Differential equations satisfied by generating functions of 5-, 6-, and 7-regular labelled graphs: a reduction-based approach

Frédéric Chyzak*[1] and Marni Mishna†[2]

[1]*Inria, France*
frederic.chyzak@inria.fr
[2]*Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada*
mmishna@sfu.ca

Submitted: May 6, 2025; Accepted: Sep 15, 2025; Published: Apr 20, 2026

Abstract. By a classic result of Gessel, the exponential generating functions for k-regular graphs are D-finite. Using Gröbner bases in Weyl algebras, we compute the linear differential equations satisfied by the generating function for 5-, 6-, and 7- regular graphs. The method is sufficiently robust to consider variants such as graphs with multiple edges, loops, and graphs whose degrees are limited to fixed sets of values.

Keywords. Regular graph, enumeration, Weyl algebra, reduction-based integration

Mathematics Subject Classifications. 05C30, 12H05

1. Introduction

1.1. A short history of k-regular graph enumeration

A graph is said to be *regular* if every vertex is incident to the same number of edges, that is, each vertex has the same degree. If that degree is k, we call the graph k-regular. One of the earliest graph enumeration problems considered was the number of non-isomorphic unlabelled k-regular graphs on n vertices. It is a relatively attainable problem for many reasons, including the fact that the number of edges is fixed in these graphs, which yields a significant simplification. For example, according to Gropp [Gro92], Jan de Vries determined the number of non-isomorphic cubic (3-regular) graphs up to 10 vertices, and shared them in a letter to Vittorio Martinetti, which

*Supported in part by the French ANR grant *De rerum natura* (ANR-19-CE40-0018) and by the French-Austrian ANR-FWF grant EAGLES (ANR-22-CE91-0007 & FWF-I-6130-N).

†Supported by Discovery Grant RGPIN-2017-04157 from the National Science and Engineering Research Council of Canada.

was eventually published in a journal in 1891. The proofs were descriptions of the graphs. Here we consider the slightly easier problem of labelled graphs, specifically the number of labelled k-regular graphs on n vertices, which we denote by $r_n^{(k)}$.

In the labelled case, the work of Read in the 1950s established enumeration formulas using the cycle index series, a relatively new machinery at the time. He gives a compact, structural equation in [Rea59a, Eq. 5.11] that is not immediately suitable for enumeration purposes for $k > 3$. He notes,

> *"It may readily be seen that to evaluate the above expressions in particular cases may involve an inordinate amount of computation."*

For $k = 3$, the equation is sufficiently manageable to give rise to a nice asymptotic formula.

One can distill from his work a formula in terms of coefficient extraction of a multivariable polynomial. This is the starting point of most modern approaches as it is easy to interpret, and there are numerous possibilities for analysis. Using the notation of square brackets to isolate the coefficient of the indicated term in a series expansion of the product we can write

$$r_n^{(k)} = [x_1^k x_2^k \dots x_n^k] \prod_{1 \leqslant i < j \leqslant n} (1 + x_i x_j). \tag{1.1}$$

The multiplication accounts for all possibilities of an edge $\{i, j\}$ to be in the graph or not. The coefficient of the indicated monomial is the number of graphs that have vertices 1 to n, such that each vertex is incident to exactly k other vertices: this is precisely $r_n^{(k)}$.

Regarding exact enumeration, Eq. (1.1) has long been used to exhibit values for $r_n^{(k)}$ for concrete integers n and k. In [McK83], McKay proposes a theoretical algorithm that proceeds by summing the evaluations of a polynomial in a way parametrized by all compositions of n. This induces a complexity that is more than exponential with respect to n, and can be proved to be in $O(2^{n^2/2})$. The method applies to several combinatorial classes, including k-regular graphs. Other works also obtain an exponential complexity, either by a variant formula for regular graphs or by applying to regular graphs an approach for more general graphs. Kaygun [Kay21] achieves a better complexity in $O(n^{kn}) = 2^{O(n \log n)}$ by a recurrence guided by the evolution of the degree sequence when a vertex is removed. By design, the approach is capable of more types of graphs than just regular graphs. Caizergues and Panafieu [CdP23] propose a variant formula to represent the number of k-regular graphs on n vertices as the Hadamard product of the nth power of some polynomial. Viewing this formula as an algorithm results in a calculation that has exponential complexity with respect to n. As a side-product of their work [EH24], Evnin and Horinouchi proceed by a term-by-term integration of the nth power of some polynomial, which, again, has a complexity that is exponential with respect to n.

On the other hand, for k-regular graphs the existence of a linear recurrence relations of finite order (see below) will ensure the possibility to compute $r_n^{(k)}$ as the nth term of a sequence in time complexity that is quasi-linear[1] in n, by methods of the late 1980s [CC89]. In this perspective, obtaining an ODE is therefore (a part of) the pre-processing before a quasi-linear time calculation for exact enumeration, and our paper is about improving the speed of this pre-processing.

[1]technically, $O(n \log^{3+\epsilon}(n))$ where the infinitesimal ϵ is there to hide powers of $\log(\log(n))$.

The formula of Eq. (1.1) can also be used in the service of asymptotic enumeration. There are a variety of strategies to consider, and other direct arguments, and the problem is well studied, remarkably even for problems with k a function of n (see, e.g., [MW91]). Wormald's 2018 ICM survey has many details on the state of asymptotic enumeration of regular graphs and related objects [Wor19]. The main goal of the more recent work [EH24] is to derive full asymptotic expansions for the number of regular graphs, which Evnin and Horinouchi realize by means of the saddle-point method. While this reference insists in working with convergent power series, Panafieu [Pan24] bases on the variant formulation of [CdP23] to develop a similar approach with the same goal of full asymptotic expansions by Laplace transforms of divergent series. However, asymptotic expansions is not our main interest in the present article and we will not dive deeper into the topic except for a comment to recall how to leverage known single-term asymptotic formulas and recurrences satisfied by the sequences to find additional asymptotic terms.

Regarding exact enumeration results, Wormald also notes in his 2018 survey that no recurrence (of finite fixed order) to enumerate k-regular graphs for k beyond 4 had appeared since the recurrences for 4-regular graphs published in the early 1980s. The entry point of the present article is also Eq. (1.1), but we follow a different lineage to contribute fixed-length linear recurrence formulas to count 5-, 6-, and 7-regular graphs, ending the drought.

The fact that there are recurrences to find at all is related to a question of Stanley [Sta80] in his foundational article on P-recursive sequences. The existence of a recurrence is equivalent to asking whether or not the exponential generating function for $r_n^{(k)}$, defined as $R^{(k)}(t) := \sum_{n \geqslant 0} r_n^{(k)} \frac{t^n}{n!}$, is *D-finite*. In other words, does $R^{(k)}(t)$ satisfy a linear differential equation with polynomial coefficients? Read had already given a recurrence for 3-regular graphs in his PhD thesis [Rea59b], and Read and Wormald used a combinatorial analysis to produce recurrences for 4-regular graphs [RW80]. Goulden, Jackson and Reilly [GJR83] were also able to determine explicit linear differential equations satisfied by $R^{(3)}(t)$ and $R^{(4)}$ using tools that dated back to MacMahon at the turn of the 20th century, called Hammond operators. But, they noted that[2]

> *"… the H-series theorem enables us to write down the system of partial differential equations for the H-series for arbitrary p without difficulty. However, the reduction of this system to a single ordinary differential equation in y_p is a technical task which we are unable to carry out for the general case."*

Their work fuelled speculation that $R^{(k)}$ should be D-finite for all k. Gessel compared their approach to his own method by the scalar product of symmetric functions and algebraic substitutions [Ges87]:

> *"… Hammond operators are undesirable for two reasons. First, they disguise the symmetry of the scalar product. Second, they can be represented as differential operators. Although this might seem like an advantage, it seems to be of little use, but misleads by directing attention in the wrong direction."*

[2] In our notation, $p = k$ and $y_p = R^{(k)}$.

Instead of working with differential equations, he recast the extraction in terms of symmetric functions, and used algebraic arguments to establish that indeed $R^{(k)}(t)$ is D-finite for all k. His framework is sufficiently simple and robust that it can be used to establish the D-finiteness of many related regular graph and hypergraph cases. Gessel was able to advance on the general case thanks to concurrent work on multivariable P-recursiveness of Lipshitz [Lip89]. The work of Lipshitz was not sufficiently straightforward to convert into an algorithm or even make computation effective beyond $k = 2$. It was over a decade before the computer algebra implementations using differential operators caught up to his theoretical results. In 2005 Chyzak, Mishna and Salvy [CMS05] made both the Hammond method and the Gessel strategy effective for any k using Gröbner bases for D-modules and non-commutative polynomial elimination, in a sort of variant of Creative Telescoping, a method for symbolic integration. The implementation quickly found differential equations up to, and including, 4-regular objects. The growth of data in the skew polynomial elimination involved in the 5-regular graph case requires computational resources that even today are insufficient to have the algorithm terminate. However, in the intervening 20 years, there have been remarkable improvements and insights to Creative Telescoping. This lead us to an evolved algorithm that terminates also in practice, and indeed we could find the linear differential equations satisfied by $R^{(5)}(t)$, $R^{(6)}(t)$, and $R^{(7)}(t)$. Our present approach can be applied to find the differential equations satisfied by the other graph, hypergraph and graph-like classes for higher degrees of regularity than were previously obtained [Mis07, Mis18].

The following theorem is the main result of this article. It appears below, rephrased, as Corollary 4.10(1).

Theorem 1.1. *For each graph model in Table A.1, there exists a known linear differential equation with polynomial coefficients satisfied by the exponential generating function, with explicit order given by column ∂_t of the table, and maximum coefficient degree given by column t of the table.*

The graph models in the table include those with

1. *only simple edges permitted (denoted 'se' in the table) or multiple edges allowed (denoted 'me');*
2. *loops forbidden ('ll'), loops allowed and contributing 2 to vertex degrees ('la'), or loops allowed and contributing 1 to vertex degrees ('lh');*
3. *degrees restricted to some finite set, including: $\{k\}$ for $2 \leqslant k \leqslant 7$, $\{1, \dots, k\}$ for $2 \leqslant k \leqslant 6$, and $\{k, \ell\}$ for $2 \leqslant k < \ell \leqslant 6$, among others.*

In particular, the order of the linear differential equation for simple, loopless k-regular graphs (coded 'se' and 'll') is summarized in the table:

k	2	3	4	5	6	7
order	1	2	2	6	6	20

1.2. The scalar product[3] of symmetric functions

The coefficient extraction in Eq. (1.1) can be placed into an infinite product, symmetric in all variables, which can be readily encoded in terms of symmetric functions. The setup of Gessel [Ges90] uses the scalar product in the ring of symmetric functions to model the extraction. Describing the method requires a small detour through symmetric function terminology and basics. There are many excellent introductions. We highlight some notation, but refer readers to Stanley [Sta99, Chapter 7] for details.

We say $\lambda = (\lambda_1, \lambda_2 \dots, \lambda_q)$ such that $\sum_{i=1}^{q} \lambda_i = n$ and $\lambda_i \geqslant \lambda_{i+1}$ is a partition of n into q parts, and write $\lambda \vdash n$ to indicate that λ is a partition of n. The monomial symmetric function is defined as $m_\lambda := \sum_{\alpha\sim\lambda} x^\alpha$ where x^α denotes the product $x_1^{\alpha_1} x_2^{\alpha_2} \cdots$ and where $\alpha \sim \lambda$ if the (finitely many) non-zero entries of α are a rearrangement of the parts of λ. (Note that only finitely many factors of the infinite product x^α are different from 1.) Using m_λ we can describe the complete homogeneous symmetric function $h_n := \sum_{\lambda\vdash n} m_\lambda$ and the power-sum symmetric function $p_n := m_{(n)} = x_1^n + x_2^n + \dots$. Products are denoted respectively $h_{n_1\, n_2 \dots\, n_\ell} := h_{n_1} h_{n_2} \dots h_{n_\ell}$ and $p_{n_1\, n_2 \dots\, n_\ell} := p_{n_1} p_{n_2} \dots p_{n_\ell}$. The vector space of symmetric functions of order n has several important bases, including $\{m_\lambda \mid \lambda \vdash n\}$, $\{h_\lambda \mid \lambda \vdash n\}$, and $\{p_\lambda \mid \lambda \vdash n\}$. For any $\lambda \vdash n$, z_λ denotes the number

$$z_\lambda := 1^{r_1} r_1!\, 2^{r_2} r_2! \, \dots n^{r_n} r_n! \tag{1.2}$$

provided λ has r_1 ones, r_2 twos, etc. We set $\delta_{\lambda=\nu}$ to 1 if $\lambda = \nu$ is true and to 0 otherwise. The *scalar product of symmetric functions* is classically defined by

$$\langle p_\lambda, p_\nu \rangle := \delta_{\lambda=\nu} z_\lambda, \quad \text{from which we deduce} \quad \langle m_\lambda, h_\nu \rangle = \delta_{\lambda=\nu}. \tag{1.3}$$

The connection to the graph enumeration problem is as follows. We can extract the coefficient of a particular monomial in a symmetric function with a judiciously chosen scalar product. Write $\bar{F} := \prod_{i<j}(1 + x_i x_j)$ and consider an example. This product is fundamental in the study of symmetric functions, particularly its expression in the various bases. Now, since $r_4^{(3)} = [x_1^3 x_2^3 x_3^3 x_4^3]\bar{F}$, to actually compute this write $\bar{F}$ as a sum of monomial symmetric functions, and determine the coefficient of $m_{3,3,3,3}$ (which is the only basis element to contain the term $x_1^3 x_2^3 x_3^3 x_4^3$). As the monomial and complete homogenous bases are orthogonal under the usual scalar product of symmetric functions, this coefficient is precisely $\langle \bar{F}, h_{3,3,3,3} \rangle = \langle \bar{F}, h_3^4 \rangle$.

