdots, x_n) = 0$$

for x = (x_1, x_2, ···, x_n). This problem is very common in many fields of science and engineering, such as formula construction, geometric intersection problems, inverse

kinematics, power flow problems with PQ-specified bases, computations of equilibrium states, etc. Elimination theory-based methods, most notably the Burcberger algorithm Buchberger (1985) for constructing Grobner bases, are the classical approach to solving Eq.(1.0), but their reliance on symbolic manipulation makes those methods seem somewhat unsuitable for all but small problems. In 1977, Garcia and Zangwill (1979) independently presented theorems suggesting that homotopy continuation could be used to find numerically the full set of isolated solutions of Eq.(1.0). During the last two decades, this method has been developed into a reliable and efficient numerical algorithm for approximating all isolated zeros of polynomial systems Li (1997). Thus, if f(x) is the system of nonlinear equations to be solved and g(x) is a second simpler system of the same number of equations, the homotopy function might be constructed as

$$H(x, \ t) \ = \ tf(x) \ + \ (1 \ - \ t)g(x) \ = \ 0 \qquad 0 \ \leq \ t \ \leq \ 1, \ x \ \in \ R^n$$
(1.1)

where t is a scalar homotopy parameter which is gradually varied from 0 to 1 as the path is tracked from starting point to a solution. One of the great advantages of the homotopy method is that, under some conditions, they offer a way to have a globally convergent method to find the zeros of any function f: $R^n \to R^n$. In general, iterative methods for solving a nonlinear equation in R^n depend strongly on the selection of the initial data. In order to reduce this dependence, the continuation process uses a family of equations given in Eq. (1.2) which for t = 1 contains the given equation. If for each t $\in$ [0, 1] a solution x(t) of Eq.(1.2) exists that varies continuously with t, then the function x : [0, 1] $\to R^n$ constitute a curve in R^n between the-assumed point to be given-point $x^0 = x(0)$ and the unknown solution $x* = x(1)$ of the original equation. Hence iterative processes may be considered which use the curve as a guide and channel their iterates in its proximity from x^0 to the intended limit x*. For further discussions of how homotopy continuation works see the papers of Rheinboldt (1980), Garcia and Zangwill (1981), Rafiq and Muhammad (2008) and M. Abdullahi and B. Yusuf (2006). For introduction to homotopy continuation methods specific for polynomial systems, we recommend Li (2003) and Sommese *et al* (2005). The books of Allgower and George (2003) and Govaerts (2000) provide introduction to path following methods applied to general nonlinear system and systems of differential equations. Dumortier *et al* (2006) studied polynomial differential systems in the real plane and developed software to allow phase portraits. Computer algebra is used to compute all singularities, but it is noted in Dumortier *et al* (2006) that for high degrees this can take a long time. Recent related symbolic methods are described in Lazard and Rouillier (2007) and Li and Wang (1993).

In most applications, one is mainly interested in real solutions. However, a complex curve of a polynomial system may have isolated real solutions. Such a real solution on a complex curve will be isolated in the real space and will show itself as a single solution on the curve. Homotopy continuation method in the last two decades has been developed into a reliable and efficient numerical algorithm for solving all isolated zeros of polynomial system. During the last few years, major computational breakthrough has emerged in the area, that was investigated by Li (2003). Tzong-Mou (2005) also used the technique to search for all the roots of inverse Kinematics Problem of Robot. Base on the Bernshtein theory on root count,

the polyhedral homotopy is established to considerably reduce the number of homotopy path that need to be traced to find all the isolated roots, making the method much more powerful. Li (2003) reports the most recent development of this new method along with future considerations. PHCpack is a software to solve polynomial systems via homotopy continuation methods. It is a general-purpose solver for polynomial systems by homotopy continuation. The software package PHC implements homotopy continuation methods to compute numerically approximations to all isolated solutions of a system of n polynomial equations in n unknowns. The name Polynomial Homotopy Continuation unites the three key concepts of the method. Since we solve polynomial systems we exploit the algebraic structure to count the roots and to construct a start system. By continuation methods, the known solutions of the start system are extended to the desired solutions of the target system. This deformation is defined by homotopy, that is, a family of start systems connecting start and target system. For an overview on how the algorithm in PHC are used in practice to solve polynomial systems, see [Verschelde, 1999]

METHODOLOGY
In the year 1225, Leonardo of Pisa studied the equation
$$f(x) = x^3 + 2x^2 + 10x - 20 \tag{1.1}$$
and produced x = 1.368 808 107. Nobody knows by what method Leonardo found this value, but it is remarkable result for this time [Francis, 1968].

