Using symbolic computation
to prove nonexistence of distance-regular graphs

Janoš Vidali*
Faculty of Mathematics and Physics
University of Ljubljana, 1000 Ljubljana, Slovenia
janos.vidali@fmf.uni-lj.si

July 29, 2021
Mathematics Subject Classification: 05E30
© Janoš Vidali. Released under the CC BY license (International 4.0).

Abstract
A package for the Sage computer algebra system is developed for checking feasibility of a given intersection array for a distance-regular graph. We use this tool to show that there is no distance-regular graph with intersection array
\[
\{(2r+1)(4r+1)(4t-1), 8r(4rt - r + 2t), (r + t)(4r + 1); \\
1, (r + t)(4r + 1), 4r(2r + 1)(4t - 1)\} \quad (r, t \geq 1),
\]
\[
\{135,128,16; 1,16,120\}, \{234,165,12; 1,30,198\} \text{ or } \{55,54,50,35,10; 1,5,20,45,55\}. \text{ In all cases, the proofs rely on equality in the Krein condition, from which triple intersection numbers are determined. Further combinatorial arguments are then used to derive nonexistence.}

Keywords: distance-regular graphs, Krein parameters, triple intersection numbers, nonexistence, symbolic computation

1 Introduction
Distance-regular graphs were introduced around 1970 by N. Biggs [4]. As they are intimately linked to many other combinatorial objects, such as finite simple groups, finite geometries, and codes, a natural goal is trying to classify them.

Many distance-regular graphs are known, however constructing new ones has proved to be a difficult task. A number of feasibility conditions for distance-regular graphs have been found, which allows us to compile a list of feasible intersection arrays for small distance-regular graphs (or related structures, such as $Q$-polynomial association schemes),

---

*This work is supported in part by the Slovenian Research Agency (research program P1-0285).
see Brouwer et al. [8, 9, 10] and Williford [63]. However, feasibility is no guarantee for existence, so proofs of nonexistence of distance-regular graphs with feasible intersection arrays are also a contribution to the classification. In certain cases, single intersection arrays have been ruled out [40, 43], while other proofs may show nonexistence for a whole infinite family of feasible intersection arrays [18, 37, 58]. In this paper we give proofs of nonexistence for distance-regular graphs belonging to a two-parameter infinite family, as well as for graphs with intersection arrays

\{135, 128, 16; 1, 16, 120\},
\{234, 165, 12; 1, 30, 198\},
\{55, 54, 50, 35, 10; 1, 5, 20, 45, 55\}.

We develop a package called sage-drg [60] for the Sage computer algebra system [54]. Sage is free open-source software written in the Python programming language [53], with many functionalities deriving from other free open-source software, such as Maxima [48], which Sage uses for symbolic computation. The sage-drg package is thus also free open-source software available under the MIT license, written in the Python programming language, making use of the Sage library. The package can be used to check for feasibility of a given intersection array against known feasibility conditions, written in the Python programming language, and up to the latest available version as of 2019. Furthermore, using equality in the Krein condition (see Theorem 1), restrictions on triple intersection numbers can be derived. In this paper, we use them to derive some nonexistence results. The sage-drg package also includes Jupyter notebooks demonstrating its use to obtain these results, as well as the notebook jupyter/Demo.ipynb giving some general examples of use of the package. A more detailed description of the sage-drg package is given in Appendix A.

The results from Sections 3, 4 and 6 appeared in the author’s PhD thesis [59], where computation was done using a Mathematica [64] notebook originally developed by M. Urlep. Thus, the sage-drg package can be seen as a move from closed-source proprietary software to free open-source software, which allows one to check all code for correctness, thus making the results verifiable.

## 2 Preliminaries

In this section we review some basic definitions and concepts. See Brouwer, Cohen and Neumaier [9] for further details.

Let \( \Gamma \) be a connected graph with diameter \( d \) and \( n \) vertices, and let \( \partial(u, v) \) denote the distance between the vertices \( u \) and \( v \) of \( \Gamma \). The graph \( \Gamma \) is distance-regular if there exist constants \( p_{ij}^h \) (\( 0 \leq h, i, j \leq d \)), called the intersection numbers, such that for any pair of vertices \( u, v \) at distance \( h \), there are precisely \( p_{ij}^h \) vertices at distances \( i \) and \( j \) from \( u \) and \( v \), respectively. In fact, all intersection numbers can be computed given only the intersection numbers \( b_i = p_{1,i+1}^i \) and \( c_{i+1} = p_{i+1,i}^i \) (\( 0 \leq i \leq d-1 \)) [9, §4.1A]. These intersection numbers are usually gathered in the intersection array \( \{b_0, b_1, \ldots, b_{d-1}; c_1, c_2, \ldots, c_d\} \). We also define the valency \( k = b_0 \) and \( a_i = k - b_i - c_i \) (\( 0 \leq i \leq d \)), where \( b_d = c_0 = 0 \). A connected
noncomplete strongly regular graph with parameters \((v, k, \lambda, \mu)\) is a distance-regular graph of diameter 2 with \(v\) vertices, valency \(k\) and intersection numbers \(a_1 = \lambda, c_2 = \mu\).

Let \(A_i (0 \leq i \leq d)\) be a binary square matrix indexed with the vertices of a graph \(\Gamma\) of diameter \(d\), with entry corresponding to vertices \(u\) and \(v\) equal to 1 precisely when \(\partial(u, v) = i\). The matrix \(A = A_1\) is the adjacency matrix of \(\Gamma\). The graph \(\Gamma\) is called primitive if all \(A_i (1 \leq i \leq d)\) are adjacency matrices of connected graphs. A distance-regular graph of valency \(k \geq 3\) that is not primitive is bipartite or antipodal (or both) \([9, \text{Thm. 4.2.1}]\). The spectrum of \(\Gamma\) is defined to be the spectrum of a \(k\)-regular graph of valency \(d\) with \(\partial\).

Suppose that \(\Gamma\) is distance-regular. Let \(M\) be the Bose-Mesner algebra, i.e., the algebra generated by \(A\). The matrices \(\{A_i\}_{i=0}^d\) form a basis of \(M\), which also has a second basis \(\{E_i\}_{i=0}^d\) consisting of projectors to the eigenspaces of \(A\) \([9, \S 2.2]\). Note that the indexing in this second basis depends on the ordering of eigenvalues. The descending ordering of eigenvalues is known as the natural ordering. We define the eigenmatrix \(P\) and dual eigenmatrix \(Q\) as \((d + 1) \times (d + 1)\) matrices such that \(A_j = \sum_{i=0}^d P_{ij} E_i\) and \(E_j = n^{-1} \sum_{i=0}^d Q_{ij} A_i\). The graph \(\Gamma\) is called formally self-dual \([9, \text{p. 49}]\) if \(P = Q\) holds for some ordering of eigenvalues. The graph \(\Gamma\) is called \(Q\)-polynomial \([9, \S 2.7]\) with respect to some ordering of eigenvalues if there exist real numbers \(z_0, \ldots, z_d\) and polynomials \(q_j\) of degree \(j\) such that \(Q_{ij} = q_j(z_i) (0 \leq i, j \leq d)\). Finally, we define the Krein parameters \(q_{ij}^h\) \([9, \S 2.3]\) as such numbers that \(E_i \circ E_j = n^{-1} \sum_{h=0}^d q_{ij}^h E_h\), where \(\circ\) represents entrywise multiplication of matrices. A formally self-dual distance-regular graph is also \(Q\)-polynomial with respect to the corresponding ordering of eigenvalues and has \(p_{ij}^h = q_{ij}^h\) \((0 \leq i, j, h \leq d)\). In this paper, we will use the natural ordering for indexing, noting when a graph is \(Q\)-polynomial or formally self-dual for some other ordering.

