Partial Fractions

Contents

1 Partial fractions

1.1 Relevant facts about polynomials

Partial fractions is useful whenever you want to reduce a rational function (that is one polynomial divided by another) to a sum of minimal pieces. We first need two facts about polynomials (see MATH 340 for proofs). For those with some field theory, fix a field and view all polynomials, factors, etc below as over that field.

By convention we will say that the degree of the constant zero polynomial is $-\infty$.

Proposition (Division algorithm for polynomials)**.** *Let f and g be polynomials. Then there exist polynomials q and r with $\deg r < \deg g$ and*

$$f(x) = g(x)q(x) + r(x)$$

The proof is by long division.

Proposition (Extended Euclidean algorithm for polynomials)**.** *Let f and g be polynomials with no common factor then there exist polynomials h and t such that*

$$f(x)h(x) + g(x)t(x) = 1$$

The proof of this is too involved to summarize in a sentence, but it works the same way as the extended Euclidean algorithm for integers, and, as in that case, it yields a good algorithm. In general the algorithm is $O(n^2)$ where n is the maximum degree of f and g, but key parts of it, which are sufficient for many purposes, are faster (see `http://planetmath.org/encyclopedia/HalfGCDAlgorithm.html`).

1.2 The partial fractions theorem

Now we are ready for the partial fraction decomposition, first for two factors and then in general. You can find many presentations of this result online, this presentation is based on `http://marasingha.org/mathspages/partialfrac/html/node2.html`.

Proposition. *Let f_1, f_2, and g be polynomials such that f_1 and g_1 have no common factor and $\deg g < \deg f_1 + \deg f_2$. Then we can find polynomials g_1 and g_2 with*

$$\frac{g(x)}{f_1(x)f_2(x)} = \frac{g_1(x)}{f_1(x)} + \frac{g_2(x)}{f_2(x)}$$

and $\deg g_1 < \deg f_1$, $\deg g_2 < \deg f_2$.

Proof. By the extended Euclidean algorithm we can find polynomials h_1 and h_2 such that

$$1 = f_1(x)h_2(x) + f_2(x)h_1(x)$$

Thus

$$g(x) = f_1(x)(h_2(x)g(x)) + f_2(x)(h_1(x)g(x))$$

By the division algorithm we can also write

$$h_1(x)g(x) = f_1(x)q(x) + g_1(x)$$

where $\deg(g_1) < \deg(f_1)$. Let

$$g_2(x) = h_2(x)g(x) + q(x)f_2(x)$$

then

$$\begin{aligned} g(x) &= f_1(x)g_2(x) - f_1(x)f_2(x)q(x) + f_2(x)f_1(x)g(x) + f_2(x)g_1(x) \\ &= f_1(x)g_2(x) + f_2(x)g_1(x) \end{aligned}$$

so

$$\frac{g_1(x)}{f_1(x)} + \frac{g_2(x)}{f_2(x)} = \frac{f_2(x)g_1(x) + f_1(x)g_2(x)}{f_1(x)f_2(x)} = \frac{g(x)}{f_1(x)f_2(x)}$$

It remains to show that $\deg(g_2) < \deg(f_2)$. Towards a contradiction suppose that $\deg(g_2) \geq \deg(f_2)$. Then

$$\deg(f_1g_2) \geq \deg(f_1f_2)$$

but

$$\deg(f_2g_1) < \deg(f_1f_2)$$

So in $f_1(x)g_2(x) + f_2(x)g_1(x)$ the f_1g_2 term dominates, so

$$\deg(g) = \deg(f_1g_2 + f_2g_1) = \deg(f_1g_2) \geq \deg(f_1f_2) = \deg f_1 + \deg f_2$$

which is a contradiction, completing the proof. □

Theorem (partial fraction decomposition)**.** *Let f and g be polynomials and write $f = f_1^{a_1} \cdots f_r^{a_r}$ where f_i and f_j have no common factors for all $i \neq j$. Suppose $\deg g < \deg f$, then we can write*

$$\frac{g(x)}{f(x)} = \sum_{i=1}^{r} \sum_{j=1}^{a_r} \frac{g_{ij}(x)}{(f_i(x))^r}$$

for some polynomials g_{ij} with $\deg g_{ij} < \deg f_i$.