Leonardo's equation using iterative method
Theorem 1.2
If r is a root of f(x) = 0 and if the equation is rewritten in the form x = g(x) in such a way that $|g'(x)| < 1$ in an interval I centered at x = r, then the sequence $x_n = g(x_{n-1})$ with arbitrary but in the interval I has lim x_n= r.
Applying the above theorem to Leonardo's Eq. (1.1), the equation can be put into the form x = g(x) in many ways. We take

$$x = g(x) = \frac{20}{x^2 + 2x + 10}$$

with $x_0 = 1$, we find that

$$x_1 = \frac{20}{13} \quad \square \quad 1.538\,461\,538.$$

Continuing the iteration produces the sequence of Table 1.1. Sure enough, on the twenty – fourth round Leonardo's value appears.

n	x_n	n	x_n
1	1.538 461 538	13	1.368 817 874
2	1.295 019 157	14	1.368 803 773
3	1.401 825 309	15	1.368 810 031
4	1.354 209 390	16	1.368 807 253
5	1.357 298 092	17	1.368 808 486
6	1.365 929 788	18	1.368 807 940
7	1.370 086 003	19	1.368 808 181

8	1.368 241 023	20	1.368 808 075
9	1.369 059 812	21	1.368 808 122
10	1.368 696 397	22	1.368 808 101
11	1.368 857 688	23	1.368 808 101
12	1.368 786 102	24	1.368 808 107

Table 1.1

Leonardo's equation Newton's method

Newton's method is the method of choice to solve nonlinear systems numerically; however, its convergent is only local. Beginning with the Taylor's formula

$$f(r) = f(x_{n-1}) + (r - x_{n-1})f'(x_{n-1}) + \tfrac{1}{2}(r - x_{n-2})^2 f''(\xi) \qquad (1.2.1)$$

Retaining the linear part, recall that f(r) = 0 and define x_n by putting it in place of the remaining r to obtain

$$0 = f(x_{n-1}) + (x_n - x_{n-1})f'(x_{n-1})$$

which rearranges at once into

$$r \Box \; x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \qquad (1.2.2)$$

This is the Newton's iteration formula.
Now we use Newton's formula to solve Leonardo's Eq. (1.1),
with

$$f(x) = x^3 + 2x^2 + 10x - 20,$$

we find that

$$f'(x) = 3x^2 + 4x + 10$$

and the iteration formula

$$x_n = x_{n-1} - \frac{x_{n-1}^3 + 2x_{n-1}^2 - 10x_{n-1} - 20}{3x_{n-1}^2 + 4x_{n-1} + 10} \qquad (1.2.3)$$

Once more choosing $x_0 = 1$, we obtain the results in the table below:

N	1	2	3	4
x_n	1.411 764 706	1.369 336 471	1.368 808 189	1.368 808 108

Table 1.2

The speed of convergence is remarkable. In four iterations we have essentially Leonardo's value. In fact, the computation shows that

$$f(1.368808107) \approx -0.000000016$$

$$f(1.368808108) \approx -0.000000005$$

(1.2.4)
which suggest that the Newton's result is the winner by a nose.

Leonardo's equation using phcpack

```
1    1
x^3+2*x^2+10*x-20;
```

47

ROOT COUNTS
total degree : 3

HOMOTOPY PARAMETERS :
  d : 16
  k : 2
  a : 9.98168055857237E-01   6.05023327334112E-02
  t : 1.00000000000000E+00   0.00000000000000E+00
  no projective transformation

