# Some issues in solving non-linear polynomial equations

Peter K. Anderson

**Abstract**

A simple geometrical problem generates a degree 8 polynomial function after firstly applying Pythagoras' Theorem and then squaring the resulting equation to derive a more elegant polynomial equation. Real and complex solutions are explored by root finding functions available in R packages together with other readily available software. The degree 8 polynomial has 4 real and 4 complex solutions as expected. Squaring introduces extraneous solutions and only one of the final 8 solutions solves the Crossed Ladders problem explored in the text.

**Keywords:** similar triangles, polynomials, R, 1[D] root finding.

## Introduction

Some mathematical problems are innately complicated, and some, perversely, need to be made so, according to the dictum of a great mathematician: "Mathematics makes easy things hard in order to make hard things easy"[1]. There are some, however, which are framed in simple terms, but which turn out to be unexpectedly difficult. From Archimedes (c. 500 BC) is said to have originated the famous "cattle problem" (Dickson, 1919). It involved numbers of cattle of different sorts and different colours and involved solving an equation with only two squares[2]. However, an integer solution consisted of 41 digits! The NQueens problem (Campbell, 1977) involving the positioning 8 queens on an 8 x 8

[1] TG Room, Advice to students, Professor of pure mathematics University of Sydney 1935-40, 1945-68.

[2] https://mathworld.wolfram.com/ArchimedesCattleProblem.html

chessboard and had 12 fundamental solutions[3], yet it took Gauss (1777 – 1855) 2 years to complete the solution.

In the light of such experiences, this paper will consider a seemingly simple geometrical problem, the Crossed Ladders Problem[4] (source unknown), which although hardly meriting the description of a 'problem', nonetheless, is not without its interest, generating, as it does, an 8 degree polynomial equation from a simple analysis using the well-known Pythagoras' Theorem. What might once have taken years to solve in detail, is now readily solvable with computer software.

## Crossed ladders problem

We provide a context for solving polynomial equations with equations resulting from applying Pythagoras' theorem to solve the Crossed Ladders Problem. Consider two ladders of respective lengths 3m and 4m leaning across a narrow path by making angles with vertical walls on each side (Figure 1). Their crossing point is at a height of 1m above the path. We are required to find the width of the path.

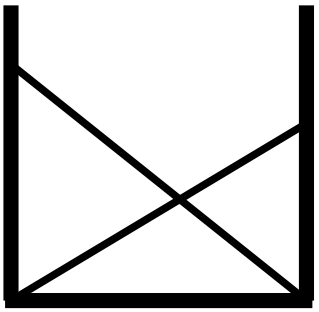

**Figure 1** Two ladders of respective lengths 3m and 4m lean across a narrow path by making angles with vertical walls on each side.

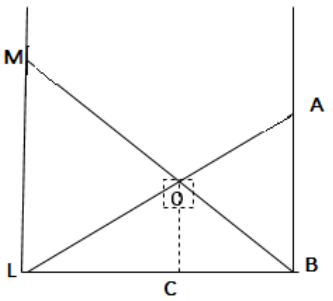


**Figure 2** shows again the two ladders with a perpendicular line drawn down from their intersection point O.

[3] https://www.geeksforgeeks.org/n-queen-problem-backtracking-3/

[4] https://mathworld.wolfram.com/CrossedLaddersProblem.html

The corresponding labelled diagram (Figure 2) is shown with a perpendicular line drawn down from the intersection point O, set at 1m above the path. We let $A0 = x$, $B0 = y$, $AB = p$, $OL = 3 - x$, $OM = 4 - y$, $ML = q$, $BL = d$, and lastly, $BC = l$. We are required to find $d$.