Proof. By the proposition applied $r-1$ times we an write

$$\frac{g(x)}{f(x)} = \sum_{i=1}^{r} \frac{h_i(x)}{(f_i(x))^{a_i}}$$

with $\deg(h_i) < \deg(f_i^{a_i}) = a_i \deg(f_i)$. (Just this much is enough for the application we have in mind below).

To continue the proof consider

$$\frac{h_i(x)}{f_i(x)^{a_i}}.$$

If $a_i = 1$ then we're done with this value of i. Assume $a_i > 1$. By the division algorithm

$$h_i(x) = q(x)f_i(x) + r(x) \tag{1}$$

with $\deg r < \deg f_i$. Therefore

$$\frac{h_i(x)}{f_i(x)^{a_i}} = \frac{q(x)}{f_i(x)^{a_i-1}} + \frac{r(x)}{f_i(x)^{a_i}} \tag{2}$$

Furthermore either $q = 0$ or $q(x)f_i(x)$ is the dominant term in (1) in which case $\deg h_i = \deg qf_i = \deg q + \deg f_i$. But $\deg h_i < a_i \deg f_i$ so $\deg q < (a_i - 1)\deg f_i$. Thus repeating (2) inductively we get the desired decomposition.

i was arbitrary and so the result follows. □

Example. Rewrite $\frac{x}{(x+1)(x-1)^2}$ by partial fractions:

By the above theorems we know this can be written

$$\frac{x}{(x+1)(x-1)^2} = \frac{A}{x+1} + \frac{B}{x-1} + \frac{C}{(x-1)^2}$$

How should we find A, B, and C? We could follow the proof and use the Euclidean algorithm, but for small hand examples it is usually easier to multiply out

$$\begin{aligned} x &= A(x-1)^2 + B(x+1)(x-1) + C(x+1) \\ &= A(x^2 - 2x + 1) + B(x^2 - 1) + C(x+1) \end{aligned}$$

Now this must be true for all x, so it must be true coefficient by coefficient. That is

$$\begin{aligned} 0 &= A - B + C \quad \text{coefficient of } x^0 \\ 1 &= -2A + C \quad \text{coefficient of } x \\ &0A + B \quad \text{coefficient of } x^2 \end{aligned}$$

This is a system of linear equations which you can solve by your favorite method, say Gaussian elimination

$$\begin{bmatrix} 1 & -1 & 1 & 0 \\ -2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & -1 & 1 & 0 \\ 0 & -2 & 3 & 1 \\ 0 & 2 & -1 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & -1 & 1 & 0 \\ 0 & -2 & 3 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$
$$\to \begin{bmatrix} 1 & 0 & 0 & -\frac{1}{4} \\ 0 & 1 & 0 & \frac{1}{4} \\ 0 & 0 & 1 & \frac{1}{2} \end{bmatrix}$$

Thus

$$\frac{x}{(x+1)(x-1)^2} = -\frac{1}{4(x+1)} + \frac{1}{4(x-1)} + \frac{1}{2(x-1)^2}$$

1.3 Algorithmic concerns

What is the runtime of partial fractions? If we do it by solving the system of equations then the system of equations is $n \times n$ where $n = \deg(f)$. By Gaussian elimination this is $O(n^3)$.

Strassen's algorithm lets one multiply two $n \times n$ matrices in $O(n^{2.81})$; it also gives a matrix inverse in the same time, and hence a system solve in the same time. There have been improvements along similar lines and the current best matrix multiplication is $O(n^{2.3727})$ (according to Wikipedia). However Strassen's algorithm is only faster than a well optimized naive approach for $n > 1000$ and the newer ones are impractical for any matrix you could actually store in a current computer. Certainly no such approach could be better than $O(n^2)$ since one needs to use all n^2 coefficients of the matrix.

