GPU Algorithm for Enumerating PL Spheres of Picard Number 4: Application to Toric Topology

Suyoung Choi
Ajou University, Suwon, South Korea

Hyeontae Jang
Ajou University, Suwon, South Korea

Mathieu Vallée[1]
LIPN, CNRS UMR 7030, Université Sorbonne Paris Nord, Villetaneuse, France

Abstract

The *fundamental theorem for toric geometry* states a toric manifold is encoded by a complete non-singular fan, whose combinatorial structure is the one of a PL sphere together with the set of generators of its rays. The wedge operation on a PL sphere increases its dimension without changing its Picard number. The *seeds* are the PL spheres that are not wedges. A PL sphere is *toric colorable* if it comes from a complete rational fan. A result of Choi and Park tells us that the set of toric seeds with a fixed Picard number p is finite. In fact, a toric PL sphere needs its facets to be bases of some binary matroids of corank p with neither coloops, nor cocircuits of size 2. We present and use a GPU-friendly and computationally efficient algorithm to enumerate this set of seeds, up to simplicial isomorphism. Explicitly, it allows us to obtain this set of seeds for Picard number 4 which is of main importance in toric topology for the characterization of toric manifolds with small Picard number. This follows the work of Kleinschmidt (1988) and Batyrev (1991) who fully classified toric manifolds with Picard number ≤ 3.

2012 ACM Subject Classification Mathematics of computing → Enumeration; Mathematics of computing → Combinatoric problems; Computing methodologies → Shared memory algorithms; Computing methodologies → Massively parallel algorithms; Theory of computation → Computational geometry

Keywords and phrases PL sphere, simplicial sphere, toric manifold, Picard number, weak pseudo-manifold, characteristic map, binary matroid, parallel computing, GPU programming

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.41

Related Version *Full Version*: `https://arxiv.org/abs/2301.00806`

Supplementary Material *Software (Source Code and Database)*: `https://github.com/MVallee1998/GPU_handle` [31], archived at `swh:1:dir:ecc6f4940dc55c6a4175e78f2cfa47ccfd3ce009`

Funding This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2019R1A2C2010989).

Acknowledgements The authors are very grateful to Dr. Axel Bacher, who introduced the last-named author to CUDA programming allowing Algorithm 2 to fit with GPU computing terminate in a reasonable time.

1 Introduction

Our interest is located at the intersection of discrete mathematics, with the enumeration of PL spheres, and geometry, with the classification of non-singular complete toric varieties.

---

[1] corresponding author

State-of-the-art known PL spheres. A *PL sphere* is a pure simplicial complex possessing a subdivision piecewise linearly homeomorphic to the boundary of a standard simplex. A PL sphere is said to be *polytopal* if it is isomorphic to the boundary complex of a simplicial polytope. Let us fix a dimension $n-1 \in \mathbb{Z}_{>0}$ of a PL sphere K, and its number of vertices $m = n+p$. We call the number p the *Picard number* of K. Starting from the end of the 19th century, the first direction for enumerating (polytopal) PL spheres was to focus on small dimensions n, namely $n \leq 4$:

n	m	Polytopal PL sphere	General PL sphere
2	$m \geq 3$	**Characterization:** m-gon	
3		**Characterization:** Steinitz theorem, equivalent to 3-connected planar graphs, 1922	
		Enumeration:	
	$m \leq 13$	Brückner by hand, 1897-1931 [9, 10]	
	$m = 11$	Corrected by Grace, 1965 [18]	
	$m = 12$	Corrected by Bowen and Fisk, 1967 [6]	
	$m = 13$	Corrected by Royle, program `plantri` by Brinkmann and McKay, 1999 [8]	
	$m \leq 23$	Brinkmann, also using `plantri`, 2007 [7]	
4		**Characterization:** unknown	**Characterization:** unknown
		Enumeration:	**Enumeration:**
	$m = 8$	Brückner, 1909 [11], Grünbaum and Sreedharan, 1967 [21]	Non-polytopal sphere by Barnette, 1969 [5], Altshuler and Steinberg, 1985 [3]
	$m = 9$	Altshuler and Bokowski and Steinberg, 1980 [1]	Altshuler and Steinberg, 1976 [2]
	$m = 10, 11$	Miyata and Padrol, 2015 [27], (neighbourly polytopes), using oriented matroids	Sulanke and Lutz, 2008-2009 [24, 30], using lexicographic enumeration

Notice that for $n \leq 3$, all PL spheres are polytopal.

Around the same period, the complete characterization of PL spheres with small p, namely $p \leq 3$, was computed. To any polytopal PL sphere K, one can associate a configuration of $(p-1)$-dimensional vectors which stores the combinatorial structure of K and is called a *Gale diagram*. We know that if $p \leq 3$, then all PL spheres are polytopal ([25]) and they are thus characterized by their Gale diagram (see [20] for details):

p	Polytopal PL spheres
1	**Characterization:** The boundary of an n-simplex
2	**Characterization:** Repeated pyramid over a free sum of two simplices, Grünbaum [20]
3	**Characterization:** Regular n-gonal Gale diagram, with n odd, Perles [20]

However, for $p = 4$, Grünbaum and Sreedharan [21] gave an example of non-polytopal PL sphere with $p = 4$. Characterizing or enumerating PL spheres is important in toric geometry since they are among the cornerstone combinatorial objects for this theory. In this article, we use the same base point as in [24, 30] which is that PL spheres are weak pseudo-manifolds, which are simplicial complexes whose structure is easier to recognize.

State-of-the-art known toric varieties. A *toric variety* of complex dimension n is a normal algebraic variety over the field of complex numbers $\mathbb{C}$ which admits an effective algebraic action of $(\mathbb{C}^*)^n$ having a dense orbit. The *fundamental theorem for toric geometry* states that the classification of toric varieties of complex dimension n is equivalent to the one of *fans* in $\mathbb{R}^n$. In particular, compact smooth toric varieties correspond to complete non-singular fans. A complete non-singular fan Σ in $\mathbb{R}^n$ having m rays can be described by a pair (K, λ), where:

- K is the underlying simplicial complex of Σ which is an $(n-1)$-dimensional PL sphere on $[m] = \{1, \dots, m\}$, and
- $\lambda\colon [m] \to \mathbb{Z}^n$ is a non-singular *fan-giving* map that is bijectively assigning a vertex of K to the primitive generator of a ray of Σ.

A non-singular fan-giving map λ should satisfy the following condition, known as the *non-singularity condition* over K; for any simplex $\{i_1, \dots, i_n\}$ in K, $\{\lambda(i_1), \dots, \lambda(i_n)\}$ is unimodular. A map $\lambda\colon [m] \to \mathbb{Z}^n$ is called a *characteristic map* over K if it satisfies the non-singularity condition over K. We call a PL sphere *toric colorable* if it supports a characteristic map. The following fundamental question appears.