For vertices \(u, v, w\) of the distance-regular graph \(\Gamma\) and integers \(i, j, h (0 \leq i, j, h \leq d)\) we denote by \([u \ v \ w \ i \ j \ h]\) (or simply \([i \ j \ h]\) when it is clear which triple \((u, v, w)\) we have in mind) the number of vertices of \(\Gamma\) that are at distances \(i, j, h\) from \(u, v, w\), respectively. We call these numbers triple intersection numbers. They have first been studied in the case of strongly regular graphs \([15]\), and later also for distance-regular graphs, see for example \([18, 36, 37, 38, 58]\). Unlike the intersection numbers, these numbers may depend on the particular choice of vertices \(u, v, w\) and not only on their pairwise distances. We may however write down a system of \(3d^2\) linear Diophantine equations with \(d^3\) triple intersection numbers as variables, thus relating them to the intersection numbers, cf. \([37]\):

\[
\sum_{\ell=1}^d [\ell \ j \ h] = p_{j h}^\ell - [0 \ j \ h], \quad \sum_{\ell=1}^d [i \ \ell \ h] = p_{i h}^\ell - [i \ 0 \ h], \quad \sum_{\ell=1}^d [i \ j \ \ell] = p_{j}^\ell - [i \ j \ 0], \quad (1)
\]

where \(U = \partial(v, w), V = \partial(u, w), W = \partial(u, v),\) and

\[
[0 \ j \ h] = \delta_{j W} \delta_{h V}, \quad [i \ 0 \ h] = \delta_{i W} \delta_{h U}, \quad [i \ j \ 0] = \delta_{i V} \delta_{j U}.
\]

Furthermore, we can use the triangle inequality to conclude that certain triple intersection numbers must be zero. Moreover, the following theorem sometimes gives additional equations.
Theorem 1. ([18, Theorem 3], cf. [9, Theorem 2.3.2]) Let $\Gamma$ be a distance-regular graph with diameter $d$, dual eigenmatrix $Q$ and Krein parameters $q_{ij}^h$ $(0 \leq i,j,h \leq d)$. Then,

$$q_{ij}^h = 0 \iff \sum_{r,s,t=0}^{d} Q_{ri}Q_{sj}Q_{th}\begin{bmatrix} u & v & w \\ r & s & t \end{bmatrix} = 0 \text{ for all } u,v,w \in V\Gamma.$$ 

Together with integrality and nonnegativity of triple intersection numbers, we can use all of the above to either derive that the system of equations has no solution, or arrive at a small number of solutions, which gives us new information on the structure of the graph and may lead to proving its nonexistence.

3 A two-parameter family of primitive graphs of diameter 3

In [37], graphs meeting necessary conditions for the existence of extremal codes were studied. One of the families of primitive graphs of diameter 3 for which these conditions were met was

$$\{ a(p+1), (a+1)p, c; 1, c, ap \},$$

where $a = a_3$, $c = c_2$ and $p = p_3^3$. Graphs belonging to this family are $Q$-polynomial with respect to the natural ordering of eigenvalues precisely when the Krein parameter $q_{11}^3$ is zero, which is equivalent to

$$c = \frac{1}{4} \left( (p+1)^2 + \frac{2a(p+1)}{p+2} \right).$$

Hence, $p + 2$ must divide $2a$ for $c$ to be integral. If $p = 2r - 1$, then $a = t(2r + 1)$ and $c = r(r + t)$ for some positive integers $r, t$, which gives us the two-parameter family

$$\{ 2rt(2r+1), (2r-1)(2rt+t+1), r(r+t); 1, r(r+t), t(4r^2-1) \}.$$

In [37], nonexistence was shown for a feasible subfamily with $r = t \geq 2$. If, on the other hand, $p$ is even, integrality of the multiplicity of the second largest eigenvalue implies that we must have $p = 4r$, $a = (2r+1)(4t-1)$ and $c = (r+t)(4r+1)$ for some positive integers $r, t$, giving the family

$$\{ (2r+1)(4r+1)(4t-1), 8r(4rt-r+2t), (r+t)(4r+1); 1, (r+t)(4r+1), 4r(2r+1)(4t-1) \}.$$ 

We find two one-parameter infinite subfamilies of feasible intersection arrays by setting $t = 4r^2$ or $t = 4r^2 + 2r$:

$$\{ (2r+1)(4r+1)(16r^2-1), 8r^2(16r^2+8r-1), r(4r+1)^2; 1, r(4r+1)^2, 4r(2r+1)(16r^2-1) \},$$

$$\{ (2r+1)(4r+1)(16r^2+8r-1), 8r^2(4r+1)(4r+3), r(4r+1)(4r+3); 1, r(4r+1)(4r+3), 4r(2r+1)(16r^2+8r-1) \}.$$
There are also other feasible cases – for instance, when \( r = 2 \), we have, besides the cases from the two subfamilies above, feasible examples when \( t \in \{4, 7, 196\} \). The case with \( r = 1 \) and \( t = 4 \) belonging to the first subfamily above is also listed in the list of feasible parameter sets for 3-class \( Q \)-polynomial association schemes by J. S. Williford [63].

We now prove that a graph \( \Delta \) with intersection array (4) does not exist. The proof parallels that of [37, Lems. 1, 3] – in fact, a significant part of the proof may be extended to the entire family (3), as it has been done in [59]. The computation needed to obtain the results in this section is illustrated in the \texttt{jupyter/DRG-d3-2param.ipynb} notebook included in the \texttt{sage-drg} package [60].

**Lemma 2.** Let \( \Delta \) be a distance-regular graph with intersection array (4), and \( u', v, w \) be vertices of \( \Delta \) with \( \partial(u', v) = 1 \), \( \partial(u', w) = 2 \) and \( \partial(v, w) = 3 \). Then \( \begin{bmatrix} u' & v & w \\ 3 & 3 & 3 \end{bmatrix} = 1 \).

**Proof.** Let \( u \) be a vertex of \( \Delta \) at distance 3 from both \( v \) and \( w \) (such a vertex exists since \( p_{33}^3 = 4r > 0 \)). We consider the triple intersection numbers \( [i \ j \ h] \) that correspond to \( (u, v, w) \). As \( q_{11}^3 = q_{13}^3 = q_{31}^3 = 0 \), Theorem 1 gives three additional equations to the system (1), allowing us to express its solution in terms of a single parameter \( \alpha = [3 \ 3 \ 3] \).

Let us express the counts of vertices at distance 1 or 2 from one of \( u, v, w \) and at distance 3 from the other two vertices:

\[
\begin{align*}
[3 \ 3 \ 1] &= [3 \ 1 \ 3] = [1 \ 3 \ 3] = \frac{(\alpha - 4r + 1)(4r + 1)}{4r - 1}, \\
[3 \ 3 \ 2] &= [3 \ 2 \ 3] = [2 \ 3 \ 3] = \frac{8r(4r - 1 - \alpha)}{4r - 1}.
\end{align*}
\]

For the values above to be nonnegative, we must have \( \alpha = 4r - 1 \), which means that they are all zero. As the choice of \( u, v, w \) was arbitrary, this implies that any pair of vertices at distance 3 induces a set of \( 4r + 2 \) vertices pairwise at distance 3 – in the terminology of [37], this is a maximal 1-code in \( \Delta \). Since we have \( a_{33}^3p_{33}^3 = 4r(2r + 1)(4t - 1) = c_3 \), it follows by [37, Prop. 2] that \( \begin{bmatrix} u' & v & w \\ 3 & 3 & 3 \end{bmatrix} = 1 \) holds.

**Theorem 3.** A distance-regular graph \( \Delta \) with intersection array (4) does not exist.

**Proof.** Let \( u', v, w \) be vertices of \( \Delta \) with \( \partial(u', v) = 1 \), \( \partial(u', w) = 2 \) and \( \partial(v, w) = 3 \) (such vertices exist, since we have \( p_{13}^2 = b_2 = (r + t)(4r + 1) > 0 \)). We consider the triple intersection numbers \( [i \ j \ h] \) that correspond to \( (u', v, w) \). By Lemma 2, we have \( [3 \ 3 \ 3] = 1 \). Using \( q_{11}^3 = 0 \), Theorem 1 gives an additional equation which allows us to obtain a unique solution to the system (1). However, we obtain \( [1 \ 1 \ 3] = 2t - 1/2 \), which is nonintegral for all integers \( t \). Therefore, the graph \( \Delta \) does not exist.

### 4 A primitive graph with diameter 3 and 1360 vertices

Let \( \Lambda \) be a distance-regular graph with intersection array

\[
\{135, 128, 16; 1, 16, 120\}. \tag{5}
\]
This intersection array can be obtained from (2) by setting \(a = 15, c = 16\) and \(p = 8\). The graph \(\Lambda\) has diameter 3 and 1360 vertices. It is not \(Q\)-polynomial, however its Krein parameter \(q_{33}\) is zero. We show that such a graph does not exist. The computation needed to prove Theorem 4 is illustrated in the \texttt{jupyter/DRG-135-128-16-1-16-120.ipynb} notebook included in the \texttt{sage-drg} package [60].

**Theorem 4.** A distance-regular graph \(\Lambda\) with intersection array (5) does not exist.

**Proof.** Let \(u, v, w\) be three pairwise adjacent vertices of \(\Lambda\) (such vertices exist, since we have \(p_{11}^1 = 6 > 0\)). We consider triple intersection numbers \([i j h]\) that correspond to \((u, v, w)\). As \(q_{33}^3 = 0\), Theorem 1 gives an additional equation to the system (1), allowing us to express its solution in terms of a single parameter \(\alpha = [1 \ 1 \ 1]\). In particular, we obtain

\[
[3 \ 3 \ 3] = \frac{71 - 27\alpha}{8}.
\]

Clearly, \(\alpha\) must be a nonnegative integer. For \([3 \ 3 \ 3]\) to be nonnegative, we must have \(\alpha \in \{0, 1, 2\}\). However, \([3 \ 3 \ 3]\) is still nonintegral in these cases, showing that the graph \(\Lambda\) does not exist. \(\square\)

5 A primitive graph with diameter 3 and 1600 vertices

Let \(\Xi\) be a distance-regular graph with intersection array

\[
\{234, 165, 12; 1, 30, 198\}.
\]

The graph \(\Xi\) has diameter 3 and 1600 vertices. The intersection array (6) has been found as an example of a feasible parameter set for a distance-regular graph which is formally self-dual for an ordering of eigenvalues distinct from the natural ordering – in fact, \(\Xi\) is \(Q\)-polynomial for the ordering 0, 2, 3, 1, so its Krein parameters \(q_{12}^1, q_{12}^2\) and \(q_{21}^2\) are zero. The intersection array (6) is also listed in the list of feasible parameter sets for 3-class \(Q\)-polynomial association schemes by J. S. Williford [63]. We show that such a graph does not exist. The computation needed to prove Theorem 5 is illustrated in the \texttt{jupyter/DRG-234-165-12-1-30-198.ipynb} notebook included in the \texttt{sage-drg} package [60].

