



**HAL**  
open science

## Drawing $K_n$ in Three Dimensions with One Bend per Edge, revisited

Olivier Devillers, Sylvain Lazard

► **To cite this version:**

Olivier Devillers, Sylvain Lazard. Drawing  $K_n$  in Three Dimensions with One Bend per Edge, revisited. 2023. <hal-04182069>

**HAL Id: hal-04182069**

**<https://inria.hal.science/hal-04182069v1>**

Preprint submitted on 16 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# Drawing $K_n$ in Three Dimensions with One Bend per Edge, revisited \*

Olivier Devillers      Sylvain Lazard

## Abstract

We present a drawing of  $K_n$  in three dimensions in which vertices are placed at integer grid points and edges are drawn crossing-free with at most one bend per edge; the bend points are also placed at integer grid points. The drawing is determined by an incremental algorithm and the observed behavior of the algorithm produces an output of volume close to quadratic.

## 1 Introduction

Although graphs are usually drawn in two dimensions, several authors have considered 3D-drawings. It is clear that if we place vertices in general positions and draw edges as line segments, then the edges are disjoint and crossing free drawings in this context is a trivial problem. The problem becomes interesting if vertices are constrained to the integer grid and we try to minimize the *volume* of the axis aligned bounding box of the drawing.<sup>1</sup>

Of course the 3D-drawing problem arises for any kind of graph and several classes of graphs has been considered. Such classes are defined by parameters or properties such as outerplanarity [10], planarity [1, 3], colorability [3, 13], stack-number, queue-number, track-number [7], or prescribed position for vertices [6, 9, 11, 14], In this paper we focus on the case of the complete graph  $K_n$ .

If the edges are constrained to be straight line segments, Cohen, Eades, Lin and Ruskey [4] proposed a crossing free drawing of the complete graph  $K_n$  within an optimal bounding box of volume  $\Theta(n^3)$ . Some papers consider the possibility of allowing some bends per edge. Bose, Czyzowicz, Morin, and Wood [2] showed that the number of edges is a lower bound on the volume of the bounding box, even in the presence of bends, establishing a quadratic lower bound on the volume of the complete graph. Dyck, Joevenazzo, Nickle, Wilsdon and Wismath [8] achieved such a quadratic construction with two bends per edge. If we restrict the number of bends to at most one per edge, Morin and Wood [12] presented a one-bend drawing of  $K_n$  with a volume

---

\*Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France.

Olivier.Devillers@inria.fr, Sylvain.Lazard@inria.fr

<sup>1</sup>To deal with integer bounding boxes, we define the volume of  $[x, x'] \times [y, y'] \times [z, z']$  where all these bounds are integers to be  $(x' - x + 1)(y' - y + 1)(z' - z + 1)$  so that the volume is exactly the number of grid points in (and on the boundary of) the box.

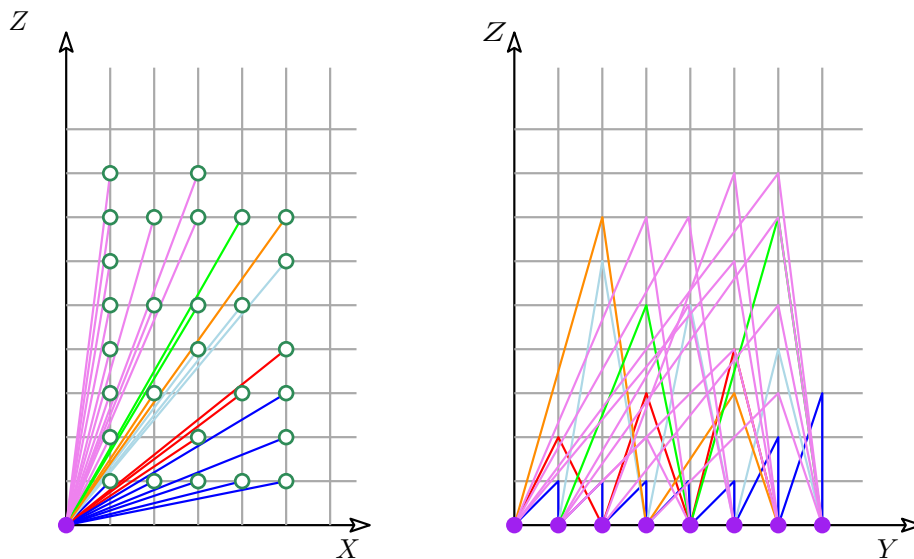


Figure 1: A collinear drawing of  $K_8$ .

of  $O(\frac{n^3}{\log^2 n})$ , later improved by Devillers, Everett, Lazard, Pentcheva, and Wismath to  $O(n^{2.5})$  [5].

In this paper, we analyze the techniques that yielded these results for the one-bend drawings and their limitation, and we propose an algorithm that bypass these limitations and provides good experimental results. The actual computation of  $K_n$  for  $n$  up to 100 shows a behavior close to quadratic and we conjecture that the actual volume of our construction is  $O(n^2 \log^2 n)$ .

## 2 Preliminaries

### 2.1 Collinear drawing of the complete graph

The first result proposed by Morin and Wood is about a *collinear drawing* of the graph, that is a drawing where all vertices lie on a line.

**Lemma 1** ([12, Lemmas 3 & 4]). *A one-bend collinear drawing of  $K_n$  with vertices on the  $Y$ -axis can be achieved within a box of size  $N_X \times n \times O(\frac{n^2}{N_X}) \forall N_X \in [1, n^2]$  of volume  $\Theta(n^3)$ .*

*Sketch.* Refer to Figure 1. We assume that the vertices are placed at coordinates  $(0, i, 0)$  for  $0 \leq i < N_Y$  with  $N_Y > n$ . Then in projection in the  $XZ$  plane, the two pieces of an edge look like a single line segment between the origin and a grid point: the projection of the bend.