▶ **Question 1.** Which pairs (K, λ) *are* complete non-singular fans?

Firstly, the simplicial complex K has to be a PL sphere. However, not all PL spheres are toric colorable. It is well-known that all PL spheres of Picard number ≤ 2 are toric colorable while also supporting non-singular complete toric varieties, see [23]. The ones of Picard number 3 may not be toric colorable; a PL sphere whose Gale-diagram is a regular $(2k+1)$-gon is toric colorable if and only if $k \leq 3$ [15], and it supports a non-singular complete toric variety if and only if $k \leq 2$ [19]. There did not exist any characterization for higher Picard numbers since there lacks a combinatorial description in such cases. Moreover, brute force algorithms for obtaining the list of PL spheres for big n and $p \geq 4$ have too high complexity making their use worthless.

One noticeable step for solving Question 1 is the work of Choi and Park [14] by translating this problem into a finite one. The *wedge* of K at a vertex v is the simplicial complex given by $\operatorname{wed}_v(K) := (I * \operatorname{Lk}_K(v)) \cup (\partial I * K \setminus \{v\})$, where I is an interval (the details will be given in Section 3). A *seed* is a PL sphere that is not the wedge (at some vertex) of any lower dimensional PL sphere. The wedge operation keeps the property of being a toric colorable PL sphere, see [16], and the Picard number. As a consequence, if we fix a Picard number p, then the complete characterization of PL spheres of Picard number p is principally given by the seeds of Picard number p. The result of Choi and Park [14] is that there are only finitely many toric colorable seeds of Picard number p whereas there are infinitely many seeds of Picard number $p \geq 3$. More precisely, Statement 2. of Theorem 10 says that the facets of these seeds must be the bases of a binary matroid without coloops nor cocircuits of size 2. Statement 3. states if an $(n-1)$-dimensional toric colorable seed is of Picard number $p \geq 3$, then p and n must satisfy the inequality

$$n + p \leq 2^p - 1.$$

In particular, if an $(n-1)$-dimensional seed of Picard number 4 is toric colorable, then $n \leq 11$.

After the full classification of non-singular complete toric varieties of Picard number $p = 1, 2, 3$, we take one more step here and complete the characterization problem for toric colorable PL spheres of Picard number 4.

The goal of the paper. The essential part of this characterization problem is to find PL spheres satisfying specific conditions up to $n = 11$. To check up all possible candidates of $n = 11$, we have to consider approximately $2^{\binom{15}{11}} \approx 10^{410}$ cases, that is not computable in reasonable time, and we additionally have to check their isomorphism classes. To obtain the complete list, we aggressively use the finiteness of the problem and construct an algorithm profiting from parallel computing and GPUs whose performances are skyrocketing from their increasing development in the last decade, mainly from the popularity of *machine learning* requiring a lot of linear algebra computation. The use of GPU is not new in discrete geometry, its main use is for speeding up sequential algorithms such as in topological data analysis, see [32], or in computing applied to molecular structural biology, see [26]. In Section 2, we provide a GPU-friendly algorithm (Algorithm 2) for obtaining all weak pseudo-manifolds whose facets are all in an input set of facets satisfying given conditions written in terms of affine functions.

A PL sphere is $\mathbb{Z}_2^n$*-colorable* if there is a map $\lambda^{\mathbb{R}}\colon [m] \to \mathbb{Z}_2^n$ such that for any simplex $\{i_1, \dots, i_n\}$ in K, $\lambda^{\mathbb{R}}(i_1), \dots, \lambda^{\mathbb{R}}(i_n)$ is linearly independent over $\mathbb{Z}_2$. The map $\lambda^{\mathbb{R}}$ is called a *mod 2 characteristic map.* To find all $\mathbb{Z}_2^n$-colorable seeds, one naive strategy is to find all seeds up to $n \leq 11$, and pick all $\mathbb{Z}_2^n$-colorable ones up. However, just like counting PL spheres, counting seeds still requires heavy computing powers. Therefore, we have to use the $\mathbb{Z}_2^n$-colorability to obtain the candidates using Algorithm 2. Section 3 and Section 4 are devoted to explain how we use Algorithm 2 for our purpose. More precisely, in Section 4, we consider a binary matroid from a mod 2 characteristic map, whose set of bases is input of Algorithm 2 to obtain all weak pseudo-manifolds supporting the mod 2 characteristic map, from which we select all seeds. Our algorithm enables us to finish the enumeration within a reasonable time for $n \leq 10$. See the full version [12] for the case $n = 11$. The main result is as follows.

▶ **Theorem 2.** *Up to simplicial isomorphisms, the number of $\mathbb{Z}_2^n$-colorable seeds of dimension $n-1$ and Picard number $p \leq 4$ is as follows:*

$p\backslash\ n$	1	2	3	4	5	6	7	8	9	10	11	>11	*total*
1	1												1
2		1											1
3		1	1	1									3
4		1	$3+1$	$20+1$	$141+1$	733	1190	776	243	39	4		3153

with the empty slots displaying zero, and the "+1" representing the suspension of the three $\mathbb{Z}_2^n$-colorable seeds of Picard number 3.

▶ Remark 3. We checked that each of the 3153 Picard number 4 seed actually supports at least one characteristic map by replacing some 1 entries to -1 in their mod 2 characteristic map, this yields they all are toric colorable.

We also obtain this corollary of Theorem 2.

▶ **Corollary 4.** *The toric (or $\mathbb{Z}_2^n$-)colorable PL spheres of dimension $n-1$ and Picard number $p \leq 4$ are exactly the ones obtained after consecutive wedge operations on the toric (or $\mathbb{Z}_2^n$-)colorable seeds (of Theorem 2).*

In short, the set of toric (or $\mathbb{Z}_2^n$-)colorable PL spheres of dimension $n-1$ and Picard number $p \leq 4$ is *finitely generated* from using multiple wedge operations on the explicit $1+1+3+3153$ seeds of Theorem 2.

One of the long-standing open problems in toric geometry is to characterize non-singular complete toric varieties and the present characterization of toric colorable seeds of Picard number 4 provides the pathway for the case of Picard number 4.

2 Classification of weak pseudo-manifolds by GPU computing

In this section, we provide a general approach to how to use GPU parallel computing capability for classifying weak pseudo-manifolds with given properties.