**Theorem 5.** A distance-regular graph \(\Xi\) with intersection array (6) does not exist.

**Proof.** Let \(u, v, w\) be three vertices of \(\Xi\) that are pairwise at distance 3 (such vertices exist, since we have \(p_{33}^3 = 8 > 0\)). We consider triple intersection numbers \([i j h]\) that correspond to \((u, v, w)\). As \(q_{22}^1 = q_{12}^2 = q_{21}^2 = 0\), Theorem 1 gives three additional equations to the system (1), allowing us to express its solution in terms of a single parameter \(\alpha = [3 \ 3 \ 3]\). In particular, we obtain

\[
[3 \ 3 \ 2] = [3 \ 2 \ 3] = [2 \ 3 \ 3] = -17 - 4\alpha.
\]

Clearly, \(\alpha\) must be nonnegative, but then we have \([3 \ 3 \ 2] = [3 \ 2 \ 3] = [2 \ 3 \ 3] < 0\), a contradiction. We conclude that the graph \(\Xi\) does not exist. \(\square\)
Figure 1: The partition of vertices of $\Sigma$ by distance from a pair of vertices $u, v$ at distance 2. The part that is at distance $i$ from $u$ and distance $j$ from $v$ has size $p_{ij}^2$. As the graph is bipartite, the intersection number $p_{ij}^2$ is nonzero only when $i + j$ is even. Moreover, there are no edges within each part. It is natural to consider $[1 1 1]$ for $w$ at distance 2 from both $u$ and $v$, see Lemma 6.

6 A bipartite graph with diameter 5

Let $\Sigma$ be a distance-regular graph with intersection array

$$\{55, 54, 50, 35, 10; 1, 5, 20, 45, 55\}. \quad (7)$$

This intersection array appears in the list of feasible intersection arrays for bipartite non-antipodal distance-regular graphs of diameter 5 by Brouwer et al. [9, p. 418] as an open case. The existence of such a graph would give a counterexample to a conjecture by MacLean and Terwilliger [45], cf. Lang [44]. The computation needed to obtain the results in this section is illustrated in the jupyter/DRG-55-54-50-35-10-bipartite.ipynb notebook included in the sage-drg package [60].

The graph $\Sigma$ has diameter 5 and 3500 vertices. The partition of $\Sigma$ corresponding to two vertices at distance 2 is shown in Figure 1. The graph is $Q$-polynomial for the natural ordering of eigenvalues, see for example [9, p. 418]. Moreover, as the graph is bipartite, it is also $Q$-antipodal [9, Thm. 8.2.1]. Many Krein parameters are zero, in particular $q_{11}^3$ and $q_{11}^4$ due to the triangle inequality. We use this fact in the proof of the following statement.

**Lemma 6.** Let $\Sigma$ be a distance-regular graph with intersection array (7), and $u, v, w$ be vertices of $\Sigma$ that are pairwise at distance 2. Then $[u \, v \, w \, 1 \, 1 \, 1] \leq 1$.

**Proof.** We consider the triple intersection numbers $[i \, j \, h]$ that correspond to $(u, v, w)$. Since the graph $\Sigma$ is bipartite, we have $[i \, j \, h] = 0$ whenever any of the sums $i + j$, $j + h$, $h + i$ is odd. As $q_{11}^3 = q_{11}^4 = 0$, Theorem 1 gives us two additional equations to the system (1), thus allowing us to express the solution of the system in terms of a single parameter $\alpha = [1 \, 1 \, 1]$. In particular, we obtain

$$[5 \, 5 \, 5] = 20 - 12\alpha.$$

The integrality and nonnegativity of $[5 \, 5 \, 5]$ now gives $\alpha \leq \lfloor 5/3 \rfloor = 1$. \qed

7
Note. It can also be shown with a method similar to the one used in Lemma 6 that the graph $[\Sigma_5(u)]_2$ for a vertex $u \in V \Sigma$ (i.e., the graph of vertices at distance 5 from a vertex $u$, with adjacency corresponding to distance 2 in $\Sigma$) is strongly regular with parameters $(v, k, \lambda, \mu) = (210, 99, 48, 45)$. A strongly regular graph with such parameters has been constructed by M. Klin [39].

Theorem 7. A distance-regular graph $\Sigma$ with intersection array (7) does not exist.

Proof. Let $u$ and $v$ be vertices of $\Sigma$ at distance 2, see Figure 1, and let $\{i j\}$ denote the set of vertices at distances $i$ and $j$ from $u$ and $v$, respectively. There are $p_{11}^2(k - 2) = 5 \cdot 53 = 265$ edges between the sets $\{1 1\}$ and $\{2 2\}$. However, the cardinality of the latter set is $p_{22}^2 = 243 < 265$, so there is a vertex $w \in \{2 2\}$ that has at least two neighbours in $\{1 1\}$, i.e., $[u \ v \ w]_{1 \ 1 \ 1} \geq 2$, which is in contradiction with Lemma 6. Hence, the graph $\Sigma$ does not exist.

Acknowledgements

I would like to thank Michael Lang for bringing the intersection array (7) to my attention and for noticing some bugs and proposing new functionality for the sage-drg package.

Appendix A Description of the sage-drg package

A.1 Installation

The sage-drg package [60] can be installed by cloning the git repository or extracting the ZIP file and making sure that Sage sees the drg directory (e.g., by starting it from the package’s root directory, or by copying or linking the drg directory into the package library used by the copy of Python in Sage’s installation directory). Once Sage is run, the package can be imported.

```
sage: import drg
```

The central class is `drg.DRGParameters`, which can be given an intersection array in the form of two lists or tuples of the same length.

```
sage: syl = drg.DRGParameters([5, 4, 2], [1, 1, 4])
sage: syl
Parameters of a distance-regular graph with intersection array
{5, 4, 2; 1, 1, 4}
```

Instead of an intersection array, parameters $(k, \lambda, \mu)$ for a strongly regular graph or classical parameters $(d, b, \alpha, \beta)$ (see [9, §6]) may also be specified.
sage: petersen = drg.DRGParameters(3, 0, 1)
sage: petersen
Parameters of a distance–regular graph with intersection array \{3, 2; 1, 1\}
sage: q7 = drg.DRGParameters(7, 1, 0, 1)
sage: q7
Parameters of a distance–regular graph with intersection array
\{7, 6, 5, 4, 3, 2, 1; 1, 2, 3, 4, 5, 6, 7\}

The intersection array (given in any of the forms above) may also contain variables. Substitution of variables is possible using the \texttt{subs} method. Note that the diameter must be constant.

sage: r = var("r")
sage: fam = drg.DRGParameters(2*\r^2*(2*\r+1), (2*\r-1)*(2*\r^2+\r+1), 2*\r^2, [1, 2*\r^2, \r*(4*\r^2-1)])
sage: fam1 = fam.subs(r == 1)
sage: fam1
Parameters of a distance–regular graph with intersection array
\{6, 4, 2; 1, 2, 3\}

### A.2 Parameter computation

As a \texttt{drg.DRGParameters} object is being constructed, its intersection numbers are computed. If any of them is determined to be negative or nonintegral,\(^1\) then an exception is thrown and the construction is aborted. Several other conditions are also checked, for example that the sequences \(\{b_i\}_{i=0}^d\) and \(\{c_i\}_{i=0}^d\) are non-ascending and non-descending, respectively, and that \(b_j \geq c_i\) if \(i + j \leq d\) [9, Prop. 4.1.6(ii)]. The handshake lemma is also checked for each subconstituent [9, Lem. 4.3.1]. If the graph is determined to be antipodal, then the covering index is also checked for integrality.