— **Realisability.** A 2D grid point is said *visible* from the origin if its two cartesian coordinates are relatively prime. One can pick  $K$  grid points visible from the origin

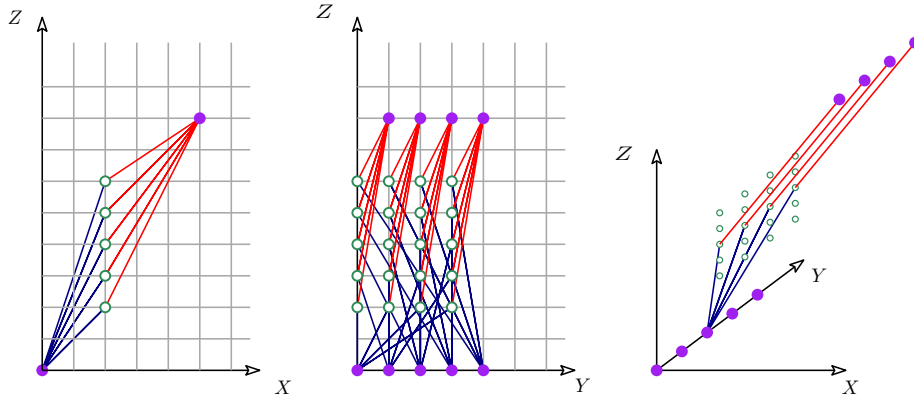


Figure 2: Drawing of  $K_{5,4}$ . Each edge has two segments, one from the vertex in the first set to its bend in blue, another from the bend to the vertex in the second set in red.

in the  $XZ$  plane with  $x \in [0, N_X]$  and  $z \in [0, N_Z]$  provided that  $N_X N_Z \geq \frac{2\pi^2}{3} K$  [12, Lemma 2]. Going to three dimensions, these grid points define “pages” around the  $Y$ -axis. Then, for each edge of the complete graph, we pick the bend in a separate page. There are  $\frac{n(n-1)}{2}$  such edges, thus the graph can be drawn in a box of size  $N_X \times N_Y \times N_Z$  of volume not bigger than  $\frac{\pi^2}{3} n^3$  if  $N_Y = n$ .

— **Lower bound.** Several edges can be drawn on the same page, but nevertheless a quadratic number of bends is necessary. Consider two edges from  $(0, i, 0)$  to  $(0, j, 0)$ , and from  $(0, k, 0)$  to  $(0, \ell, 0)$  such that  $0 \leq i, k < \frac{n}{2} < j, \ell < n$ . These two edges must have bends with different projections in  $XZ$ -plane. The lower bound follows since the number of such edges is quadratic.  $\square$

## 2.2 Bi-collinear drawing of the complete bi-partite graph

The construction proposed by Morin and Wood [12, §3] uses, as building blocks, drawings of complete bi-partite graphs with exactly one bend point per edge. These drawings are called one-bend bi-collinear drawings because the two sets of vertices of  $K_{n,m}$  are furthermore constrained to lie on two different lines parallel to the  $Y$ -axis.

**Lemma 2** ([12]). *A one-bend bi-collinear drawing of  $K_{n,m}$  can be achieved within  $\Theta(nm)$  volume.*

*Sketch.* The lower bound is trivial, since  $K_{n,m}$  has  $nm$  edges and thus must have  $nm$  bends [2]. Now refer to Figure 2. To realize the graph, we place  $n$  points at coordinates  $(0, i, 0)$  for  $0 \leq i < n$  and  $m$  points at coordinates  $(x, y + j, z)$  for  $0 \leq j < m$  and  $(x, y, z) \in \mathbb{Z}^3$ . Then, the bend point for the edge between  $(0, i, 0)$  and  $(x, y + j, z)$  is placed at  $(x', j, z' + i)$   $0 < x' < x$  and  $z' \in \mathbb{Z}$ .

By choosing suitable values, e.g.,  $x' = 1$ ,  $z' = 0$ ,  $(x, y, z) = (2, 0, 0)$ , the volume of this drawing is  $3nm$ . Notice that this construction is contained in two convex volumes, that we call *wedges*, which are the convex hulls of, respectively, the bends and the  $n$  vertices, and the bends and the  $m$  vertices.

Disjointness of pieces of edges is granted by considering the different planes parallel to the  $Y$ -axis they belong to (see Figure 2-right).  $\square$

**Remark 3.** Notice that this construction still works if we place the bends of the edges from  $(0, i, 0)$  to  $(x, y + j, z)$  at  $(x', y_{ij}, z' + i)$  (instead of  $(x', j, z' + i)$ ) where  $(y_{i,j})_j$  are sequences of integers that are strictly increasing with  $j$  (sequences may depend on  $i$ ).

## 2.3 Previous constructions

The idea of the Morin-Wood and the Devillers-Everett-Lazard-Pentcheva-Wismath constructions is to cut the set of  $n$  vertices in  $k$  groups of size  $\frac{n}{k}$ . To place the groups on some lines parallel to the  $Y$ -axis and draw for each group a drawing of its complete graph in a box using Lemma 1 and for each pair of groups a drawing of its complete bi-partite graph in two wedges using Lemma 2. These drawings are realized such that all the boxes and all the wedges are pairwise interior disjoint. The complete graphs are drawn below the vertices, the bi-partite graphs are drawn above and the number of groups is tuned to balance the heights of the constructions above and below.

### 2.3.1 The Morin-Wood construction.

In the Morin-Wood construction [12], the vertices are split in a logarithmic number of groups. The groups are placed in the  $XY$  plane such that (i) all vertices have different  $Y$ -coordinates, and (ii) vertices of the same group are at consecutive  $Y$ -coordinates and at the same (even)  $X$ -coordinate. The boxes for the complete graph of each group are below the  $XY$  plane, and the wedges are above it. They show that this construction is possible with a height of  $O(\frac{n^2}{\log^3 n})$  yielding a box of size  $[0, \log_4 n] \times [0, n] \times [-O(\frac{n^2}{\log^3 n}), O(\frac{n^2}{\log^3 n})]$  of volume  $O(\frac{n^3}{\log^2 n})$ .

### 2.3.2 The Devillers-Everett-Lazard-Pentcheva-Wismath construction.

In this construction [5], the number of groups is  $k = n^{\frac{1}{4}}$ . As in the previous construction the groups are placed with properties (i) and (ii) but at different  $Z$ -coordinates. Placing the groups of vertices on a parabola in the  $XZ$  plane allows to draw the bi-partite graphs using less height than in the previous construction. Namely the height is  $O(n^{\frac{5}{4}})$  giving a box of size  $[0, 2n^{\frac{1}{4}}] \times [0, n] \times [-O(n^{\frac{5}{4}}), O(n^{\frac{5}{4}})]$  of volume of  $O(n^{\frac{5}{2}})$ . Figure 3 presents such a drawing.

## 3 New construction: mixing the pencils

Requiring that the different wedges and boxes used in the construction are pairwise disjoint impose strong restrictions on the possible constructions using this paradigm. It is possible to imagine situations where some wedges intersect although the different line segments in the wedges remain crossing free.