Let K be a pure simplicial complex of dimension $n-1$ on the vertex set $[m] = \{1, 2, \dots, m\}$. A *facet* of K is an element of size n of K, and a *ridge* is an element of size $n-1$ of K. We denote by $\mathcal{F}(K)$ and $\mathcal{R}(K)$ the sets of facets and ridges of K, respectively. Also, we will often say a facet and a ridge without specifying a simplicial complex to refer to a subset of size n and a subset of size $n-1$ of $[m]$. We provide an algorithm as follows:

- **Inputs:** A set $\mathcal{F} \subseteq \binom{[m]}{n}$, and a collection $\mathcal{G}$ of affine functions on the subsets of $\mathcal{F}$, called *properties*.
- **Output:** The set of weak pseudo-manifolds K such that $\mathcal{F}(K) \subset \mathcal{F}$ and $g(\mathcal{F}(K)) > 0$ for all $g \in \mathcal{G}$, namely, K *satisfies* all the properties.

2.1 Enumerating weak pseudo-manifolds

In this subsection, we give some computational results which allow us to provide an algorithmic description of how to enumerate weak pseudo-manifolds.

Provided any set of facets $\mathcal{F} = \{F_1, \dots, F_M\}$, we can compute the set $\mathcal{R} = \{r_1, \dots, r_N\}$ of all ridges which come from these facets. We then construct the *ridge-facet incidence matrix* $A(\mathcal{F}) = (a_{i,j})$ of size $N \times M$ as follows:

$$a_{i,j} = \begin{cases} 1 & r_i \subset F_j \\ 0 & \text{otherwise} \end{cases},$$

for $i = 1, \dots, N$ and $j = 1, \dots, M$. A simplicial complex K whose facets are all in some set of facets $\mathcal{F} = \{F_1, \dots, F_M\}$ can be regarded as a *characteristic vector* $K = (k_1, \dots, k_M)^t \in \mathbb{Z}^M$ with

$$k_j = \begin{cases} 1 & F_j \in K \\ 0 & F_j \notin K \end{cases},$$

for $j = 1, \dots, M$. The pure simplicial complex K is a *weak pseudo-manifold* if any ridge of K is in exactly two facets of K. That reflects in the following property:

▶ **Proposition 5.** *Let $\mathcal{F}$ be a set of facets, $A = A(\mathcal{F})$ the ridge-facet incidence matrix of $\mathcal{F}$, and K a pure simplicial complex whose facets are all in $\mathcal{F}$. Then K is a weak pseudo-manifold if and only if the coordinates of the product AK are all in $\{0, 2\}$.*

From that, the characteristic vectors in $\mathbb{Z}_2^M$ of weak pseudo-manifolds are all included in the $\mathbb{Z}_2$-kernel of the matrix A seen as a linear map $A\colon \mathbb{Z}_2^M \to \mathbb{Z}_2^N$.

Let $B = \begin{bmatrix} K_1 & \cdots & K_s \end{bmatrix}$ be a matrix whose columns form a $\mathbb{Z}_2$-basis of $\ker_{\mathbb{Z}_2} A$. Every weak pseudo-manifold K is uniquely expressed as one of the 2^s possible $\mathbb{Z}_2$-linear combinations of $K_1, \dots, K_s$, namely $K = \sum_{i=1}^{s} x_i K_i \pmod 2 = BX$, for $X = (x_1, \dots, x_s)^t \in \mathbb{Z}_2^s$. Restricting the set of facets $\mathcal{F}$ well enough, we are hoping that s will be small. Furthermore, we also can find a suitable basis $\tilde{K}_1, \dots, \tilde{K}_s$ to reduce the number of cases to compute.

We first explain how to construct this basis when the set $\mathcal{F}$ contains all possible facets of $[m]$ and $\mathcal{R}$ all the ridges. There are $\binom{m}{n}$ facets and $\binom{m}{n-1}$ ridges. For a ridge r, we will write as $(AK)_r$ the coordinate of AK corresponding to r. Let us denote by $\mathcal{P}(r) := \{j \in [M] \colon r \subset F_j\}$ the set of the indexes in $\mathcal{F}$ of the facets containing r, called the *parents* of r, which are the only facets contributing to $(AK)_r$. In this first case, any ridge has $m-n+1$ parents. For a kernel matrix B whose row are indexed by $\mathcal{F}$, let us denote by $B_{\mathcal{P}(r)}$ the matrix whose rows are the ones of B taken at indexes $\mathcal{P}(r)$. For every $r \in \mathcal{R}$, for every $t = 1, \ldots, s$, the tth column of $B_{\mathcal{P}(r)}$ has an even number of ones since the basis element K_t has an even number of facets containing r. Performing a mod 2 Gaussian elimination on the columns of $B_{\mathcal{P}(r)}$ yields a matrix of the following form

$$B_{\mathcal{P}(r)} E = \begin{bmatrix} Z_{m-n} & \mathbf{0} \end{bmatrix},$$

with the $(k+1) \times k$-matrix

$$Z_k = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ 1 & \cdots & 1 & 1 \end{bmatrix},$$

for some integer k, and $E \in \mathrm{GL}(s, \mathbb{Z}_2)$ corresponding to the operations performed in the Gaussian elimination. The columns of the new matrix BE corresponds to another basis of the $\mathbb{Z}_2$-kernel of A with a convenient description for which facets containing the ridge r each generator possesses. Only the first $m-n$ ones have facets contributing to $(AK)_r$. Moreover, one can see that taking the mod 2 linear combination of strictly more than two of them would lead to $(AK)_r$ being strictly greater than 2, which is a case we want to avoid computing since we focus on weak pseudo-manifolds, see Proposition 5. Thus this decreases the number of mod 2 combinations containing the first $m-n$ new generators that we need to compute from 2^{m-n} to $\binom{m-n}{0} + \binom{m-n}{1} + \binom{m-n}{2} = 1 + (m-n) + \binom{m-n}{2}$.

By writing $r^1 := r$ and $E_1 := E$, one can inductively repeat the latter process by taking care at step $k+1$ of:

- Choosing each time a new ridge r^{k+1} such that for all $i = 1, \ldots, k, \mathcal{P}(r^i) \cap \mathcal{P}(r^{k+1}) = \emptyset$,
- Starting the Gaussian pivot at columns index $k(m-n)+1$ so that the structure of the generators of previous columns is not lost.