From the formulas $\log(1+u) = \sum_{k\geqslant 1} (-1)^{k+1} u^k / k$ and $2\sum_{i<j} x_i^k x_j^k = \sum_{i,j} x_i^k x_j^k - \sum_i x_i^{2k}$ it follows

$$\bar{F} = \exp\left(\sum_{i<j} \log(1 + x_i x_j)\right) = \exp\left(\sum_{i<j}\sum_{k\geqslant 1} (-1)^{k+1} \frac{x_i^k x_j^k}{k}\right) = \exp\left(\sum_{k\geqslant 1} (-1)^{k+1} \frac{p_k^2 - p_{2k}}{2k}\right). \tag{1.4}$$

Henceforth we will only work with the power-sum basis, specifically, we work in a ring generated by t and a finite number of the the p_i variables. To continue the example, to determine $R^{(3)}(t)$

[3] We follow the usual terminology of a "scalar product" in combinatorics, although the presence of a formal indeterminate t would require to speak more properly of a "pairing".

we first write $h_3 = \frac{p_3}{3} + \frac{p_2 p_1}{2} + \frac{p_1^3}{6}$, and thus obtain the following expression for the generating function:

$$R^{(3)}(t) = \left\langle \bar{F}, \sum_{n\geqslant 0} h_3^n \frac{t^n}{n!} \right\rangle = \left\langle \exp\left(\sum_{k\geqslant 1} (-1)^{k+1} \frac{p_k^2 - p_{2k}}{2k} \right), \exp\left(\left(\frac{p_3}{3} + \frac{p_2 p_1}{2} + \frac{p_1^3}{6} \right) t \right) \right\rangle. \tag{1.5}$$

Since the second argument has only p_1, p_2, p_3, all terms with other p_i contribute 0:

$$R^{(3)}(t) = \left\langle \exp\left(\frac{p_1^2}{2} - \frac{p_2}{2} - \frac{p_2^2}{4} + \frac{p_3^2}{6} \right), \exp\left(\left(\frac{p_3}{3} + \frac{p_2 p_1}{2} + \frac{p_1^3}{6} \right) t \right) \right\rangle. \tag{1.6}$$

For future reference, we note the following formula, which leads to generalizations of Eqs. (1.5) and (1.6):

$$R^{(k)}(t) = \left\langle \bar{F}, \sum_{n\geqslant 0} h_k^n \frac{t^n}{n!} \right\rangle = \langle \bar{F}, \exp(h_k t) \rangle. \tag{1.7}$$

This formula will be proven and extended in Lemma 3.1.

1.3. Earlier computational approaches

As we mentioned above, Gessel [Ges90] proved the existence of linear differential equations for scalar products like Eq. (1.7), and earlier work [CMS05] proposed algorithms to compute them. In there, a non-commutative ring W_p of so-called skew polynomials in the variables $p_1, \dots, p_k$ and in the corresponding derivatives $\partial_1, \dots, \partial_k$ is introduced to represent linear differential operators. Its elements act on formal power series: for each i, p_i acts by multiplication by p_i, and ∂_i by the usual derivation d/dp_i. (See Eq. (4.1) in Section 4 for the formal definition of W_p.) For a given series S in the variables $p_1, \dots, p_k$ we also consider the set, denoted $\operatorname{ann}(S)$, of all skew polynomials of W_p representing linear differential operators that annihilate S. This set $\operatorname{ann}(S)$ is closed under multiplication by any element of W_p on the left and is thus a left ideal of W_p. As is customary in effective literature, a left ideal is best represented by a non-commutative analogue of a Gröbner basis, that is, by a finite set of skew polynomials that can algorithmically divide a given ideal element, resulting into a uniquely defined remainder that is zero if and only if the given polynomial is in the ideal. We refer to [SST00, Chapter 1], [Kau23, Section 4.6], [Bah20] for textbooks and tutorial presentations.

Given a number k, we henceforth write $p = (p_1, \dots, p_k)$ and $\partial = (\partial_1, \dots, \partial_k)$. Given a series F in p and a series G in (t, p), the differential equations with respect to t satisfied by the scalar product $\langle F, G \rangle$ are to be found as those elements free of (p, ∂) in the (vector space) sum of the left ideal $\operatorname{ann}(G)$ and of the right ideal $\operatorname{ann}(F)^\dagger$ obtained by taking the adjoints of all elements in $\operatorname{ann}(F)$ [CMS05] (see the definitions in Section 4). A first algorithm in [CMS05], based on linear algebra, consists: *(i)* in fixing an integer d; *(ii)* in determining representatives of $\operatorname{ann}(F)^\dagger$ and $\operatorname{ann}(G)$ for each possible leading monomial of total degree at most d with respect to $(p, \partial, \partial_t)$; *(iii)* and in using a non-commutative variant of Gaussian elimination over $\mathbb{Q}(t)$ to eliminate (p, ∂), repeating the whole process with a larger d if elimination results in no non-trivial output. Because there are $\binom{d}{2k+1} = O(d^{2k+1})$ monomials of degree at most d, and almost as many representatives to determine for each ideal, this process is very inefficient in practice.

A second algorithm in [CMS05] is tailored to a certain form for the argument G in the scalar product: if $G = \exp(h_k t)$, the theory of Hammond series, as developed in [GJR83], provides the formula

$$\langle F, \exp(h_k t_k) \rangle = \mathcal{H}(F)(0, \dots, 0, t_k),$$

where $\mathcal{H}(F)(t_1, \dots, t_k)$ is a transform of F known as its Hammond series. A simple replacement of the p_i and the ∂_i in $\operatorname{ann}(F)$ with suitable polynomials in $t_1, \dots, t_k$ and corresponding derivatives ∂_{t_i} provides $\operatorname{ann}\big(\mathcal{H}(F)\big)$. The specialization of $t_1, \dots, t_{k-1}$ to 0 is then obtained by restriction, an operation dual to integration. One way to implement it would have been to first eliminate the $k-1$ variables $\partial_{t_1}, \dots, \partial_{t_{k-1}}$, e.g., by a Gröbner basis calculation, before setting all of the $k-1$ variables $t_1, \dots, t_{k-1}$ to zero and taking a generator of the resulting principal ideal in $\mathbb{Q}(t_k)\langle \partial_{t_k} \rangle$. But a simultaneous elimination in this way leads to high degrees and is also inefficient in practice. More generally, in the 2000s, no good algorithm was known for integration with respect to several variables considered simultaneously, so one had to resort to iterated integrations, one variable after the other. Correspondingly, for multiple restriction one had to perform specializations one variable after the other, and this is what is proposed in [CMS05], in a way that is reminiscent of elimination by successive resultants. This approach, too, fails for $k = 5$: all steps are fast until the last elimination, which should eliminate ∂_{t_1} from two degree-9 polynomials in the four variables $t_1, t_5, \partial_{t_1}, \partial_{t_5}$, and this fails in practice.

In both old approaches, the culprit is elimination in too many variables: eliminating $2k$ variables between polynomials in $2k+1$ variables over $\mathbb{Q}(t)$ in the first approach; eliminating $k-1$ variables between polynomials in $2k-1$ variables over $\mathbb{Q}(t_k)$ in the second approach. The second is an improvement in that it reduces the number of variables, and this is assisted by specializations to zero along the process.

A turning point in the theory of Creative Telescoping was the introduction of reduction-based algorithms, starting with the integration of bivariate rational functions [BCCL10] in 2010, and followed by many articles in the literature. Inspiration for the present work came from a more recent reduction-based algorithm [BCLS18] for the integration with regard to one variable p of general D-finite functions $f(t, p)$, leading to integrals parametrized by t. In a nutshell, reduction-based algorithms: *(i)* set up a reduction process that corresponds to simplifying a function to be integrated modulo derivatives with respect to p of other functions, in such a way that the resulting remainder lies in a finite-dimensional vector space; *(ii)* find a linear relation between the remainders of successive higher-order derivatives with respect to the parameter t of the function to be integrated. In situations where integrals of derivatives are zero, the output linear relation reflects a differential equation in t of the parametrized integral. Although the symmetric scalar product cannot be represented as an integral of a D-finite function, the method of [BCLS18] can be adapted to the present situation, in a way that the reduction with respect to the k variables $p_1, \dots, p_k$ is possible simultaneously and that most of the calculations involve polynomials in $k+1$ variables over $\mathbb{Q}(t)$.

1.4. Contributions

Besides presenting in Algorithm 4.1 a method that adapts reduction-based algorithms to a simultaneous reduction with respect to several integration variables, our main contribution in the present work is to obtain differential equations satisfied by various models of graphs with vertex degrees restricted to be in a fixed subset of $\{1, \ldots, 7\}$ (see Theorem 1.1 and Corollary 4.10). We cannot guarantee the termination of our method, but any differential equation it outputs is correct, as proven by Theorem 4.9. In Table A.1, we list for a few dozens of models the order of a differential equation satisfied by the counting generating function and the order of a recurrence equation satisfied by its sequence of coefficients, together with corresponding degrees of their coefficients. All those equations are proven correct by the computer calculations (see Corollary 4.10). To the best of our knowledge, this is the first time differential equations are presented for $R^{(5)}(t)$, $R^{(6)}(t)$ and $R^{(7)}(t)$, or more generally graphs where degrees 5, 6, or 7 are considered.

The recurrences we find are linear, with polynomial coefficients and hence can be unravelled quickly to get data for graphs of high order. For example, it takes less than 14 minutes[4] to determine the number of 7-regular graphs on 2000 vertices from the ODE of order 20 that we found:

$$r^{(7)}_{2000} = 80680697 \ldots 04296875 \approx 8.068069734 \times 10^{18572}.$$

This is after the preprocessing step of computing the ODE, which was almost 9 hours (see Table A.1), so to better appreciate the computational complexity of the calculation, let us add that it takes not more than four times[5] as long (less than 49 minutes) to obtain

$$r^{(7)}_{3000} = 71432976 \ldots 18359375 \approx 7.143297604 \times 10^{29711}.$$

More generally, we are able to significantly increase the number of known terms compared to the state of the art in the On-line Encyclopedia of Integer Sequences (OEIS) [OEI] for sequences A338978 and A339847, and we have contributed a new sequence, A374842 counting 7-regular graphs.

Finally, in Section 7 we recall how to use the recurrences to easily find additional terms in the asymptotic expansions in cases where the first term expansion was known. Expansions are given in Table B.1 in Appendix B.

The generated enumerative data, recurrences, differential equations and Maple code implementing our strategy are all available at `https://files.inria.fr/chyzak/kregs/`.

---

[4] We used `gfun[rectoproc]` as shipped with Maple 2024 and the same computer as for the calculations of Table A.1. The calculation is not optimized: we unrolled the recurrence, essentially computing all numbers up to index 2000, before multiplying the obtained rational number by 2000!. Computing with `gfun[nth_term]`, the command dedicated to computing a single term in more recent versions of `gfun`, is unfortunately not faster for this calculation, presumably because the recurrence order is 1684, not much below 2000.

[5] Again, our implementation is not optimized to be quasi-linear.

2. Worked example: 4-regular graphs

Before introducing our procedure in a systematic way in Section 4, we illustrate it with the class of 4-regular graphs, allowing single edges and no loops. (The case $k = 3$ is too simple to demonstrate important points of our method.) Specializing Eq. (1.7) to $k = 4$, we consider the scalar product $\langle F, G\rangle$, which represents $R^{(4)}(t)$ when the exponential functions $F = \exp(f)$ and $G = \exp(tg)$ are given by

$$f := \frac{p_1^2}{2} - \frac{p_2^2}{4} + \frac{p_3^2}{6} - \frac{p_4^2}{8} - \frac{p_2}{2} + \frac{p_4}{4}, \qquad g := \frac{p_1^4}{24} + \frac{p_1^2 p_2}{4} + \frac{p_2^2}{8} + \frac{p_1 p_3}{3} + \frac{p_4}{4}.$$

2.1. A reduction procedure

We begin by explaining a procedure to normalize expressions of the form $\langle F, sG\rangle$ for a polynomial $s \in \mathbb{Q}(t)[p]$: without changing the value of the scalar product, the polynomial s will be replaced with an element in $\mathbb{Q}(t) + \mathbb{Q}(t)p_1 + \mathbb{Q}(t)p_2$.

From the definition of F, we get that annihilating operators for F are

$$P_1 := \partial_1 - p_1, \qquad P_2 := 2\partial_2 + p_2 + 1, \qquad P_3 := 3\partial_3 - p_3, \qquad P_4 := 4\partial_4 + p_4 - 1. \quad (2.1)$$

In Section 4, we will define two transformations on differential operators, namely adjoints (†) and twists ($^\sharp$). Applying them to Eq. (2.1), we obtain

$$P_1^\dagger := p_1 - \partial_1, \qquad P_2^\dagger := p_2 + 2\partial_2 + 1, \qquad P_3^\dagger := p_3 - 3\partial_3, \qquad P_4^\dagger := p_4 + 4\partial_4 - 1,$$

and

$$P_1^\sharp := p_1 - \partial_1 - \frac{t}{6}(p_1^3 + 3p_1p_2 + 2p_3), \qquad P_2^\sharp := p_2 + 2\partial_2 + \frac{t}{2}(p_1^2 + p_2) + 1,$$
$$P_3^\sharp := p_3 - 3\partial_3 - tp_1, \qquad P_4^\sharp := p_4 + 4\partial_4 + t - 1.$$

We will prove in Section 4 that $\langle F, (P_j^\sharp \cdot \bar{s})\, G\rangle$ is zero for any $\bar{s} \in \mathbb{Q}(t)[p]$ and any j, motivating that we will try to adjust s by a linear combination of polynomials of the form $P_j^\sharp \cdot \bar{s}$.