We thus propose the following strategy (refer to Figure 4):

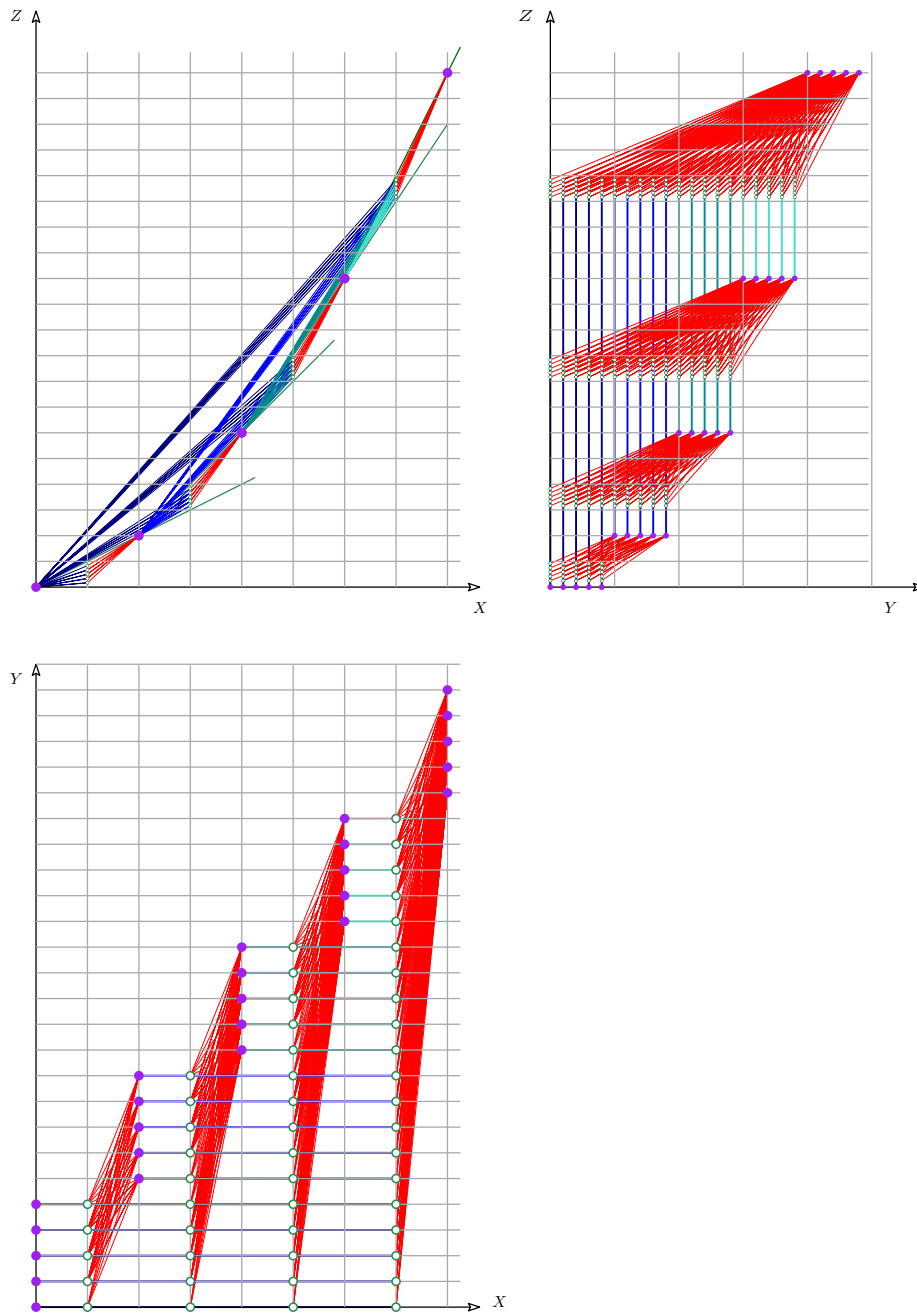


Figure 3: Drawing of the bi-partite graphs for 5 groups of 5 vertices in the Devillers-Everett-Lazard-Pentcheva-Wismath construction. Each edge has two segments, one from the vertex in the lower group to its bend in blue, another from the bend to the vertex in the upper group in red.

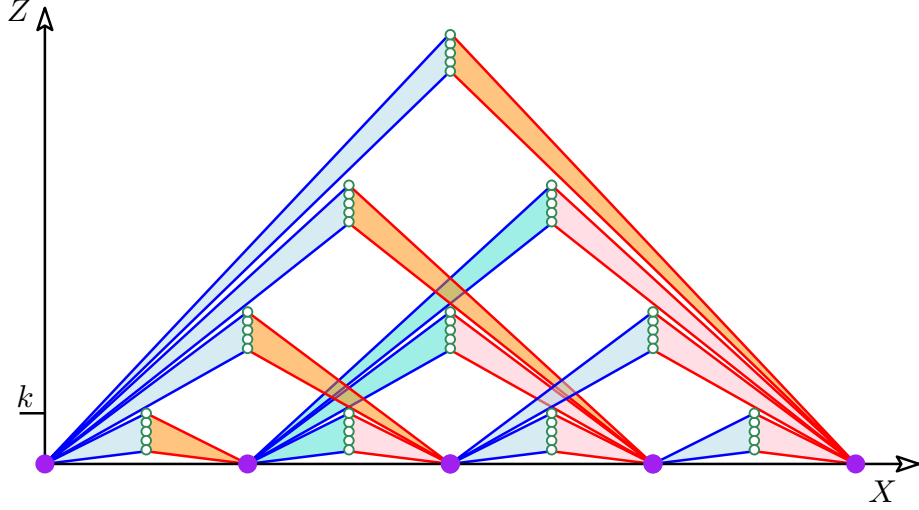


Figure 4: New construction in projection in the  $XZ$  plane. Up-going wedges (in blue) are pairwise disjoint. Down-going wedges (in red or orange) are pairwise disjoint.

— We split the vertices in  $k = \sqrt{n}$  groups of  $k$  vertices (without loss of assumption, we assume  $n$  to be the square of an integer).

— Vertex  $v_{i,j}$ ,  $0 \leq i, j < k$  is placed at coordinates  $(2i, j, 0)$  in  $XY$ -plane.

— The complete graph of group  $v_{i,*}$  is drawn in the box  $[2i, 2i+1] \times [0, k] \times [-k^2, 0]$  using Lemma 1. Notice that these boxes are pairwise disjoint.

— The bend for the edge from  $v_{i,j}$  to  $v_{i',j'}$  with  $i < i'$  is placed at  $e_{i,j,i',j'} = (i' + i, y_{i,j,i',j'}, z_{i'-i}k - j)$  where  $z_1 = 1$ ,  $z_i = \left\lceil \frac{i}{i-1} z_{i-1} \right\rceil + 1$ , and  $y_{i,j,i',j'}$  will be determined algorithmically later (before Theorem 7).

Figure 4 depicts this construction for  $k = 5$ . Before studying the disjointness of the edges, we start by a preliminary lemma on the heights of the bend points.