This process terminates at some step $k_{\max}$ whenever one of the former conditions cannot be satisfied. We obtain a final matrix, whose columns are the new basis elements $\tilde{K}_1, \ldots, \tilde{K}_s$, and, up to reordering, whose rows are according to the sets $\mathcal{P}(r^1), \ldots, \mathcal{P}(r^{k_{\max}})$ looks as follows:

$$BE_1 \cdots E_{k_{\max}} = \left[\begin{array}{ccccc} Z_{m-n} & 0 & \cdots & \cdots & 0 \\ 0 & Z_{m-n} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & Z_{m-n} & 0 \\ \hline \star & \star & \star & \star & \star \end{array}\right] = \begin{bmatrix} \tilde{K}_1 & \cdots & \tilde{K}_s \end{bmatrix}.$$

In this case, we decrease the total number of mod 2 combinations from 2^s to $(1 + (m-n) + \binom{m-n}{2})^{k_{\max}} 2^{s - k_{\max}(m-n)}$ since we should take at most 2 basis elements for each block Z_{m-n} and since there remains $s - k_{\max}(m-n)$ generators $\tilde{K}_i$.

As for the general case, there may be ridges having less than $m - n + 1$ parents. In this case, we try to wisely choose some ridges $r^1, \ldots, r^{k_{\max}}$ such that the blocks Z_k are of the maximum possible size so we minimize the number of mod 2 combinations BX of the generators we need to compute. That provides a partition $I_1, \ldots, I_l$ of $\{1, \ldots, s\}$ such that if we are to sum more than two basis elements with indexes in I_k, for $k = 1, \ldots, l$, we are sure not to obtain a weak pseudo-manifold. We can split the vector X in the mod 2 combinations BX as blocks according to this partition: $X = \sum_{k=1}^{l} X_k$, with X_k representing the part of X whose only nonzero coordinates are in I_k. Let us denote by $\mathcal{X}_k$ the set of all such possible X_k, for $k = 1, \ldots, l$.

If we recap our process, given a set of facets $\mathcal{F}$, we constructed

- The ridge-facet incidence matrix A whose $\mathbb{Z}_2$-kernel contains all weak pseudo-manifolds,
- A matrix B whose columns form a convenient basis $\tilde{K}_1, \ldots, \tilde{K}_s$ of $\ker_{\mathbb{Z}_2}(A)$,
- A partition $I_1, \ldots, I_l$ of $\{1, \ldots, s\}$,
- Sets $\mathcal{X}_1, \ldots, \mathcal{X}_l$ of partitions of the vectors of $\mathbb{Z}_2^s$ such that for all $k = 1, \ldots, l$, $X_k \in \mathcal{X}_k$ has a maximum of two nonzero coordinates which are all in I_k ,

such that any weak pseudo-manifold whose facets are in $\mathcal{F}$ is of the form $K = BX$, with $X = \sum_{k=1}^{l} X_k$ for some $(X_1, \ldots, X_l) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_l$, satisfying thatsatisfying that the entries of AK corresponding to the chosed ridges are in $\{0, 2\}$. Moreover, given any affine function $K \mapsto g(K)$, it is easy to check using computer programming that $g(K) > 0$ is verified. We provide in the next subsection some concepts about GPU programming.

2.2 Generalities about GPU programming

In this article, we used Nvidia CUDA [28] whose syntax and vocabulary may differ from other GPU languages. The general idea behind GPU computing is that it allows parallelizing tasks with two layers of parallel programming without needing a supercomputer. Parallel programming takes several forms, and the two we will use are the following:

- Data parallelism: one has a list of elements $\mathcal{X}$ and wants to apply the same function g to every element $X \in \mathcal{X}$. In this case, each call of the function g is independent.
- Task parallelism: one has an element X and wants to apply a set of similar functions $g_1, \ldots, g_k$ on X in order to obtain the result as a list $(g_1(X), \ldots, g_k(X))$. The simplest example is a matrix product AX, and if each row of A is denoted by a_i, then the functions g_i are the inner products with the a_is.

In all that follows, a *thread (of execution)* will be a processing unit that computes machine operations linearly, and a GPU will be a two-layered structure of threads. Namely, a GPU will be a set of p *grids*, and each grid will be a set of q threads. Therefore a GPU can be seen as $p \times q$ threads organized for parallel programming, as in Figure 1. The number $p \times q$ of GPU threads that can run simultaneously is roughly the number of CUDA cores (if we consider Nvidia GPUs) and is around eighteen thousand for the current architectures (as of 2023). Thus a single GPU would be approximately equivalent to at least a thousand CPU threads. In CUDA programming we use this two-layered structure as follows:

- **First layer (blocks):** Let $\mathcal{X} = \{X_1, \ldots, X_N\}$ be the set of data on which we want to apply the same function g, called the *kernel.* We create some list of N *blocks* indexed by an integer i. Each block embodies the function call $g(X_i)$. A block has three possible states: *on hold*, *active*, and *completed.* In the beginning, every block is on hold. Then the p grids of the GPU are filled with some blocks which will be running, these are active, and the rest are waiting to be launched on the grid and remain on hold. Whenever some active block has completed, the GPU replaces it with a block on hold. The program terminates when all blocks are "completed".

- **Second layer (threads):** Whenever we send a block to a grid, the operations made in the block are split into threads using task parallelism, and we distribute any procedures in g into q functions which will run simultaneously on all q threads of the grid. Notice that we need every thread to finish its tasks to obtain the result. We can explicitly require this condition by *synchronizing* the threads.

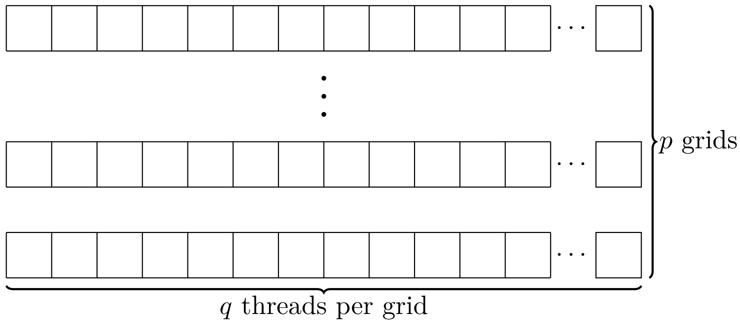

Figure 1 The two layered parallel structure of a GPU.

In all that follows, we will use such notations:

- A set $\mathcal{X}$ will be denoted as a list `list_X`,
- A matrix $A = [a_{i,j}]$ will be represented as an array whose coefficient at index i, j is `A[i][j]`,
- A binary vector $X \in \mathbb{Z}_2^k$ will be represented as a binary variable `x` on k bits.

We will use the following processor instructions on binary variables [28]:

- The *and* operation `x&y`, 64 operations per cycle,
- The *exclusive or* operation `x^y`, 64 operations per cycle,
- The *population count* operation `popcount(x)` which counts the number of "1" bits in the value of `x`, 16-32 operations per cycle.
- Atomic operations, that we use to avoid memory access errors when many threads may want to write at the same memory location concurrently. The processor scheduler creates a queue of all atomic operation calls.

