Using the Rational Root Test to Factor with the TI-83

Timothy W. Jones

October 6, 2019

Abstract

The rational root test gives a way to solve polynomial equations. We apply the idea to factoring quadratics (and other polynomials). A calculator speeds up the filtering through possible rational roots.

Introduction

The rational root test (RRT) is typically taught in college algebra classes [1]. It says that a polynomial will have a rational root, p/q, if p divides its constant term and q divides its leading coefficient. Typically students are given a high degree polynomial and asked to test a series of such numbers using synthetic division. If a root is found, then the process can be repeated on the quotient derived.

The rational root test is not used, in the college and elementary algebra books I've read, to factor a quadratic. In this article we will make a case that it can be used to factor quadratics.

The potentially tedious task of trial and error in finding rational roots with the RRT is speeded up by writing program in a calculator. The program requires a *for* loop, an *if* statement, and various other features of the language; it is a good programming exercise. It is especially nice in that the theory is proven to be useful in a practical way. All in all one can give to students a feeling of the power of exhaustive, clear thinking leading to perfect factoring.

We'll first give the theory, in its simplest form, and then present the program. The program is short and sweet.

The theory

We'll make it simple. Consider the linear case: solve $ax + b = 0$. The solution is $x = -b/a$. That means the constant divides the numerator of the solution rational and the leading coefficient divides its denominator. This generalizes to the quadratic case. So, for $ax^2 + bx + c = 0$ we notice that the roots of

$$ax^2 + bx + c = 0 \text{ are the same as } x^2 + \frac{b}{a}x + \frac{c}{a} = 0.$$

So if the roots are the rationals d/e and f/g, we have

$$(x - d/e)(x - f/g) = x^2 - (d/e + f/g) + \frac{df}{eg}$$

and hence c divides df and a divides eg. This means the roots must be of the form

$$\frac{\text{divisor of } df \text{ or } c}{\text{divisor of } eg \text{ or } a}.$$

Here's an example. Consider that $(x-3/5)(x+2/7)$ has the same roots as $(5x - 3)(7x + 2) = 35x^2 - 11x - 6$. The roots are $3/5$ and $-2/7$; the product of their denominators is 35 and that of the numerators is -6. So the form of rational roots must be

$$\frac{p}{q} = \frac{\pm\text{divisor of } 6}{\text{divisor of } 35}$$

The rational root test yields that any rational root of $ax^2 + bx + c$ must be between $-|c|$ and $|c|$. This follows as the c/a is at its max when $a = 1$, giving c, assuming here that $c > 0$.

The net is we can get a calculator to cycle through all combinations of fractions using

$$x\frac{1}{a},$$

where x an integer is such that $-|ac| \le x \le |ac|$.

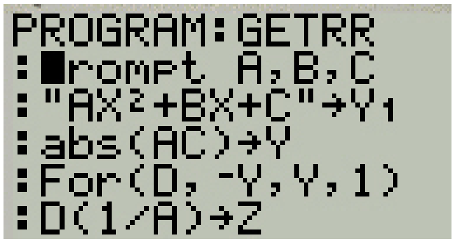
Continuing with our example, we check for roots of the form

$$-6 \times 35 \le x \le 6 \times 35 \text{ with } x \times \frac{1}{35}.$$

All possible rational roots are included – and then some. Eventually we hit the roots of our example: $3/5$ is $21 \cdot 1/35$ and $-2/7$ is $-10 \cdot 1/35$.

The code

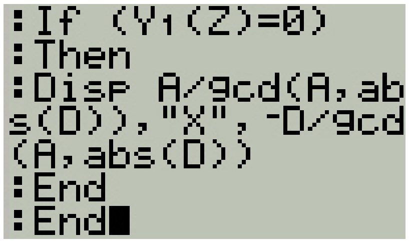
Screen captures on a TI-83 calculator are given in Figures 1 and 2



```
PROGRAM:GETRR
:█rompt A,B,C
:"AX²+BX+C"→Y₁
:abs(AC)→Y
:For(D,⁻Y,Y,1)
:D(1/A)→Z
```

Figure 1: Code for getrr, get rational roots program.

The number of iterations is equal to the number of fractions with $1/a$ denominator between $-c$ and c: $2|ac|$.



```
:If (Y₁(Z)=0)
:Then
:Disp A/gcd(A,ab
s(D)),"X",⁻D/gcd
(A,abs(D))
:End
:End█
```

Figure 2: Code continued for getrr.

$16x^2 - 16x + 3$	$(4x-1)(4x-3)$
$35x^2 + 11x - 10$	$(5x-2)(7x+5)$
$35x^2 - 11x - 10$	$(5x+2)(7x-5)$
$10x^2 + 29x + 21$	$(2x+3)(5x+7)$
$6x^2 - 19x + 15$	$(2x-3)(3x-5)$
$10x^2 + x - 21$	$(2x+3)(5x-7)$
$10x^2 - x - 21$	$(2x-3)(5x+7)$

Table 1: Test cases.

Conclusion

It is possible to speed the given program up and further tweak it. On a TI-83, an older version of this calculator, it can take a full minute to cycle through all the combinations. Certainly a while loop could be used with a STOP command, after two roots are found. Also if A, B, and C are all positive the roots can't be positive, so the range of values tested can be further shortened for such cases. These tweaks make for nice extra credit possibilities.

One can generalize the program to check for all rational roots of a general polynomial. The fastest way is to prompt for the degree of the polynomial and then fill an array or a matrix with coefficients as necessary. An interesting challenge is to get Descartes rule for signs involved to speed up this checking and to adjust window dimensions to see a graph for the polynomial.

References

[1] R. Blitzer, *Algebra and Trigonometry*, 4th ed., Upper Saddle, NJ, 2010.