**Lemma 4.**  $\forall i \geq 3, \quad 1 + i \ln i < z_i < 2i \ln i.$

*Proof.* First, this is true for  $z_3 = 6 < 2 \cdot 3 \ln 3 \simeq 6.59$  and  $z_3 > 1 + 3 \ln 3 \simeq 4.3$ . By induction, we have:

$$\begin{aligned}
 z_i &= \left\lceil \frac{i}{i-1} z_{i-1} \right\rceil + 1 \\
 \frac{i}{i-1} z_{i-1} + 1 &\leq z_i \leq \frac{i}{i-1} z_{i-1} + 2 \\
 \frac{i}{i-1} (1 + (i-1) \ln(i-1)) + 1 &< z_i < \frac{i}{i-1} 2(i-1) \ln(i-1) + 2 \quad \text{by induction hypothesis} \\
 1 + i \ln i < i \left( \frac{1}{i-1} + \ln(i-1) \right) + 1 &< z_i < 2i \left( \ln(i-1) + \frac{1}{i} \right) < 2i \ln i.
 \end{aligned}$$

where we use for the last leftmost and rightmost inequalities the fact that  $\frac{1}{i} < \int_{i-1}^i \frac{dx}{x} < \frac{1}{i-1}$  and thus that  $\ln i = \ln(i-1) + \int_{i-1}^i \frac{dx}{x} \in (\ln(i-1) + \frac{1}{i}, \ln(i-1) + \frac{1}{i-1})$ .  $\square$

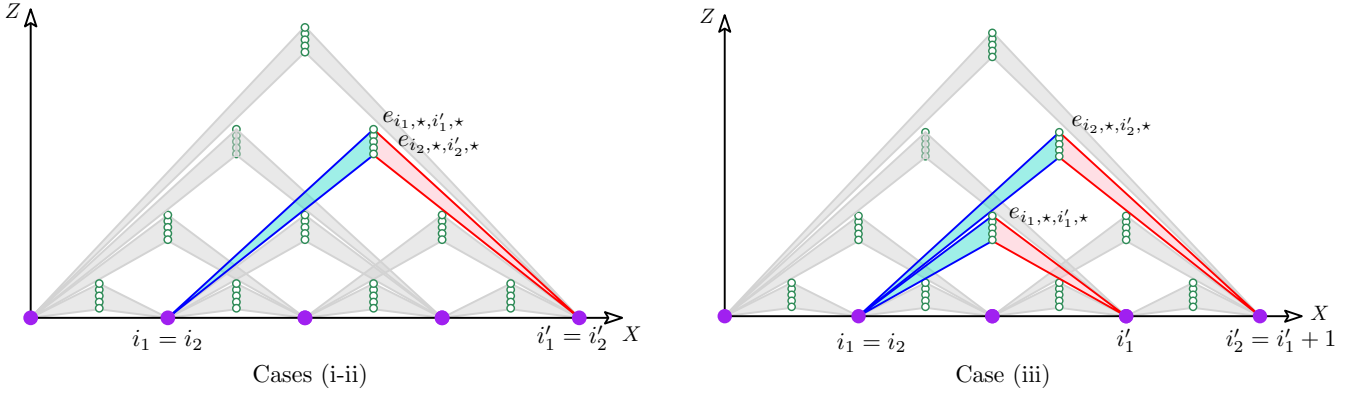


Figure 5: For the proof of Lemma 5.

The edge from  $v_{i,j}$  to  $v_{i',j'}$  consists of an upgoing line segment between  $v_{i,j}$  and  $e_{i,j,i',j'}$  and a downgoing line segment between  $e_{i,j,i',j'}$  and  $v_{i',j'}$ . We first show the following lemma:

**Lemma 5.** *Two edges starting (resp. finishing) within the same group of vertices are interior disjoint.*

*Proof.* As in previous construction, one can compare the wedges in  $XZ$  projection. Refer to Figure 5. Consider two segments:  $[v_{i_1,j_1}; e_{i_1,j_1,i'_1,j'_1}]$  and  $[v_{i_2,j_2}; e_{i_2,j_2,i'_2,j'_2}]$  and assume without loss of generality that  $i_1 = i_2$  and  $i'_1 \leq i'_2$ . We distinguish several cases:

(i) If  $i'_1 = i'_2$  and  $j_1 = j_2$ , then the two segments coincide in  $XZ$  projection and intersect only at their origin in 3D (if the two edges are distincts).

(ii) If  $i'_1 = i'_2$  and  $j_1 \neq j_2$ , then the two segments are in the same wedge. In  $XZ$  projection, the two segments have the same origin and different targets, so they are interior disjoint in projection and also in 3D.

(iii) If  $i'_2 = i'_1 + 1$ , then the two segments are in two consecutive wedges issued from the same group. The upper segment of the first wedge is obtained for  $j_1 = 0$  and has slope  $\frac{z_{i'_1} - i_1 k}{i'_1 - i_1}$  (in the  $XZ$  projection plane). The lower segment of the second wedge is obtained for  $j_2 = k - 1$  and has slope  $\frac{z_{i'_2} - i_2 k}{i'_2 - i_2} = \frac{z_{i'_1 - i_1 + 1} k - k + 1}{i'_1 - i_1 + 1}$ . Denoting  $i = i'_1 - i_1$  we ensure that the first wedge is below the second wedge by proving that

$$\frac{z_i k}{i} < \frac{z_{i+1} k - (k-1)}{i+1},$$

which follows easily from

$$\frac{z_{i+1} k - k + 1}{i+1} > \frac{z_{i+1} k - k}{i+1} = \left( \left[ \frac{i+1}{i} z_i \right] + 1 - 1 \right) \frac{k}{i+1} \geq \left( \frac{i+1}{i} z_i \right) \frac{k}{i+1} = \frac{z_i k}{i}.$$

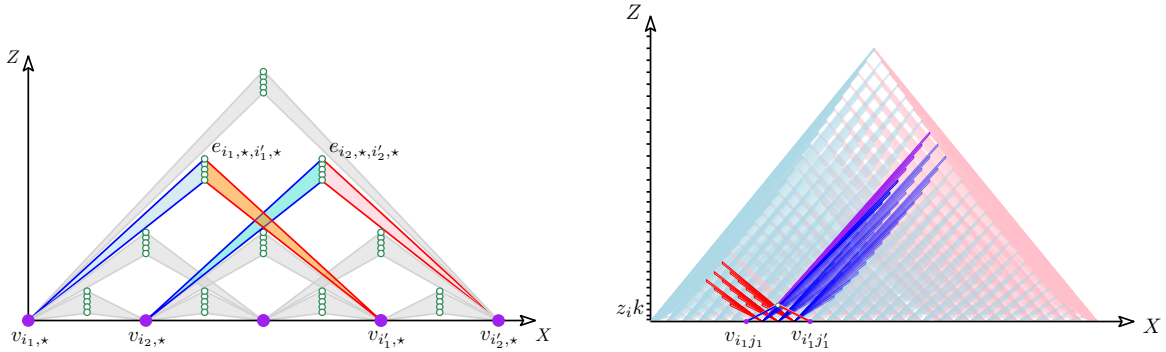


Figure 6: Segment intersections in projection in the  $XZ$  plane.