Examination of the figure shows a number of relationships, the first stemming from Pythagoras' theorem:

$$l^2 + 1 = y^2 \tag{1}$$

$$d^2 + q^2 = 16 \tag{2}$$

$$\frac{1}{y} = \frac{q}{4} \tag{3}$$

Then, combining (1), (2) and (3) we have:

$$\frac{1}{\sqrt{l^2+1}} = \frac{\sqrt{16-d^2}}{4} \tag{4}$$

Similarly, triangles LCO and LBA yield the further relationships:

$$(d-l)^2 + 1 = (3-x)^2 \tag{5}$$

$$d^2 + p^2 = 9 \tag{6}$$

$$\frac{1}{3-x} = \frac{p}{3} \tag{7}$$

Combining these three equations we have:

$$\frac{1}{\sqrt{(d-l)^2+1}} = \frac{\sqrt{9-d^2}}{3} \tag{8}$$

Now equations (4) and (8) contain only $l$ and $d$. Since the former is not needed, an equation in $d$, the required width, may be formed by making $l$ the subject of each equation. Thus, after separately squaring (which will, of course, introduce unwanted solutions) and inverting each of equations (4) and (8), we obtain:

$$l = \sqrt{\frac{16}{16-d^2} - 1} \tag{9}$$

and so

$$l = d - \sqrt{\frac{d^2}{9-d^2}} \tag{10}$$

whence

$$\sqrt{\frac{d^2}{16-d^2}} = d - \sqrt{\frac{d^2}{9-d^2}}. \quad (11)$$

Eliminating $d$ (presuming $d \neq 0$) from the numerators yields the following equation:

$$\sqrt{\frac{1}{16-d^2}} + \sqrt{\frac{1}{9-d^2}} - 1 = 0. \quad (12)$$

which, although not in a particularly elegant form for solving, at least, contains only $d$.

**Initial inspection**

The following section of the paper uses command lines from the R statistical computing and graphics programming language. Denoting equation (12) as a function *f*, graphing is obtained using the following two commands from the Curve function[5] in R:

(i) curve(f, from = -3, to = 3, col = "red", lwd = 2, xname = "d")

(ii) abline(h = 0, lty = 3).

The second command (ii) contains standard R graphical parameters; in the first command (i) the limits -3 and 3 come from the obvious physical constraints of the problem. Thus, initial plotting of

$$f12(d) = (1/(16-d^2))^{0.5} + (1/(9-d^2))^{0.5} - 1$$

(Figure 3) indicates 2 possible solutions to eqn. (12) here designated as *f12(d)* = 0. The negative value of d, of course, is not physically tenable and was introduced by the squaring process.

[5] https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/curve

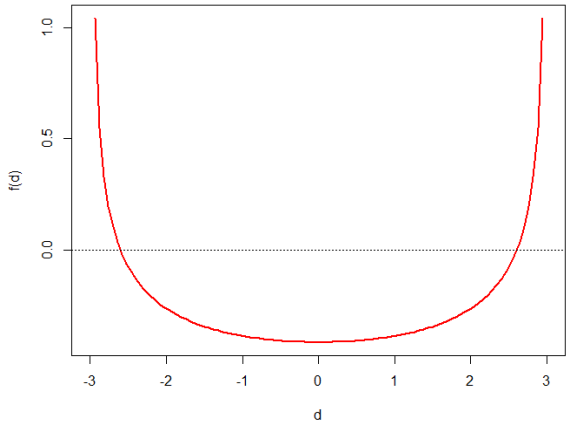


**Figure 3** Initial plotting to search for solutions to equation (12) where *f12_(d)* = 0. Two solutions are evident with the negative value of d, of course, not physically tenable and introduced by the squaring process

## Solutions from uniroot in R

A root-finding technique for polynomials within an interval is available with the function *uniroot* in the R *rootSolve* package. Based on the well-known bisection method of successive approximation, it requires the input of a function, and an interval, at one end of which the function is positive, at the other, negative and so passing through zero. This is a strong assumption, often calling for the inspection of the function by plotting as provided above (Figure 3).

Depending on the difficulty of the function, plotting may be required to find two points which may enclose a root. Then the R command would be of the following form:

uniroot(f, c(0,3)).