***************** CURRENT CONTINUATION PARAMETERS *****************
Global Monitor:
  1. The condition of the homotopy   : 0
  2. Number of paths tracked simultaneously   : 1
  3. Maximum number of steps along a path   : 500
  4. Distance from target to start end game   : 1.000E-01
  5. Order of extrapolator in end game   : 0
  6. Maximum number of re-runs   : 1
Step Control (Predictor) :   along path : end game
  7:8. type (x:Sec,t:Rea):(x:Sec,t:Rea)   : 0   : 0
  9:10. minimum step size   : 1.000E-06 : 1.000E-08
 11:12. maximum step size   : 1.000E-01 : 1.000E-02
 13:14. reduction factor for step size   : 7.000E-01 : 5.000E-01
 15:16. expansion factor for step size   : 1.250E+00 : 1.100E+00
 17:18. expansion threshold   : 1   : 1
Path Closeness (Corrector) :   along path : end game
 19:20. maximum number of iterations   : 4   : 4
 21:22. relative precision for residuals   : 1.000E-09 : 1.000E-11
 23:24. absolute precision for residuals   : 1.000E-09 : 1.000E-11
 25:26. relative precision for corrections   : 1.000E-09 : 1.000E-11
 27:28. absolute precision for corrections   : 1.000E-09 : 1.000E-11
Solution Tolerances: along path : end game
 29:30. inverse condition of Jacobian   : 1.000E-04 : 1.000E-12
 31:32. clustering of solutions   : 1.000E-04 : 1.000E-12
 33:34. solution at infinity   : 1.000E+08 : 1.000E+12
****

Output Information during Continuation:
  0 : no intermediate output information during continuation

Timing Information for Continuation
The elapsed time in seconds was   0.000000000 = 0h 0m 0s 0ms

The Solutions:

3   1
==
=======
Solution 1:   start residual : 3.553E-15 #iterations : 1 success
t : 1.00000000000000E+00 0.00000000000000E+00
m : 1
the solution for t :
 x : -1.68440405391069E+00 3.43133135019769E+00
== err : 1.591E-16 = rco : 1.000E+00 = res : 3.553E-15 = complex regular ==
solution 2 :   start residual : 3.553E-15 #iterations : 1 success
t : 1.00000000000000E+00 0.00000000000000E+00
m : 1
the solution for t :
 x : -1.68440405391069E+00 -3.43133135019769E+00
== err : 1.591E-16 = rco : 1.000E+00 = res : 3.553E-15 = complex regular ==
solution 3 :   start residual : 3.174E-35 #iterations : 1 success
t : 1.00000000000000E+00 0.00000000000000E+00
m : 1
the solution for t :
 x : 1.36880810782137E+00 0.00000000000000E+00
== err : 1.505E-36 = rco : 1.000E+00 = res : 0.000E+00 = real regular ==
==
========
A list of 3 solutions has been refined :

Number of regular solutions   : 3.
Number of singular solutions   : 0.
Number of real solutions   : 1.
Number of complex solutions   : 2.
Number of clustered solutions   : 0.
Number of solutions at infinity   : 0.
Number of failures   : 0.
==
========
Frequency tables for correction, residual, and condition numbers :
FreqCorr : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3   : 3
FreqResi : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3   : 3
FreqCond : 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   : 3
Small correction terms and residuals counted to the right.
Well conditioned and distinct roots counted to the left.

TIMING INFORMATION for Root Refining
The elapsed time in seconds was   0.000000000 = 0h 0m 0s 0ms

TIMING INFORMATION for solving the polynomial system
The elapsed time in seconds was   220.953000000 = 0h 3m40s953ms

PHC ran from 9 August 2011, 17:23:08 till 9 August 2011, 17:28:02.
The total elapsed time is 295 seconds = 4 minutes 55 seconds.

DISCUSSIONS

The result obtained by Leonardo for solving Eq. (1.1) was quite remarkable for his time, regardless of what method used in obtaining the result. The iteration method, in this setting seems to move at a snail speed. The function f(x) = 0 is transformed to the form x = g(x) and have $x_{n+1} = g(x_n)$ as its iteration scheme with the condition that $|g'(x)| < 1$. See Theorem 1.2 above. Looking at Table 1.1, we observed that the Leonardo's value appears on the twenty-fourth iteration. This means that iteration method is very slow in convergence.
Newton's method is the method of choice to solve nonlinear systems numerically; however, its convergent is only local, that is, it gives only one approximation to the solution as does the iteration method. Newton's method in this problem uses initial guess $x_0 = 1$. The method depends on the initial choice of the guess as close as possible to the actual root to guarantee speedy convergence. We observed that the speed of convergence of Newton's method is remarkable. The Leonardo's value is obtained in just four iteration, as observed from Table 1.2. In fact, the computation in Eq.(1.2.4) shows that the Newton's method is the winner by a nose. PHCpack, a general-purpose solver for polynomial systems by homotopy continuation. The software is as a result of a PhD work of the author [Verschelde, 1999]. It is proved to be a very reliable and efficient way to solving any polynomial or polynomial systems of any dimension. The software uses the predictor-corrector approach as seen in Section 2.3. The *phc* gives all the three expected solutions of Eq (1.1), two of which are complex solutions.