(iv) If  $i'_2 > i'_1 + 1$ , the two segments are in different wedges issued from the same group. Their disjointness is granted by an easy induction from Case (iii).

The disjointness of the downgoing parts of the edges (i.e., segments  $[e_{i_1, j_1, i'_1, j'_1}; v_{i'_1, j'_1}]$  and  $[e_{i_2, j_2, i'_2, j'_2}; v_{i'_2, j'_2}]$ ) is obtained with similar arguments in Cases (i-ii) and, in Cases (iii-iv), because, since  $i'_2 > i'_1$ , vertex  $v_{i'_2, *}$  is to the right of  $v_{i'_1, *}$  and bend point  $e_{i_2, \dots}$  is also to the right of bend point  $e_{i_1, \dots}$  (see Figure 5). The case of edges finishing within the same group of vertices is obtained by symmetry.  $\square$

**Lemma 6.** *Refer to Figure 6. Two edges  $E_1 = [v_{i_1, j_1}; e_{i_1, j_1, i'_1, j'_1}; v_{i'_1, j'_1}]$  and  $E_2 = [v_{i_2, j_2}; e_{i_2, j_2, i'_2, j'_2}; v_{i'_2, j'_2}]$  properly intersects in  $XZ$  projection if and only if their endpoints are interlaced, that is,  $i_1 < i_2 < i'_1 < i'_2$  or  $i_2 < i_1 < i'_2 < i'_1$ . If the bend or  $E_2$  is above that of  $E_1$  and  $i_1 < i_2 < i'_1 < i'_2$  (resp.  $i_2 < i_1 < i'_2 < i'_1$ ), then the intersection occurs in the upgoing (resp. downgoing) part of  $E_2$ .*

*Proof.* We first show that, if  $i_1 \leq i_2$ , the two edges properly intersects in  $XZ$  projection if and only if  $i_1 < i_2 < i'_1 < i'_2$ ; refer to Figure 6(left). Since  $i_1 \leq i_2 < i'_2$ , integer  $i'_1$  is in  $(i_1, i_2)$ ,  $(i_2, i'_2)$  or  $[i'_2, \infty)$ . In the first case,  $i_1 < i'_1 \leq i_2 < i'_2$ , the edges are interior disjoint in  $X$  projection.

In the third case,  $i_1 \leq i_2 < i'_2 \leq i'_1$ , that is, both endpoints of  $E_2$  are in between those of  $E_1$ . If  $i_1 = i_2$  or  $i'_1 = i'_2$  the result follow from Lemma 5. Thus  $i_1 < i_2 < i'_2 < i'_1$ . Since  $i'_1 - i_1 > i'_2 - i_2$ , the bend of  $E_1$  is above that of  $E_2$ . Thus, the left segment of  $E_1$  is steepest than the left segment of  $E_2$  (as in the proof of Lemma 5) and thus they do not intersect in  $XZ$  projection. Symmetrically, the downgoing edges of  $E_1$  and  $E_2$  do not intersect in  $XZ$  projection. It follows that the two edges do not intersect in  $XZ$  projection since their endpoints are nested.

Hence, the second case holds, that is  $i_1 < i_2 < i'_1 < i'_2$ . The endpoints of  $E_1$  and  $E_2$  are thus interlaced and thus the edges intersect. The above argument on the slopes of the edge segments proves the second part of the lemma; see Figure 6(right).  $\square$

Then, we have to determine the  $Y$  coordinate of the bends to avoid crossings. We do this algorithmically: for  $i, j, i', j'$  from 0 to  $k$ . Using Remark 3, we choose the  $Y$ -coordinate  $y_{i, j, i', j'}$  of the bend point on the edge from  $v_{i, j}$  to  $v_{i', j'}$  to be the smallest

non-negative integer such that the edge properly intersects none of the edges that have been previously defined in the loop. Since the number of already drawn edges is  $O(n^2)$ ,  $y_{i,j,i',j'}$  is granted to be also not bigger than  $O(n^2)$ , even if we expect it to be much smaller. A similar incremental process is used by Wood [14] to determine the bends in a different context.

It should be noted that the bend points depend on the order in which we consider  $i, j, i', j'$ . To optimize the resulting volume of the drawing, we define the bends in a lexicographic order, by first decreasing  $Z$ , then increasing  $X$  and increasing  $Y$ . In other words, in  $XZ$  projection (Figure 4), the bend points are defined from top to down and then from left to right. This is illustrated on Figure 6(right), downgoing wedges to consider against  $[v_{i_1 j_1} e_{i_1 j_1 i'_1 j'_1}]$  are in red, upgoing wedges to consider against  $[v_{i_1 j_1} e_{i_1 j_1 i'_1 j'_1}]$  are in purple, and upgoing wedges to consider against  $[e_{i_1 j_1 i'_1 j'_1} v_{i'_1 j'_1}]$  are in blue. The corresponding pseudo-code can be found in Appendix A.

**Theorem 7.** *A one-bend non-crossing 3D drawing of  $K_n$  can be achieved within a box of size  $2\sqrt{n} \times Y_n \times O(n \log n)$ , with  $Y_n$  a function of  $n$  smaller than  $O(n^2)$ .*

*Proof.* The distinct boxes that enclose the drawings of the complete graphs of each group are disjoint and below the plane  $Z = 0$ . The drawings of the bi-partite graphs between two groups are above the plane  $Z = 0$  and thus bi-partite graphs and complete graphs do not interfere. Edges starting or finishing at the same place are pairwise disjoint by Lemma 5. Edges whose starting and finishing place are not interlaced are pairwise disjoint by Lemma 6. The remaining edges are pairwise disjoint by construction.

The  $X$ -dimension of our construction is  $2k = 2\sqrt{n}$  by construction. The  $Y$ -dimension is some finite value  $Y_n = O(n^2)$ . The height is  $O(k^2) = O(n)$  below the plane  $Z = 0$  by Lemma 1 and it is  $O(n \log n)$  above the plane  $Z = 0$  by Lemma 4.  $\square$

**Conjecture 8.**  *$Y_n = O(\sqrt{n} \log n)$  and thus the volume of the drawing in Theorem 7 is  $O(n^2 \log^2 n)$  reaching the quadratic optimal bound up to log factors.*

## 4 Experimental study

We implemented our construction to determine crossing-free 3D drawings of  $K_n$  for  $n$  up to 10 000; precisely, we determined crossing-free 3D drawings of  $K_{k^2}$  for  $k \in [4, 100]$ . The code is simple enough so that we are confident in its correctness; the code is given in Appendix D.

