DOI 10.12694/scpe.v14i1.823
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 14, Number 1, pp. 5–15. http://www.scpe.org © 2013 SCPE

GENERATION OF CATALOGUES OF PL N-MANIFOLDS: COMPUTATIONAL ASPECTS ON HPC SYSTEMS

ALESSANDRO MARANI[*], MARZIA RIVI[*] AND PAOLA CRISTOFORI[†]

Abstract. Within mathematical research, Geometric Topology deals with the study of piecewise-linear n-manifolds, i.e. triangulable spaces which appear locally as the n-dimensional Euclidean space. This paper reports on the computational aspects of an algorithm for generating triangulations of PL 3- and 4-manifolds represented by edge-coloured graphs. As the number of graph vertices is increased the algorithm becomes computationally expensive very quickly, making it a natural candidate for the usage of HPC resources. We present an optimized, parallel version of the algorithm that is suitable for deployment of multi-core systems. Scalability results are discussed on two different platforms, namely an IBM iDataPlex Linux cluster and the IBM supercomputer BlueGene/Q.

Key words: High Performance Computing, n-manifolds, coloured triangulations, edge-coloured graphs.

AMS subject classifications. 57Q15 - 57M15 - 68W10.

1. Introduction. Geometric Topology deals with piecewise-linear (PL) n-manifolds [2], i.e. compact topological manifolds for which there is a triangulation such that each point has a neighbourhood which is piecewise-linearly isomorphic (*PL-homeomorphic*) to an affine n-simplex. Catalogues of triangulations of PL n-manifolds are valuable sources of data: they yield examples to test conjectures, calculate invariants and make comparisons; they can offer insight into the structural properties of the represented manifolds and may suggest ideas for further theoretical investigation.

In particular, since each compact (topological) 3-manifold admits a PL structure and any two PL structures on the same topological 3-manifold are equivalent (i.e. PL-homeomorphic, see [2]), the study of triangulations of PL 3-manifolds is naturally related to the problem of classification, which is still one of the main topics of 3-dimensional topology. The possibility of representing manifolds by combinatorial structures, together with recent advances in computing power, enabled topologists to construct exhaustive tables of small (i.e. obtained by a small number of simplices) 3-manifolds based on different representation methods. In the closed case (i.e. compact and without boundary), catalogues have already been produced and analysed by many authors [8, 19, 20], with a particular focus on combinatorial properties of minimal triangulations.

On the other hand, the problem of classification in dimension four must take into account that a topological 4-manifold not always admits PL structures or may admit non-equivalent ones. For example, although there exists a classification of simply-connected topological 4-manifolds, long established by Freedman [7], the study of (PL) equivalence classes of such structures, especially with regard to their minimal representatives, is an interesting and still open subject of research. Several examples of different PL 4-manifolds triangulating the same topological 4-manifold have recently been presented and the subject is being continuously updated (see for example [9]), but no exhaustive catalogue is available, yet.

Crystallization theory is a representation method for PL n-manifolds by means of a particular class of edge-coloured graphs (*gems*, i.e. Graphs Encoding Manifolds), which are dual 1-skeletons of vertex-coloured (pseudo)triangulations. Topological properties of a manifold are thus encoded in the combinatorial structure of its crystallizations, which are particularly suitable for computer manipulation. Furthermore, crystallization theory allows to develop an algorithmic approach to the generation of censuses of triangulations of compact PL n-manifolds represented by edge-coloured graphs. Unfortunately, the scope of tables of triangulations of n-manifolds is limited by the significant amounts of time required to generate them: in general, a census of triangulations (coloured triangulations are no exception) formed by t n-simplices requires computing time at least exponential in t.

The problem of reducing the generation time can be faced from a topological point of view by excluding some typical graph configurations, which can be removed without affecting the list of represented manifolds (as it happens with dipoles and ρ-pairs, see Sect. 2) or which forbid the graph to represent a manifold, such as those

[*] Department of SuperComputing Applications and Innovation, CINECA, via Magnanelli 6/3, 40033 Casalecchio di Reno, Bologna, Italy

[†] Department of Computer, Mathematical and Physical Sciences, University of Modena and Reggio Emilia, via Campi 213/B, 41125 Modena, Italy

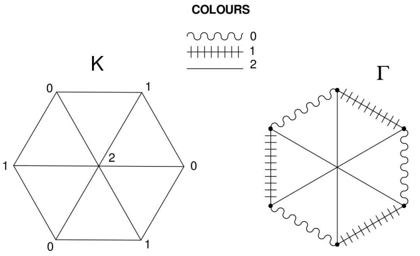


Figure 2.1. *Representation of a 2-dimensional torus by a triangulation and its corresponding edge-coloured graph.*

not satisfying Eqs. 3.2 and 3.3 in Sect. 3. As a consequence, positive effects on the generating algorithm come from a sort of "branch and bound" technique. Another direction of improvement relies on the parallelization of the algorithm in order to exploit high performance computing (HPC) infrastructures, which is the focus of the work presented in this paper.

The usage of supercomputing resources has allowed the generation of catalogues of 3-manifold crystallizations with up to 32 vertices, which have already been completely classified up to 30 vertices (both for orientable [13] and non-orientable [14] cases), and the generation of catalogues of 4-manifold crystallizations up to 20 vertices, for which some classification results have already been obtained from their analysis, as reported in [16].

