CDM

Interpolation and Expansion

Klaus Sutner
Carnegie Mellon University

Interpolation

A standard problem:

Interpolation:
Given data points $(a_0, b_0), \ldots, (a_n, b_n)$, $a_i < a_{i+1}$, find a degree n polynomial p such that $p(a_i) = b_i$.

There are two classical ways of doing this:

- Lagrange interpolation
- Newton interpolation

Lagrange

Lagrange's method uses a linear combination of special polynomials of degree n, so-called Lagrange interpolants:

$$L_i^n(x) = \prod_{j \neq i} \frac{x - a_j}{a_i - a_j}$$

Proposition

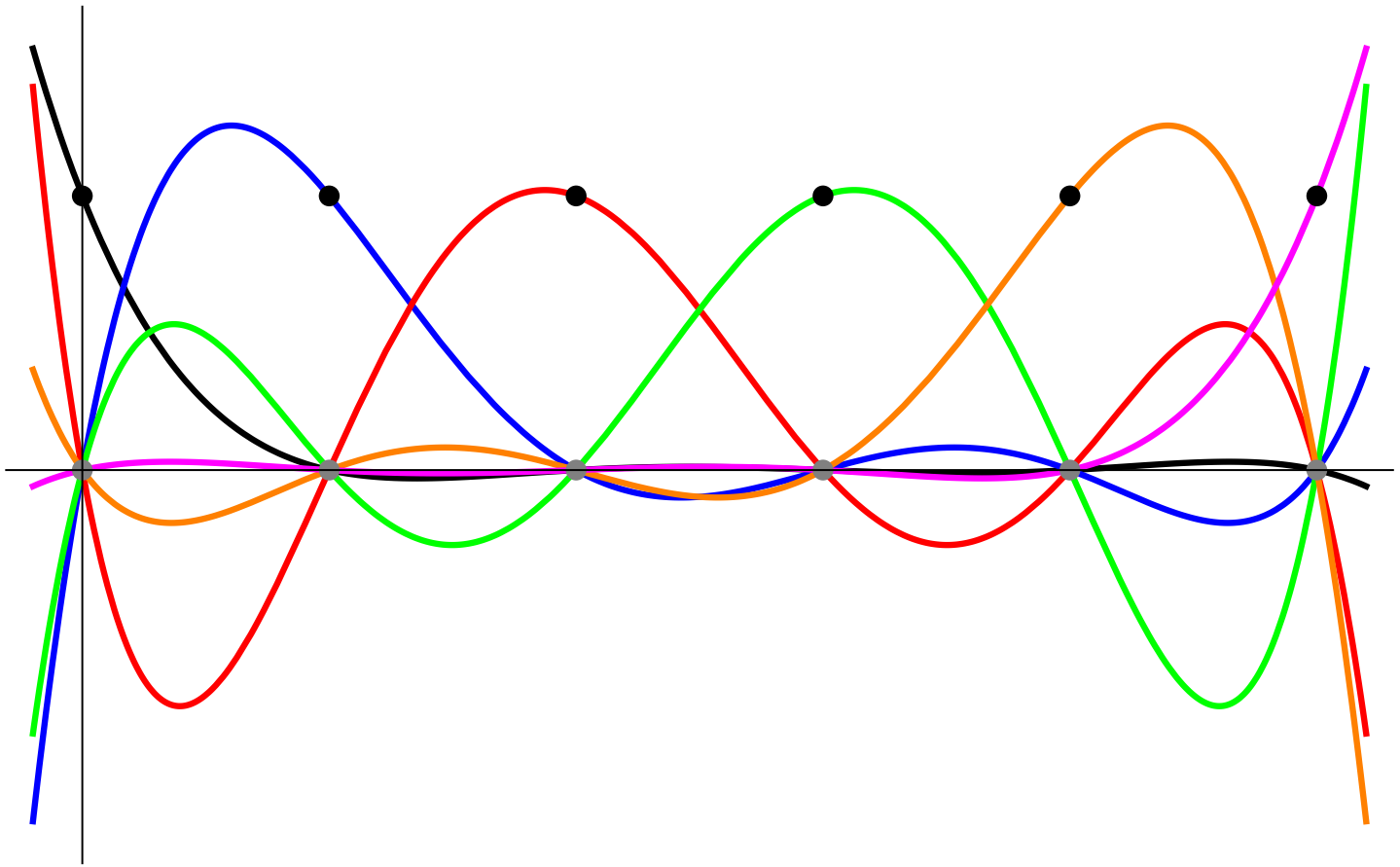
$L_i^n(a_i) = 1$ and $L_i^n(a_j) = 0$ for $i \neq j$.