The values of  $Y_n$  for  $n \leq 10000$ , given in Figure 7 and Appendix C, show a very reasonable increase of  $Y_n$  in terms of  $n$ . Actually plotting  $\frac{Y_n}{\sqrt{n} \log_2 n}$  (see Figure 8) supports Conjecture 8.

We draw  $K_{8^4} = K_{4096}$  within a box  $[0, 127] \times [0, 629] \times [-2048, 24192]$  of volume about 2.3 billions. Comparing to the result by Devillers et al. [5], their construction fits within box  $[0, 32] \times [0, 4095] \times [-8192, 28672]$ , which has four time the volume of our construction and a worse aspect ratio.

For  $K_{10^4} = K_{10000}$ , we obtain bounding box  $[0, 199] \times [0, 1044] \times [-5000, 66500]$  of volume about 16 billions, while Devillers et al. box is  $[0, 40] \times [0, 10000] \times$

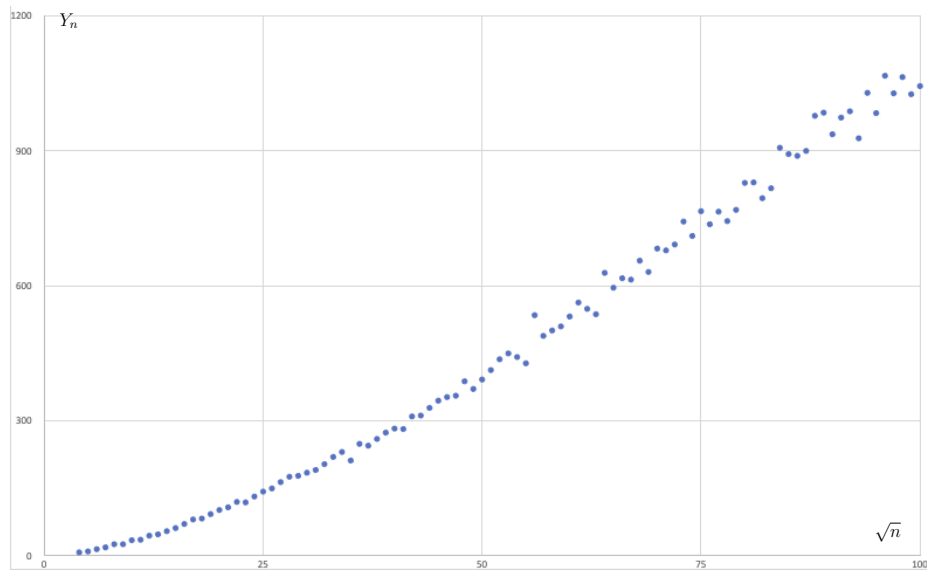


Figure 7:  $Y_n$  for  $n \leq 10^4$ .

$[-25000, 90000]$  of volume about 6 times greater and also much worse aspect ratio. See Figure 9 for a more complete comparison of volume used by the two methods.

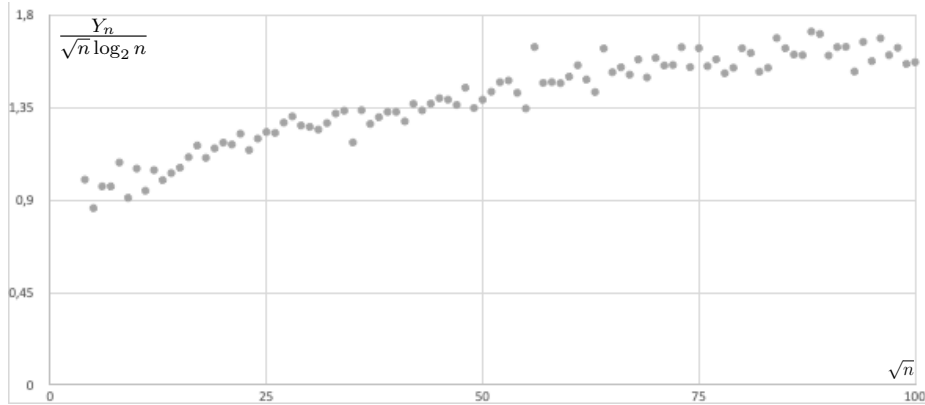


Figure 8:  $\frac{Y_n}{\sqrt{n} \log_2 n}$  for  $n \leq 10000$ .

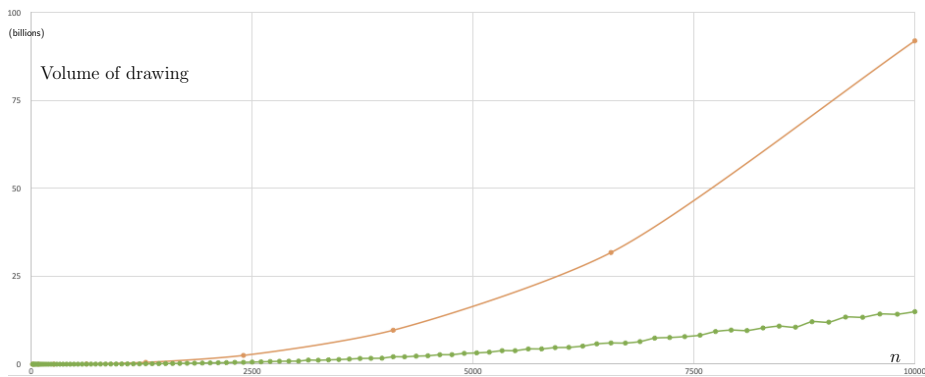


Figure 9: Volumes obtained in our construction (green) and Devillers et al. [5] (orange).

## References

- [1] G. Di Battista, F. Frati, and J. Pach. On the queue number of planar graphs. *SIAM Journal on Computing*, 42(6):2243–2285, 2013. doi:10.1137/130908051.
- [2] P. Bose, J. Czyzowicz, P. Morin, and D. R. Wood. The maximum number of edges in a three-dimensional grid-drawing. *J. of Graph Algorithms and Applications*, 8(1):21–26, 2004. doi:10.7155/jgaa.00079.
- [3] T. Calamoneri and A. Sterbini. 3D straight-line grid drawing of 4-colorable graphs. *Information Processing Letters*, 63(2):97–102, 1997.
- [4] R. F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. *Algorithmica*, 17(2):199–208, 1997. doi:10.1007/BF02522826.