The number of vertices, their valency and the diameter of the graph can be obtained with the \texttt{order}, \texttt{valency}, and \texttt{diameter} methods.

sage: syl.order()
36
sage: syl.valency()
5
sage: syl.diameter()
3

\(^1\)Non-constant expressions are also checked for nonnegativity – if there is no assignment of real numbers to the variables such that the value of the expression is nonnegative, then the expression is marked as negative and cannot appear as, e.g., an intersection number.
The entire array of intersection numbers can be obtained with the `pTable` method, which returns a `drg.Array3D` object implementing a three-dimensional array.

```
sage: syl.pTable()
0: 
[ 1  0  0  0]
[ 0  5  0  0]
[ 0  0 20  0]
[ 0  0  0 10]
1: 
[ 0  1  0  0]
[ 0  4  0  0]
[ 0  8  8  0]
[ 0  8  2  0]
2: 
[ 0  0  1  0]
[ 0  1  2  2]
[ 0 11  6  0]
[ 0  2  2  2]
3: 
[ 0  0  0  1]
[ 0  1  4  1]
[ 0  4 12  4]
[ 0  4  4  4]
```

The subsets of intersection numbers \( \{a_i\}_{d=1}^{i}, \{b_i\}_{d=0}^{i-1}, \{c_i\}_{d=1}^{i}, \) and \( \{k_i\}_{d=0}^{i} \) (where \( k_i = P_{0i}^d \) is the number of vertices at distance \( i \) from any vertex of the graph) can be obtained as tuples with the `aTable`, `bTable`, `cTable`, and `kTable` methods, respectively. There is also a method `intersectionArray` returning the entire intersection array as a pair of tuples.

```
sage: syl.aTable()
(0, 2, 1)
sage: syl.bTable()
(5, 4, 2)
sage: syl.cTable()
(1, 1, 4)
sage: syl.kTable()
(1, 5, 20, 10)
sage: syl.intersectionArray()
((5, 4, 2), (1, 1, 4))
```

Eigenvalues can be computed using the `eigenvalues` method.

```
sage: syl.eigenvalues()
(5, 2, -1, -3)
```

Eigenvalues are sorted in the decreasing order; if there is a variable in the intersection array, then the order is derived under the assumption that the variable takes a (large
enough) positive value. If there is more than one variable, a warning is issued that the
given ordering is not necessarily correct. This can be avoided by explicitly specifying an
order of the variables using the \texttt{set.vars} method (see the \texttt{jupyter/DRG-d3-2param.ipynb}
notebook for an example). The ordering of eigenvalues (and thus the ordering of cor-
responding parameters) can also be changed later using the \texttt{reorderEigenvalues} method,
which accepts an ordering of the indices of the nontrivial eigenvalues (i.e., integers from
1 to \(d\)).

Once the ordering of eigenvalues is determined, the cosine sequences and multiplicities
of the eigenvalues can be computed using the \texttt{cosineSequences} and \texttt{multiplicities} methods.
The multiplicities are checked to be integral.

\begin{verbatim}
\texttt{sage: syl.cosineSequences()}
\begin{bmatrix}
 1 & 1 & 1 & 1 \\
 1 & 2/5 & -1/20 & -1/5 \\
 1 & -1/5 & -1/5 & 2/5 \\
 1 & -3/5 & 1/5 & -1/5
\end{bmatrix}
\texttt{sage: syl.multiplicities()}
(1, 16, 10, 9)
\end{verbatim}

The eigenmatrix and dual eigenmatrix can be computed using the \texttt{eigenmatrix} and
dualEigenmatrix methods. The \texttt{is_formallySelfDual} method checks whether the two matri-
ces are equal (using the current ordering of eigenvalues).

\begin{verbatim}
\texttt{sage: syl.eigenmatrix()}
\begin{bmatrix}
 1 & 5 & 20 & 10 \\
 1 & 2 & -1 & -2 \\
 1 & -1 & -4 & 4 \\
 1 & -3 & 4 & -2
\end{bmatrix}
\texttt{sage: syl.dualEigenmatrix()}
\begin{bmatrix}
 1 & 16 & 10 & 9 \\
 1 & 32/5 & -2 & -27/5 \\
 1 & -4/5 & -2 & 9/5 \\
 1 & -16/5 & 4 & -9/5
\end{bmatrix}
\texttt{sage: syl.is_formallySelfDual()}
False
\end{verbatim}

The Krein parameters can be computed using the \texttt{kreinParameters} method, which
returns a \texttt{drg.Array3D} object. The Krein parameters are checked to be nonnegative.

\begin{verbatim}
\texttt{sage: syl.kreinParameters()}
0: \begin{bmatrix}
 1 & 0 & 0 & 0 \\
 0 & 16 & 0 & 0 \\
 0 & 0 & 10 & 0 \\
 0 & 0 & 0 & 9
\end{bmatrix}
1: \begin{bmatrix}
 0 & 1 & 0 & 0
\end{bmatrix}
\end{verbatim}
Classical parameters can be computed using the is_classical method, which returns a list of all tuples of classical parameters, or False if the graph is not classical.

\[
\text{sage: fam1.is_classical()}
\]
\[
[(3, 1, 0, 2)]
\]

The method genPoly_parameters returns the tuple \((g, s, t)\) if the parameters correspond to those of a collinearity graph of a generalized \(g\)-gon of order \((s, t)\), or \((False, None, None)\) if there is no such generalized \(g\)-gon. See \[57\] and \[9, \S 6.5\] for definitions of generalized polygons and some results.

\[
\text{sage: drg.DRGParameters([6, 4, 4], [1, 1, 3]).genPoly_parameters()}
\]
\[
(6, 2, 2)
\]

Note that the existence of a strongly regular graph for which \((g, s, t)\) are defined does not imply the existence of a corresponding generalized quadrangle. A distance-regular graph \(\Gamma\) of diameter at least 3 has these parameters defined precisely when \(\Gamma\) is isomorphic to the collinearity graph of a corresponding generalized \(g\)-gon.

All the methods mentioned above store their results, so subsequent calls will not redo the computations. Note that one does not need to call the methods in the order given here – if some required computation has not been done before, it will be performed when needed. Where applicable, the methods above also take three named boolean parameters expand, factor, and simplify (all set to False by default), which control how the returned expression(s) will be manipulated. In the case when there are no variables in use, setting these parameters has no effect.

### A.3 Parameters of derived graphs

In some cases, the parameters of a distance-regular graph imply the existence of another distance-regular graph which can be derived from the original graph. This is true for imprimitive graphs (i.e., antipodal or bipartite), but sometimes, new distance-regular graphs can be obtained by taking subgraphs or by merging classes.
The antipodality of a graph can be checked with the `is_antipodal` method, which returns the covering index for antipodal graphs, and `False` otherwise. The parameters of the antipodal quotient of an antipodal graph can then be obtained with the `antipodalQuotient` method.

```
sage: q7.is_antipodal()
2
sage: q7.antipodalQuotient()
Parameters of a distance–regular graph with intersection array
{7, 6, 5; 1, 2, 3}
```

The bipartiteness of a graph can be checked with the `is_bipartite` method. The parameters of the bipartite half of a bipartite graph can then be obtained with the `bipartiteHalf` method.

```
sage: q7.is_bipartite()
True
sage: q7.bipartiteHalf()
Parameters of a distance–regular graph with intersection array
{21, 10, 3; 1, 6, 15}
```

In some cases, distance-regularity of the local graph can be established (for instance, for tight graphs, see [36]). In these cases, the parameters of the local graph can then be obtained with the `localGraph` method.

```
sage: drg.DRGParameters([27, 10, 1], [1, 10, 27]).localGraph()
Parameters of a distance–regular graph with intersection array
{16, 5; 1, 8}
```

Similarly, the distance-regularity of a subconstituent (i.e., a graph induced by vertices at a given distance from a vertex) can be established in certain cases. Their parameters can be obtained using the `subconstituent` method. Usually, distance-regularity is derived from triple intersection numbers (see Subsection A.5), which are not computed by default. To force this computation, the parameter `compute` can be set to `True`.

```
sage: drg.DRGParameters([204, 175, 48, 1], [1, 12, 175, 204]).subconstituent(2, compute = True)
Parameters of a distance–regular graph with intersection array
{144, 125, 32, 1; 1, 8, 125, 144}
```

Note that calling `localGraph()` is equivalent to calling `subconstituent(1)`. The `localGraph` method also accepts the `compute` parameter.

The complement of a strongly regular graph is also strongly regular. If the complement is connected, its parameters can be obtained with the `complementaryGraph` method.

```
sage: petersen.complementaryGraph()
Parameters of a distance–regular graph with intersection array
{6, 2; 1, 4}
```
Sometimes, merging classes of the underlying association scheme yields a new distance-regular graph. Its parameters (or the parameters of a connected component if the resulting graph is disconnected) can be obtained with the `mergeClasses` method, which takes the indices of classes which will be merged into the first class of the new scheme (i.e., the distances in the original graph which will correspond to adjacency in the new graph).

```
sage: q7.mergeClasses(2, 3, 6)
Parameters of a distance-regular graph with intersection array
{63, 30, 1; 1, 30, 63}
```

Note that `mergeClasses(2)` gives the parameters of the bipartite half for bipartite graphs, and of the complement for non-antipodal strongly regular graphs.