However partial fractions has more structure. The proof suggests a different algorithm, one using the extended Euclidean algorithm. In fact partial fractions doesn't even need the full power of the extended Euclidean algorithm, and so with this basic approach and some additional cleverness one can obtain $O(\log n M(n))$ where $M(n)$ is the runtime needed to multiply two (slightly special) polynomials of degree n. Naively $M(n)$ would be n^2, which already gets us better that the system solving approach, but using fast fourier transforms one can obtain $M(n) = O(n \log n)$ and hence partial fractions can be done in $O(n(\log n)^2)$. The details of all of this are beyond the scope of this course. Reference for the algorithm: Kung, H. T. and Tong, D. M., "Fast algorithms for partial fraction decomposition" (1976). Computer Science Department. Paper 1675. `http://repository.cmu.edu/compsci/1675`.

One final algorithmic concern. What if the factorization of f is not given? This is a whole different story; fast polynomial factorization, say over the rationals, is done by looking modulo primes and then putting it back together. The good news is that it is polynomial time. In the case we're interested in we want to factor down to linear factors so we actually need to factor over the complex numbers, but not all polynomials can have their roots expressed in terms of square roots, cube roots etc, which leads in to the very interesting world of Galois theory.

1.4 Algorithm for coefficient extraction in rational functions

From the coefficient extraction theorems, we can compute $[z^n]\frac{1}{(1-mz)^r} = m^r\binom{r+n-1}{n}$ for real numbers m and r. Let us reduce the computation of coefficient extraction for rational functions to these kinds of terms.

1. Partial fraction Decomposition Simplify the form of the rational function. This is the most computationally intensive part of the computation. Here $\alpha_{i,k}(z)$ is a polynomial of degree less than k. The

$$R(z) = \sum_{i,k} \frac{\alpha_{i,k}(z)}{(1-m_i z)^k}$$

2. Separate Use sum rule to separate into terms of the form $[z^n](1-mz)^k$.

3. Apply the binomial theorem Compute these values

4. Sum Take the sum of the resulting terms.

Example (Simple rational example)**.** Let $R(x) = \frac{2x^2-4x+1}{1-18x+129x^2-460x^3+816x^4-576x^5}$. Compute $[x^n]R(x)$.

1. First we compute that

$$\frac{2x^2-4x+1}{1-18x+129x^2-460x^3+816x^4-576x^5} = \frac{-16}{(1-4x)} + \frac{12}{(1-3x)} + \frac{2}{(1-4x)^3} + \frac{3}{(1-3x)^2}.$$

   We can use several techniques to do this manually, or the Maple command `convert(R(x), parfrac)`.

2. We expand and solve.

$$\begin{aligned}
[x^n]R(x) &= [x^n]\frac{-16}{(1-4x)} + \frac{12}{(1-3x)} + \frac{2}{(1-4x)^3} + \frac{3}{(1-3x)^2} \\
&= -16[x^n]\frac{1}{1-4x} + 12[x^n]\frac{1}{1-3x} + 2[x^n]\frac{1}{(1+(-4x))^3} + 3[x^n]\frac{1}{(1+(-3x))^2} \\
&= -16\cdot 4^n + 12\cdot 3^n + 2\binom{-3}{n}(-1)^n(-4)^n + 3\binom{-2}{n}(-1)^n(-3)^n \\
&= -16\cdot 4^n + 12\cdot 3^n + 2\cdot 4^n\binom{n+2}{2} + 3^{n+1}\binom{n+1}{1} \\
&= (-16+(n+1)(n+2))\cdot 4^n + (12+3(n+1))\cdot 3^n.
\end{aligned}$$