A *cycle* is the shortest time interval considered in a processor unit that it performs at its frequency f: if the frequency is 1GHz the processor realizes 10^9 cycles per second. The thread synchronization allows us to manage how the threads behave in parallel as follows:

- The `syncthreads()` command asks all the threads to wait for all of them to come across the same line in the algorithm code of the kernel.
- For a local thread variable `t`, the `syncthreads_and(t)` and `syncthreads_or(t)` commands allows us to manage the *and* and the *or* operation between all of the `t` variables existing in each thread of a grid. For example, if a thread encounters a condition that should stop the current case in a loop, then all the threads should stop at once since it is useless to compute this case.

2.3 The GPU algorithm for classifying weak pseudo-manifolds

To simplify our explanations, we suppose that the number of generators $s = 64$ and that we can write the product $\mathcal{X}_1 \times \cdots \times \mathcal{X}_l$ as $\mathcal{X}_a \times \mathcal{X}_b$ such that $\mathcal{X}_a$ and $\mathcal{X}_b$ describe the 32 first or last generators, respectively. We thus decompose K as $K_a + K_b$, with $K_a = BX_a$ and $K_b = BX_b$ for every $(X_a, X_b) \in \mathcal{X}_a \times \mathcal{X}_b$. Both vectors X_a and X_b are binary vectors whose nonzero coordinates are in the 32 first or last coordinates, respectively, which we store as 32 bits variables `xa` and `xb`, more precisely as unsigned integers.

The dot product in $\mathbb{Z}$ of the binary forms of two integers x, y is the number of active bits in the & operation: $|x\&y|$. Its mod 2 reduction is the value of its least significant bit: $|x\&y|\&1$.

The main idea of the algorithm is to use M threads to compute each coordinate of $K \in \mathbb{Z}_2^M$, with M being the number of facets in $\mathcal{F}$, as provided in Algorithm 2 whose GPU kernel is given in Algorithm 1.

Algorithm 1 The GPU kernel of Algorithm 2.

```
Shared memory: Integer array r of size N, such that r[k] stores the kth coefficient
  of the product AK.
Global variable: The list list_K in which we store the weak pseudo-manifolds.
Function Kernel(xa,Ka):
    Let i be the local thread index
    b←list_b[i]
    ka←list_Ka[i]
    for xb in list_Xb do
        skip←False
        Ki←(popcount(b&xb)^ka)&1
        syncthreads()
        for g in list_G do
            compute g(K) using the thread values Ki
            if g(K) ≤ 0 then
                skip←True
                break
        if syncthreads_or(skip) then
            continue to the next xb
        Reinitialize each value of r to 0 using the threads
        if Ki=1 then
            for k= 1,...,n do
                increment r[A[k][i]] using the atomic add operation
                if r[A[k][i]] ≥ 3 then
                    skip←True
                    break
        if syncthreads_or(skip) then
            continue to the next xb
        Add K to the list of results list_K
```

▶ Remark 6. When we say *using the threads*, we mean we evenly distribute the operations to perform among the threads. For example, to reinitialize the array r, we use the fact that we have q threads that can set to zero q coordinates simultaneously until all coordinates reset. Thus, it requires $\lceil \frac{N}{q} \rceil$ iteration, for N the number of ridges. We use a similar process for calculating the image by the affine functions $g \in \mathcal{G}$.

▶ Remark 7. We use the atomic add operation for incrementing values in r since many threads may write at the same memory location r[k].

Algorithm 2 The algorithm for classifying weak pseudo-manifolds whose facets are in a facet set $\mathcal{F}$.

Input : The list `list_F`, corresponding to the set of facets $\mathcal{F}$, and the list `list_G`, corresponding to the set of affine functions $\mathcal{G}$.

Output : The list `list_K` of weak pseudo-manifolds `K` with facets in `list_F` and which satisfy `g(K)>0` for every `g` in `list_G`.

Initialization:

Compute the ridge-facet incidence matrix $A = A(\mathcal{F}) \in \mathbb{Z}_2^{N\times M}$ and store it in `A`, a column sparse matrix: `A[k][i]` represents the index of the `k`th nonzero coordinate of the `i`th column of A.

Compute $B = \begin{bmatrix}\tilde{K}_1 & \cdots & \tilde{K}_{64}\end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_M & b_M \end{bmatrix}$ and store it as two lists `list_a` and `list_b` of integers, where `list_a[k]` and `list_b[k]` represent the binary value of the row vectors a_k and b_k, respectively.

Enumerate $\mathcal{X}_a$ and $\mathcal{X}_b$, and store them as two lists `list_Xa` and `list_Xb`.

Create a list `list_Ka` of all the K_as:

for *xa in list_Xa* **do**
  for $k = 1, \dots, M$ **do**
    `Ka[k]`←`popcount(a[k]&xa)&1`

main : Launch the $|\mathcal{X}_a|$ blocks which correspond to all the pairs `(xa,Ka)` on the `Kernel` (Algorithm 1).

The global complexity of this algorithm is:

$$\mathcal{O}\left(\frac{|\mathcal{X}_a|}{p} \times |\mathcal{X}_b| \times \frac{N}{q} \times (\alpha|\mathcal{G}|+1)\right) = \mathcal{O}\left(\frac{|\mathcal{X}_a \times \mathcal{X}_b| \times N \times (\alpha|\mathcal{G}|+1)}{pq}\right),$$

with α representing the average complexity of the atomic operation when called multiple times for a given $g \in \mathcal{G}$.

3 Preparation for applying the algorithm

We refer the reader to the full version of the article for the proofs in this section.

3.1 Finiteness of the problem

Let K be an $(n-1)$-dimensional simplicial complex on $[m] = \{1, 2, \dots, m\}$. The *join* $K * L$ of two simplicial complexes K and L is the simplicial complex $\{\sigma \cup \tau \mid \sigma \in K, \tau \in L\}$. The *link* $Lk_K(\sigma)$ of a face σ in K is the simplicial complex $\{\tau \setminus \sigma \mid \sigma \subset \tau \in K\}$. For the sake of simplicity, we denote the simplicial complex consisting of a single maximal simplex σ by just σ. The (simplicial) *wedge* $\mathrm{wed}_v(K)$ of K at a vertex v is $(I * \mathrm{Lk}_K(v)) \cup (\partial I * K \setminus v)$, where I is a 1-simplex, and $K \setminus v$ is the simplicial complex consisting of the facets of K which do not contain v. We call K a *seed* if K is not a wedge of L for some simplicial complex L. The *suspension* of K is $K * \partial I$ for I some 1-simplex. We can also define it as the wedge of K at some ghost vertex, however we distinguish both here. Therefore, wedging preserves the Picard number while suspending does not.