We see that if we compare the first two schemes with *phc*, we observe that Iterative and Newton's schemes give only an approximation to the root and both schemes provides only one root. The convergent in the two schemes are only local, that is, they provide only one solution. Nothing is said about the other two remaining roots of Eq. (1.1). On the other hand, to achieve a global convergence, we use homotopy. PHCpack provides all the three solutions along with their characteristics. In our problem, we have the three solutions indicating a refinement of the solutions, of the three regular solutions; one is real and the other two, complex. Nevertheless, the *phc* also gives an approximation. Though it is most reliable and efficient. As a research interest, one can apply interval analysis of [(Moore, 1996), (Kearfott, 1994) and (Alefeld, 1983)] to the end game to obtain exact solutions. This is an area of interest.

REFERENCES

A. Barari, Abdoul R. Ghotbi, F. Farrokhzad and D.D. Ganji, 2008. Variational Iteration Method and Homotopy-Perturbation Method for Solving Different Types of Wave Equations. *Journal of Applied Sciences, 8: 120-126.*

A.J. Choobbasti , F. Farrokhzad , A. Barari and D.D. Ganji , 2008. Application of Wave Equation in Pile Foundation Using Homotopy Perturbation Method and Variational Iteration Method. *Journal of Applied Sciences, 8: 2158-2162.*

Alefeld, G. and J. Herzberger, 1983. Introduction to Interval Analysis. Academic Press, New York.

Allgower, E. L. and K. George, 2003. Introduction to Numerical Continuation Methods. Volume 45 of Classics in Applied Mathematics. SIAM.

Burchburger, B., 1985. Grobner Bases; An Algorithmic Method in Polynomial Ideal Theory. Multidimensional System Theory, N. K. Bose, ed., D. Reidel Publishing Company, Dordrech, Boston, Lancaster, *184 − 232*.

D.D. Ganji , H. Mirgolbabaei , Me. Miansari and Mo. Miansari , 2008. Application of Homotopy Perturbation Method to Solve Linear and Non-Linear Systems of Ordinary Differential Equations and Differential Equation of Order Three. *Journal of Applied Sciences, 8: 1256-1261.*

Dumortier, F., J. Llibre, and J. C. Artes, 2006. Qualitative Theory of Planner Differential Systems. Springer - Verlag.

Francis Scheid, 1968. Scaum's Outline Series, Theorey and Problems of Numerical Analysis. McGRaw − Hill Company. *314-316.*

Garcia, C. B. and W. I. Zangwill, 1979., Determining All Solution to Cartesian Systems of Nonlinear Equations. Mathematics of Operations Research, Vol. 4, *1 − 14*.

Garcia, C. B. and W. I. Zangwill, 1981. Pathways to Solutions and Equilibria. Prentice − Hill, Inc. Englewood Cliffs, N. J. 07632.

Govaerts, W. J. F., 2000. Numerical Methods for Bifurcations of Dynamic Equilibria. SIAM.

H. Mirgolbabaei and D.D. Ganji, 2009. Application of Homotopy Perturbation Method to Solve Combined Korteweg de Vries-Modified Korteweg de Vries Equation. Journal of Applied Sciences, 9: 3587-3592.

Kearfott, R. B., 1994. An Interval Step Control for Continuation Methods. SIAM J. Numerical Analysis, 31(3):   *892 − 914*.

Keller, H. B., 1978. Global Homotopies and Newton's Method: Recent Advances in Numerical Analysis. C.de Boor and G Golub, eds., Academic Press, New York.

