Math 421 • 13 September 2006

# The Factor Theorem and a corollary of the Fundamental Theorem of Algebra



For your convenience, the input cells in this notebook are already evaluated, so that you may read it with *MathReader*. When you have access to *Mathematica* itself, you should carry out the evaluations one-by-one.

As you work through this notebook in *Mathematica*, if you come across a *Mathematica* function you don't understand, try using the `?` information command to find out something about it. (And follow the hyperlink in it, if any, to HelpBrowser help.) For example, to learn about `PolynomialQuotient`, used in the first section below, evaluate:

```
In[1]:= ?PolynomialQuotient

        PolynomialQuotient[p, q, x] gives the quotient of p
           and q, treated as polynomials in x, with any remainder dropped. More…
```

## The Factor Theorem

The theorem is:

**The Factor Theorem.** Let $P(z)$ be a polynomial in $z$ (with real or complex coefficients) of degree $n > 0$. Then a (real or complex) number $z_0$ is a root of $P(z)$ if and only if

$$P(z) = (z - z_0)\, Q(z)$$

for some polynomial $Q(z)$ of degree $n - 1$.

**Proof.** First assume that $z_0$ is a root of $P(z)$. By long division, there is a quotient polynomial $Q(z)$ and a remainder polynomial $R(z)$ for which $P(z) = (z - z_0)\, Q(z) + R(z)$. Since the divisor $z - z_0$ is of degree 1, the remainder polynomial $R(z)$ is of degree 0, that is, a constant $r$. Then $P(z) = (z - z_0)\, Q(z) + r$. Take $z = z_0$ in both sides of this equation to obtain $0 = P(z_0) = 0\, Q(z_0) + r$, so that $r = 0$. Hence $P(z) = (z - z_0)\, Q(z)$. Since $P(z)$ has degree $n$ whereas $z - z_0$ has degree 1, then $Q(z)$ has degree $n - 1$.

Conversely, suppose $P(z) = (z - z_0)\, Q(z)$ for a polynomial $Q(z)$ of degree $n - 1$. Take $z = z_0$ in this equation to obtain $P(z_0) = 0\, Q(z_0) = 0$. Hence $z_0$ is a root of $P(z)$. ∎

### ■ Example of the Factor Theorem

```
In[2]:= p[z_] := -13 + 17 z - 5 z^2 + z^3
```

By inspection, 1 is a root of this polynomial:

```
In[3]:= p[1]

Out[3]= 0
```

Give this root a name:

```
In[4]:= z0 = 1;
```

The question is how to form the quotient polynomial $Q(z)$ in *Mathematica*. (As usual, begin user-defined *Mathematica* names with use lower-case letters.)

One way to obtain the quotient:

```
In[5]:= q[z_] := PolynomialQuotient[p[z], z - z0, z]
        q[z]

Out[6]= 13 - 4 z + z^2
```

Another way to obtain the quotient:

```
In[7]:= Simplify[p[z] / (z - z0)]

Out[7]= 13 - 4 z + z^2
```

In turn find roots of the quotient:

```
In[8]:= quadraticRoots = Solve[q[z] == 0, z]

Out[8]= {{z → 2 - 3 i}, {z → 2 + 3 i}}
```

Factor the quotient:

```
In[9]:= Factor[q[z]]

Out[9]= 13 - 4 z + z^2
```

Ugh! That does nothing. Here's one way to factor the quotient:

```
In[10]:= quadraticProduct = Apply[Times, z - w /. (quadraticRoots /. z → w)]

Out[10]= ((-2 - 3 i) + z) ((-2 + 3 i) + z)
```

Whoa—that expression looks rather complicated! How does it work? To see, build up the entire expression in pieces, starting with:

```
In[11]:= quadraticRoots /. z → w

Out[11]= {{w → 2 - 3 i}, {w → 2 + 3 i}}
```

So that piece merely replaces `z` by `w` in the list of rules `{{z→2-3 i},{z→2+3 i}}`. The reason for doing that is we want to use `z` for another purpose in a moment. Next:

```
In[12]:= w /. (quadraticRoots /. z → w)

Out[12]= {2 - 3 ⅈ, 2 + 3 ⅈ}
```

So that expression converted the list of rules to an actual list of numbers, namely, the roots of the quadratic quotient. Next:

```
In[13]:= z - w /. (quadraticRoots /. z → w)

Out[13]= {(-2 + 3 ⅈ) + z, (-2 - 3 ⅈ) + z}
```

So that expression forms the list consisting of the two factors `z - (2 - 3ⅈ)` and `z - (2 + 3ⅈ)` of the quadratic quotient. Finally, the functional programming construct `Apply[Times, …]` is going to apply the `Times` (multiplication) function to that list of two factors, in other words, to form:

```
In[14]:= Times[z - (2 - 3 ⅈ), z - (2 + 3 ⅈ)]

Out[14]= ((-2 - 3 ⅈ) + z) ((-2 + 3 ⅈ) + z)
```

Thus the expression `Apply[Times, z - w /. (quadraticRoots /. z → w)]` forms all at once this product of the two linear factors of the quadratic quotient polynomial. That result was assigned to `quadraticProduct`.

```
quadraticRoots /. z → w
```

That expression simply changes z to w in the list of rules {{z→2-3 *i*},{z→2+3 *i*}}

It so happens that the two complex roots of the quadratic quotient polynomial have integers as real and complex parts; in other words, these two complex roots belong to the set of **Gaussian integers**. Then in this example, another way to factor the quotient polynomial is to use `Factor` after all, but with the option `GaussianIntegers → True`:

```
In[15]:= Factor[q[z], GaussianIntegers → True]

Out[15]= ((-2 - 3 ⅈ) + z) ((-2 + 3 ⅈ) + z)
```

Factor the original cubic:

```
In[16]:= (z - z0) quadraticProduct

Out[16]= ((-2 - 3 ⅈ) + z) ((-2 + 3 ⅈ) + z) (-1 + z)

In[17]:= p[z] == ComplexExpand[%]

Out[17]= True

In[18]:= Clear[p, q]
```

## Corollary to the FTA

**The Fundamental Theorem of Algebra** (**FTA**). Every non-constant polynomial with real or complex coefficients has at least one real or complex root.

**Corollary.** Let $P(z)$ be a polynomial with complex coefficients, of degree $n > 0$. Then $P(z)$ has exactly $n$ (not necessarily distinct) complex roots.

The easiest way to make sense for now of the "not necessarily distinct" part of the conclusion there is to rephrase the corollary as follows:

**Corollary** (restated). Let $P(z)$ be a polynomial with complex coefficients, of degree $n > 0$. Then there are $n$ complex numbers $z_1, z_2, \ldots, z_n$, not necessarily different from one another, for which

$$P(z) = (z - z_0)(z - z_1) \cdots (z - z_n) .$$

## ■ Example of the Corollary

```
In[19]:= p[z_] := -4 - (12 + 4 i) z - (11 + 12 i) z^2 - (1 + 12 i) z^3 + (3 - 4 i) z^4 + z^5

In[20]:= Factor[p[z]]

Out[20]= (-2 i + z)^2 (1 + z)^3

In[21]:= Solve[p[z] == 0, z]

Out[21]= {{z → -1}, {z → -1}, {z → -1}, {z → 2 i}, {z → 2 i}}

In[22]:= Clear[p]
```