A *PL manifold* is a simplicial complex such that the link of each of its vertices is a PL sphere. A PL sphere is a PL manifold ([22, Lemma 1.17]). By the definition of a wedge, $\mathrm{wed}_v(K)$ contains an isomorphic copy of K as the link of both new vertices. This observation implies that if $\mathrm{wed}_v(K)$ is a PL sphere, so is K. The converse is also true.

▶ **Proposition 8.** *Let K be a PL sphere and v a vertex of K. Then $\mathrm{wed}_v(K)$ is a PL sphere.*

A *characteristic map* over K is a map $\lambda\colon [m] \longrightarrow \mathbb{Z}^n$ satisfying that for each facet σ of K, the set of integer vectors $\lambda(\sigma)$ is a unimodular, that is the determinant of the associated square matrix is ± 1. A simplicial complex K is *toric colorable* if K admits a characteristic map. One can consider its mod 2 analogue as well. A *mod* 2 *characteristic map* over K is a map $\lambda\colon [m] \longrightarrow \mathbb{Z}_2^n$ such that $\lambda(\sigma)$ is a mod 2 independent set. Similarly, K is $\mathbb{Z}_2^n$*-colorable* if K admits a mod 2 characteristic map.

▶ **Proposition 9** ([16], [13])**.** *Let K be a PL sphere and v a vertex of K. Then K is toric colorable if and only if so is $\mathrm{wed}_v(K)$. In addition, K is $\mathbb{Z}_2^n$-colorable if and only if $\mathrm{wed}_v(K)$ is $\mathbb{Z}_2^{n+1}$-colorable.*

Notice that the composition of a characteristic map over K and mod 2 reduction $\mathbb{Z}^n \to \mathbb{Z}_2^n$ becomes a mod 2 characteristic map over K. As a consequence, we can focus firstly on $\mathbb{Z}_2^n$-colorable seeds.

We often see a mod 2 characteristic map λ as a matrix $\begin{bmatrix}\lambda(1) & \lambda(2) & \cdots & \lambda(m)\end{bmatrix}$. Up to simplicial isomorphisms, we may assume that the facet $\{1, 2, \dots, n\}$ is in K. With this assumption, to check $\mathbb{Z}_2^n$-colorability, it is enough to consider characteristic maps of the form $\lambda = \begin{bmatrix} I_n & A\end{bmatrix}$ since the non-singularity on its facets is preserved by the left multiplication of an element of $\mathrm{GL}(n, \mathbb{Z}_2)$.

Let us define *dual characteristic maps (DCM)* over K. For $\lambda = \begin{bmatrix} I_n & A\end{bmatrix}$, the DCM associated with λ is a map $\bar\lambda\colon [m] \longrightarrow \mathbb{Z}_2^{m-n}$ such that $\bar\lambda = \begin{bmatrix}\bar\lambda(1) & \bar\lambda(2) & \cdots & \bar\lambda(m)\end{bmatrix}^t = \begin{bmatrix} A \\ I_{m-n}\end{bmatrix}$. We shorten the term injective DCM to IDCM.

▶ **Theorem 10** ([14])**.** *Let K be an $(n-1)$-dimensional PL sphere with m vertices and v, w distinct vertices of K. Then the following are true.*

1. *If every facet of K contains v or w, then K is a wedge or a suspension with respect to v and w.*
2. *If K is a seed that is not a suspension, then every DCM over K must be an IDCM.*

Statements (2) and (3) imply:

resume *If K is a seed and $m - n \geq 3$ then $m \leq 2^{m-n} - 1$.*

We will call a seed that is not a suspension a *regular* seed. We conclude from Statement 3. of Theorem 10 that there are only finitely many $\mathbb{Z}_2^n$-colorable seeds to enumerate for a fixed integer $p := m - n$, that we match here with the Picard number of K written $Pic(K)$. For $p \leq 3$, it is known that PL spheres all are boundaries of polytopes [25] and were already completely enumerated by Perles [20]. We easily check the $\mathbb{Z}_2^n$-colorability by an algorithm described in [17], and verify the seedness with the following lemma.

▶ **Lemma 11** (seedness)**.** *A PL sphere K is a seed if and only if it has no edge $\{v, w\}$ such that every facet of K has either v or w.*

We now focus on the case $p = 4$. By Statement 3. of Theorem 10, we have $n \leq 11$, implying that it is enough to enumerate PL spheres of dimension up to 10 ($n = 11$). Since PL spheres of Picard number ≥ 4 are not necessarily polytopal [5], we cannot make use of the Gale diagram. We thus need to put our concern on checking PL sphereness.

3.2 Collecting PL spheres among weak pseudo-manifolds

We first have a compelling criterion for a PL manifold to be a PL sphere when its Picard number is small enough.

▶ **Theorem 12** ([4]). *Let K be a PL manifold with* $\mathrm{Pic}(K) \le 7$*. If K is a $\mathbb{Z}_2$-homology sphere, then K is a PL sphere.*

By using the above theorem, we obtain the following lemma.

▶ **Lemma 13** (PL sphereness). *A weak pseudo-manifold K with Pic(K) ≤ 7 is a PL sphere if and only if the link of any face (possibly the empty face) of K is a $\mathbb{Z}_2$-homology sphere.*

We now have the tools for checking:

- The PL sphereness of a weak pseudo-manifold of Picard number 4 with Lemma 13
- The seedness of a PL sphere with Lemma 11.

4 Toric colorable PL spheres of Picard number 4

We devote this section to enumerate all $(n-1)$-dimensional toric colorable seeds of Picard number 4. One could intuitively try to input Algorithm 2 with all n subsets of $[m]$, check PL sphereness, $\mathbb{Z}_2^n$-colorability, and seedness. However, it is hopeless when we consider high dimensions. Indeed, we manage to obtain results up to $n = 6$, but the program takes too long to finish with $n = 7$. On the other hand, Theorem 10 states that there are only two kinds of $\mathbb{Z}_2^n$-colorable seeds: regular or suspended. At first, we consider how to enumerate the regular seeds which only supports IDCM.

We recall here some definitions from matroid theory. A *matroid* M is a simplicial complex with the so-called *augmentation property*: for any $\tau, \sigma \in M$ with $|\tau| < |\sigma|$, there exists $x \in \sigma \setminus \tau$ such that $\tau \cup \{x\} \in M$. Although the facets of a matroid are called the *bases*, we will keep the simplicial complex terminology here and call them facets. The *dual matroid* $\overline{M}$ of a matroid M is the matroid on the same vertex set as M and whose facets are the complement of each facet of M, called *cofacets* of M. For an $n \times m$ matrix λ over $\mathbb{Z}_2$ of full row rank n, the simplicial complex M_λ whose facets are the sets of column indexes of n independent columns of λ is a matroid, called the *binary matroid* associated to λ. Then we can rephrase that a pure simplicial complex K supports a mod 2 characteristic map λ as K is in the binary matroid associated to λ. By linear Gale duality [16], the dual $\overline{M_\lambda}$ is equal to $M_{\bar{\lambda}^t}$. We effortlessly verify the following proposition by the definition of M_λ and $M_{\bar{\lambda}^t}$.