In order to determine how to do so more precisely, observe first that for any monomial p^α,

$$P_1^\sharp \cdot p^\alpha = -\frac{t}{6} p_1^{\alpha_1+3} p_2^{\alpha_2} p_3^{\alpha_3} p_4^{\alpha_4} + \cdots, \qquad P_2^\sharp \cdot p^\alpha = \frac{t}{2} p_1^{\alpha_1+2} p_2^{\alpha_2} p_3^{\alpha_3} p_4^{\alpha_4} + \cdots,$$
$$P_3^\sharp \cdot p^\alpha = -tp_1^{\alpha_1+1} p_2^{\alpha_2} p_3^{\alpha_3} p_4^{\alpha_4} + p_1^{\alpha_1} p_2^{\alpha_2} p_3^{\alpha_3+1} p_4^{\alpha_4} + \cdots, \qquad P_4^\sharp \cdot p^\alpha = p_1^{\alpha_1} p_2^{\alpha_2} p_3^{\alpha_3} p_4^{\alpha_4+1} + \cdots,$$

where in each case, the dots represent a polynomial with lower total degree. We will base our calculation on these forms. Consider for example any monomial ordering for which p_4 is lexicographically higher than all other variables. Given a polynomial $s \in \mathbb{Q}(t)[p]$ with leading term cp^β for $\beta_4 \geqslant 1$, the choice $\alpha = \beta - (0,0,0,1)$ ensures that $s - P_4^\sharp \cdot (cp^\alpha)$ has a leading monomial less than p^β. As a consequence, s can be reduced by a series of like transformations to a polynomial $s - P_4^\sharp \cdot \bar{s}$ that does not involve p_4: here $\bar{s}$ is a polynomial that adds up all the cp^α observed during the reduction process. In other words, one can eliminate p_4 from s. One can similarly use $P_3^\sharp$ to reduce the degree with respect to p_3: this essentially introduces p_1 as a replacement of p_3, but one can eliminate p_3 as well. By continuing with transformations based on $P_2^\sharp$, which do not reintroduce either p_3 or p_4, one could hope to eliminate p_1 as well (after p_3 and p_4) from s. It turns out that one cannot fully eliminate p_1, but that degrees with respect to p_1 can be reduced down to at most 1. On the other hand, it is not immediately evident that degrees with respect to p_2 can be kept under control.

To explain how controlling p_2 can be done, we continue our informal presentation by recombining the $P_i^\sharp$ in the following way into elements of the right ideal they generate:

$$\begin{aligned}
P_1^\sharp + P_3^\sharp \frac{t}{3} &= -\frac{t}{6}p_1^3 - \frac{t}{2}p_1p_2 + \left(1 - \frac{t^2}{3}\right)p_1 - \partial_1 - t\partial_3,\\
P_2^\sharp &= \frac{t}{2}p_1^2 + \left(1 + \frac{t}{2}\right)p_2 + 1 + 2\partial_2,\\
\tilde P_5 := P_1^\sharp + P_3^\sharp \frac{t}{3} + P_2^\sharp \frac{p_1}{3} &= \frac{1-t}{3}p_1p_2 + \frac{4-t^2}{3}p_1 + \frac{2}{3}p_1\partial_2 - \partial_1 - t\partial_3,\\
\tilde P_6 := \tilde P_5 \frac{t}{2}p_1 + P_2^\sharp \frac{t-1}{3}p_1 &= \frac{(4-t^2)t}{6}p_1^2 + \frac{t^2+t-2}{6}p_2^2 + \frac{t-1}{3}p_2 + \frac{t}{3}p_1^2\partial_2\\
&\quad + \frac{t-4}{6} - \frac{t}{2}p_1\partial_1 + \frac{2(t-1)}{3}p_2\partial_2 - \frac{t^2}{2}p_1\partial_3,\\
\tilde P_7 := \tilde P_6 + P_2^\sharp \frac{t^2-4}{3} &= \frac{t^2+t-2}{6}p_2^2 + \frac{t^3+2t^2-2t-10}{6}p_2 + \frac{t}{3}p_1^2\partial_2\\
&\quad + \frac{2t^2+t-4}{6} - \frac{t}{2}p_1\partial_1 + \frac{2(t-1)}{3}p_2\partial_2 - \frac{t^2}{2}p_1\partial_3 + \frac{2(t^2-4)}{3}\partial_2.
\end{aligned}$$

Observe how at each line, one can determine precisely the action of the operator on a monomial $p_1^{\alpha_1}p_2^{\alpha_2}$ and thus predict the leading monomial of the result for the monomial ordering refining total degree by $p_1 > p_2$:

$$\begin{aligned}
\tilde P_5 \cdot p_1^{\alpha_1}p_2^{\alpha_2} &= \frac{1-t}{3}p_1^{\alpha_1+1}p_2^{\alpha_2+1} + \cdots,\\
\tilde P_6 \cdot p_1^{\alpha_1}p_2^{\alpha_2} &= \frac{(4-t^2)t}{6}p_1^{\alpha_1+2}p_2^{\alpha_2} + \cdots,\\
\tilde P_7 \cdot p_1^{\alpha_1}p_2^{\alpha_2} &= \frac{t^2+t-2}{6}p_1^{\alpha_1}p_2^{\alpha_2+2} + \cdots.
\end{aligned}$$

Considering in particular $\tilde P_7$, one obtains that degrees with respect to p_2 can be reduced down to at most 1. Note that the $\tilde P_7 \cdot p_1^{\alpha_1}p_2^{\alpha_2}$ luckily do not reintroduce the variables p_3 and p_4. So at this point, any polynomial $s \in \mathbb{Q}(t)[p]$ in an expression $\langle F, sG\rangle$ can be replaced with a linear combination of 1, p_1, p_2, and p_1p_2 over $\mathbb{Q}(t)$, that is, with some polynomial confined to a 4-dimensional vector space. Finally, because $\tilde P_5 \cdot 1 = \frac{1-t}{3}p_1p_2 + \frac{4-t^2}{3}p_1$, the monomial p_1p_2 can be replaced with p_1 in such linear combinations, bringing the finite dimension down to 3. In the end, for any $s \in \mathbb{Q}(t)[p]$, a sequence of transformations results: first in an element $\check s \in \mathbb{Q}(t) + \mathbb{Q}(t)p_1 + \mathbb{Q}(t)p_2$ and elements $\tilde s_j \in \mathbb{Q}(t)[p]$ for $j = 0, \dots, 4$ such that $\langle F, sG\rangle = \langle F, \check{s}G\rangle$ and

$$s - \check{s} = \sum_{i=0}^{4} G_i \cdot \tilde{s}_i \quad \text{for} \quad (G_0, \dots, G_4) = (P_4^\sharp, P_3^\sharp, P_2^\sharp, \tilde{P}_7, \tilde{P}_5);$$

next, because $\tilde{P}_5$ and $\tilde{P}_7$ are in the right ideal, in elements $\bar{s}_j \in \mathbb{Q}(t)[p]$ for $j = 1, \dots, 4$ such that $s - \check{s} = P_1^\sharp \cdot \bar{s}_1 + P_2^\sharp \cdot \bar{s}_2 + P_3^\sharp \cdot \bar{s}_3 + P_4^\sharp \cdot \bar{s}_4$.

Eliminating variables one after the other in this presentation was chosen for the sake of the informal explanation. In the next section and in our implementation, we use an optimized elimination strategy that bases more strongly on total degree.

2.2. Recombining normal forms for a differential equation

We now explain how the reduction step of the previous section can be used to derive a differential equation with respect to t for $\langle F, G\rangle$.

For any $i \in \mathbb{N}$, the identity $\partial_t^i \cdot \langle F, G\rangle = \langle F, g^i G\rangle$ follows from the definition $G = \exp(tg)$. By the reduction of previous section, the polynomial g^i can be replaced with some element $\check{g}_i$ from the 3-dimensional vector space $\mathbb{Q}(t) + \mathbb{Q}(t)p_1 + \mathbb{Q}(t)p_2$. So, the family $\{\check{g}_0, \check{g}_1, \check{g}_2, \check{g}_3\}$ is obviously linearly dependent over $\mathbb{Q}(t)$, and a linear relation $q_0\check{g}_0 + \dots + q_3\check{g}_3 = 0$ with $q_i \in \mathbb{Q}(t)$ provides a linear differential relation $(q_0 + q_1\partial_t + q_2\partial_t^2 + q_3\partial_t^3) \cdot \langle F, G\rangle = 0$.

Performing these calculations on our worked example, we start with $g^0 = 1$, so that $\check{g}_0 = 1$ as 1 is already reduced. Next, reducing g yields $g = \check{g}_1 + \sum_{i=0}^4 G_i \cdot \tilde{s}_i$ with

$$\check{g}_1 = -\frac{(t^5 + 2t^4 + 2t^2 + 8t - 4)}{4(t^2 + t - 2)t^2}(p_2 + 1)$$

$$\text{and} \quad (\tilde{s}_0, \dots, \tilde{s}_4) = \left(\frac{1}{4}, \frac{p_1}{3}, \frac{p_1^2}{12t} + \frac{(5t - 2)p_2}{12t^2} + \frac{4t^2 - 1}{6t^2}, -\frac{t^2 + 4t - 2}{2t^2(t^2 + t - 2)}, 0\right).$$

At this point, a more heavy calculation yields $g^2 = \check{g}_2 + \sum_{i=0}^4 G_i \cdot \tilde{\tilde{s}}_i$ with

$$\check{g}_2 = -\frac{t^{12} - 14t^{10} - 20t^9 - 36t^8 - 200t^7 - 356t^6 - 48t^5 + 200t^4 - 336t^3 - 240t^2 + 416t - 96}{16(t^2 + t - 2)^2(t - 1)t^4(t + 2)} - \frac{(t^{13} + 4t^{12} - 16t^{10} - 10t^9 - 36t^8 - 220t^7 - 348t^6 - 48t^5 + 200t^4 - 336t^3 - 240t^2 + 416t - 96)}{16(t^2 + t - 2)^2(t - 1)t^4(t + 2)}p_2$$

and quotients $\tilde{\tilde{s}}_i$ that we refrain from displaying. After finding a linear dependency between the $\check{g}_i$ over $\mathbb{Q}(t)$, we obtain the annihilating operator

$$\begin{aligned} &16t^2(t + 2)^2(t - 1)^2(t^5 + 2t^4 + 2t^2 + 8t - 4)\partial_t^2 \\ &\quad + (-4t^{13} - 16t^{12} + 64t^{10} + 40t^9 + 144t^8 + 880t^7 + 1392t^6 \\ &\qquad + 192t^5 - 800t^4 + 1344t^3 + 960t^2 - 1664t + 384)\partial_t \\ &\qquad\qquad - t^4(t^5 + 2t^4 + 2t^2 + 8t - 4)^2. \end{aligned}$$

Getting an order 2 less than the dimension 3 could not be predicted.

For efficiency, the remainders $\check{g}_i$ can be obtained in a more incremental way: the formula

$$\partial_t^{i+1} \cdot \langle F, G\rangle = \partial_t \cdot \langle F, \check{g}_i G\rangle = \langle F, \partial_t \cdot (\check{g}_i G)\rangle = \langle F, (\check{g}_i \times g + \partial_t \cdot \check{g}_i) G\rangle$$

suggests one can obtain $\check{g}_{i+1}$ by reducing $\check{g}_i \times g + \partial_t \cdot \check{g}_i$, which is much smaller than g^{i+1}. This makes calculations generally faster, although in the present example $\check{g}_1 \times g + \partial_t \cdot \check{g}_1$ is messier than g^2.

3. Applicability to various models of graphs

As we remarked in the introduction, there are many enumeration problems that can be expressed using the scalar product, and have the potential to be solved with our strategy. The computational limits are directly related to the maximal index i of all p_i that appear in the expressions, and this leaves substantial flexibility. Although in the work above (namely Section 1.2 and Section 2) we have focused on the case of simple, loopless graphs, with only minor modifications of $\bar{F}$ in Eq. (1.7) we can consider graphs with multiple edges, or loops, or both, as we will prove in Lemma 3.1. The form is still an exponential of a polynomial in the p_i. Similarly, it is straightforward to consider graph classes where the possible vertex degrees come from a finite set K. To this end, it suffices to replace $\exp(th_k)$ with $\exp(t(\sum_{j\in K} h_j))$ (as per the lemma again) and to express the h_j in the power-sum basis. For the lemma and future discussions, we label generalized regular graph models according to three parameters:

- e encodes the model of allowed edges: 'se' is used for graphs with single edges; 'me' is used for generalized structures with multiple edges allowed (usually called "multigraphs").
- l encodes how loops are allowed and counted:
  - 'll' is used for loopless structures, like "graphs" in the usual terminology;
  - 'la' is used for structures with loops allowed and contributing 2 each to the degree of a vertex, in other words, those models enumerate structures according to the number of adjacent half-edges;
  - 'lh' is used for structures with loops allowed and contributing 1 each to the degree of a vertex, in other words, those models enumerate structures according to the number of adjacent edges.
- K denotes the set of allowed degrees of vertices, whether it be counting adjacent edges with 'lh' models or counting adjacent half-edges with 'la' models; usual k-regular graphs are obtained by setting K to the singleton $\{k\}$; models with K of larger cardinality allow different vertices of a graph to have different degrees as long as they are in K; for example, $K = \{1, 2, \ldots, k\}$ can be used to describe a class of graphs with vertex degree bounded by k; unless otherwise clear by the context, we make $k = \max K$.