Since *f* is a simple function, other possible arguments for *uniroot* (dealing with tolerance and maximum number of iterations) need not, in this instance, be specified. Their default values are displayed in the output, the key parts of which are the value of the root, and the value of the function at the root, typically not exactly zero, but approximately so.

Thus, applied to equation (12) with a search domain of [0,3] uniroot readily gives d = 2.603288, which finishes the problem there, noting that, while d = −2.603288 may be also a mathematical solution, it is non - applicable for the physical situation, being a negative of distance.

**Further development**

With some further algebraic development, it is possible to produce an explicit expression for *d*, i.e. one that is free of radicals. Thus, rearranging equation (12) and squaring (again raising the possibility of the further addition of superfluous solutions) gives:

$$\frac{1}{16-d^2} = -2\sqrt{\frac{1}{9-d^2}} + \frac{1}{9-d^2} + 1 \qquad (13)$$

which, after squaring again, yields:

$$\left[\frac{-151+25d^2-d^4}{(16-d^2)(9-d^2)}\right]^2 = \frac{4}{9-d^2}. \qquad (14)$$

Then after multiplying out and re-arranging we have the equation:

$$d^8 - 46\,d^6 + 763\,d^4 - 5374\,d^2 + 13585 = 0. \quad (15)$$

This is the degree 8 polynomial mentioned previously in the introduction, and there is some interest to be had in determining its roots. Denoting eqn. (15) as function *g15(d)* and again applying *uniroot*, after some plotting, the function yields *d* = 2.60329, as previously, with a function value at the zero of -0.0031118, although no other root.

**Initial inspection**

Initial plotting of

$$g15(d) = d^8 - 46\,d^6 + 763\,d^4 - 5374\,d^2 + 13585$$

(Figure 4) indicates 8 possible solutions (4 real and 4 complex) to eqn. (15) located symmetrically about the y axis and shown in greater detail in graphical displays (Figures 5 & 6).

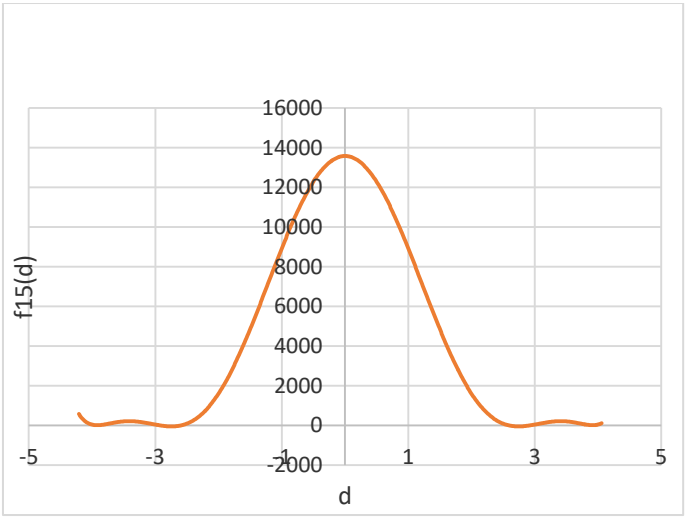


**Figure 4** Initial graph of polynomial *g15(d)* permitting overall inspection.

There are 4 minimum turning points which would produce 8 real solutions if all minima lay below the x-axis and the curves cut the axis in 8 places. As only two minima lie above the x-axis, 4 out of the 8 possible solutions are complex.

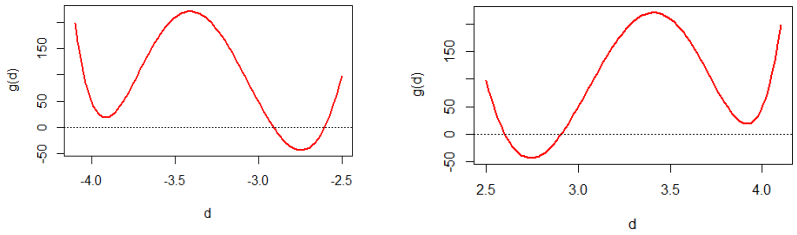


**Figure 5** Plot of eqn. (15) LHS as a function of d locating approximate negative (LHS) and positive (RHS) positions of roots or real and complex zero solutions.

