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

Timothy W. Jones

October 1, 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 $ax^2 + bx + c = 0$, a rational root, p/q will be such that p divides the constant c and q divides the leading coefficient a. We can actually read this off of the quadratic formula:

$$\frac{p}{q} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{1}$$

implies that if one root is rational, so is the other. It also implies that

$$\left(x - \frac{-b + \sqrt{b^2 - 4ac}}{2a}\right)\left(x - \frac{-b - \sqrt{b^2 - 4ac}}{2a}\right)$$

$$= x^2 + \frac{b}{a}x + \frac{c}{a}$$

has the same roots as $ax^2 + bx + c$. So rational roots, if they exist, will be products of two denominators equal to c and two numerators equal to a.[1]

The rational root test yields that any rational root must be between $-|c|$ and $|c|$ as the smallest and largest rational with the divisor for numerators and denominators is c is the largest divisor of c and 1 is the smallest divisor of a. We then know that a rational root will be some multiple of the form

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

where x an integer is such that $-|c| \leq x \leq |c|$. This looks like a for loop for a computer program. We have our theory.

[1] One can make a program that gives these two roots in fraction form and asks for denominators of displayed fractions. This entry is converted into a $ax + b$ form and is listed as one factor. Done twice per (1) the quadratic is factored. We don't supply this code; it's more cumbersome than using the RRT.

The code

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

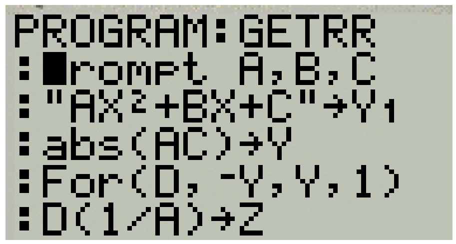

Figure 1: Code for getrr, get rational roots program.

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

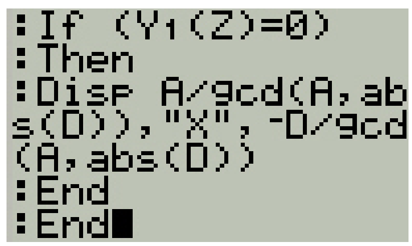

Figure 2: Code continued for getrr.

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. As the program is written in cycles through the entire list even after two roots have been found. Also if A, B, and C are all positive the roots can't be negative, so the range of values tested can be further shortened for such cases. These tweaks make for nice extra credit possibilities.

References

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