A dictionary mapping the merged indices to parameters of a new graphs for all possibilities can be obtained using the `distanceGraphs` method.

```
sage: q7.distanceGraphs()
{(1, 2): Parameters of a distance-regular graph with intersection array
{28, 15, 6, 1; 1, 6, 15, 28},
(1, 2, 3, 4, 5, 6): Parameters of a distance-regular graph
with intersection array {126, 1; 1, 126},
(1, 3, 5): Parameters of a distance-regular graph with intersection array
{63, 62, 1; 1, 62, 63},
(1, 3, 5, 7): Parameters of a distance-regular graph
with intersection array {64, 63; 1, 64},
(1, 4, 5): Parameters of a distance-regular graph with intersection array
{63, 32, 1; 1, 32, 63},
(1, 5): Parameters of a distance-regular graph with intersection array
{28, 27, 16; 1, 12, 28},
(1, 7): Parameters of a distance-regular graph with intersection array
{8, 7, 6, 5; 1, 2, 3, 8},
(2,): Parameters of a distance-regular graph with intersection array
{21, 10, 3; 1, 6, 15},
(2, 3, 6): Parameters of a distance-regular graph with intersection array
{63, 30, 1; 1, 30, 63},
(2, 4, 6): Parameters of a distance-regular graph with intersection array
{63; 1},
(2, 6): Parameters of a distance-regular graph with intersection array
{28, 15; 1, 12},
(3, 7): Parameters of a distance-regular graph with intersection array
{36, 35, 16; 1, 20, 36},
(4,): Parameters of a distance-regular graph with intersection array
{35, 16; 1, 20},
(6,): Parameters of a distance-regular graph with intersection array
{7, 6, 5; 1, 2, 3},
(7,): Parameters of a distance-regular graph with intersection array
{1; 1}}
```
A.4 Feasibility checking

To check whether a given parameter set is feasible, the check_feasible method may be called. This method calls other check_∗ methods which perform the actual checks. Selected checks may also be skipped by providing a parameter skip with a list of strings identifying checks to be skipped.

sporadic. The check_sporadic method checks whether the intersection array matches one from a list of intersection arrays for which nonexistence of a corresponding graph has been proven, but does not belong to any infinite family. If so, the parameter set is reported as infeasible. Currently, the list includes:

- \{14, 12; 1, 4\}, cf. Wilbrink and Brouwer [62],
- \{16, 12; 1, 6\}, cf. Bussemaker et al. [14],
- \{21, 18; 1, 7\}, cf. Haemers [31],
- \{30, 21; 1, 14\}, cf. Bondarenko, Prymak and Radchenko [6],
- \{32, 21; 1, 16\}, cf. Azarija and Marc [2],
- \{38, 27; 1, 18\}, cf. Degraer [22],
- \{40, 27; 1, 20\}, cf. Azarija and Marc [1],
- \{57, 56; 1, 12\}, cf. Gavrilyuk and Makhnev [26],
- \{67, 56; 1, 2\}, cf. Brouwer and Neumaier [12],
- \{116, 115; 1, 20\}, cf. Makhnev [47],
- \{153, 120; 1, 60\}, cf. Bondarenko et al. [5],
- \{165, 128; 1, 66\}, cf. Makhnev [46],
- \{486, 320; 1, 243\}, cf. Makhnev [46],
- \{5, 4, 3; 1, 1, 2\}, cf. Fon-Der-Flaass [24],
- \{11, 10, 10; 1, 1, 11\} (projective plane of order 10), cf. Lam, Thiel and Swiercz [42],
- \{13, 10, 7; 1, 2, 7\}, cf. Coolsaet [16],
- \{18, 12, 1; 1, 2, 18\} (generalized quadrangle of order (6,3) minus a spread), cf. [9, Prop. 12.5.2] and [52, 6.2.2],
- \{20, 10, 10; 1, 1, 2\} (projective plane of order 10), cf. Lam, Thiel and Swiercz [42],
- \{21, 16, 8; 1, 4, 14\}, cf. Coolsaet [17],
- \{22, 16, 5; 1, 2, 20\}, cf. Sumalroj and Worawannotai [56],
- \{27, 20, 10; 1, 2, 18\}, cf. Brouwer, Sumalroj and Worawannotai [13],
- \{36, 28, 4; 1, 2, 24\}, cf. Brouwer, Sumalroj and Worawannotai [13],
- \{39, 24, 1; 1, 4, 39\}, cf. Bang, Gavrilyuk and Koolen [3],
- \{45, 30, 7; 1, 2, 27\}, cf. Gavrilyuk and Makhnev [28],
family. The check_family method checks whether the intersection array matches one from a list of infinite families of intersection arrays for which nonexistence of corresponding graphs has been proven. If so, the parameter set is reported as infeasible. Currently, the list includes:

- \(\{52, 35, 16; 1, 4, 28\}\), cf. Gavrilyuk and Makhnev [27],
- \(\{55, 36, 11; 1, 4, 45\}\), cf. Gavrilyuk [25],
- \(\{56, 36, 9; 1, 3, 48\}\), cf. Gavrilyuk [25],
- \(\{69, 48, 24; 1, 4, 46\}\), cf. Gavrilyuk and Makhnev [27],
- \(\{74, 54, 15; 1, 9, 60\}\), cf. Coosaeet and Jurišić [18],
- \(\{105, 102, 99; 1, 2, 35\}\), cf. De Bruyn and Vanhove [21],
- \(\{130, 96, 18; 1, 12, 117\}\), cf. Jurišić and Vidali [38],
- \(\{135, 128, 16; 1, 16, 120\}\), see Theorem 4,
- \(\{234, 165, 12; 1, 30, 198\}\), see Theorem 5,
- \(\{4818, 4248, 192; 1, 72, 4672\}\), cf. Jurišić and Vidali [38],
- \(\{5928, 5920, 5888; 1, 5, 741\}\), cf. De Bruyn and Vanhove [21],
- \(\{120939612, 120939520, 120933632; 1, 65, 1314561\}\), cf. De Bruyn and Vanhove [21],
- \(\{9757175, 97571080, 97569275; 1, 20, 1027065\}\), cf. De Bruyn and Vanhove [21],
- \(\{290116365, 290116260, 290100825; 1, 148, 2763013\}\), cf. De Bruyn and Vanhove [21],
- \(\{5, 4, 3, 3; 1, 1, 1, 2\}\), cf. Fon-Der-Flaass [23],
- \(\{10, 9, 1, 1; 1, 9, 10\}\), cf. [9, Prop. 11.4.5],
- \(\{32, 27, 6, 1; 1, 6, 27, 32\}\), cf. Soicher [55],
- \(\{32, 27, 9, 1; 1, 3, 27, 32\}\), cf. Soicher [55],
- \(\{56, 45, 20, 1; 1, 4, 45, 56\}\), cf. [9, Prop. 11.4.5],
- \(\{55, 54, 50, 35, 10; 1, 5, 20, 45, 55\}\), see Theorem 7, and
- \(\{15, 14, 12, 6, 1; 1, 1, 3, 12, 14, 15\}\), cf. Ivanov and Shpectorov [33].

The check_family method checks whether the intersection array matches one from a list of infinite families of intersection arrays for which nonexistence of corresponding graphs has been proven. If so, the parameter set is reported as infeasible. Currently, the list includes:

- \(\{r^2(r + 3), (r + 1)(r^2 + 2r - 2); 1, r(r + 1)\}\) with \(r \geq 3, r \neq 4\), cf. Bondarenko and Radchenko [7],
- \(\{(2r^2 - 1)(2r + 1), 4r(r^2 - 1), 2r^2; 1, 2(r^2 - 1), r(4r^2 - 2)\}\) with \(r \geq 2\), cf. Jurišić and Vidali [37],
- \(\{2r^2(2r + 1), (2r - 1)(2r^2 + r + 1), 2r^2; 1, 2r^2, r(4r^2 - 1)\}\) with \(r \geq 2\), cf. Jurišić and Vidali [37],
- \(\{4r^3 + 8r^2 + 6r + 1, 2r(r + 1)(2r + 1), 2r^2 + 2r + 1; 1, 2r(r + 1), (2r + 1)(2r^2 + 2r + 1)\}\) with \(r \geq 1\), cf. Coosaeet and Jurišić [18],
• \{ (2r+1)(4r+1)(4t-1), 8r(4rt-r+2t), (r+t)(4r+1), 4r(2r+1)(4t-1) \} with \( r, t \geq 1 \), see Theorem 3,

• \{ (r+1)(r^3-1), r(r-1)(r^2+r-1), r^2-1; 1, r(r+1), (r^2-1)(r^2+r-1) \} with \( r \geq 3 \), cf. Urlep [58],

• \{ r^2(rt+t+1), (r^2-1)(rt+1), r(r-1)(t+1), 1; 1, r(t+1), (r^2-1)(rt+1), r^2(rt+t+1) \} with \( r \geq 3 \) and \((r, t) \neq (3, 1), (3, 3), (4, 2), \) cf. Jurišić and Koolen [35], and

• \{ 2r^2 + r, 2r^2 + r - 1, r^2, r, 1; 1, r, r^2, 2r^2 + r - 1, 2r^2 + r \} with \( r \geq 2 \), cf. Coolsaet, Jurišić and Koolen [19].