## Solver optimisation tool

Solver, available as an Excel add-in, is an optimisation tool finding maximum, minimum, or zero values in an objective cell, subject to given constraints or limits[6]. With a given starting point,

[6] https://www.solver.com/excel-solver-online-help

Solver will stop after finding the first point satisfying the set conditions. Thus, it might find relative minima or maxima but not necessarily absolute values. The choice, therefore, has to be made of starting points for iterative searches after the overall shape of the polynomial curve has been determined (Figure 4).

Results of searches of polynomial *g15(d)* using the Solver optimisation tool (Table 1) show two zeros where a minimum lies below the x axis and a minimum only where it does not (Figure 5, RHS). In the latter case, we can expect to find complex roots. Initial starting points were determined by inspection of Figure 4. Solutions for –ve x starting points are not necessary because of the symmetry of the curve.

**Table 1** Results of searches of polynomial *g15(d)* using the Solver optimisation tool showing two zeros where a minimum lies below the x axis and a minimum where it does not (Figure 5).

| **Initial value of d (Search starting point)** | **Solution value of d** |
|---|---|
| 2 | 2.60328781925415 (zero) |
| 3 | 2.90907216330945 (zero) |
| 4 | 3.91635300301455 (min) |
| 5 | 3.91635300301455 (min) |

## Uniroot.all

Within R's rootSolve package there is also a function *uniroot.all* which is designed specifically to identify multiple function roots within an interval, by division into sub-intervals[7]. It needs to be made available first via the command *library(rootSolve)* (see Appendix). Then, the minimum syntax (omitting some optional arguments) is similar to that required for *uniroot*:

```
uniroot.all( g, c(2.90970, 2.90975)),
```

which provides a +ve root (Figure 5) as $d = 2.909072$.

[7] https://www.rdocumentation.org/packages/rootSolve/versions/1.8.2.1/topics/uniroot.all

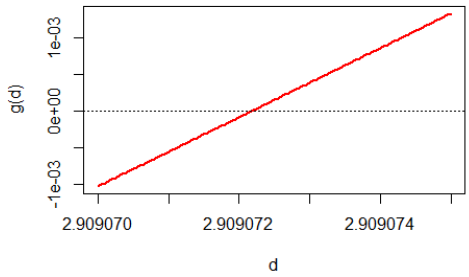


**Figure 6** A closer examination again of shows the zero position between 2.909072 and 2.909073.

The syntax:

uniroot.all( g, c(2.0, 2.9)),

provides a second root (Figure 5) as $d$ = 2.603288. Corresponding –ve roots are readily available by symmetry**.**

**Polyroot**

Also within R there is the *polynom* package[8], with a function **solve**[9] which is designed to reveal the complex roots of a polynomial. Here we enter the polynomial considered in this paper:

$$g15(d) = d^8 - 46\,d^6 + 763\,d^4 - 5374\,d^2 + 13585$$

in the following format:

```
polynomial(8:1)
8 + 7*x + 6*x^2 + 5*x^3 + 4*x^4 + 3*x^5 + 2*x^6 + x^7
> p <- as.polynomial(c(13585,0,-5374,0,763,0,-46,0,1))
> p13585 - 5374*x^2 + 763*x^4 - 46*x^6 + x^8
> solve(p)
```

[8] https://www.rdocumentation.org/packages/polynom/versions/1.4-0

[9] https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/solve

Both real and complex solutions are obtained as follows (c.f. Figure 5):

(i) Real Solutions where *g15(d)* cuts the x axis:

-2.909072 -2.603288
2.603288 2.909072,

(ii) Complex solutions (Figure 7) where *g15(d)* minima lie above, and so does not cut, the x axis:

$z_1$ = 3.922411 - 0.072176i $z_2$ = 3.922411 + 0.072176i
$z_3$ = -3.922411 - 0.072176i $z_4$ = -3.922411 + 0.072176i.