Hence we can choose

$$p(x) = \sum_{i \leq n} b_i L_i^n(x)$$

By construction, p has degree bound n.

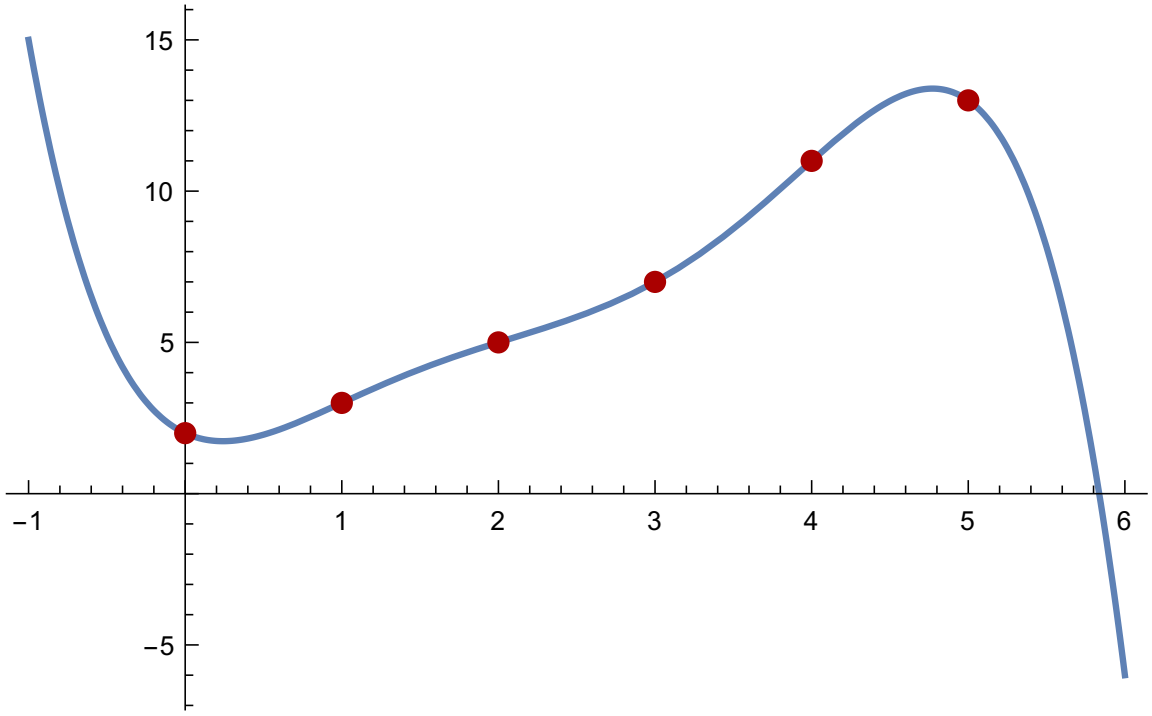
Lagrange Interpolants



Example: Primes

Suppose we want $f(i) =$ the ith prime for $i = 0, \ldots, 5$. The Lagrange interpolation looks like

$$\begin{aligned} f(x) &= 2L_0^6 + 3L_1^6 + 5L_2^6 + 7L_3^6 + 11L_4^6 + 13L_5^6 \\ &= 2 - \frac{143x}{60} + \frac{49x^2}{8} - \frac{85x^3}{24} + \frac{7x^4}{8} - \frac{3x^5}{40} \end{aligned}$$



Lagrange interpolants all have degree n, the same as the target polynomial. By contrast, Newton interpolants have increasing degrees:

$$N_k(x) = \prod_{i<k} x - a_i$$

with the understanding that $N_0(x) = 1$.

Thus N_k has degree k and we have $N_k(a_i) = 0$ for $i < k$. Hence we can write the interpolation polynomial in the form

$$p(x) = \prod_{i \leq n} c_i N_i(x)$$

So the problem is to find the coefficients c_i efficiently.