\textbf{2graph.} The \texttt{check_2graph} method checks conditions related to two-graphs, cf. [9, Thm. 1.5.6]. For strongly regular graphs with parameters \((v, k, \lambda, \mu)\), for which \( v = 2(2k - \lambda - \mu) \) holds, it records the parameters \((v - 1, 2(k - \mu), k + \lambda - 2\mu, k - \mu)\) of a strongly regular graph to be checked for feasibility later. For Taylor graphs (i.e., antipodal double covers of diameter 3) for which \( a_1 > 0 \) holds, it checks whether \( a_1 \) is even and whether the number of vertices \( n \) is a multiple of 4, and then records the parameters \((k, a_1, (3a_1 - k - 1)/2, a_1/2)\) of a strongly regular graph as those of the local graph to be checked for feasibility later, cf. [9, Thm. 1.5.3].

\textbf{classical.} The \texttt{check_classical} method checks whether any of the classical parameters for the parameter set match some from a list of infinite families of classical parameters for which nonexistence of corresponding graphs has been proven. If so, the parameter set is reported as infeasible. Currently, the list only includes two sets of classical parameters:

• \((d, b, \alpha, \beta) = (d, -2, -2, ((-2)^{d+1}-1)/3)\) with \( d \geq 4 \), cf. Huang, Pan and Weng [32], and

• \((d, b, \alpha, \beta) = (d, -r, -r/(r-1), r + r^2((-r)^{d-1} - 1)/(r^2 - 1))\) with \( d \geq 4, r \geq 2 \), cf. De Bruyn and Vanhove [21].

Additionally, the method checks whether nonexistence can be derived from one of the following characterizations:

• a characterization of Grassmann graphs by Metsch [49, Thm. 2.3],

• a characterization of bilinear forms graphs by Metsch [50, Prop. 2.2],

• a characterization of graphs with classical parameters \((d, b, \alpha, \beta)\) and \( d \geq 4, b < 0 \) by Weng [61, Thm. 10.3], and

• a characterization of graphs with classical parameters \((d, b, \alpha, \beta)\) and \( d \geq 3, a_1 = 0, a_2 > 0 \) by Pan and Weng [51, Thm. 2.1].

\textbf{combinatorial.} The \texttt{check_combinatiorial} method checks various combinatorial conditions:

• a graph with \( b_1 = 1 \) must be a cycle or a cocktail party graph,

• Godsil’s diameter bound [9, Lem. 5.3.1],

• a lower bound for \( c_3 \) [9, Thm. 5.4.1],

17
• a condition for \(b_1 = b_i\) [9, Prop. 5.4.4],

• a lower bound for \(a_1\) [9, Prop. 5.5.1],

• a handshake lemma for Pappus subgraphs, cf. Koolen [40]

• Turán’s theorem [9, Lem. 5.6.4],

• a counting argument by Lambeck [43],

• two lower bounds for the size of the last subconstituent [9, Props. 5.6.1, 5.6.3],

• handshake lemmas for the numbers of edges and triangles [9, Lem. 4.3.1], and

• a condition for \(p_{dd}^2 = 0\) [9, Prop. 5.7.1].

Additionally, the method checks whether a condition for the existence of cliques of size \(a_1 + 2\) is satisfied [9, Prop. 4.3.2] and performs additional checks:

• a divisibility check if the last subconstituent is a union of cliques [9, Prop. 4.3.2(ii)],

• a handshake lemma for maximal cliques [9, Prop. 4.3.3], and

• two inequalities from counting arguments [9, Prop. 4.3.3].

conference. For a strongly regular graph with parameters \((v,k,\lambda,\mu)\), where \(k = 2\mu\) and \(\mu = k − \lambda − 1\), the check conference method checks whether \(n \not\equiv 1 \pmod{4}\) and \(n\) is a sum of two squares, cf. [9, §1.3].

geodeticEmbedding. For a distance-regular graph with intersection array \(\{2b, b, 1; 1, 1, 2b\}\), the check geodeticEmbedding method checks whether \(b \leq 4\), cf. [9, Prop. 1.17.3].

2design. For a distance-regular graph with intersection array \(\{r\mu + 1, (r − 1)\mu, 1; 1, 1, \mu, r\mu + 1\}\), the check 2design method checks that a corresponding 2-design exists, cf. [9, Prop. 1.10.5].

hadamard. For a distance-regular graph with intersection array \(\{2\mu, 2\mu − 1, \mu, 1; 1, 1, 2\mu − 1, 2\mu\}\) with \(\mu > 1\), the check hadamard method checks whether \(\mu\) is even, i.e., whether a Hadamard matrix of order \(2\mu\) can exist, cf. [9, Cor. 1.8.2].

antipodal. For an antipodal cover of even diameter at least 4, the check antipodal method checks whether its quotient satisfies necessary conditions for the existence of a cover, cf. [9, Prop. 4.2.7].

genPoly. The check genPoly method checks conditions related to generalized polygons. First, it checks whether the conditions for the existence of cliques of size \(a_1 + 2\) have been checked, and calls check combinatorial otherwise to obtain this information. If the existence of cliques has been established and the intersection array matches that of a collinearity graph of a generalized \(g\)-gon with parameter \((s,t)\), then we know that we should indeed have such a graph, and the following conditions are checked:

• Feit-Higman theorem [9, Thm. 6.5.1],

• \((s + t)|st(s + 1)(t + 1)\) for generalized quadrangles [52, 1.2.2], and

• Bruck-Ryser theorem for thin generalized hexagons [9, Thm. 1.10.4].
An antipodal distance-regular $r$-cover of diameter 3 with $k = (r - 1)(c_2 + 1)$ such that the existence of cliques of size $a_1 + 2$ has been established corresponds to the collinearity graph of a generalized quadrangle of order $(s, t) = (r - 1, c_2 + 1)$ with a spread removed. Therefore, in this case, check_genPoly checks for feasibility of such a generalized quadrangle (first two conditions above), and also checks whether it can contain a spread, see [9, Prop. 12.5.2] and [52, 1.8.3].

clawBound. For a strongly regular graph, the check_clawBound method checks the claw bound, i.e., whether the graph must be the point graph of an infeasible partial geometry, cf. Brouwer and Van Lint [11].

terwilliger. The check_terwilliger method checks conditions related to Terwilliger graphs and induced quadrangles, cf. [9, §1.16]. First, it checks the coclique bound by Koolen and Park [41, Thm. 3]; if it is met with equality, it checks whether the graph can be a Terwilliger graph [9, Cor. 1.16.6]. If the parameters imply that the graph contains an induced quadrangle, then Terwilliger’s diameter bound is checked [9, Thm. 5.2.1].

secondEigenvalue. For a distance-regular graph with an eigenvalue equal to $b_1 - 1$, the check_secondEigenvalue method checks whether the graph belongs to the classification given in [9, Thm. 4.4.11].

localEigenvalue. The check_localEigenvalue method checks conditions related to the eigenvalues of the local graph. For a distance-regular graph of diameter 3 that is not the dodecahedron, the following conditions are checked:

- general bounds for the second largest and smallest eigenvalue of the local graph [9, Thm. 4.4.3],
- a bound on eigenvalues of the local graph of a non-bipartite graph by Jurišić and Koolen [34],
- the fundamental bound by Jurišić, Koolen and Terwilliger [36], and
- bounds on multiplicities of eigenvalues, see [9, Thm. 4.4.4], [29] and [30].

If equality is met in the bounds of the second or third point above, then the local graph is determined to be strongly regular and is thus stored as such to be checked for feasibility later.

absoluteBound. The check_absoluteBound method checks the absolute bound on the multiplicities of eigenvalues [9, Thm. 2.3.3].

After running all the checks described above, the check_feasible method calls itself on all already derived graphs (antipodal quotient, bipartite half, complement, 2-graph derivation, subconstituents where applicable), and then also on each parameter set for distance-regular graphs obtained by merging classes. To avoid repetitions, a list of checked intersection arrays is maintained. This step can be skipped by setting the derived parameter to False.