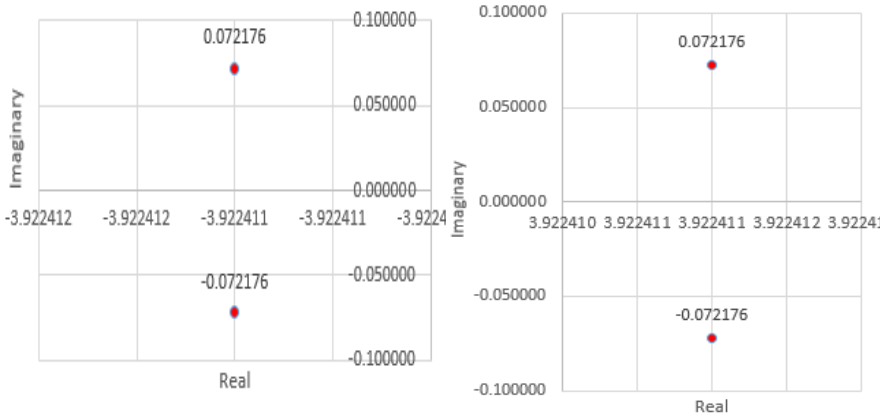


**Figure 7** Complex roots (3.922411 ± 0.072176i, -3.922411 ± 0.072176i) shown on Real and Imaginary axes of an Argand diagrams to be compared with Figure 5 where minima do not cut the x axis and so complex or imaginary roots are formed.

These solutions are complex (imaginary) rather that real, but will still solve the g15(d) = 0 polynomial, with the imaginary components becoming real upon squaring.

**Allroots( ) function in Maxima**

Maxima is a computer algebra system used for the manipulation of symbolic and numerical expressions. It yields high precision

numeric results and plots functions and data in two and three dimensions[10].

For solving again the polynomial:

$$g15(d) = d^8 - 46\,d^6 + 763\,d^4 - 5374\,d^2 + 13585,$$

the format used is:

```
eqn: x^8-46*x^6+763*x^4-5374*x^2+13585;
          8         6             4             2
       x - 46 x + 763 x    - 5374 x  +  13585
soln: allroots (eqn);
```

giving the Real solutions (c.f. Figure 5):

x = 2.603287754423185, x = - 2.603287754423191,
x = - 2.909072186343945, x = 2.909072186343939.

and the Complex solutions (c.f. Argand diagram, Figure 7):

x = 0.07217649359882254 i + 3.92241066021693,
x = 3.92241066021693 - 0.07217649359882254 i,
x = 0.0721764935987751 i - 3.922410660216924,
x = - 0.0721764935987751 i - 3.922410660216924.

all of which satisfy *g15(d).* Thus, all 8 roots of the 8 degree polynomial are accounted for.

**Summary and conclusion**

A simple geometrical problem was shown to generate a degree 8 polynomial function after firstly applying Pythagoras' Theorem and then squaring the resulting equation to derive a more elegant polynomial equation. Real and complex solutions were found by root finding functions available in R packages together with other readily available software.

[10] https://swmath.org/software/560

The polynomial has 4 real and 4 complex solutions as expected. Squaring introduces extraneous solutions and only one of the final 8 solutions solves the Crossed Ladders problem. We notice that while the value d = ± 2.909072 satisfies equation (15), it does not satisfy equation (12), and so is not a solution to the original problem. There is, however, an interesting feature of this extraneous root: it is hidden away between 2.909070 where g15(d) is negative, and 2.909075 (Figure 6) where the function is positively making its existence virtually undetectable by plotting.