3.1. Theoretical flexibility

Lemma 3.1. *The exponential generating function of a graph model given by some tuple* (e, l, K) *is the scalar product* $\langle \exp(f), \exp(tg)\rangle$ *for the polynomials* f *and* g *in* $p_1, \ldots, p_k$ *(*$k = \max K$*) defined by Eqs. (4.7) and (4.8).*

The following proof generalizes Eq. (1.7) to handle more graph classes, and degree restrictions. For each graph class we define a symmetric function encoding of graphs without degree restrictions, which we shall denote by $\bar{F}$, expressed in the power sum basis (see Table 3.1). In all cases $\bar{F}$ can be written as an exponential of an infinite sum $\bar{f}$ of terms in the p_i, and the wanted generating function takes the form of a scalar product $\langle \bar{F}, G\rangle$. We show that the coefficient extractor G has the form $G = \exp(tg)$ for a symmetric polynomial g. Writing g in the power sum basis involves only a finite number of p_i, and hence $\langle \bar{F}, G\rangle = \langle F, G\rangle$, where F is obtained from $\bar{F}$ by setting all but a finite number of the p_i to 0, and hence is of the form $F = \exp(f)$ where f is obtained from $\bar{f}$ in the same way.

Proof. First we consider the extraction operators and the corresponding series G. For any symmetric function S, the coefficient of the monomial m_λ in S is $\langle S, h_\lambda\rangle$. To get the desired form of G we use the decomposition $h_\lambda = h_{\lambda_1} h_{\lambda_2} \cdots$, and the linearity of the scalar product.

Thus, in order to count k-regular objects, that is, for the case $K = \{k\}$, as above with Eq. (1.7) we have to use $\lambda = (k, \ldots, k)$, with n equal parts, for each size n. This yields the extraction formula

$$\sum_{n \geqslant 0} \left\langle S, h_{k^n} \right\rangle \frac{t^n}{n!} = \sum_{n \geqslant 0} \left\langle S, h_k^n \frac{t^n}{n!} \right\rangle = \langle S, \exp(h_k t)\rangle,$$

and so $G = \exp(th_k)$.

In the case where the degree can be from a finite set K of integers, we need to extract the coefficients $\langle S, m_\lambda\rangle$ for partitions λ with all parts in K. We note that, by the classic correspondence between the exponential function and labelled set constructions, $G = \exp(t \sum_{k \in K} h_k)$ gives the correct set of monomials with the correct weighting. Using the change of basis formula $h_n = \sum_{\lambda \vdash n} \frac{p_\lambda}{z_\lambda}$, we see that written in the power sum basis, G uses a finite number of p_i, the maximum index of which is at most the maximum element of K.

Next, let us consider the symmetric functions $\bar{F}$ encoding the different unconstrained graph classes corresponding to each choice for (e, l). We can build up the generating function for all six combinations, and use some basic symmetric-function identities to express them using power sums. The results are derived from [Sta99, Proposition 7.7.4], which is proved in a manner similar to Eq. 1.4 and states

$$\prod_{i,j} \frac{1}{1 - x_i y_j} = \exp\left(\sum_{n \geqslant 1} \frac{1}{n} p_n(x) p_n(y)\right), \quad \prod_{i,j} (1 + x_i y_j) = \exp\left(\sum_{n \geqslant 1} \frac{(-1)^{n-1}}{n} p_n(x) p_n(y)\right), \tag{3.1}$$

where $p_n(x)$ is the same series $p_n = \sum_i x_i^n$ as before and $p_n(y)$ is its analogue $\sum_i y_i^n$. We exploit these equations using two key evaluations. Setting $y_i = x_i$ in Eq. (3.1) (and hence writing p_n

Graph type	x-expression $\bar F$	$\bar f$ in power-sum basis
('se', 'll')	$\prod_{i<j}(1+x_ix_j)$	$\sum_{n\geqslant 1}(-1)^{n-1}\frac{p_n^2-p_{2n}}{2n}$
('se', 'la')	$\prod_{i\leqslant j}(1+x_ix_j)$	$\sum_{n\geqslant 1}(-1)^{n-1}\frac{p_n^2+p_{2n}}{2n}$
('se', 'lh')	$\prod_{i<j}(1+x_ix_j)\times\prod_i(1+x_i)$	$\sum_{n\geqslant 1}(-1)^{n-1}\left(\frac{p_n^2-p_{2n}}{2n}+\frac{p_n}{n}\right)$
('me', 'll')	$\prod_{i<j}(1-x_ix_j)^{-1}$	$\sum_{n\geqslant 1}\frac{p_n^2-p_{2n}}{2n}$
('me', 'la')	$\prod_{i\leqslant j}(1-x_ix_j)^{-1}$	$\sum_{n\geqslant 1}\frac{p_n^2+p_{2n}}{2n}$
('me', 'lh')	$\prod_{i<j}(1-x_ix_j)^{-1}\times\prod_i(1-x_i)^{-1}$	$\sum_{n\geqslant 1}\left(\frac{p_n^2-p_{2n}}{2n}+\frac{p_n}{n}\right)$

Table 3.1: Product expressions to encode labelled graphs of the six types considered in Eq. (4.7). The product expression $\bar F$ is equal to the exponential $\exp(\bar f)$ where $\bar f$ is the summation in the final column. The polynomial f in the hypotheses is equal to $\bar f$ where all p_n are set to zero, for $n > k$.

for $p_n(x)$) we get

$$\prod_{i,j}\frac{1}{1-x_ix_j}=\exp\left(\sum_{n\geqslant 1}\frac{p_n^2}{n}\right)\quad\text{and}\quad\prod_{i,j}(1+x_ix_j)=\exp\left(\sum_{n\geqslant 1}(-1)^{n-1}\frac{p_n^2}{n}\right).\tag{3.2}$$

Next, setting $y_1 = 1$ and $y_k = 0$, for $k > 1$, into Eq. (3.1), thus forcing $p_n(y) = 1$, for $n \geqslant 1$, we have

$$\prod_{i}\frac{1}{1-x_i}=\exp\left(\sum_{n\geqslant 1}\frac{p_n}{n}\right)\quad\text{and}\quad\prod_{i}(1+x_i)=\exp\left(\sum_{n\geqslant 1}(-1)^{n-1}\frac{p_n}{n}\right).$$

Remark that for $p_n = p_n(x_1, x_2, \dots)$, we have $p_n(x_1^2, x_2^2, \dots) = p_{2n}(x_1, x_2, \dots) = p_{2n}$, hence

$$\prod_{i}\frac{1}{1-x_i^2}=\exp\left(\sum_{n\geqslant 1}\frac{p_{2n}}{n}\right)\quad\text{and}\quad\prod_{i}(1+x_i^2)=\exp\left(\sum_{n\geqslant 1}(-1)^{n-1}\frac{p_{2n}}{n}\right).\tag{3.3}$$

All six cases can be derived using these in various products, and the results are summarized in Table 3.1. For example, multiplying Eqs. (3.2) and (3.3) yields the squares of the x-expressions for ('se','la') and ('me','la'), and correspondingly the doubles of the power-sum expressions $\bar f$. These calculations give the values in Eq. (4.7) once we recall that the maximum index of a power-sum symmetric function in g is bounded, thus the scalar product will be unchanged if the power sums with indices higher than that bound are set to 0. Each truncated expression comes in two sums to accommodate the parts in p_n and p_{2n} separately. □

It is worth it to recall that given two combinatorial classes, and differential equations satisfied by the generating function of each class, we can determine the differential equations satisfied by both the sum and the product of the two generating functions. This sum and product are respectively the generating functions of the union and the cartesian product of the two classes.

3.2. Practical calculations

We implemented our method as summarized in Algorithm 4.1 and ran it successfully in Maple. Table A.1 presents the results. All of our calculations are for sets K included in $\{1,2,3,4,5,6,7\}$, allowing several edge and loop variations. For example, we have computed the differential equation satisfied by the set of labelled graphs with degree bounded by $k = 7$, that is, for $K = \{1, 2, 3, 4, 5, 6, 7\}$, and the differential equation satisfied by the set of labelled graphs with degree exactly $k = 7$, that is, for $K = \{7\}$.

Following up the remark at the end of the previous section, from our existing results we could for example easily determine the differential equations satisfied by the set of graphs that are either 5- or 6-regular. (This calculation by the closure property of D-finite functions under addition contrasts with the case of graphs whose vertices are of degree either 5 or 6, which we can determine directly by the method of the present paper.)

4. Description of the approach

We now provide more formal details on our method, which will lead to Algorithm 4.1. Fix a number k and, again, write $p = (p_1, \dots, p_k)$ and $\partial = (\partial_1, \dots, \partial_k)$ as a shorthand. The number k is the level of regularity of graphs, that is, with the k variables in p we will be able to express the enumerative series of k-regular graphs and variants with degree bounded by k.

Introduce the Weyl algebra

$$W_p := \mathbb{Q}\langle p_1, \dots, p_k, \partial_1, \dots, \partial_k; \partial_i p_j = p_j \partial_i + \delta_{i,j},\ 1 \leqslant i, j \leqslant k\rangle, \tag{4.1}$$

where $\delta_{i,j}$ is one if and only if $i = j$, zero otherwise. Each ∂_i acts on $\mathbb{Q}[t][[p]]$ as the usual derivation operator with respect to p_i. The following relations are easily derived for any two series U and V in $\mathbb{Q}[t][[p]]$ and any $i \in \{1, \dots, k\}$:

$$\langle p_i U, V\rangle = \langle U, i\partial_i \cdot V\rangle, \qquad \langle i\partial_i \cdot U, V\rangle = \langle U, p_i V\rangle.$$

By bilinearity and symmetry, proving these relations reduces indeed to proving the identity

$$\langle p_i p_\lambda, p_\nu\rangle = \langle p_\lambda, i\partial_i \cdot p_\nu\rangle \tag{4.2}$$

for any i, λ, and ν. We prove it for completeness. First, the identity holds if ν does not involve i, both sides being zero. So we continue assuming i appears in ν. Define λ^+ as the partition obtained by adjoining i to λ and consider the integers r_i as in Eq. (1.2), so that the analog of Eq. (1.2) for λ^+ is obtained by incrementing r_i. Therefore, z_{λ^+} is equal to $z_\lambda i(r_i + 1)$. Define as well ν^- as the partition obtained by removing i from ν, so that $\partial_i \cdot p_\nu = s_i p_{\nu^-}$ where s_i denotes the number of occurrences of i in ν. In particular, $s_i = r_i + 1$ if $\nu = \lambda^+$. Next,

$$\langle p_i p_\lambda, p_\nu\rangle = \langle p_{\lambda^+}, p_\nu\rangle = \delta_{\lambda^+,\nu} z_\lambda i(r_i+1) = i(r_i+1)\delta_{\lambda,\nu^-} z_\lambda = i(r_i+1)\langle p_\lambda, p_{\nu^-}\rangle = i\langle p_\lambda, \partial_i \cdot p_\nu\rangle$$

and Eq. (4.2) is proved. More generally, for any linear differential operator L, $\langle L \cdot U, V\rangle = \langle U, L^\dagger \cdot V\rangle$, where the adjoint $L^\dagger$ of L is the result of applying the algebra anti-automorphism of W_p defined by $p_i^\dagger = i\partial_i$ and $\partial_i^\dagger = i^{-1} p_i$. This adjoint operation is an involution.

Note that $W_p[t]$ acts on $\mathbb{Q}[t][[p]]$ as well, but we will restrict the use of this action to right-hand arguments of scalar products.

With Theorem 4.9, we will only able to prove the correctness of Algorithm 4.1, but not its completeness. This is why we now proceed to progressively develop sufficient properties satisfied by intermediate Gröbner bases in calculations for Algorithm 4.1 to be successful and return correct outputs.