Differences

To this end we define divided differences $\nu(i,j)$ for $0 \leq i \leq j \leq n$ as follows:

$$\nu(i,i) = b_i$$

$$\nu(i,j) = \frac{\nu(i+1,j) - \nu(i,j-1)}{a_j - a_i}$$

In practice, this is done by dynamic programming in quadratic time.

We get the interpolation polynomial

$$p(x) = \sum \nu(0,i)\, N_i(x)$$

Symbolic Coefficients

$$b_0$$

$$\frac{-b_0+b_1}{-a_0+a_1}$$

$$\frac{-\frac{-b_0+b_1}{-a_0+a_1}+\frac{-b_1+b_2}{-a_1+a_2}}{-a_0+a_2}$$

$$\frac{-\frac{-\frac{-b_0+b_1}{-a_0+a_1}+\frac{-b_1+b_2}{-a_1+a_2}}{-a_0+a_2}+\frac{-\frac{-b_1+b_2}{-a_1+a_2}+\frac{-b_2+b_3}{-a_2+a_3}}{-a_1+a_3}}{-a_0+a_3}$$

$$\frac{-\frac{-\frac{-\frac{-b_0+b_1}{-a_0+a_1}+\frac{-b_1+b_2}{-a_1+a_2}}{-a_0+a_2}+\frac{-\frac{-b_1+b_2}{-a_1+a_2}+\frac{-b_2+b_3}{-a_2+a_3}}{-a_1+a_3}}{-a_0+a_3}+\frac{-\frac{-\frac{-b_1+b_2}{-a_1+a_2}+\frac{-b_2+b_3}{-a_2+a_3}}{-a_1+a_3}+\frac{-\frac{-b_2+b_3}{-a_2+a_3}+\frac{-b_3+b_4}{-a_3+a_4}}{-a_2+a_4}}{-a_1+a_4}}{-a_0+a_4}$$

The first few Newton coefficients $\nu(0,k)$.

To Falling

Difference computations become particularly simple when the support points a_i are equidistant, say, $a_{i+1} = a_i + h$.

Even better is $a_i = i$, $i = 0, \ldots, n$, then

$$N_k(x) = \prod_{i<k} x - i = x^{\underline{k}}$$

In this special case, Newton interpolants turn directly into falling factorials.

Primes, Again

For the prime example, the Newton coefficients are

$$2 \quad 1 \quad \frac{1}{2} \quad \frac{-1}{6} \quad \frac{1}{8} \quad \frac{-3}{40}$$

Producing the unexpanded polynomial

$$2 + x + \frac{1}{2}(x-1)x - \frac{1}{6}(x-2)(x-1)x + \\ \frac{1}{8}(x-3)(x-2)(x-1)x - \frac{3}{40}(x-4)(x-3)(x-2)(x-1)x$$

Our definition of a polynomial in terms of coproducts is very elegant and it brings out the algebraic properties of polynomials very clearly. It even suggests an implementation in terms of (higher-dimensional) coefficient lists.

Unfortunately, it coexists somewhat uneasily with computation. The problem is that an expression like

$$(x + y)^5 - 1$$

morally ought to be a polynomial as well: We can expand out the expression to obtain an explicit polynomial in the strict sense.

In other words, we would like to extend our definition to capture additional types of expressions that can be expanded to yield actual polynomials.

Implicit Polynomials

To this end we need to define a class of terms that can be converted to polynomials by straightforward algebraic manipulations.

Since polynomials are nicely closed under addtion, it suffices to handle multiplication. Any expression

$$P_1 \cdot P_2 \cdot P_3 \cdot \ldots \cdot P_{k-1} \cdot P_k$$

where all the P_i are polynomials should be considered to be an implicit polynomial.

This convention also covers exponential terms (with exponents in $\mathbb{N}$) as in the example from the last slide.

Old Hat

We have used this implicit representation in several places already, without any protest from the audience.

- The expressions $c(x - a_1)(x - a_2) \dots (x - a_n)$ encountered in the root decomposition.
- The description of the determinant of a Vandermonde matrix $\prod_{i<j}(x_j - x_i)$.
- Using interpolation and evaluation to retrieve the secret in Shamir's method.

Expansion

Obviously we can recover the explicit polynomial (i.e. the coefficient list) from these implicit representations. E.g., the implicit polynomial

$$p = (x_1 - x_2)\,(x_3 - x_4)\,(x_5 - x_6)$$

expands to

$$x_1\,x_3\,x_5 - x_2\,x_3\,x_5 - x_1\,x_4\,x_5 + x_2\,x_4\,x_5 - x_1\,x_3\,x_6 + x_2\,x_3\,x_6 + x_1\,x_4\,x_6 - x_2$$

in coefficient list form.