If the parameter set is feasible (i.e., it passes all checks), then check_feasible returns without error. Otherwise, a drg.InfeasibleError exception is thrown indicating the reason for nonexistence and providing a reference.
Details on the given references are available in the `drg.references` submodule.

```python
sage: import drg.references
sage: drg.references.refs["GavrilyukMakhnev05"]
{
    'authors': [('Gavrilyuk', ('Alexander', 'L.')), ('Makhnev', ('Alexander', 'Alexeevich'))],
    'fjournal': 'Doklady Akademii Nauk',
    'journal': 'Dokl. Akad. Nauk',
    'number': 6,
    'pages': (727, 730),
    'title': 'Krein graphs without triangles',
    'type': 'article',
    'volume': 403,
    'year': 2005}
```

Details on the nonexistence may also be extracted from the exception.

```python
sage: try:
    ....:     drg.DRGParameters([65, 44, 11], [1, 4, 55]).check_feasible()
    ....: except drg.InfeasibleError as ex:
    ....:     print("Part: %s" % (ex.part , ))
    ....:     print("Reason: %s" % ex.reason)
    ....:     for r, thm in ex.refs:
    ....:         ref = drg.references.refs[r]
    ....:         print("Authors: %s" % ref["authors"])
    ....:         print("Title: %s" % ref["title"])
    ....:         print("Theorem: %s" % thm)
    ....:
Part: ()
Reason: coclique bound exceeded
Authors: [(["Koolen", ('Jack', 'H.'))], ["Park", ('Jongyook',))]}
Title: Shilla distance-regular graphs
Theorem: Thm. 3.
```

## A.5 Partitions and triple intersection numbers

For a given parameter set, the distance partition corresponding to a vertex can be obtained with the `distancePartition` method, which returns a graph representing the distance partition.
sage: dp = syl.distancePartition()
sage: dp
Distance partition of {5, 4, 2; 1, 1, 4}
sage: dp.show()

A distance partition corresponding to two vertices can be obtained by passing the
distance between them as an argument to **distancePartition**.

sage: syl.distancePartition(1).show()

Note that edges are shown between any pair of cells such that their distances from
either of the initial vertices differ by at most 1. To show all the distance partitions
Corresponding to at most two vertices, the **show_distancePartitions** method may be used
(see below).

For a given triple of distances \((U, V, W)\) such that \(p_{U,V}^W > 0\), the method **tripleEquations**
gives the solution to the system of equations (1) augmented by equations derived from
Theorem 1 for each triple \((i,j,h)\) \((1 \leq i,j,h \leq d)\) such that \(q_{ij}^h = 0\). The solution is
returned as a **drg.Array3D** object.

sage: syl.tripleEquations(1, 1, 2)
0: [[0 0 0 0]
   [0 1 0 0]
   [0 0 0 0]
   [0 0 0 0]]
1: [[0 0 1 0]
   [0 0 0 0]
   [1 0 3 0]
   [0 0 0 0]]
2: [[0 0 0 0]
   [0 0 2 2]
   [0 2 2 4]
   [0 2 4 2]]
3: [[0 0 0 0]]
If the solution is not unique, one or more parameters will be present in the solution.

\[
\begin{align*}
\text{sage: } & \text{syl.tripleEquations}(1, 2, 3) \\
0: & \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} \\
1: & \begin{bmatrix}
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 \\
0 & 1 & 2 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix} \\
2: & \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & r^2 & r^2 + 4 & -2r^2 + 4 \\
1 & -r^2 + 2 & -r^2 + 4 & 2r^2 + 1
\end{bmatrix} \\
3: & \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & -r^2 + 3 & -r^2 + 6 & 2r^2 - 1 \\
0 & r^2 - 1 & r^2 -2r^2 + 3
\end{bmatrix}
\end{align*}
\]

Parameters may also be set explicitly by passing a \texttt{params} argument with a dictionary mapping the name of the parameter to the triple of distances it represents.

\[
\text{sage: } \text{syl.tripleEquations}(1, 3, 3, \text{params} = \{"a": (3, 3, 3)\}) \\
0: \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} \\
1: \begin{bmatrix}
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 \\
0 & 4 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} \\
2: \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 4 & 0 \\
0 & -1/2a + 4 & -1/2a + 4 & a \\
0 & 1/2a & 1/2a + 4 & -a + 4
\end{bmatrix} \\
3: \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 1/2a & 1/2a + 4 & -a + 4 \\
1 & -1/2a + 1 & -1/2a & a
\end{bmatrix}
\]

If the appropriate triple intersection number is computed to be zero, an edge will not be shown in the distance partition.
References

[1] J. Azarija and T. Marc. There is no (95,40,12,20) strongly regular graph, 2016. arXiv:1603.02032.

[2] J. Azarija and T. Marc. There is no (75,32,10,16) strongly regular graph. Linear Algebra Appl., 557:62–83, 2018. doi:10.1016/j.laa.2018.07.019.

[3] S. Bang, A. L. Gavrilyuk, and J. H. Koolen. Distance-regular graphs without 4-claws. European J. Combin., 2018. doi:10.1016/j.ejc.2018.02.022.

[4] N. L. Biggs. Intersection matrices for linear graphs. In D. J. A. Welsh, editor, Combinatorial Mathematics and its applications, pages 15–23. Academic Press, London, 1971.

[5] A. V. Bondarenko, A. Mellit, A. V. Prymak, D. V. Radchenko, and M. S. Viazovska. There is no strongly regular graph with parameters (460, 153, 32, 60). In Contemporary computational mathematics – a celebration of the
80th birthday of Ian Sloan, volume 1–2, pages 131–134. Springer, Cham, 2018. doi:10.1007/978-3-319-72456-0_7.

[6] A. V. Bondarenko, A. V. Prymak, and D. V. Radchenko. Non-existence of (76, 30, 8, 14) strongly regular graph, 2017. doi:10.1016/j.laa.2017.03.033.

[7] A. V. Bondarenko and D. V. Radchenko. On a family of strongly regular graphs with \( \lambda = 1 \). J. Combin. Theory Ser. B, 103(4):521–531, 2013. doi:10.1016/j.jctb.2013.05.005.

[8] A. E. Brouwer. Parameters of distance-regular graphs, 2011. http://www.win.tue.nl/~aeb/drg/drgtables.html.

[9] A. E. Brouwer, A. M. Cohen, and A. Neumaier. Distance-regular graphs, volume 18 of Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]. Springer-Verlag, Berlin, 1989. doi:10.1007/978-3-642-74341-2.

[10] A. E. Brouwer, A. M. Cohen, and A. Neumaier. Corrections and additions to the book ‘Distance-regular graphs’, 1994. http://www.win.tue.nl/~aeb/drg/.

[11] A. E. Brouwer and J. H. van Lint. Strongly regular graphs and partial geometries. In Enumeration and design (Waterloo, Ont., 1982), pages 85–122. Academic Press, Toronto, 1984. https://ir.cwi.nl/pub/1817/1817A.pdf.

[12] A. E. Brouwer and A. Neumaier. Strongly regular graphs where \( \mu = 2 \) and \( \lambda \) is large. Afdeling Zuivere Wiskunde [Department of Pure Mathematics], Stichting Mathematisch Centrum, Amsterdam, 1981. https://ir.cwi.nl/pub/6792/6792A.pdf.

[13] A. E. Brouwer, S. Sumalroj, and C. Worawannotai. The nonexistence of distance-regular graphs with intersection arrays \( \{27, 20, 10; 1, 2, 18\} \) and \( \{36, 28, 4; 1, 2, 24\} \). Australas. J. Combin., 66:330–332, 2016. http://ajc.maths.uq.edu.au/pdf/66/ajc_v66_p330.pdf.

[14] F. C. Bussemaker, W. H. Haemers, R. A. Mathon, and H. A. Wilbrink. A (49, 16, 3, 6) strongly regular graph does not exist. European J. Combin., 10(5):413–418, 1989. doi:10.1016/S0195-6698(89)80014-9.

[15] P. J. Cameron, J.-M. Goethals, and J. J. Seidel. Strongly regular graphs having strongly regular subconstituents. J. Algebra, 55(2):257–280, 1978. doi:10.1016/0021-8693(78)90220-X.

[16] K. Coolsaet. Local structure of graphs with \( \lambda = \mu = 2, a_2 = 4 \). Combinatorica, 15(4):481–487, 1995. doi:10.1007/BF01192521.
[17] K. Coolsaet. A distance regular graph with intersection array $(21, 16, 8; 1, 4, 14)$ does not exist. European J. Combin., 26(5):709–716, 2005. doi:10.1016/j.ejc.2004.04.005.

[18] K. Coolsaet and A. Jurišić. Using equality in the Krein conditions to prove nonexistence of certain distance-regular graphs. J. Combin. Theory Ser. A, 115(6):1086–1095, 2008. doi:10.1016/j.jcta.2007.12.001.

[19] K. Coolsaet, A. Jurišić, and J. Koolen. On triangle-free distance-regular graphs with an eigenvalue multiplicity equal to the valency. European J. Combin., 29(5):1186–1199, 2008. doi:10.1016/j.ejc.2007.06.010.

[20] E. R. van Dam, J. H. Koolen, and H. Tanaka. Distance-regular graphs. Electron. J. Combin., DS:22, 2016. http://www.combinatorics.org/ojs/index.php/eljc/article/view/DS22.

[21] B. De Bruyn and F. Vanhove. On $Q$-polynomial regular near $2d$-gons. Combinatorica, 35(2):181–208, 2015. doi:10.1007/s00493-014-3039-x.