Let f and g be two polynomials in $\mathbb{Q}[p] \setminus \mathbb{Q}$, which we do not want to fix at this point to the polynomial needed for the k-regular models provided in Lemma 3.1. Introduce

$$F := \exp(f) \in \mathbb{Q}[[p]], \qquad G := \exp(tg) \in \mathbb{Q}[p][[t]] \cap \mathbb{Q}[t][[p]], \qquad S := \langle F, G \rangle \in \mathbb{Q}[[t]].$$

We will write f_i for $\partial_i \cdot f$ and g_i for $\partial_i \cdot g$. Given an element $P \in W_p$, we define its twist $P^\sharp \in W_p[t]$ by

$$P^\sharp(p_1, \dots, p_k, \partial_1, \dots, \partial_k) = P^\dagger(p_1, \dots, p_k, \partial_1 + tg_1, \dots, \partial_k + tg_k). \tag{4.3}$$

Finally, let $H \subset \mathbb{Q}(t)[p]$ denote the vector space

$$H := \sum_{P \in \operatorname{ann}(F)} P^\sharp \cdot \mathbb{Q}(t)[p] = \sum_{P \in \operatorname{ann}(F)^\sharp} P \cdot \mathbb{Q}(t)[p]. \tag{4.4}$$

Lemma 4.1. *For any polynomial $h \in H \subset \mathbb{Q}(t)[p]$, the scalar product $\langle F, hG \rangle$ is zero.*

Proof. If $P \in W_p$ annihilates F, then for any $s \in \mathbb{Q}[p]$,

$$0 = \langle P \cdot F, sG \rangle = \langle F, P^\dagger \cdot (sG) \rangle = \langle F, (P^\sharp \cdot s) G \rangle, \tag{4.5}$$

where $P^\sharp(p_1, \dots, p_k, \partial_1, \dots, \partial_k) \in W_p[t]$ is defined by Eq. (4.3). Note that when P runs over the left ideal of annihilating operators of F, denoted $\operatorname{ann}(F)$, the transform $P^\dagger$ runs over the right ideal $\operatorname{ann}(F)^\dagger$, and likewise $P^\sharp$ runs over the right ideal $\operatorname{ann}(F)^\sharp$. The result follows by linearity over $\mathbb{Q}(t)$. □

Introduce the derivation operator ∂_t with respect to t as well as the Weyl algebra

$$W_t := \mathbb{Q}\langle t, \partial_t; \partial_t t = t\partial_t + 1 \rangle.$$

Observe

$$\partial_t^j \cdot S = \langle F, \partial_t^j \cdot G \rangle = \langle F, g^j G \rangle, \tag{4.6}$$

so that, as a consequence of Lemma 4.1, any annihilator $Q = \sum_{j=0}^r q_j(t) \partial_t^j \in W_t$ of S satisfies

$$0 = Q \cdot S = \langle F, Q \cdot G \rangle = \left\langle F, \sum_{j=0}^r q_j g^j G \right\rangle = \left\langle F, \sum_{j=0}^r q_j (g^j + \ell_j) G \right\rangle$$

for any polynomials ℓ_j that are elements of the vector space H defined by Eq. (4.4). In what follows, for each g^j we (implicitly) obtain ℓ_j in such a way that the computed $g^j + \ell_j$ is "reduced" and confined in a finite-dimensional $\mathbb{Q}(t)$-vector space. This makes it possible to derive the q_j. The procedure is therefore to deal with Eq. (4.6) for each j separately, by reducing the coefficient g^j modulo H.

Although, as we will see, we will only be able to reduce by a subspace of H, we continue our analysis by expressing the space H as a finite sum of spaces.

Lemma 4.2. *The vector space* H *is the sum* $\sum_{i=1}^{\ell} L_i \cdot \mathbb{Q}(t)[p]$ *for any finite family* $\{L_i\}_{i=1}^{\ell}$ *generating* $\operatorname{ann}(F)^\sharp$.

Proof. The finite sum $\tilde{H} := \sum_{i=1}^{\ell} L_i \cdot \mathbb{Q}(t)[p]$ is a subspace of H. Writing any P of $\operatorname{ann}(F)^\sharp$ in the form $P = \sum_{i=1}^{\ell} L_i U_i$ yields the inclusion of the space H into $\tilde{H}$, and thus the equality $\tilde{H} = H$. □

Eq. (4.5) holds in particular for $P = P_i := i(\partial_i - f_i)$, in which case $P^\sharp$ is given as $P_i^\sharp$ in Eq. (4.9).

To define the reduction that was announced, we proceed by exchanging the generating family $\{P_i^\sharp\}$ of $\operatorname{ann}(F)^\sharp$ for a family $\{G_i\}_{i=1}^{\ell}$ satisfying the property that any term cm to be reduced (c a coefficient, m a monomial) will be obtained for some $(j, s) \in \{1, \dots, \ell\} \times \mathbb{Q}(t)[p]$ as the leading monomial of $G_j \cdot s$, where leading monomials are decided by some monomial ordering of $\mathbb{Q}(t)[p]$ that is compatible with the choice of the family $\{G_i\}_{i=1}^{\ell}$. To make this possible, we ensure that $G_j = m_j + \cdots \in W_p(t)$ for a monomial m_j in p, with the property that, for any $\tilde{s} \in \mathbb{Q}(t)[p]$, the leading monomial of $m_j \tilde{s}$ is larger than the leading monomial of $(G_j - m_j) \cdot \tilde{s}$. In practice, the polynomial s used to reduce cm will be set to the term cm/m_j, so that m is reduced into $m - G_j \cdot (cm/m_j)$. Observing that only finitely many monomials are divisible by none of the m_j will then ensure the wanted confinement in finite dimension.

The following lemma describes a situation in which a skew polynomial G has the property we require from the G_j. For future reference, we give a name to this property.

Definition 4.3. A skew polynomial $G \in W_p(t)$ is said to be *dominant* if it is of the form $G = m + R$ for a non-zero monomial $m \in \mathbb{Q}[p]$ and some skew polynomial R involving only monomials $p^\alpha \partial_p^\beta$ for which $\sum_{i=1}^{k} (\alpha_i - \beta_i)$ is less than the total degree of m. We call m the *dominant monomial* of G.

Lemma 4.4. *Let* $G \in W_p(t)$ *be dominant with dominant monomial* m. *Fix a monomial ordering on* $\mathbb{Q}(t)[p]$ *that is graded by total degree. Then, for any non-zero polynomial* s, *the leading monomial of* $G \cdot s$ *is the product of* m *with the leading monomial of* s.

Proof. The quantity $\sum_{i=1}^{k} (\alpha_i - \beta_i)$ is what is added to the degree of any monomial q when applying $p^\alpha \partial_p^\beta$ to it. So because G is dominant and the ordering is graded by total degree, the leading monomial of $G \cdot s$ is the product mq. The result then follows from the general properties of monomial orderings. □

In general, we do not know how to ensure the existence of a family $\{G_j\}_{j=1}^{\ell}$ consisting of dominant G_j, but for k-regular graphs, a module Gröbner basis calculation will in practice compute such a family for k up to 7, as we will show in Proposition 4.7. To describe how to engineer the construction of such a family, we now introduce a few compatible monomial orderings for the structures that we will use.

Definition 4.5. 1. Let $\prec$ denote a monomial ordering on $W_p(t)$ that compares the p_i by a total degree order and that eliminates p by making p_i lexicographically higher than ∂_j for all i and j, Here, lexicographically higher means $p_i \succ \partial_j^n$ for all i, j, and n.

2. Let $\prec_h$ denote the monomial ordering on the free right module $\eta_0 W_p(t) \oplus \eta_1 W_p(t)$ that eliminates η_1 by making η_1 lexicographically higher than η_0 and by sorting monomials in the same η_i by means of $\prec$. Here, η_0 and η_1 are new names denoting elements of a basis and lexicographically higher means $p^\alpha \partial^\beta \eta_1 \succ_h p^\gamma \partial^\delta \eta_0$ for all α, β, γ, and δ.

3. Let $\prec_p$ denote the monomials ordering on $\mathbb{Q}(t)[p]$ that is induced by $\prec$. This is just notation to stress the polynomial situation, as we could use the symbol $\prec$ as well.

For each i, write $P_i^\sharp = Q_i(p) + R_i(p, \partial)$, where Q_i does not involve any ∂_j and each monomial of R_i involves at least one ∂_j. Then, consider $M_i := \eta_1 Q_i + \eta_0 R_i$. Consider a Gröbner basis for the right[6] module over $W_p(t)$ generated by the M_i with respect to $\prec_h$. Those elements $\eta_1 Q + \eta_0 R$ of the Gröbner basis satisfying $Q \neq 0$ need not make $Q + R$ dominant by general properties of Gröbner bases, but as we start from the dominant elements $P_i^\sharp$, the Gröbner basis elements can be hoped to make the $Q + R$ dominant, at least if the Gröbner basis calculation does not modify too much the higher monomials of the input polynomials. Indeed, we are in the nice situation that the $Q + R$ are the G_i we are looking for when $k \leqslant 7$ for all variant models described in Section 3. Because the goal of the calculation is to reveal a zero-dimensional ideal in $\mathbb{Q}(t)[p]$, the module structure over $W_p(t)$ can be replaced with a module structure over $\mathbb{Q}(t)[p]$, that is, one would like to consider only polynomial recombinations guided by the coefficients of η_1, without continuing with non-commutative recombinations of the coefficients of η_0 between generators with zero coefficient with respect to η_1. This is achieved by viewing the M_i as elements of a free $\mathbb{Q}(t)[p]$-module with a finite basis consisting of η_1 and some $\partial^\alpha \eta_0$, for the same ordering $\prec_h$. For all models considered when $2 \leqslant k \leqslant 6$, this has the nice consequence of speeding up the calculation. For the models we considered with $k = 7$ we therefore directy used this improvement.

Proposition 4.6. *The operators G_i obtained at Step e. of Algorithm 4.1 are such that the ideal $\sum_{i=1}^{\rho} G_i W_p(t)$ is a subideal of $\operatorname{ann}(F)^\sharp$, including the degenerate case 0 obtained if $\rho = 0$.*

Proof. Given a graph model (e, l, K), Step a. implements the formula announced by Lemma 3.1, so that $\langle \exp(f), \exp(tg) \rangle$ is the exponential generating function of the model and F is fixed to $\exp(f)$. Next, Step b. computes generators of the right $W_p(t)$-ideal $\operatorname{ann}(F)^\sharp$ because the map $\sharp$ is a linear anti-homomorphism from W_p to $W_p(t)$ and the P_i generate the left W_p-ideal $\operatorname{ann}(F)$. Steps c. to e. produce operators G_i that are elements of $\operatorname{ann}(F)^\sharp$. This proves the result. □

[6] As Maple only computes Gröbner bases for left structures, the actual computer calculation computes a Gröbner basis for the left module generated by the $Q_i^\dagger \eta_1 + R_i^\dagger \eta_0$, then returns the adjoints $(Q + R)^\dagger$ obtained from the elements $Q\eta_1 + R\eta_0$ of the Gröbner basis satisfying $Q \neq 0$.

Input: a graph model (e, l, K),
where $e \in \{\text{‘se’}, \text{‘me’}\}$, $l \in \{\text{‘ll’}, \text{‘la’}, \text{‘lh’}\}$, $K \subset \mathbb{N}_{>0}$.

Output: an operator of minimal order in ∂_t
that cancels the exponential generating function of the model.

---

a. Set $k = \max K$, then compute f and g by the formulas

$$f = \sum_{i=1}^{k}\Big((-1)^{i+1}\delta_{e,\text{‘se’}} + \delta_{e,\text{‘me’}}\Big)\left(\frac{p_i^2}{2i} + \delta_{l,\text{‘lh’}}\frac{p_i}{i}\right) - (-1)^{\delta_{l,\text{‘la’}}}\sum_{i=1}^{\lfloor k/2\rfloor}\Big((-1)^{i+1}\delta_{e,\text{‘se’}} + \delta_{e,\text{‘me’}}\Big)\frac{p_{2i}}{2i}, \tag{4.7}$$

$$g = \sum_{j\in K}\sum_{\lambda\vdash j}\frac{p_\lambda}{z_\lambda}, \tag{4.8}$$

where the expression $\delta_{x,y}$ is equal to 1 if $x = y$ and to 0 otherwise.

b. Get generators of the right $W_p(t)$-ideal $\operatorname{ann}(F)^\sharp$ by computing $P_i^\sharp$ for $1 \leqslant i \leqslant k$ by the formula

$$P_i^\sharp = (i\partial_i - if_i)^\sharp = p_i - if_i(\partial_1 + tg_1, 2(\partial_2 + tg_2), \dots, k(\partial_k + tg_k)). \tag{4.9}$$

Here, the right-hand side is obtained by the non-commutative substitution of p_1 with $\partial_1 + tg_1$, of p_2 with $2(\partial_2 + tg_2)$, …, of p_k with $k(\partial_k + tg_k)$, in the polynomial $f_i = f_i(p_1, \dots, p_k)$.

c. Transform each $P_i^\sharp$ by the map

$$\sum_\alpha c_\alpha \partial^\alpha \mapsto c_0\eta_1 + \sum_{\alpha\neq 0} c_\alpha\partial^\alpha\eta_0$$

to get a system of generators of a right $\mathbb{Q}(t)[p]$-module with basis $\{\eta_1\} \cup \{\partial^\alpha\eta_0\}_\alpha$. Here, α ranges in the finite set of non-zero exponents involved in the $P_i^\sharp$.

d. Compute a Gröbner basis of this right module for an ordering $\prec_h$ that makes η_1 lexicographically higher than η_0, and p lexicographically higher than ∂, using Definition 4.5.

e. Obtain elements $G_1, \dots, G_\rho$ of $W_p(t)$ by setting $\eta_1 = \eta_0 = 1$ in those elements of the Gröbner basis that involve η_1 with a non-zero coefficient, then write each G_i in the form $Q_i(p) + R_i(p, \partial)$ where each monomial of R_i involves at least one ∂_j.

f. If the polynomial ideal $I = (Q_1, \dots, Q_\rho)$ of $\mathbb{Q}(t)[p]$ has positive dimension, then return 'FAIL', else determine the monomials $p^{\beta_1}, \dots, p^{\beta_\delta}$ under the stair of I with respect to $\prec_p$.

g. Set $\check{g}_0 = 1$, then for i from 2 to δ, set $\check{g}_i = \mathrm{red}(g\check{g}_{i-1} + \partial_t \cdot \check{g}_{i-1}, (G_i)_{i=1}^{\rho}, \prec)$.

h. Compute the matrix M with rows indexed by $0 \leqslant i \leqslant \delta$ and columns indexed by $1 \leqslant j \leqslant \delta$, whose entry at position (i, j) is the coefficient of p^{β_j} in $\check{g}_i$.

i. Compute a basis of the left kernel of M, then combine its elements to obtain a non-zero row vector $(q_0, \dots, q_\delta) \in \mathbb{Q}(t)^{\delta+1}$ with maximal number of zeros to the right.

j. Return $q_0 + q_1\partial_t + \dots + q_\delta\partial_t^\delta$.