It is a healthy exercise to formalize this method, but the basic idea is perfectly natural:

- Use distributivity to flatten out the products.
- Then use cancellation to remove monomials with coefficient 0 and collect terms as usual.

The Problem

Expansion is straightforward in principle, but there is an efficiency problem:
It may take exponential time to perform the necessary operations.

Careful, though: Just because the obvious method takes exponential time does not necessarily mean there is a computational hardness issue—even though it is far from clear how one could speed things up.

As we will see, there is a close connection between 3-colorability, a well-known $\mathbb{NP}$-hard problem, and expanding polynomials.

Colorability

Recall that an undirected graph $G = \langle V, E \rangle$ is k-colorable iff one can assign k colors to the vertices so that no edge is monochromatic.

2-Colorability is special: it means the same as being bipartite and can be checked in polynomial time.

Theorem

It is $\mathbb{NP}$-complete to test if a graph is 3-colorable.

So $k = 3$ is a threshold, things are already as bad as they could be.

Complex Colors

A k-coloring is typically expressed as a map

$$\gamma : V \longrightarrow [k]$$

Of course, we could also use *red*, *green*, *blue* and so on.

In our case, a good choice of colors is a subset $C \subseteq \mathbb{C}$, $|C| = k$.

$$\gamma : V \longrightarrow C \subseteq \mathbb{C}$$

In particular let ω be a kth root of unity, say, $\omega = e^{2\mathbf{i}\pi/k}$.

We will use colors $\Omega_k = \{1, \omega, \omega^2, \ldots, \omega^{k-1}\}$.

Graph Polynomials

For simplicity assume $V = [n]$. Define the graph polynomial $P_G \in \mathbb{Z}[x_1, x_2, \ldots, x_n]$ by

$$P_G(\boldsymbol{x}) = \prod_{ij \in E} x_i - x_j$$

We assume $i < j$ in the product.

Clearly, P_G does not vanish since our graph has no loops.

There is a simple connection to colorability: if G is not k-colorable and $S \subseteq \mathbb{C}$ has cardinality k, then P_G vanishes on S:

$$\forall\, \boldsymbol{a} \in S^n \left(P_G(\boldsymbol{x}) = 0 \right)$$

Smashing Exponents

We will use the smash operator $\mathcal{R}_k$:

$$x_1^{e_1} x_2^{e_2} \dots x_n^{e_n} \quad \rightsquigarrow \quad x_1^{e_1 \bmod k} x_2^{e_2 \bmod k} \dots x_n^{e_n \bmod k}$$

After smashing, each variable has degree at most $k-1$.

Smashing all monomials will typically result in cancellations.

Also, in general, evaluating $p(\boldsymbol{a})$ produces different results from evaluating $\mathcal{R}_k(p)(\boldsymbol{a})$.

Colorability Polynomial

The next step is to associate $P_G(\boldsymbol{x})$ with a colorability polynomial $Q_{G,k}$ that encodes the k-colorability of graph G more directly.

To this end, multiply out the graph polynomial to get a sum of 2^m, $m = |E|$, monomials of the form