Kuno, M and J. D. Seader, 1988. Computing All real solutions to systems of nonlinear equations with homotopy continuation. Ind. Eng: Chem. Res,Vol. 27, *1320 − 1329*.

Lahaye, E,, 1948. Solution of Systems of Transcendental Equations. Acad. Roy. Belg. Bull. Cl. Sci. 5Vol. 5. *805 − 822*.

Lazard, D. and F. Rouillier, 2007. Solving Parametric Polynomial Systems. Jour. Symbolic Computation, 42(6). *636 − 667.*

Li, T. Y. and X. Wang, 1993. Solving Real Polynomial systems with Real Homotopies. Math. Comp. Vol. 60:*669 − 680*.

Li, T. Y., 1997. Numerical Solution of Multivariate Polynomial Systems by Homotopy Continuation Methods. Acta Numerica, Cambridge University Press.*399 − 436.*

Li, T. Y., 2003. Numerical Solution of Polynomial Systems by Homotopy Continuation Methods. In F. Cucker editor, Handbook of Numerical Analysis. Volume XI. Special Volume: Foundations of Computational Mathematics, North − Holland, *209 − 304*.

M. Abdullahi and Bukar Y., 2006. The Difference between Euler and Newton Solution Methods, The path following   perspective. Journal of Science and Information Tech. (JOSIT), Vol. 5, No. 1: *148 − 159*.

M. Fazeli, S.A. Zahedi and N. Tolou, 2008. Explicit Solution of Non-Linear Fourth-Order Parabolic Equations via Homotopy Perturbation Method . *Journal of Applied Sciences, 8: 2619-2624.*

Moore, R. E., 1966. Interval Analysis. Prentice Hall, Englewood Cliffs, N. J.

N. Tolou, J. Mahmoudi, M. Ghasemi, I. Khatami, A. Barari and D.D. Ganji, 2008. On the Non-Linear Deformation of Elastic Beams in an Analytic Solution. *Asian Journal of Scientific Research, 1: 437-443.*

Rafiq, A. and Muhammad Awais, 2008. Convergence on Homotopy Continuation Method. Innt. J. of Appl. Math and Mech. **4**(6):*62 − 70*.

Rheinboldt, W. C., 1977. Numerical Continuation Methods for Finite Element Applications. Formulations and Computational Algorithms in Finite Element Analysis, K. J. Bathe, J. T. Oden and W. Wunderlich, eds., MIT Press, Cambridge, MA.

Rheinboldt, W. C., 1980. Adaptive Continuation Process for Solving Systems of Nonlinear Equations. Banach Center Publication, Vol.3: *129 – 142*.

P.R. Sharma and Giriraj Methi, 2011. Applications of Homotopy Perturbation Method to Partial Differential Equations. *Asian Journal of Mathematics & Statistics, 4: 140-150.*

Sazzad Hossien Chowdhury , 2011. A Comparison between the Modified Homotopy Perturbation Method and Adomian Decomposition Method for Solving Nonlinear Heat Transfer Equations. *Journal of Applied Sciences, 11: 1416-1420.*

Sommese, J., J. Verschelde, and C. W. Wampler, 2005. The Numerical Solution of Systems of Polynomials Arising in Engineering and Science. World Scientific Press. Singapore.

Tzong-Mou Wu , 2004. The Kinematics Design Problems. *Journal of Applied Sciences, 4: 398-405.*

Tzong-Mou Wu , 2005. Searching All the Roots of Inverse Kinematics Problem of Robot by Homotopy Continuation Method. *Journal of Applied Sciences, 5: 666-673.*

Tzong-Mou Wu , 2005. A Technique to Avoid Divergence for the Planar and Spatial Newton-homotopy Continuation Method. *Journal of Applied Sciences, 5: 1036-1040.*

Twinkle Patel, M.N. Mehta and V.H. Pradhan, 2012. The Numerical Solution of Burger's Equation Arising into the Irradiation of Tumour Tissue in Biological Diffusing System by Homotopy Analysis Method. *Asian Journal of Applied Sciences, 5: 60-66.*

Verschelde, J., 1999. Algorithm 795: PHCpack: A general-Purpose Solver for Polynomial Systems by Homotopy Continuation. ACM Transaction on Mathematical Software 25(2): *251 – 276*. Software available at http://www.math.uic.edu/jan.