To summarize, there was an extra effort in the simple project of casting equation (12) into an elegant form, and in finding and discarding a spurious root, but there were also some useful learnings involved.

## References

Campbell, P. J. (1977). Gauss and the Eight Queens Problem. *Historica Mathematica, 4*, 397-404.

Dickson L. E., (1919). *The history of the theory of numbers* Carnegie Institute of Washington: Washington, US.

R Foundation. (2020). The R project for statistical computing Retrieved 25 August 2020, from https://www.r-project.org

Wolfram. (2020). Mathematica Retrieved 25 August 2020, from https://wolfram.com

Swmath. (2020). Maxima Retrieved 25 August 2020, from https://swmath.org/software/560

## Acknowledgements

The author acknowledges the assistance of Dr R King for helpful and useful discussions and assistance with the initial R script.

## Author

Peter K. Anderson, PhD
Head, Department of Mathematics & Computing Science, DWU
Email: panderson@dwu.ac.pg

## Appendix – Code used

```
library(rootSolve)
#--------------------------------#
uniroot
#If equation (12) is described as f
        f <- function(d) { 1/sqrt(16 - d^2) + 1/sqrt(9 - d^2) -1 }
# and the search interval is 0 -> 3
        uniroot(f,c(0,3)) $root
# gives d = 2.603288 and for more information, enter:
        uniroot(f,c(0,3))
# showing that the function value at the root is of the order of 1e-
07.
# Thus:
        f(2.603288)
# satisfies equation (12), as by symmetry so does (-2.603288)
#--------------------------------#
The two following commands produce a plot of f:
        curve(f, from = 0, to = 3)
        abline(h = 0, lty = 3)
#--------------------------------#
if equation (15) is described as g
        g <- function(d){ d^8 -46*d^6 + 763* d^4 -5374*d^2 +
        13585}
# For uniroot, some experimentation with the search endpoints is
needed to produce opposite signs bracketing a root.
# Thus:
        uniroot(g,c(2.0, 2.9))$root
# gives d = 2.603288 as before.
#--------------------------------#

Applying uniroot.all using a similar syntax to uniroot:
        uniroot.all(g, c(1, 3))
# gives (i) 2.603291 which is a slightly less accurate value than
the previous 2.603288 because
```

```
        abs(g(2.603291)) > abs(g(2.603288))
# and also (ii) 2.909072, which, however, does not satisfy the original eqn (12) since f(2.909072) is not zero or approximately so.
# ------------------------------------------------#
Plotting the new root of (15)
        curve(g, from = 2.909070, to = 2.909075)
        abline(h = 0, lty = 3)
# ------------------------------------------------#
In Mathematica, g is defined as follows:
#  g[d_]: = d^8 -46*d^6 + 763* d^4 -5374*d^2 + 13585;
# The syntax for finding a root(s) of a function g of d would be:

# FindRoot[g, {d = s}] where 's' is some starting point.
# real and complex roots come from the one command
# ------------------------------------------------

# test the complex roots of g by substitution
        g(-3.92241 + 0.0721765i)
        g(-3.92241 - 0.0721765i)

        g(3.92241 + 0.0721765i)
        g(3.92241 - 0.0721765i)

# all of which give 0 (approximately).

Real & Complex solutions with Maxima
        eqn: x^8-46*x^6+763*x^4-5374*x^2+13585;
        8          6           4           2
        x  - 46 x  +   763 x  -   5374 x  +  13585

soln: allroots (eqn);
        x = 2.603287754423185, x = - 2.603287754423191,
        x = - 2.909072186343945, x = 2.909072186343939,
        x = 0.07217649359882254 i + 3.92241066021693,
```

x = 3.92241066021693 - 0.07217649359882254 i,
x = 0.0721764935987751 i - 3.922410660216924,
x = - 0.0721764935987751 i - 3.922410660216924

The result from *Maxima* was:

$$d = \pm(-3.92241 \pm 0.07217651i),$$
$$d = \pm 2.909072, \pm 2.60329,$$