$$\pm z_1 z_2 \dots z_m \qquad\qquad z_i \in \{x_1, \dots, x_n\}$$

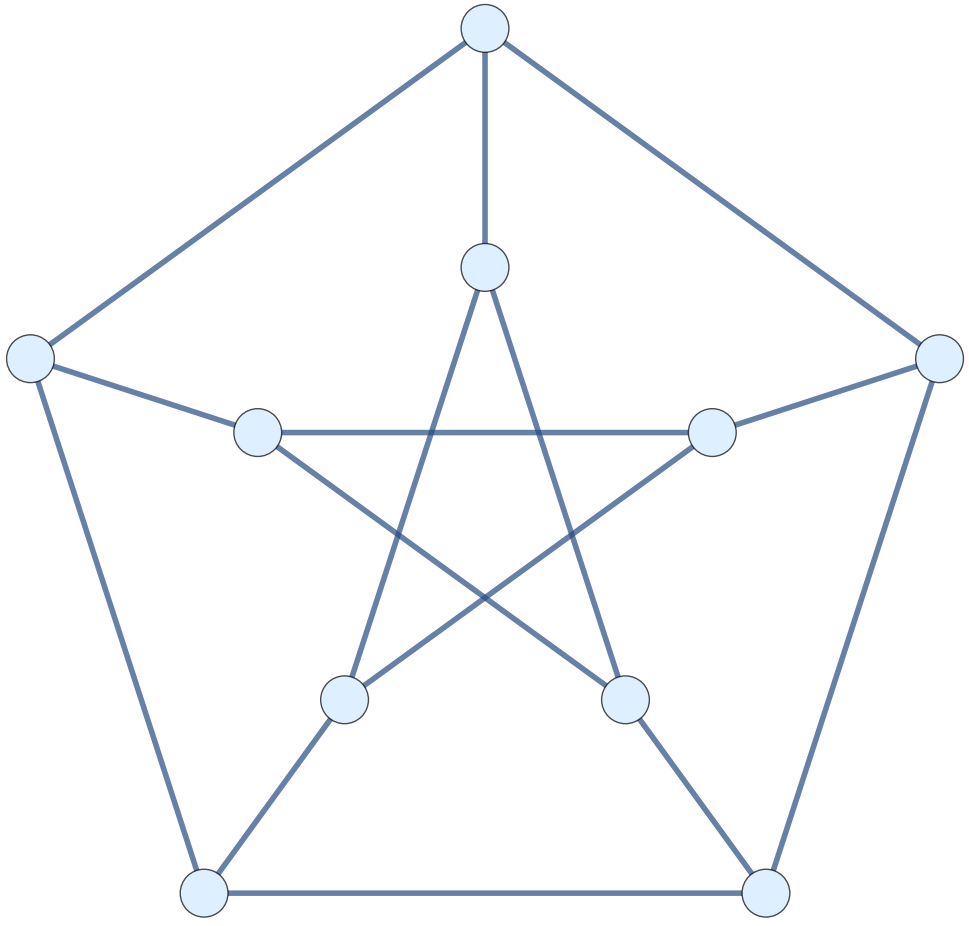
Collect the variables to get the standard form

$$\pm x_1^{e_1} x_2^{e_2} \dots x_n^{e_n} \qquad\qquad 0 \le e_i \le \deg(i)$$

Then smash the polynomial via $\mathcal{R}_k$.

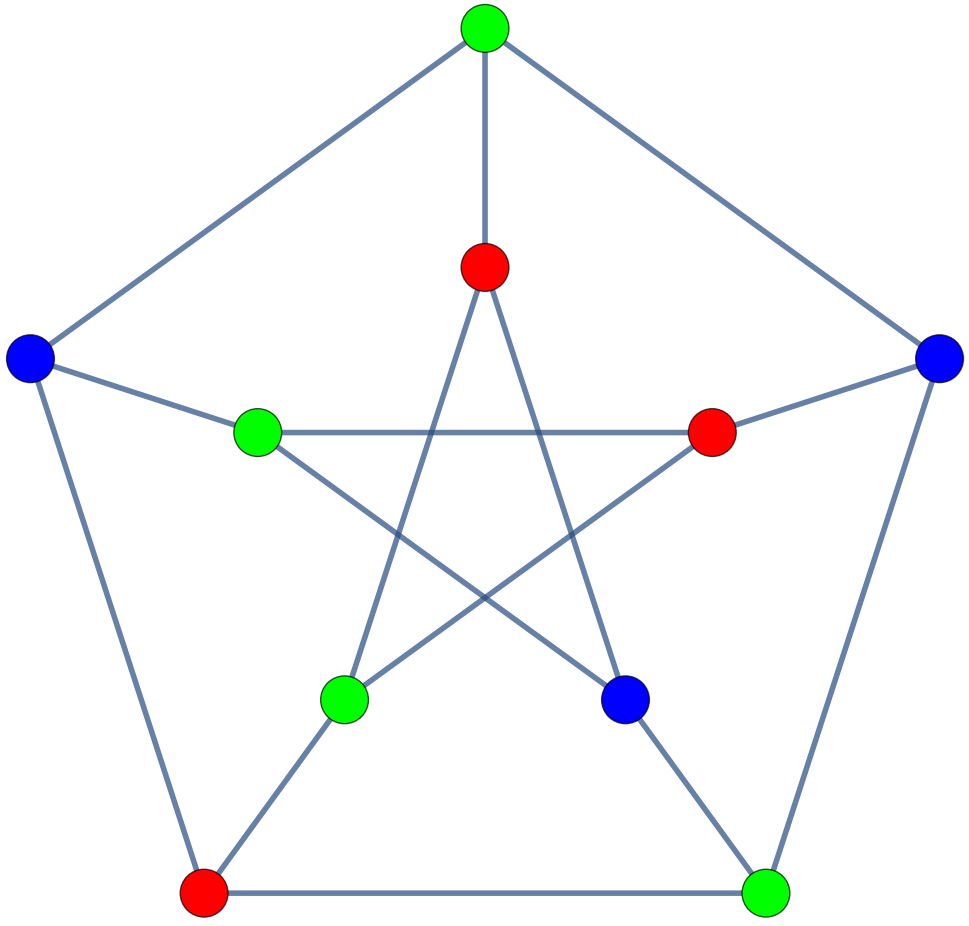
The result is the colorability polynomial $Q_{G,k}(\boldsymbol{x}) \in \mathbb{Z}[\boldsymbol{x}]$.
In fact, we have just computed the coefficient list of $Q_{G,k}$.