▶ **Proposition 14.** *Let K be an $(n-1)$-dimensional simplicial complex on $[m]$ and $\bar{\lambda}$ an $m \times (m-n)$ matrix over $\mathbb{Z}_2$ of rank $m-n$. Then K supports $\bar{\lambda}$ as a DCM if and only if it is a subcomplex of $\overline{M_{\bar{\lambda}^t}} = M_\lambda$.*

Recall that the more we reduce the number of facets in the input of Algorithm 2, the smaller the dimension of the mod 2 kernel of the ridge-facet incidence matrix will be, and the faster the algorithm will run. Proposition 14 gives us what we want: a finer set of facets. In addition, we take advantage of the upper bound theorem ([29]): the number of facets of a simplicial sphere is less than or equal to the one of the cyclic n-polytope $C(m,n)$ with m vertices. This condition is embodied by the following affine function: $g(K) = f_{n-1}(C(m,n)) - \|K\|_1 + 1$. Fix an injective map $\bar{\lambda} \colon [m] \longrightarrow \mathbb{Z}_2^4$ and set $\mathcal{F}(\lambda) = \mathcal{F}(M_\lambda)$. Algorithm 2 with inputs being the set of facets $\mathcal{F}(\lambda)$ and the affine function g outputs the set of all weak pseudo-manifolds which support $\bar{\lambda}$ and satisfy the upper bound theorem.

At first sight, it seems that we need to run the algorithm on each of the $\binom{11}{n} \times (n!)$ injective maps $\bar{\lambda}$ even if we fix $\begin{bmatrix}\bar{\lambda}(n+1) & \bar{\lambda}(n+2) & \bar{\lambda}(n+3) & \bar{\lambda}(n+4)\end{bmatrix} = I_4$. However, we will drastically reduce this large number of cases to compute by noticing that many injective maps provide the same outputs up to simplicial isomorphism.

Let $\Lambda(n,p)$ be the set of all $(n+p)\times p$ matrices over $\mathbb{Z}_2$ of the form $\begin{bmatrix} A \\ I_p \end{bmatrix}$ such that each matrix has no repeated rows. Consider the product of two symmetric groups $\mathfrak{S}_n \times \mathfrak{S}_p$. This group gives a group action on $\Lambda(n,p)$ by $\left(\begin{bmatrix} A \\ I_p \end{bmatrix}, (s,t)\right) \mapsto \begin{bmatrix} P_s^t A P_t \\ I_p \end{bmatrix}$, where P_s and P_t are column permutation matrices corresponding to s and t. Let us call each element of $\Lambda(n,p)/\mathfrak{S}_n \times \mathfrak{S}_p$ an *IDCM orbit*.

▶ **Proposition 15.** *For $(s,t) \in \mathfrak{S}_n \times \mathfrak{S}_p$, there is a simplicial isomorphism between binary matroids associated to $\bar{\lambda} \in \Lambda(n,p)$ and $\bar{\lambda} \circ (s,t) \in \Lambda(n,p)$.*

Let $\Lambda^\circ(n,4) \subseteq \Lambda(n,4)$ be a set containing one representative of each IDCM orbit. By Proposition 15, it is enough to input Algorithm 2 with $\mathcal{F}(\lambda)$, for all $\lambda \in \Lambda^\circ(n,4)$. Table 1 shows the number of IDCM orbits of $\Lambda(n,4)$ and the computation time of Algorithm 2.

Table 1 Data table for Picard number 4 and $n = 2, \dots, 11$. The time spent refers to by Algorithm 2 running on an Nvidia Quadro A5000. *Estimated.

n	2	3	4	5	6	7	8	9	10	11
Number of IDCM orbits	7	16	28	35	35	28	16	7	3	1
$\max_{\bar{\lambda}}(\dim\ker A(\mathcal{F}(\lambda)))$	7	13	21	24	28	34	42	48	56	64
$\max_{\bar{\lambda}} \lvert\mathcal{X}(\bar{\lambda})\rvert$	56	3e3	5e5	1e6	2e7	9e8	1e11	3e12	4.4e14	4.2e16
Time spent for one orbit	1ms	10ms	0.1s	0.6s	1.3s	3m	15m	2h	12d	3y*

We now provide the global strategy for enumerating all $\mathbb{Z}_2^n$-colorable seeds of Picard number 4 by recapping the case of regular seeds and explaining the one of suspended seeds. This allows us to obtain Theorem 2.

Strategy.

- **CASE I: Regular seeds.** For every representative $\lambda \in \Lambda^\circ(n,4)$, run Algorithm 2 with inputs being the set of facets $\mathcal{F}(\lambda)$ and the affine function $g(K) = f_{n-1}(C(m,n)) - \|K\|_1 + 1$. After reducing isomorphic ones, we obtain the list of $\mathbb{Z}_2^n$-colorable weak pseudo-manifolds on $[m]$ satisfying the upper bound theorem up to isomorphism. We then apply Lemma 11 and Lemma 13 to collect the seeds.
- **CASE II: Suspended seeds.** From Theorem 10, a $\mathbb{Z}_2^n$-colorable seed without IDCM is a suspension. From the definition of a wedge, the suspension of a wedge is again a wedge, and the suspension operation increases the Picard number by one. Therefore, it is enough to consider suspensions of seeds of Picard number 3.
  Let $L = \partial[v,w] * K$ for an $(n-2)$-dimensional simplicial complex K, and λ a characteristic map over L. Without loss of generality, we may assume $v = 1$ so that $\lambda(v) = \begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}^t$. Then for any facet $\{v\} \cup \{v_1, \dots, v_{n-1}\}$ of L, the $(1,1)$ minor of the matrix $\begin{bmatrix}\lambda(v) & \lambda(v_1) & \cdots & \lambda(v_{n-1})\end{bmatrix}$ is equal to 1. This implies that $\mathrm{Lk}_L(1) = K$ is $\mathbb{Z}_2^{n-1}$-colorable.
  We know there are three $\mathbb{Z}_2^n$-colorable seeds of the Picard number 3; 5-gon, 3-cube, and the cyclic polytope $C(7,4)$. The suspension of the 3-cube does not support any IDCM but does support a DCM, while the suspensions of the others support IDCM.