Algorithm 4.1: Outline of the method. Uses the reduction $\mathrm{red}(\cdot)$ of ALGORITHM 4.2.

Input: a polynomial $s \in \mathbb{Q}(t)[p]$ to be reduced and dominant operators $G_1, \dots, G_\rho$ of $W_p(t)$.

Output: a polynomial $\check{s} \in \mathbb{Q}(t)[p]$.

---

a. For each i, set m_i to the leading monomial of G_i.

b. If no monomial of s is divisible by any of the m_i, return s.

c. Set m to the maximal monomial in s that is divisible by some m_i and choose j such that m_j divides m.

d. Set c to the coefficient of m in s and c_j to the leading coefficient of G_j.

e. Set t to the term $\frac{c}{c_j}\frac{m}{m_j}$ and return $\mathrm{red}(s - G_j \cdot t, (G_i)_{i=1}^{\rho})$.

Algorithm 4.2: Reduction $\mathrm{red}(\cdot)$ used by ALGORITHM 4.1.

Proposition 4.7. *For each model in Table A.1:*

1. *the elements $Q_i(p) + R_i(p, \partial)$ of the Gröbner basis obtained at Step e. of Algorithm 4.1 are all dominant and so satisfy the property of G in Lemma 4.4, for $m = Q_i(p)$.*

2. *the ideal I generated by the $Q_i(p)$ at Step f. has dimension zero.*

Proof. For each model, the proof is by inspection after computing the Gröbner basis at Step d.: testing Point (1) consists in comparing each monomial of $Q_i + R_i$ with the leading monomial m_i of Q_i; testing Point (2) is done by computing the dimension of the commutative polynomial ideal by a classical algorithm, after observing that the commutative polynomials Q_i obtained as parts of the $Q_i + R_i$ are already a Gröbner basis for the ordering induced on $\mathbb{Q}(t)[p]$ by the ordering $\prec_h$ used at Step d. □

The polynomials $G_i = Q_i + R_i$ obtained at Step e. of Algorithm 4.1 will be used to reduce polynomials to a finite-dimensional vector space at Step g. So we first analyse this reduction separately. To this end, we stress a possibly confusing fact: although $\operatorname{ann}(F)^\sharp$ is a right-module, we make it act to the left of polynomials during reductions.

Proposition 4.8. *The reduction algorithm, Algorithm 4.2, terminates if the ideal I generated by the leading monomials with respect to $\prec$ of the dominant inputs G_i is zero-dimensional. It returns a polynomial $\check{s} \in \mathbb{Q}(t)[p]$ whose monomials are under the stairs of I with respect to $\prec_p$. The difference $s - \check{s}$ is in $\sum_{i=1}^{\rho} G_i \cdot \mathbb{Q}(t)[p]$, and therefore it is an element of the vector space H defined by Eq. (4.4).*

Proof. The term t at Step e. is chosen so that leading terms satisfy:

$$\operatorname{lt}(G_j \cdot t) = \operatorname{lt}(\operatorname{lt}(G_j) \cdot t) = \operatorname{lt}(c_j m_j \cdot t) = \operatorname{lt}(c_j m_j t) = \operatorname{lt}(cm) = cm,$$

where the first equality is by the assumed dominant character of G_j and by Lemma 4.4, and where the third equality is because m_j does not involve ∂. So all monomials appearing in the difference $s - G_j \cdot t$ and susceptible of reduction by the m_i are less than m. Because of the 0-dimensionality of I, the recursive calls to $\operatorname{red}(\cdot)$ terminate. The output has no monomial reducible by any m_i, hence all its monomials are under the stairs of I. Finally, each $G_j \cdot t$ considered by some recursive call is in H, hence so is $\check{s} - s$. □

We note that the choice of $\prec_p$ to be induced by $\prec_h$ and the fact that $\prec_h$ reduces to a graded order on monomials in p are not for termination, but for efficiency.

Theorem 4.9. *Algorithm 4.1 is correct, that is, if it terminates, then either this is by giving up, returning 'FAIL' at Step f., or this is by outputting at Step j. a differential equation that annihilates the scalar product S.*

Proof. By Proposition 4.6, we obtain that, after Step e., $\sum_{i=1}^{\rho} G_i W_p(t)$-is a subideal of $\operatorname{ann}(F)^\sharp$. So, if the algorithm terminates without returning 'FAIL' at Step f., then the ideal I, which is generated by the $Q_i = G_i - R_i$, must have dimension 0. Proposition 4.8 shows that $\operatorname{red}(\cdot)$ modifies its input s by adding to it an element $\check{s} - s$ of the vector space H defined by Eq. (4.4). So, Step g. is an incremental calculation of the $\partial_t^i \cdot S$ defined by Eq. (4.6), as justified by Lemma 4.1. The final steps compute a non-trivial $\mathbb{Q}(t)$-linear relation between the $\partial_t^i \cdot S$. The corresponding differential equation is finally returned. □

We emphasize that we do not claim that the case of 0-dimensionality of the ideal I at Step f. implies that termination of the subsequent steps. In fact, this is just a necessary condition for those steps to be well defined. A sufficient condition for termination, knowing that the dimension of I is 0, is that for each i in $\{1, \dots, \rho\}$ the (total) degree of Q_i in p is larger than the (partial) degree of R_i in p (cf. Lemma 4.4). This is a condition that we have observed in all of our experiments.

Because we observe that our implementation of Algorithm 4.1 terminates and outputs a differential equation on the models listed in Table A.1, we get the following corollary.

Corollary 4.10. *For each model in Table A.1, there exist:*

1. *a linear differential equation annihilating the generating function of the model, with order given in column '∂_t',*
2. *a linear recurrence equation annihilating the enumerative sequence of the model, with order given in column '∂_n minimized', the latter being the minimal order of a recurrence if it is not starred.*

Proof. Only the minimality of recurrence order requires a proof: this is by the correctness of van Hoeij's `LREtools[MinimalRecurrence]` implementation, whose proven method is described in his student Zhou's PhD thesis [Zho22, Chapter 6]. □

5. No computation of initial conditions is needed

For all classes, after computing the ODE one readily proves by observation that it possesses the only exponent $n = 0$ at $t = 0$. Consequently, the series solutions form a 1-dimensional vector space, for which a possible basis is the family with only entry our combinatorial series, e.g., the singleton family $(R^{(k)}(t))$ in the k-regular case. Note that the empty graph is k-regular for any k, implying the identity $R^{(k)}(t) = 1 + O(t)$, and an analogous result holds for all degree sets K. Converting the ODE to a recurrence relation satisfied by the coefficient sequence $(c_n)_{n\in\mathbb{Z}}$ of any of its series solution $\sum_{n\in\mathbb{Z}} c_n t^n$ and forcing $c_0 = 1$ and $c_n = 0$ for all $n < 0$ therefore uniquely determines all c_n for $n > 0$. This observation generalizes to any of the models of edges and loops presented in Section 3.

So, a complete proof of correctness of the method for computing an ODE satisfied by the scalar product, together with the observation above, makes it unnecessary to apply a resource-consuming calculation of first terms of the series. Nonetheless, for all the classes we did verify our series solution by direct computation of scalar product. For example for simple k-regular graphs, we directly determined $r_n^{(k)}$ for $n \leqslant 1284$ when $k = 3$, $n \leqslant 216$ when $k = 4$, $n \leqslant 90$ when $k = 5$, $n \leqslant 46$ when $k = 6$, and $n \leqslant 31$ when $k = 7$. Furthermore, McKay provided values for $k = 5$ and $n \leqslant 600$, all consistent with our computations.

6. Minimal recurrences

Table A.1 shows that in many instances the obtained ODE has low order and high degree, leading to a recurrence of high order and low degree. This is one possible motivation for searching for recurrences of lower orders. To this end, we have used Mark van Hoeij's Maple implementation of his algorithm to reduce the order of recurrences satisfied by a specific solution to a given initial recurrence[7], which is available in Maple as `LREtools[MinimalRecurrence]`.

For roughly half of the models, we could reduce the order. For a small quarter of the list, minimizing was too much calculation. Fortunately in those cases, the generating series is even, the initial recurrence relates every second term of the sequence, and a change of indexes to consider the sub-sequence of even terms led to a recurrence that Maple could reduce. These are marked with a star in the table.

However, for models with $k = \max K = 6$ and loops described by $l =$ 'lh', no recurrence of lower order could be found.

In all relevant cases, the recurrence of reduced order is larger than the initial recurrence, as the degree increase outbalances the order decrease. This is amplified by an increase in the size of integers that occur in the reduced recurrence. For example, for the model defined by $e =$ 'se', $l =$ 'll', $K = \{6\}$, the product "degree $\times$ order" is multiplied by 4.8, raising from 876 to 4176, while the length of the longest integer raises from 50 to 236. As a consequence, computing first terms as with Maple's command `gfun[rectoproc]` results in slower calculations with the reduced recurrence: for the same example, computing up to the 1000th term takes more than five times as much with the reduced recurrence, with times raising from below 30 seconds to above 150 seconds.

7. Asymptotic expansions

In general, obtaining an asymptotically equivalent estimate starting from a recurrence relation and initial values can be tricky, with a key difficulty to determine the constant coefficient in front of a dominant asymptotic behavior. On the other hand, using the recurrence to expand a known single-term asymptotic equivalent estimate into to a higher-order asymptotic expansion is almost routine. This favorable situation occurs for k-regular graphs, for which the asymptotic estimate

$$\frac{1}{e^{(k^2-1)/4}} \frac{(kn)!}{(kn/2)!\,(2^{k/2}k!)^n}(1 + O(1/n)) \tag{7.1}$$

for fixed k has been known since 1978 with the independent works of Bender and Canfield, and of Wormald (see [McK85, Theorem 5.2] and the references in that article). The O-term is expected to be a series in $1/n$, so using undetermined coefficients "readily" yields an asymptotic formula. We listed such formulas for $k \in \{3, 4, 5, 6\}$ in Table B.1. They match and prolong the formulas obtained by a different approach in [Pan24].

[7]The documentation of the procedure promises to achieve the minimal order, but to the best of our knowledge no formal publication is available yet.

The only difficulty in the method lies in the size of the recurrence relation: for calculations (with Maple) to terminate in reasonable time[8], one has to predict common denominators and avoid any calculation with rational functions—as is just common sense. It is also not clear that a minimal-order recurrence relation can help as its size (order times degree) is more than the size of the recurrence relation obtained directly from the ODE. We could not deal with the case $k = 7$ in reasonable time.

8. Conclusion

8.1. Computational considerations

For each graph class we considered with degrees bounded by 6, it did not require more than 10 minutes to determine the differential equation satisfied the generating function. In contrast, for models with $k = \max K = 7$, the very same implementation requires hours to terminate (between 4.5 and $\simeq 30$). For example, for the model defined by $e =$ 'se', $l =$ 'll', $K = \{7\}$, the time breaks down as follows: a Gröbner basis that supports the reduction by the vector space H of Eq. (4.4). can be obtained in less than 1 hour (Steps a. to f. in Algorithm 4.1). From this, one can predict that reduced forms of scalar products will be confined in dimension 20. Twenty successive reductions are then performed, taking longer and longer, for a total duration of about 9 hours (Step g.). After this, the linear algebra (Steps h. and i.) is comparatively fast. The resulting ODE satisfied by the generating series $R^{(7)}(t)$ for 7-regular graphs has order 20.

Also, the generating function for all regular graphs is conjectured to not be D-finite. An argument in support of this is the increase of the order of the ODE found for increasing k. By prolonging the sequence of known differential equations, presumably beyond first possibly irregular terms, we corroborated this belief. On the other hand, are there properties of the presented ODEs that can help us understand if the generating function of all regular graphs is differentially algebraic or not, and if so how to find the differential equation? The question remains open.

8.2. Combinatorial considerations

We have used symmetric function identities to get an expression in the power-sum basis for the graph generating functions. In fact, there is a well developed combinatorial theory to get these expressions directly using cycle index series, and other machinery from species theory. The underlying combinatorial framework is developed and rigourous in [M9́3] and is interpreted for this context in [Mis18]. Roughly, this means that we can express the generating functions of a wide family of combinatorial objects such as hypergraphs, and weighted graphs under restrictions of graph degree in a form similar to those expressed in Lemma 3.1. Furthermore, the f are very simply deduced using the plethysm operator of symmetric functions.

The symmetric function $S := \prod_{i\geqslant 1} \frac{1}{1-x_i} \prod_{1\leqslant i<j} \frac{1}{1-x_i x_j}$ encodes graphs models with $e =$ 'me', $l =$ 'lh'. This is a very well studied symmetric function, as it also encodes semi-standard Young tableaux by their content [Sta99, Corollary 7.13.8]. Thus, the method presented in this work determines generating functions of semi-standard Young tableaux with restrictions on the number

[8] a few hours

of times each number appears as an entry. For example, $\langle S, h_1^n \rangle$ is the number of standard Young tableaux on n boxes with entries $1, 2, \ldots, n$, each appearing exactly once, as in the absence of repetition, semi-standard is equivalent to standard. For $k \geqslant 2$, $\langle S, h_k^n \rangle$ is similarly the number of semi-standard Young tableaux with entries $1, 2, \ldots, n$ repeated k times. Some of our results then double as recurrences for the number of semi-standard Young tableaux on n boxes where the number of times an entry appears comes from a finite set K.