Example: Petersen Graph



A famous and surprisingly useful small graph: Petersen graph.

A 3-Coloring



Clearly, the graph is not 2-colorable.

Petersen Graph Polynomial

The graph polynomial here has 15 terms:

$$(x_1-x_3)(x_1-x_4)(x_2-x_4)(x_2-x_5)(x_3-x_5)(x_1-x_6)(x_2-x_7)(x_6-x_7)$$
$$(x_3-x_8)(x_7-x_8)(x_4-x_9)(x_8-x_9)(x_5-x_{10})(x_6-x_{10})(x_9-x_{10})$$

Not bad at all. Alas, the expanded polynomial with cancellation has 18,800 terms and it would take some 400 slides to display:

$$x_1^3x_2^3x_3^2x_4x_5x_6^2x_7x_8x_9 - x_1^2x_2^3x_3^3x_4x_5x_6^2x_7x_8x_9 - x_1^3x_2^2x_3^2x_4^2x_5x_6^2x_7x_8x_9 -$$
$$x_1^2x_2^3x_3^2x_4^2x_5x_6^2x_7x_8x_9 + \ldots \langle 18792 \rangle \ldots + x_1x_3x_4x_5^2x_6x_7^2x_8^2x_9^2x_{10}^3 +$$
$$x_2x_3x_4x_5^2x_6x_7^2x_8^2x_9^2x_{10}^3 + x_1x_4^2x_5^2x_6x_7^2x_8^2x_9^2x_{10}^3 - x_3x_4^2x_5^2x_6x_7^2x_8^2x_9^2x_{10}^3$$

Colorability

After smashing with $\mathcal{R}_2$, all terms cancel out and $Q_{G,2}$ vanishes.

But $\mathcal{R}_3$ produces a polynomial with 12,940 terms. Here is a small sample, somewhere in the middle:

$$\begin{aligned}\ldots - 4x_1^2x_2x_3x_4x_7^2x_8x_{10} - 2x_1x_2^2x_3x_4x_7^2x_8x_{10} + \ x_1^2x_3^2x_4x_7^2x_8x_{10} + \\ 3x_1x_2x_3^2x_4x_7^2x_8x_{10} + x_2^2x_3^2x_4x_7^2x_8x_{10} - \ x_1^2x_2x_4^2x_7^2x_8x_{10} + \\ 2x_1^2x_3x_4^2x_7^2x_8x_{10} + 3x_1x_2x_3x_4^2x_7^2x_8x_{10} - \ 2x_1x_3^2x_4^2x_7^2x_8x_{10} + \ldots\end{aligned}$$

$Q_{G,3}$ emphatically does not vanish.

The Theorem

Theorem (Alon, Tarsi 1992)

The graph G is k-colorable iff its colorability polynomial $Q_{G,k}$ does not vanish.

One direction is easy: If the graph is k-colorable, then for some coloring $\gamma : [n] \to \Omega_k$ we have $P_G(\gamma) \neq 0$. But then $Q_{G,k} = \mathcal{R}_k P_G$ cannot vanish since $\omega^e = \omega^{e \bmod k}$.

The opposite direction is hard.