[22] J. Degraer. Isomorph-free exhaustive generation algorithms for association schemes. PhD thesis, Ghent University, 2007. https://cage.ugent.be/geometry/Theses/52/degraer-phd.pdf.

[23] D. G. Fon-Der-Flaass. A distance-regular graph with intersection array $(5, 4, 3, 3; 1, 1, 1, 2)$ does not exist. J. Algebraic Combin., 2(1):49–56, 1993. doi:10.1023/A:1022476614402.

[24] D. G. Fon-Der-Flaass. There exists no distance-regular graph with intersection array $(5, 4, 3; 1, 1, 2)$. European J. Combin., 14(5):409–412, 1993. doi:10.1006/eujc.1993.1045.

[25] A. L. Gavrilyuk. Distance-regular graphs with intersection arrays $\{55, 36, 11; 1, 4, 45\}$ and $\{56, 36, 9; 1, 3, 48\}$ do not exist. Dokl. Akad. Nauk, 439(1):14–17, 2011. doi:10.1134/S1064562411040028.

[26] A. L. Gavrilyuk and A. A. Makhnev. Krein graphs without triangles. Dokl. Akad. Nauk, 403(6):727–730, 2005.

[27] A. L. Gavrilyuk and A. A. Makhnev. Distance-regular graphs with intersection arrays $\{52, 35, 16; 1, 4, 28\}$ and $\{69, 48, 24; 1, 4, 46\}$ do not exist. Des. Codes Cryptogr., 65(1–2):49–54, 2012. doi:10.1007/s10623-012-9695-1.

[28] A. L. Gavrilyuk and A. A. Makhnev. A distance-regular graph with intersection array $\{45, 30, 7; 1, 2, 27\}$ does not exist. Diskret. Mat, 25(2):13–30, 2013.

[29] C. D. Godsil and A. D. Hensel. Distance regular covers of the complete graph. J. Combin. Theory Ser. B, 56(2):205–238, 1992. doi:10.1016/0095-8956(92)90019-T.
[30] C. D. Godsil and J. H. Koolen. On the multiplicity of eigenvalues of distance-regular graphs. Linear Algebra Appl., 226–228:273–275, 1995. doi:10.1016/0024-3795(95)00152-H.

[31] W. H. Haemers. There exists no (76, 21, 2, 7) strongly regular graph. In Finite geometry and combinatorics, London Mathematical Society Lecture Note Series, pages 175–176. Cambridge Univ. Press, Cambridge, 1993. doi:10.1017/CBO9780511526336.018.

[32] Y. Huang, Y. Pan, and C. Weng. Nonexistence of a class of distance-regular graphs. Electron. J. Combin., 22(2):2.37, 2015. http://www.combinatorics.org/ojs/index.php/eljc/article/view/v22i2p37.

[33] A. A. Ivanov and S. V. Shpectorov. The $P$-geometry for $M_{23}$ has no nontrivial 2-coverings. European J. Combin., 11(4):373–379, 1990. doi:10.1016/S0195-6698(13)80139-4.

[34] A. Jurišić and J. Koolen. Nonexistence of some antipodal distance-regular graphs of diameter four. European J. Combin., 21(8):1039–1046, 2000.

[35] A. Jurišić and J. Koolen. Classification of the family AT4(qs,q,q) of antipodal tight graphs. J. Combin. Theory Ser. A, 118(3):842–852, 2011.

[36] A. Jurišić, J. Koolen, and P. Terwilliger. Tight distance-regular graphs. J. Algebraic Combin., 12(2):163–197, 2000. doi:10.1023/A:1026544111089.

[37] A. Jurišić and J. Vidali. Extremal 1-codes in distance-regular graphs of diameter 3. Des. Codes Cryptogr., 65(1–2):29–47, 2012. doi:10.1007/s10623-012-9651-0.

[38] A. Jurišić and J. Vidali. Restrictions on classical distance-regular graphs. J. Algebraic Combin., 46(3–4):571–588, 2017. doi:10.1007/s10801-017-0765-3.

[39] M. Klin, C. Pech, S. Reichard, A. Woldar, and M. Ziv-Av. Examples of computer experimentation in algebraic combinatorics. Ars Math. Contemp., 3(2):238–248, 2010. http://amc-journal.eu/index.php/amc/article/view/119.

[40] J. H. Koolen. A new condition for distance-regular graphs. European J. Combin., 13(1):63–64, 1992. doi:10.1016/0195-6698(92)90068-B.

[41] J. H. Koolen and J. Park. Shilla distance-regular graphs. European J. Combin., 31(8):2064–2073, 2010. doi:10.1016/j.ejc.2010.05.012.

[42] C. W. H. Lam, L. Thiel, and S. Swiercz. The nonexistence of finite projective planes of order 10. Canad. J. Math., 41(6):1117–1123, 1989. doi:10.4153/CJM-1989-049-4.

[43] E. Lambeck. Some elementary inequalities for distance-regular graphs. European J. Combin., 14(1):53–54, 1993. doi:10.1006/eujc.1993.1008.
[44] M. S. Lang. Bipartite distance-regular graphs: the $Q$-polynomial property and pseudo primitive idempotents. *Discrete Math.*, 331:27–35, 2014. doi:10.1016/j.disc.2014.04.025.

[45] M. S. MacLean and P. Terwilliger. Taut distance-regular graphs and the subconstituent algebra. *Discrete Math.*, 306(15):1694–1721, 2006. doi:10.1016/j.disc.2006.03.046.

[46] A. A. Makhnev. On the nonexistence of strongly regular graphs with the parameters $(486, 165, 36, 66)$. *Ukrain. Mat. Zh.*, 54(7):941–949, 2002. doi:10.1023/A:1022066425998.

[47] A. A. Makhnev. The graph $Kre(4)$ does not exist. *Dokl. Math.*, 96(1):348–350, 2017. doi:10.1134/S1064562417040123.

[48] Maxima. Maxima, a computer algebra system. version 5.39.0, 2017. http://maxima.sourceforge.net/.

[49] K. Metsch. A characterization of Grassmann graphs. *European J. Combin.*, 16(6):639–644, 1995. doi:10.1016/0195-6698(95)90045-4.

[50] K. Metsch. On a characterization of bilinear forms graphs. *European J. Combin.*, 20(4):293–306, 1999. doi:10.1006/eujc.1998.0280.

[51] Y. Pan and C. Weng. A note on triangle-free distance-regular graphs with $a_2 \neq 0$. *J. Combin. Theory Ser. B*, 99(1):266–270, 2009. doi:10.1016/j.jctb.2008.07.005.

[52] S. E. Payne and J. A. Thas. *Finite generalized quadrangles*. EMS Series of Lectures in Mathematics. European Mathematical Society (EMS), Zürich, second edition, 2009. doi:10.4171/066.

[53] Python Software Foundation. *Python Language Reference, version 2.7.13*, 2017. http://www.python.org.

[54] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 7.6)*, 2017. http://www.sagemath.org.

[55] L. H. Soicher. The uniqueness of a distance-regular graph with intersection array $\{32,27,8,1;1,4,27,32\}$ and related results. *Des. Codes Cryptogr.*, 84(1–2):101–108, 2017. doi:10.1007/s10623-016-0223-6.

[56] S. Sumalroj and C. Worawannotai. The nonexistence of a distance-regular graph with intersection array $\{22,16,5;1,2,20\}$. *Electron. J. Combin.*, 23(1):1.32, 2016. http://www.combinatorics.org/ojs/index.php/eljc/article/view/v23i1p32.

[57] J. Tits. Sur la trialité et certains groupes qui s’en déduisent. *Inst. Hautes Études Sci. Publ. Math.*, (2):13–60, 1959. http://www.numdam.org/item?id=PMIHES_1959__2_13_0.

27
[58] M. Urlep. Triple intersection numbers of $Q$-polynomial distance-regular graphs. *European J. Combin.*, 33(6):1246–1252, 2012. doi:10.1016/j.ejc.2012.02.005.

[59] J. Vidali. *Codes in distance-regular graphs*. PhD thesis, University of Ljubljana, 2013. http://eprints.fri.uni-lj.si/2167/ (in Slovene).

[60] J. Vidali. jaanos/sage-drg: sage-drg Sage package v0.8, 2018. https://github.com/jaanos/sage-drg/, doi:10.5281/zenodo.1418410.

[61] C. Weng. Classical distance-regular graphs of negative type. *J. Combin. Theory Ser. B*, 76(1):93–116, 1999. doi:10.1006/jctb.1998.1892.

[62] H. A. Wilbrink and A. E. Brouwer. A $(57,14,1)$ strongly regular graph does not exist. *Nederl. Akad. Wetensch. Indag. Math.*, 45(1):117–121, 1983.

[63] J. S. Williford. Primitive 3-class Q-polynomial association schemes, 2017. Mirrored version available at https://jaanos.github.io/tables/qpoly/qprim3_table.html.

[64] Wolfram Research, Inc. *Mathematica, Version 8.0*. Champaign, Illinois, 2010. http://www.wolfram.com/mathematica/.