References

1 A. Altshuler, J. Bokowski, and L. Steinberg. The classification of simplicial 3-spheres with nine vertices into polytopes and nonpolytopes. *Discrete Math.*, 31:115–124, 1980. doi:10.1016/0012-365X(80)90029-1.

2 A. Altshuler and L. Steinberg. An enumeration of combinatorial 3-manifolds with nine vertices. *Discrete Math.*, 16:91–108, 1976. doi:10.1016/0012-365X(76)90138-2.

3 Amos Altshuler and Leon Steinberg. The complete enumeration of the 4-polytopes and 3-spheres with eight vertices. *Pac. J. Math.*, 117:1–16, 1985. doi:10.2140/pjm.1985.117.1.

4 Bhaskar Bagchi and Basudeb Datta. Combinatorial triangulations of homology spheres. *Discrete Math.*, 305(1):1–17, 2005. doi:10.1016/j.disc.2005.06.026.

5 D. W. Barnette. A simple 4-dimensional nonfacet. *Isr. J. Math.*, 7:16–20, 1969. doi:10.1007/BF02771742.

6 R. Bowen and S. Fisk. Generation of triangulations of the sphere. *Math. Comput.*, 21:250–252, 1967. doi:10.2307/2004172.

7 Gunnar Brinkmann. Fast generation of planar graphs. *MATCH Commun. Math. Comput. Chem.*, 58(2):323–357, 2007.

8 Gunnar Brinkmann and Brendan D. McKay. Fast generation of some classes of planar graphs. *Electron. Notes Discret. Math.*, 3:28–31, 1999. doi:10.1016/S1571-0653(05)80016-2.

9 J. M. Brückner. Geschichtliche Bemerkungen zur Aufzählung der Vielflache. Pr. (No. 578) Realgymn. Zwickau. 19 S. 4° + 7 S. Taf (1897), 1897.

10 M. Brückner. Über die Anzahl $\psi(n)$ der allgemeinen Vielflache. Atti Congresso Bologna 4, 5-11 (1931), 1931.

11 M. M. Brückner. Bemerkungen zur Morphologie der außergewöhnlichen Polyeder, erläutert durch die Sechsflache. Rom. 4. Math.-Kongr. 2, 293-295 (1909), 1909.

12 Suyoung Choi, Hyeontae Jang, and Mathieu Vallée. The characterization of $(n-1)$-spheres with $n+4$ vertices having maximal buchstaber number, 2023. arXiv:2301.00806.

13 Suyoung Choi and Hanchul Park. Wedge operations and torus symmetries. *Tohoku Math. J. (2)*, 68(1):91–138, 2016. URL: http://projecteuclid.org/euclid.tmj/1458248864.

14 Suyoung Choi and Hanchul Park. Wedge Operations and Torus Symmetries II. *Canad. J. Math.*, 69(4):767–789, 2017. doi:10.4153/CJM-2016-037-4.

15 N. Yu. Erokhovets. Moment-angle manifolds of simple n-dimensional polytopes with $n+3$ facets. *Uspekhi Mat. Nauk*, 66(5(401)):187–188, 2011. doi:10.1070/RM2011v066n05ABEH004767.

16 Günter Ewald. *Combinatorial convexity and algebraic geometry*, volume 168. Springer Science & Business Media, 1996.

17 Anne Garrison and Richard Scott. Small covers of the dodecahedron and the 120-cell. *Proc. Amer. Math. Soc.*, 131(3):963–971, 2003. doi:10.1090/S0002-9939-02-06577-2.

18 Donald W. Grace. *Computer search for non-isomorphic convex polyhedra.* PhD thesis, Stanford University, USA, 1965. URL: https://searchworks.stanford.edu/view/2043380.

19 Jörg Gretenkort, Peter Kleinschmidt, and Bernd Sturmfels. On the existence of certain smooth toric varieties. *Discrete Comput. Geom.*, 5(3):255–262, 1990. doi:10.1007/BF02187789.

20 Branko Grünbaum. *Convex polytopes*, volume 221 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, second edition, 2003. Prepared and with a preface by Volker Kaibel, Victor Klee and Günter M. Ziegler. doi:10.1007/978-1-4613-0019-9.

21 Branko Grünbaum and V. P. Sreedharan. An enumeration of simplicial 4-polytopes with 8 vertices. *J. Comb. Theory*, 2:437–465, 1967. doi:10.1016/S0021-9800(67)80055-3.

22 J.F.P. Hudson and J.L. Shaneson. *Piecewise Linear Topology.* Mathematics lecture note series. W. A. Benjamin, 1969. URL: https://books.google.fr/books?id=UAY_AAAAIAAJ.

23 Peter Kleinschmidt. A classification of toric varieties with few generators. *Aequationes Math.*, 35(2-3):254–266, 1988. doi:10.1007/BF01830946.

24 Frank H. Lutz. Enumeration and random realization of triangulated surfaces. In *Discrete differential geometry*, pages 235–253. Basel: Birkhäuser, 2008.

25 P. Mani. Spheres with few vertices. *J. Comb. Theory, Ser. A*, 13(3):346–352, 1972. `doi:10.1016/0097-3165(72)90068-4`.

26 Talha Bin Masood, Tathagata Ray, and Vijay Natarajan. Parallel computation of alpha complexes for biomolecules. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.SoCG.2020.17`.

27 Hiroyuki Miyata and Arnau Padrol. Enumerating neighborly polytopes and oriented matroids. *Exp. Math.*, 24(4):489–505, 2015. `doi:10.1080/10586458.2015.1015084`.

28 NVIDIA Corporation. NVIDIA CUDA C programming guide, 2023. Version 12.3. URL: `https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html`.

29 Richard P. Stanley. *Combinatorics and commutative algebra*, volume 41 of *Progress in Mathematics*. Birkhäuser Boston, Inc., Boston, MA, second edition, 1996.

30 Thom Sulanke and Frank H. Lutz. Isomorphism-free lexicographic enumeration of triangulated surfaces and 3-manifolds. *Eur. J. Comb.*, 30(8):1965–1979, 2009. `doi:10.1016/j.ejc.2008.12.016`.

31 Mathieu Vallée. GPU algorithm for enumerating weak pseudomanifolds. Software, swhId: `swh:1:dir:ecc6f4940dc55c6a4175e78f2cfa47ccfd3ce009`, (visited on 21/05/2024). URL: `https://github.com/MVallee1998/GPU_handle`.

32 Fan Wang, Hubert Wagner, and Chao Chen. Gpu computation of the euler characteristic curve for imaging data. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 64:1–64:16, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.SoCG.2022.64`.