8.3. Ongoing extensions and future works

The enhanced understanding achieved with the present work paved the way to the use of generic integration algorithms in computer algebra: in another work [BCL25], ODEs for the case $k = 8$ could be computed.

The scalar product formulation and the extraction method of the present work easily generalize to bivariate generating functions that enumerate various models of regular graphs counted by both edges and vertices. A variation of Algorithm 4.1 then outputs a system of linear differential equations. This will be explored in a future work, with an application to a family of regular graphs with related numbers of edges and vertices.

Acknowledgements

The authors are grateful to Nick Wormald for comments on this work, and also to Brendan McKay for motivating our interest in the problem; for providing some counting sequence data for comparision purposes; and for keen observations that arose in the study of the output recurrences. We are also grateful to the anonymous referees who provided useful comments and references.

References

[Bah20] Rouchdi Bahloul. Gröbner bases in D-modules: application to Bernstein-Sato polynomials. In *Two algebraic byways from differential equations: Gröbner bases and quivers*, volume 28 of *Algorithms Comput. Math.*, pages 75–93. Springer, Cham, 2020. doi:10.1007/978-3-030-26454-3_2.

[BCCL10] Alin Bostan, Shaoshi Chen, Frédéric Chyzak, and Ziming Li. Complexity of creative telescoping for bivariate rational functions. In *ISSAC'10: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 203–210, New York, NY, USA, 2010. ACM. doi:10.1145/1837934.1837975.

[BCL25] Hadrien Brochet, Frédéric Chyzak, and Pierre Lairez. Faster multivariate integration in D-modules, April 2025. arXiv:2504.12724.

[BCLS18] Alin Bostan, Frédéric Chyzak, Pierre Lairez, and Bruno Salvy. Generalized Hermite reduction, creative telescoping and definite integration of D-finite functions. In Éric Schost, editor, *ISSAC'18*, pages 95–102. ACM Press, 2018. doi:10.1145/3208976.3208992.

[CC89] D. V. Chudnovsky and G. V. Chudnovsky. The computation of classical constants. *Proceedings of the National Academy of Sciences*, 86(21):8178–8182, November 1989. doi:10.1073/pnas.86.21.8178.

[CdP23] Emma Caizergues and Élie de Panafieu. Exact enumeration of graphs and bipartite graphs with degree constraints. In *Proceedings of the 12th European Conference on Combinatorics, Graph Theory and Applications*, pages 254–262. Masaryk University Press, 2023. doi:10.5817/cz.muni.eurocomb23-035.

[CMS05] Frédéric Chyzak, Marni Mishna, and Bruno Salvy. Effective scalar products of D-finite symmetric functions. *Journal of Combinatorial Theory. Series A*, 112(1):1–43, 2005. doi:10.1016/j.jcta.2005.01.001.

[EH24] Oleg Evnin and Weerawit Horinouchi. A Gaussian integral that counts regular graphs. *Journal of Mathematical Physics*, 65(9):093301, 2024. doi:10.1063/5.0208715.

[Ges87] Ira Gessel. Enumerative applications of symmetric functions. *Séminaire Lotharingien de Combinatoire*, 17(B17a), 1987. Publisher: Universität Wien, Fakultät für Mathematik. 17 pages. URL: http://eudml.org/doc/121536.

[Ges90] Ira M. Gessel. Symmetric functions and P-recursiveness. *Journal of Combinatorial Theory. Series A*, 53(2):257–285, 1990. doi:10.1016/0097-3165(90)90060-A.

[GJR83] I. P. Goulden, D. M. Jackson, and J. W. Reilly. The Hammond series of a symmetric function and its application to P-recursiveness. *Society for Industrial and Applied Mathematics. Journal on Algebraic and Discrete Methods*, 4(2):179–193, 1983. doi:10.1137/0604019.

[Gro92] Harald Gropp. Enumeration of regular graphs 100 years ago. *Discrete Mathematics*, 101(1-3):73–85, 1992. doi:10.1016/0012-365X(92)90592-4.

[Kau23] Manuel Kauers. *D-Finite Functions*, volume 30 of *Algorithms and Computation in Mathematics*. Springer Nature Switzerland, Cham, 2023. doi:10.1007/978-3-031-34652-1.

[Kay21] Atabey Kaygun. Enumerating labeled graphs that realize a fixed degree sequence, 2021. arXiv:2101.02299.

[Lip89] L. Lipshitz. D-finite power series. *Journal of Algebra*, 122(2):353–373, 1989. doi:10.1016/0021-8693(89)90222-6.

[Ḿ93] Miguel Méndez. Multisets and the combinatorics of symmetric functions. *Adv. Math.*, 102(1):95–125, 1993. doi:10.1006/aima.1993.1057.

[McK83] Brendan D. McKay. Applications of a technique for labelled enumeration. In *Proceedings of the fourteenth Southeastern conference on combinatorics, graph theory and computing (Boca Raton, Fla., 1983)*, volume 40, pages 207–221, 1983.

[McK85] Brendan D. McKay. Asymptotics for symmetric 0-1 matrices with prescribed row sums. *Ars Combinatorica*, 19A:15–25, 1985.

[Mis07] Marni Mishna. Automatic enumeration of regular objects. *Journal of Integer Sequences*, 10(5), 2007. Article 07.5.5. 18 pages.

[Mis18] Marni Mishna. Regularity in weighted graphs a symmetric function approach. *Contributions to Discrete Mathematics*, 13(2):32–44, 2018. doi:10.11575/cdm.v13i2.62459.

[MW91] Brendan D. McKay and Nicholas C. Wormald. Asymptotic enumeration by degree sequence of graphs with degrees $(o(n^{1/2})$. *Combinatorica*, 11(4):369–382, December 1991. doi:10.1007/bf01275671.

[OEI] OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences. Published electronically at http://oeis.org. Accessed on July 23, 2024.

[Pan24] Élie de Panafieu. Asymptotic expansion of regular and connected regular graphs, 2024. arXiv:2408.12459.

[Rea59a] R. C. Read. The enumeration of locally restricted graphs (I). *Journal of the London Mathematical Society*, s1-34(4):417–436, 1959. doi:10.1112/jlms/s1-34.4.417.

[Rea59b] Ronald C. Read. *Some enumeration problems in graph theory*. PhD thesis, University of London (University College of the West Indies), 1959.

[RW80] R. C. Read and N. C. Wormald. Number of labeled 4-regular graphs. *Journal of Graph Theory*, 4(2):203–212, 1980. doi:10.1002/jgt.3190040208.

[SST00] Mutsumi Saito, Bernd Sturmfels, and Nobuki Takayama. *Gröbner deformations of hypergeometric differential equations*, volume 6 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2000.

[Sta80] R. P. Stanley. Differentiably finite power series. *European Journal of Combinatorics*, 1(2):175–188, 1980. doi:10.1016/S0195-6698(80)80051-5.

[Sta99] Richard P. Stanley. *Enumerative combinatorics. Vol. 2*, volume 62 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1999. doi:10.1017/CBO9780511609589.

[Wor19] Nicholas Wormald. Asymptotic enumeration of graphs with given degree sequence. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3245–3264. World Scientific, 2019. doi:10.1142/9789813272880_0180.

[Zho22] Yi Zhou. *Algorithms for factoring linear recurrence operators*. PhD thesis, Florida State University, 2022.

A. Differential equations and recurrences relations

Table A.1 gathers information related to computations we performed for a list of models:

- parameters “edges”, “loops”, and k’s are as described in Section 3;
- the obtained ODE has order provided in column ∂_t and its polynomial coefficients have degrees bounded by the number in column t;

- a first recurrence on the number of graphs of size n is directly obtained by translating the ODE by Maple's `gfun[diffeqtorec]`; it has order provided in column ∂_n and its polynomial coefficients have degrees bounded by the number in column n;
- in the majority of models, the command `LREtools[MinimalRecurrence]` could be used to minimize the first recurrence, leading to a new pair of columns n and ∂_n; the minimal order is proved unless it is starred, meaning that the minimized recurrence is only of minimal order among recurrences on even terms;
- the corresponding calculation is done in the time of column "time", measured in seconds.

Table A.1: Models up to $k = 7$. See description in Section 3. All timings obtained on the same computer (with AMD EPYC 9754 processor), running our implementation in Maple 2024.

edges	loops	k's	t	∂_t	n	∂_n	n	∂_n	time
					from ODE		minimized		
se	ll	2	2	1	1	3	1	3	0.04
me	ll	2	2	1	1	3	1	3	0.05
se	la	2	2	1	1	3	1	3	0.05
me	la	2	2	1	1	3	1	3	0.05
se	lh	2	3	1	1	4	1	4	0.05
me	lh	2	3	1	1	4	1	4	0.05
se	ll	1,2	3	1	1	4	1	4	0.05
me	ll	1,2	3	1	1	4	1	4	0.05
se	la	1,2	3	1	1	4	1	4	0.07
me	la	1,2	3	1	1	4	1	4	0.05
se	lh	1,2	3	1	1	4	1	4	0.06
me	lh	1,2	3	1	1	4	1	4	0.06
se	ll	3	11	2	2	12	4	8*	0.08
me	ll	3	11	2	2	12	4	8*	0.08
se	la	3	11	2	2	12	4	8*	0.09
me	la	3	11	2	2	12	4	8*	0.07
se	lh	3	11	2	2	12	6	8	0.09
me	lh	3	11	2	2	12	6	8	0.1
se	ll	1,3	11	2	2	12	4	8*	0.09
me	ll	1,3	11	2	2	12	4	8*	0.07
se	la	1,3	11	2	2	12	4	8*	0.09
me	la	1,3	11	2	2	12	3	8*	0.07
se	lh	1,3	11	2	2	12	6	8	0.07
					from ODE		minimized		
edges	loops	k's	t	∂_t	n	∂_n	n	∂_n	time

Table A.1: Models up to $k = 7$ (continued).

edges	loops	k's	t	∂_t	n	∂_n	n	∂_n	time
					from ODE		minimized		
me	lh	1,3	11	2	2	12	6	8	0.09
se	ll	2,3	11	2	2	12	6	8	0.08
me	ll	2,3	11	2	2	12	6	8	0.1
se	la	2,3	11	2	2	12	6	8	0.08
me	la	2,3	11	2	2	12	6	8	0.08
se	lh	2,3	11	2	2	12	6	8	0.08
me	lh	2,3	11	2	2	12	6	8	0.08
se	ll	1,2,3	11	2	2	12	6	8	0.07
me	ll	1,2,3	11	2	2	12	6	8	0.1
se	la	1,2,3	11	2	2	12	6	8	0.08
me	la	1,2,3	11	2	2	12	6	8	0.07
se	lh	1,2,3	11	2	2	12	6	8	0.08
me	lh	1,2,3	11	2	2	12	6	8	0.08
se	ll	4	14	2	2	15	7	10	0.2
me	ll	4	14	2	2	15	7	10	0.2
se	la	4	14	2	2	15	7	10	0.19
me	la	4	14	2	2	15	6	10	0.17
se	lh	4	30	3	3	31	18	16	0.19
me	lh	4	29	3	3	30	17	16	0.19
se	ll	1,4	29	3	3	30	17	16	0.29
me	ll	1,4	29	3	3	30	17	16	0.28
se	la	1,4	29	3	3	30	17	16	0.2
me	la	1,4	29	3	3	30	17	16	0.18
se	lh	1,4	30	3	3	31	18	16	0.29
me	lh	1,4	29	3	3	30	17	16	0.27
se	ll	2,4	14	2	2	15	7	10	0.2
me	ll	2,4	14	2	2	15	7	10	0.2
se	la	2,4	14	2	2	15	7	10	0.2
me	la	2,4	14	2	2	15	7	10	0.21
se	lh	2,4	29	3	3	30	17	16	0.29
me	lh	2,4	30	3	3	31	18	16	0.27
se	ll	3,4	30	3	3	31	18	16	0.3
me	ll	3,4	29	3	3	30	17	16	0.33
se	la	3,4	29	3	3	30	17	16	0.3
me	la	3,4	29	3	3	30	17	16	0.31
se	lh	3,4	30	3	3	31	18	16	0.33
me	lh	3,4	30	3	3	31	18	16	0.32
					from ODE		minimized		
edges	loops	k's	t	∂_t	n	∂_n	n	∂_n	time

Table A.1: Models up to $k = 7$ (continued).

edges	loops	k's	t	∂_t	n	∂_n	n	∂_n	time
					from ODE		minimized		
se	ll	1,2,3,4	29	3	3	30	17	16	0.32
me	ll	1,2,3,4	29	3	3	30	17	16	0.24
se	la	1,2,3,4	29	3	3	30	17	16	0.34
me	la	1,2,3,4	30	3	3	31	18	16	0.24
se	lh	1,2,3,4	30	3	3	31	18	16	0.28
me	lh	1,2,3,4	30	3	3	31	18	16	0.3
se	ll	5	125	6	6	126	53	32*	1.96
me	ll	5	125	6	6	126	53	32*	1.7
se	la	5	125	6	6	126	53	32*	1.92
me	la	5	125	6	6	126	53	32*	1.8
se	lh	5	125	6	6	126	–	–	2.42
me	lh	5	125	6	6	126	–	–	2.09
se	ll	1,5	125	6	6	126	53	32*	1.78
me	ll	1,5	125	6	6	126	53	32*	1.52
se	la	1,5	125	6	6	126	53	32*	1.5
me	la	1,5	125	6	6	126	53	32*	1.8
se	lh	1,5	125	6	6	126	–	–	2.55
me	lh	1,5	125	6	6	126	–	–	2.44
se	ll	2,5	125	6	6	126	–	–	2.27
me	ll	2,5	125	6	6	126	–	–	2.76
se	la	2,5	125	6	6	126	–	–	2.49
me	la	2,5	125	6	6	126	–	–	2.37
se	lh	2,5	125	6	6	126	–	–	2.62
me	lh	2,5	125	6	6	126	–	–	2.48
se	ll	3,5	125	6	6	126	53	32*	1.94
me	ll	3,5	125	6	6	126	53	32*	1.97
se	la	3,5	125	6	6	126	53	32*	1.82
me	la	3,5	125	6	6	126	53	32*	2.18
se	lh	3,5	125	6	6	126	–	–	2.46
me	lh	3,5	125	6	6	126	–	–	2.52
se	ll	4,5	125	6	6	126	–	–	2.87
me	ll	4,5	125	6	6	126	–	–	2.8
se	la	4,5	125	6	6	126	–	–	2.71
me	la	4,5	125	6	6	126	–	–	2.72
se	lh	4,5	125	6	6	126	–	–	2.95
me	lh	4,5	125	6	6	126	–	–	2.8
se	ll	1,3,5	125	6	6	126	53	32*	1.83
					from ODE		minimized		
edges	loops	k's	t	∂_t	n	∂_n	n	∂_n	time