- [5] O. Devillers, H. Everett, S. Lazard, M. Pentcheva, and S. Wismath. Drawing  $K_n$  in Three Dimensions with One Bend per Edge. *Journal of Graph Algorithms and Applications*, 10(2):287–295, 2006. URL: <https://inria.hal.science/inria-00103923>, doi:10.7155/jgaa.00128.
- [6] V. Dujmović, W. Evans, S. Lazard, W. Lenhart, G. Liotta, D. Rappaport, and S. Wismath. On point-sets that support planar graphs. *Computational Geometry*, 46(1):29–50, 2013. URL: <https://www.sciencedirect.com/science/article/pii/S0925772112000739>, doi:10.1016/j.comgeo.2012.03.003.
- [7] V. Dujmović and D. R. Wood. Stacks, queues and tracks: Layouts of graph subdivisions. *Discrete Mathematics & Theoretical Computer Science*, 7, 2005. doi:10.46298/dmtcs.346.
- [8] B. Dyck, J. Joevenazzo, E. Nickle, J. Wilsdon, and S. Wismath. Drawing  $K_n$  in Three Dimensions with Two Bends Per Edge. Technical report, University of Lethbridge, 2004. URL: [https://citeseerx.ist.psu.edu/doc\\_view/pid/359925ca9da74cceeef62c31a9f8461889e7e592c](https://citeseerx.ist.psu.edu/doc_view/pid/359925ca9da74cceeef62c31a9f8461889e7e592c).
- [9] H. Everett, S. Lazard, G. Liotta, and S. Wismath. Universal Sets of  $n$  Points for One-bend Drawings of Planar Graphs with  $n$  Vertices. *Discrete and Computational Geometry*, 43(2):272–288, 2010. URL: <https://inria.hal.science/inria-00431769>, doi:10.1007/s00454-009-9149-3.
- [10] S. Felsner, G. Liotta, and S. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. *J. of Graph Algorithms and Applications*, 7(4):363–398, 2003. doi:10.7155/jgaa.00075.
- [11] H. Meijer and S. Wismath. Point-set embedding in three dimensions. *J. of Graph Algorithms and Applications*, 19(1):243–257, 2015. doi:10.7155/jgaa.00355.
- [12] P. Morin and D. R. Wood. Three-dimensional 1-bend graph drawings. *J. of Graph Algorithms and Applications*, 8(3):357–366, 2004. doi:10.7155/jgaa.00095.
- [13] J. Pach, T. Thiele, and G. Tóth. Three-dimensional grid drawings of graphs. In *Proc. of 4th Int. Symposium on Graph Drawing (GD '97), Lecture Notes in Computer Science*, volume 1353, pages 47–51. Springer-Verlag, 1997.
- [14] D. Wood. Three dimensional graph drawing with fixed vertices and one bend per edge. 2016. URL: <https://arxiv.org/abs/1606.09188>.

## A Pseudo code for determining the $Y$ -coordinates of the bend points

```

for  $i = k - 1$  downto 1 //  $i = i'_1 - i_1$ 
  for  $i_1 = 0$  to  $k - i$ 
    for  $j_1 = 0$  to  $k - 1$ 
      for  $j'_1 = 0$  to  $k - 1$  {
         $i'_1 = i_1 + i$ ;
        if  $j'_1 = 0$  then  $y = 0$  else  $y_{i_1, j_1, i'_1, j'_1} = y_{i_1, j_1, i'_1, j'_1 - 1} + 1$ ;
        while check( $i_1, j_1, i'_1, j'_1$ ) unsuccessful
           $y_{i_1, j_1, i'_1, j'_1} = y_{i_1, j_1, i'_1, j'_1} + 1$ ;
      }

check( $i_1, j_1, i'_1, j'_1$ ) // return success
//if  $[v_{i_1, j_1}, y_{i_1, j_1, i'_1, j'_1}] \cup [y_{i_1, j_1, i'_1, j'_1}, v_{i'_1, j'_1}]$  has no  $\cap$  with previously drawn edges
  //check  $[v_{i_1, j_1}, e_{i_1, j_1, i'_1, j'_1}]$  with edges starting left
  for all  $(i_2, j_2, i'_2, j'_2) \in [0, i_1] \times [0, k] \times [\max(i_1 + 1, i_2 + i), i'_1] \times [0, k)$ 
    if  $e_{i_2, j_2, i'_2, j'_2}$  left of  $[v_{i_1, j_1}, e_{i_1, j_1, i'_1, j'_1}]$  in XZ projection
      return unsuccessful if
coplanar( $v_{i_1, j_1}, v_{i'_2, j'_2}, e_{i_1, j_1, i'_1, j'_1}, e_{i_2, j_2, i'_2, j'_2}$ )
  else // below
    return unsuccessful if
coplanar( $v_{i'_1, j'_1}, v_{i'_2, j'_2}, e_{i_1, j_1, i'_1, j'_1}, e_{i_2, j_2, i'_2, j'_2}$ )
  //check  $[v_{i_1, j_1}, e_{i_1, j_1, i'_1, j'_1}]$  with edges starting in between
  for all  $(i_2, j_2, i'_2, j'_2) \in (i_1, i'_1) \times [0, k] \times (\max(i'_1, i_2 + i), k) \times [0, k)$ 
    if  $e_{i_2, j_2, i'_2, j'_2}$  right of  $[v_{i_1, j_1}, e_{i_1, j_1, i'_1, j'_1}]$  in XZ projection
      return unsuccessful if
coplanar( $v_{i'_1, j'_1}, v_{i_2, j_2}, e_{i_1, j_1, i'_1, j'_1}, e_{i_2, j_2, i'_2, j'_2}$ )
  else // below
    return unsuccessful if
coplanar( $v_{i_1, j_1}, v_{i_2, j_2}, e_{i_1, j_1, i'_1, j'_1}, e_{i_2, j_2, i'_2, j'_2}$ )
  return successful

```

## B Coplanarity and XZ-leftturn predicates

We present here the computations for the predicates used in previous and next sections.

The usual determinants for 3D and 2D orientation simplify since some vertices are in the plane  $Z = 0$ .

$$\begin{aligned}
 & \text{orient3D}((x_0, y_0, 0), (x_1, y_1, 0), (x_2, y_2, z_2), (x_3, y_3, z_3)) \\
 &= \begin{vmatrix} x_0 & x_1 & x_3 \\ y_0 & y_1 & y_3 \\ 0 & 0 & z_3 \\ 1 & 1 & 1 \end{vmatrix} \\
 &= z_2 \begin{vmatrix} x_0 & x_1 & x_3 \\ y_0 & y_1 & y_3 \\ 1 & 1 & 1 \end{vmatrix} - z_3 \begin{vmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{vmatrix} \\
 &= z_2 \begin{vmatrix} x_0 & x_1 - x_0 & x_3 - x_0 \\ y_0 & y_1 - y_0 & y_3 - y_0 \\ 1 & 0 & 0 \end{vmatrix} - z_3 \begin{vmatrix} x_0 & x_1 - x_0 & x_2 - x_0 \\ y_0 & y_1 - y_0 & y_2 - y_0 \\ 1 & 0 & 0 \end{vmatrix} \\
 &= z_2((x_1 - x_0)(y_3 - y_0) - (y_1 - y_0)(x_3 - x_0)) \\
 &\quad - z_3((x_1 - x_0)(y_2 - y_0) - (y_1 - y_0)(x_2 - x_0)).
 \end{aligned}$$

The four points are coplanar if this determinant is 0.

$$\begin{aligned}
 & \text{orient2D}((x_0, 0), (x_1, z_1), (x_2, z_2)) \\
 &= \begin{vmatrix} x_0 & x_1 & x_2 \\ 0 & z_1 & z_2 \\ 1 & 1 & 1 \end{vmatrix} \\
 &= x_0(z_1 - z_2) + (x_1 z_2 - x_2 z_1).
 \end{aligned}$$

The three points turn left if this determinant is positive.

### C Values of $Y_n$

$\sqrt{n}$	$Y_n$	$\sqrt{n}$	$Y_n$	$\sqrt{n}$	$Y_n$
5	10	37	245	69	631
6	15	38	260	70	683
7	19	39	274	71	679
8	26	40	283	72	692
9	26	41	282	73	743
10	35	42	310	74	711
11	36	43	312	75	766
12	45	44	329	76	737
13	48	45	345	77	765
14	55	46	353	78	744
15	62	47	356	79	769
16	71	48	388	80	829
17	81	49	371	81	830
18	83	50	392	82	795
19	93	51	413	83	817
20	102	52	437	84	907
21	108	53	450	85	893
22	120	54	442	86	889
23	119	55	428	87	900
24	132	56	535	88	978
25	143	57	489	89	985
26	150	58	501	90	937
27	164	59	510	91	974
28	176	60	532	92	988
29	178	61	563	93	928
30	185	62	549	94	1029
31	191	63	537	95	984
32	204	64	629	96	1067
33	220	65	596	97	1028
34	231	66	617	98	1064
35	212	67	614	99	1026
36	249	68	656	100	1044

## D Source code

```

#include <stdio.h>
#include <math.h>
const int KMAX=82;
const int KMIN=4;

int
N, // number of vertices
K=KMAX, // square root of N
Z[KMAX], //
Y[KMAX][KMAX][KMAX][KMAX], //
YMAX;

int max(int i, int j) { return (i>j)?i:j; }
int coplanar(int x0,int y0, int x1,int y1, int x2,int y2,int z2, int x3,int y3,int z3){
int det = // z0=z1=0 is assumed
z2 * ( (x1-x0)*(y3-y0) - (y1-y0)*(x3-x0) ) - z3 * ( (x1-x0)*(y2-y0) - (y1-y0)*(x2-x0) );
return (det==0);
}
int leftturnXZ(int x0, int x1, int z1, int x2, int z2) {
int det = // z0 is assumed
x0 * ( z1 - z2) + (x1*z2 -x2*z1);
return (det>0);
}
int try(int i1,int j1,int i11,int j11)
{
// return ok if Y[i1,j1,i11,j11] is a correct value
// check all edges already drawn and crossing in XZ projection
for (int i2=0; i2<i1; ++i2)
for (int j2=0; j2<j1; ++j2)
for (int i12=max(i1+1,i2+i11-i1); i12<i11; ++i12)
for (int j12=0; j12<j1; ++j12){
if (leftturnXZ(2*i1,i1+i11,Z[i11-i1]-j1,i2+i12,Z[i12-i2]-j2))
if (coplanar(2*i1,j1,2+i12,j2,i1+i11,Y[i1][j1][i11][j11],Z[i11-i1]-j1,i2+i12,V[i2][j2][i12][j12],Z[i12-i2]-j2)) return 0;}
if (coplanar(2*i1,j1,2+i12,j2,i1+i11,Y[i1][j1][i11][j11],Z[i11-i1]-j1,i2+i12,V[i2][j2][i12][j12],Z[i12-i2]-j2)) return 0;}
}
for (int i2=i1+1; i2<i11; ++i2)
for (int j2=0; j2<j1; ++j2)
for (int i12=max(i11+1,i2+i11-i1+1); i12<K; ++i12)
for (int j12=0; j12<j1; ++j12){
if (!leftturnXZ(2*i11,i1+i11,Z[i11-i1]-j1,i2+i12,Z[i12-i2]-j2))
if (coplanar(2*i11,j11,2+i12,j2,i1+i11,Y[i1][j1][i11][j11],Z[i11-i1]-j1,i2+i12,V[i2][j2][i12][j12],Z[i12-i2]-j2)) return 0;}
else
if (coplanar(2*i1,j1,2+i12,j2,i1+i11,Y[i1][j1][i11][j11],Z[i11-i1]-j1,i2+i12,V[i2][j2][i12][j12],Z[i12-i2]-j2)) return 0;}
}
return 1;
}

int main(int argc, char *argv[])
{
K=KMIN;

while (1){
N= K*K; // ensure N is a perfect square
YMAX=K-1;
printf("\nComplete graph with %d vertices\n",N);
Z[0]=0;Z[1]=1;
for(int i=2; i<K; ++i){
Z[i] = ceil( 1/(double)(i-1) * Z[i-1] ) + 1;
}
for(int i=0; i<K; ++i) { Z[i]*=K; }
// V[i,j] = ( 2*i, j, 0)
// Bend[i,j,ii,jj] = ( ii+i, Y[i,j,ii,jj], Z[i1-i]-j )
for (int i1=K-1; i1>0; --i1)
for (int i=0; i<K-i1; ++i)
for (int j=0; j<K; ++j)
for (int jj=0; jj<K; ++jj){
int i11=i+i1;
Y[i][j][i1][jj] = (jj==0) ? 0 : Y[i][j][i1][jj-1]+1;
while (! try(i,j,i1,jj)){
Y[i][j][i1][jj]++;
}
if (YMAX<Y[i][j][i1][jj]) YMAX=Y[i][j][i1][jj];
}
printf("XxYxZ = [0,%d] x [0,%d] x [0,%d]\n",2*(K-1),YMAX,Z[K-1]);
}
}

```