Table A.1: Models up to $k = 7$ (continued).

edges	loops	k's	t	∂_t	n from ODE	∂_n	n minimized	∂_n	time
me	ll	1,3,5	125	6	6	126	53	32*	1.8
se	la	1,3,5	125	6	6	126	53	32*	1.85
me	la	1,3,5	125	6	6	126	53	32*	1.78
se	lh	1,3,5	125	6	6	126	–	–	2.55
me	lh	1,3,5	125	6	6	126	–	–	2.73
se	ll	1,2,3,4,5	125	6	6	126	–	–	3.17
me	ll	1,2,3,4,5	125	6	6	126	–	–	2.98
se	la	1,2,3,4,5	125	6	6	126	–	–	2.96
me	la	1,2,3,4,5	125	6	6	126	–	–	2.83
se	lh	1,2,3,4,5	125	6	6	126	–	–	3.03
me	lh	1,2,3,4,5	125	6	6	126	–	–	2.88
se	ll	6	145	6	6	146	116	36	52.3
me	ll	6	145	6	6	146	116	36	49.4
se	la	6	145	6	6	146	116	36	52.6
me	la	6	145	6	6	146	116	36	49.5
se	lh	6	425	10	10	426	–	–	250
me	lh	6	425	10	10	426	–	–	265
se	ll	1,6	417	10	10	418	–	–	182
me	ll	1,6	417	10	10	418	–	–	170
se	la	1,6	417	10	10	418	–	–	186
me	la	1,6	417	10	10	418	–	–	186
se	lh	1,6	425	10	10	426	–	–	265
me	lh	1,6	425	10	10	426	–	–	233
se	ll	2,6	145	6	6	146	116	36	55
me	ll	2,6	145	6	6	146	116	36	58
se	la	2,6	145	6	6	146	116	36	55.4
me	la	2,6	145	6	6	146	116	36	51.6
se	lh	2,6	425	10	10	426	–	–	264
me	lh	2,6	425	10	10	426	–	–	261
se	ll	3,6	423	10	10	424	–	–	262
me	ll	3,6	423	10	10	424	–	–	277
se	la	3,6	423	10	10	424	–	–	259
me	la	3,6	423	10	10	424	–	–	298
se	lh	3,6	425	10	10	426	–	–	276
me	lh	3,6	425	10	10	426	–	–	268
se	ll	4,6	145	6	6	146	116	36	63.5
me	ll	4,6	145	6	6	146	116	36	67.7
edges	loops	k's	t	∂_t	n from ODE	∂_n	n minimized	∂_n	time

Table A.1: Models up to $k = 7$ (continued).

edges	loops	k's	t	∂_t	n from ODE	∂_n from ODE	n minimized	∂_n minimized	time
se	la	4,6	145	6	6	146	116	36	60.6
me	la	4,6	145	6	6	146	116	36	65.4
se	lh	4,6	425	10	10	426	–	–	308
me	lh	4,6	425	10	10	426	–	–	311
se	ll	5,6	425	10	10	426	–	–	378
me	ll	5,6	425	10	10	426	–	–	315
se	la	5,6	425	10	10	426	–	–	361
me	la	5,6	425	10	10	426	–	–	326
se	lh	5,6	425	10	10	426	–	–	344
me	lh	5,6	425	10	10	426	–	–	302
se	ll	2,4,6	145	6	6	146	116	36	73.9
me	ll	2,4,6	145	6	6	146	116	36	68.9
se	la	2,4,6	145	6	6	146	116	36	59.2
me	la	2,4,6	145	6	6	146	116	36	69.3
se	lh	2,4,6	425	10	10	426	–	–	293
me	lh	2,4,6	425	10	10	426	–	–	300
se	ll	1,2,3,4,5,6	425	10	10	426	–	–	447
me	ll	1,2,3,4,5,6	425	10	10	426	–	–	402
se	la	1,2,3,4,5,6	425	10	10	426	–	–	387
me	la	1,2,3,4,5,6	425	10	10	426	–	–	509
se	lh	1,2,3,4,5,6	425	10	10	426	–	–	397
me	lh	1,2,3,4,5,6	425	10	10	426	–	–	547
se	ll	7	1683	20	20	1684	–	–	3.22e+04
me	ll	7	1683	20	20	1684	–	–	2.66e+04
se	la	7	1683	20	20	1684	–	–	5.16e+04
me	la	7	1683	20	20	1684	–	–	2.02e+04
se	lh	7	1683	20	20	1684	–	–	3.46e+04
me	lh	7	1683	20	20	1684	–	–	3.06e+04
se	ll	1,7	1683	20	20	1684	–	–	3.11e+04
me	ll	1,7	1683	20	20	1684	–	–	1.65e+04
se	la	1,7	1683	20	20	1684	–	–	1.82e+04
me	la	1,7	1683	20	20	1684	–	–	5.93e+04
se	lh	1,7	1683	20	20	1684	–	–	5.15e+04
me	lh	1,7	1683	20	20	1684	–	–	4.11e+04
se	ll	2,7	1683	20	20	1684	–	–	7.68e+04
me	ll	2,7	1683	20	20	1684	–	–	4.39e+04
se	la	2,7	1683	20	20	1684	–	–	4.43e+04
edges	loops	k's	t	∂_t	from ODE n	from ODE ∂_n	minimized n	minimized ∂_n	time

Table A.1: Models up to $k = 7$ (continued).

edges	loops	k's	t	∂_t	n	∂_n	n	∂_n	time
					from ODE		minimized		
me	la	2,7	1683	20	20	1684	–	–	3.22e+04
se	lh	2,7	1683	20	20	1684	–	–	4.17e+04
me	lh	2,7	1683	20	20	1684	–	–	5.04e+04
se	ll	3,7	1683	20	20	1684	–	–	9.72e+04
me	ll	3,7	1683	20	20	1684	–	–	5.89e+04
se	la	3,7	1683	20	20	1684	–	–	1.46e+04
me	la	3,7	1683	20	20	1684	–	–	1.38e+05
se	lh	3,7	1683	20	20	1684	–	–	3.73e+04
me	lh	3,7	1683	20	20	1684	–	–	3.80e+04
se	ll	4,7	1683	20	20	1684	–	–	3.65e+04
me	ll	4,7	1683	20	20	1684	–	–	3.50e+04
se	la	4,7	1683	20	20	1684	–	–	6.28e+04
me	la	4,7	1683	20	20	1684	–	–	3.75e+04
se	lh	4,7	1683	20	20	1684	–	–	3.69e+04
me	lh	4,7	1683	20	20	1684	–	–	3.61e+04
se	ll	5,7	1683	20	20	1684	–	–	3.48e+04
me	ll	5,7	1683	20	20	1684	–	–	1.09e+05
se	la	5,7	1683	20	20	1684	–	–	4.17e+04
me	la	5,7	1683	20	20	1684	–	–	4.87e+04
se	lh	5,7	1683	20	20	1684	–	–	4.39e+04
me	lh	5,7	1683	20	20	1684	–	–	4.61e+04
se	ll	6,7	1683	20	20	1684	–	–	4.67e+04
me	ll	6,7	1683	20	20	1684	–	–	4.60e+04
se	la	6,7	1683	20	20	1684	–	–	4.97e+04
me	la	6,7	1683	20	20	1684	–	–	4.76e+04
se	lh	6,7	1683	20	20	1684	–	–	4.10e+04
me	lh	6,7	1683	20	20	1684	–	–	5.31e+04
se	ll	1,3,5,7	1683	20	20	1684	–	–	2.63e+04
me	ll	1,3,5,7	1683	20	20	1684	–	–	8.89e+04
se	la	1,3,5,7	1683	20	20	1684	–	–	1.96e+04
me	la	1,3,5,7	1683	20	20	1684	–	–	2.44e+04
se	lh	1,3,5,7	1683	20	20	1684	–	–	3.90e+04
me	lh	1,3,5,7	1683	20	20	1684	–	–	3.89e+04
se	ll	1,2,3,4,5,6,7	1683	20	20	1684	–	–	4.67e+04
me	ll	1,2,3,4,5,6,7	1683	20	20	1684	–	–	4.99e+04
se	la	1,2,3,4,5,6,7	1683	20	20	1684	–	–	3.86e+04
me	la	1,2,3,4,5,6,7	1683	20	20	1684	–	–	4.95e+04
					from ODE		minimized		
edges	loops	k's	t	∂_t	n	∂_n	n	∂_n	time

Table A.1: Models up to $k = 7$ (continued).

edges	loops	k's	t	∂_t	n	∂_n	n	∂_n	time
					from ODE		minimized		
se	lh	1,2,3,4,5,6,7	1683	20	20	1684	–	–	5.53e+04
me	lh	1,2,3,4,5,6,7	1683	20	20	1684	–	–	6.12e+04

B. Improving asymptotic expansions

Table B.1: Correcting series in the asymptotic expansion of k-regular graphs.

k	series $(1+O(1/n))$ of Eq. (7.1), truncated to $O(1/n^{16})$
3	$1-\frac{35}{18n}-\frac{71}{648n^2}+\frac{58321}{34992n^3}+\frac{15251617}{2519424n^4}+\frac{4309623397}{226748160n^5}+\frac{251934087845}{4897760256n^6}+\frac{253134308612953}{3085588961280n^7}-\frac{20819936902809929}{63474972917760n^8}-\frac{3498126053079270135 23}{71980619288739840n^9}-\frac{4966616413200899569795 91}{12956511471973171200n^{10}}-\frac{25402259045023537484296823}{102615570858027515904n^{11}}-\frac{765439965993629439125774442719}{554124082633348585881600n^{12}}-\frac{808591743016375102248254949968651}{12966503533620356909629440 0n^{13}}-\frac{91243941039613574788563711002461723}{6535117780944659882453237760n^{14}}+\frac{1872585435336039675152016642956847466 39}{1260344143467898691615981568000n^{15}}$
4	$1-\frac{39}{8n}+\frac{495}{128n^2}+\frac{7113}{1024n^3}+\frac{449163}{32768n^4}+\frac{58542459}{1310720n^5}+\frac{3986508663}{20971520n^6}+\frac{201917774607}{234881024n^7}+\frac{286433371481337}{75161927680n^8}+\frac{9350023543331001}{601295421440n^9}+\frac{2384550175342571199}{48103633715200n^{10}}+\frac{3118232293077332049}{384829069721600n^{11}}-\frac{509749028387325028838 7}{2462906046218240n^{12}}-\frac{536788760332741410071640 9}{182958734861926400n^{13}}-\frac{35352640637223387441191913 33}{1103381908705771520 0n^{14}}-\frac{18430368708843963889928080108383}{5737585925270011904000n^{15}}$
5	$1-\frac{49}{5n}+\frac{1313}{50n^2}+\frac{2113}{625n^3}-\frac{554701}{25000n^4}-\frac{2996369}{125000n^5}+\frac{2273542681}{18750000n^6}+\frac{392452265731}{328125000n^7}+\frac{10282316794379}{1250000000n^8}+\frac{11850077562825427}{218750000000n^9}+\frac{2370921840124552067}{6562500000000n^{10}}+\frac{638737383696843789 67}{25781250000000n^{11}}+\frac{1904233060863531204510223}{108281250000000000n^{12}}+\frac{1304699803912488799196971 97}{1005468750000000000n^{13}}+\frac{4917460839323474756360169420 49}{492679687500000000000n^{14}}+\frac{2712110449773582528752921021299}{338378906250000000000n^{15}}$
6	$1-\frac{155}{9n}+\frac{130325}{1296n^2}-\frac{6315175}{34992n^3}-\frac{1594380125}{10077696n^4}+\frac{16940128895}{90699264n^5}+\frac{35036225450575}{39182082048n^6}+\frac{6169580836439575}{2468471169024n^7}+\frac{13307325761511531125}{1421839393357824n^8}+\frac{653823253160409195654 5}{115168990861983744n^9}+\frac{7199141815878166186779725}{16584334684125659136n^{10}}+\frac{5849667349614600574859755625}{1641849133728440254464n^{11}}+\frac{42938926555417928885579383793875}{1418557651541372379856896n^{12}}+\frac{44026150734767380680022143165683635}{165971245230340568443256832n^{13}}+\frac{4014589190031631530121826312397721826 75}{16729901519218329299080288665 6n^{14}}+\frac{7785959540317401992319706762929278608675}{347467185399149916211667533824n^{15}}$