This paper discusses the strategy adopted to optimize and parallelize the generation algorithm and shows performance results on different architectures for the 3- and 4-dimensional cases. In Sect. 2 we introduce the concepts and summarise results from crystallization theory which are required in the development of algorithmic procedures for the generation of PL n-manifolds censuses. A sketch of the generation algorithm is provided in Sect. 3, where specific conditions are detailed for dimensions 3 and 4. Section 4 contains a description of the parallelization strategy implemented, while Sect. 5 shows scalability results on two specific HPC platforms.

2. Representation of PL-manifolds by edge-coloured graphs. For basic PL-topology, topology of 3- and 4-manifolds and elementary notions about graphs, we refer to [1, 3, 4, 6]. For surveys about crystallization theory see [10, 11, 12]. All manifolds are assumed to be closed and connected, unless explicitly mentioned.

A *coloured triangulation* of a PL n-manifold M is a triangulation of M by means of a pseudo-complex whose vertices are labelled by the integers $\{0, \ldots, n\}$, so that vertices of the same simplex have different labels. The dual 1-skeleton of a coloured triangulation K of M is a (multi)graph $\Gamma = (V(\Gamma), E(\Gamma))$ whose edges inherit a coloration from K: an edge e of Γ is coloured c if and only if c is the missing colour in the vertices of the $(n-1)$-simplex of K dual to e. In this case, we say that Γ *represents* M or is a *gem* (Graph Encoding Manifold) of M. It is easy to see that M is orientable if and only if Γ is bipartite.

Note that, as a result of the above construction, the coloration of the elements of $E(\Gamma)$ is injective on each pair of adjacent edges. Any regular graph of degree $n+1$ equipped with such an edge-coloration is called an $(n+1)$-coloured graph (without boundary). The elements of the set $\Delta_n = \{0, 1, \ldots, n\}$ are called *colours*; moreover, for each $i \in \Delta_n$, we denote by $\Gamma_{\hat{i}}$ the n-coloured graph obtained from Γ by deleting all edges coloured by i. Given an $(n+1)$-coloured graph we can always construct a coloured pseudo-complex $K(\Gamma)$ by taking an n-simplex for each vertex of Γ, by colouring its vertices by Δ_n and, for each $i \in \Delta_n$, by identifying the $(n-1)$-faces of two n-simplices σ and σ' opposite to their i-coloured vertices if and only if the corresponding vertices of Γ are i-adjacent (see Fig. 2.1 for an example).

The above constructions can be generalised in order to take into account coloured triangulations of manifolds with non-empty boundary. In this case the dual 1-skeletons with the inherited edge-coloration miss some n-coloured edges and are called $(n+1)$-coloured graphs *with boundary.* If Γ is such a graph, its *boundary graph* $\partial\Gamma$ is an n-coloured graph (without boundary) whose associated pseudo-complex is the boundary of $K(\Gamma)$.

Most of the following definitions and results can be easily generalised to the boundary case but, for sake of simplicity, we will restrict them to the closed one and, except when explicitly pointed out, we will consider only graphs without boundary. We say that a $(n+1)$-coloured graph Γ is *contracted* when its associated

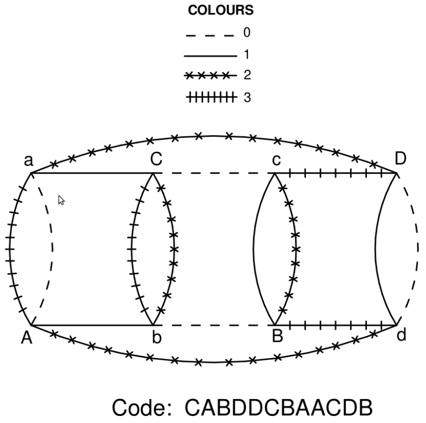


FIGURE 2.2. *A crystallization of the orientable $\mathbb{S}^2$-bundle over $\mathbb{S}^1$ together with its code and the vertex-labelling and coloration associated to it.*

triangulation $K(\Gamma)$ has the minimal number of vertices, i.e. $n+1$. A contracted gem of a PL n-manifold M is called a *crystallization* of M. Contractedness can be checked directly on the coloured graph. In fact duality establishes a bijective correspondence between the connected components of $\Gamma_{\hat{i}}$ and the i-coloured vertices of the associated triangulation for each $i \in \Delta_n$. As a consequence, a gem Γ of a n-manifold M is a crystallization of M if and only if the graph $\Gamma_{\hat{i}}$ is connected for each $i \in \Delta_n$.

Characterising gems of PL n-manifolds among $(n+1)$-coloured graphs is a problem which, by the following result, is strictly related to the recognition of gems of $(n-1)$-spheres.

PROPOSITION 2.1. [10] *An $(n+1)$-coloured graph Γ represents a PL n-manifold iff, for each $i \in \Delta_n$, all connected components of $\Gamma_{\hat{i}}$ represent $(n-1)$-spheres.*

Classical results presented in [10] guarantee that each n-manifold admits a crystallization; obviously, it generally admits many of them and it is a basic problem how to recognise crystallizations (or, more generally, gems) of the same manifold. The easiest case is when two gems are *colour-isomorphic*, i.e. there exists an isomorphism between the graphs, which preserves colours up to a permutation of Δ_n. It is quite trivial to check that two colour-isomorphic gems produce the same polyhedron. The following result assures that colour-isomorphic graphs can be effectively detected by means of a suitably defined numerical *code* [5], which can be directly computed for each of them (see [5, 17] for the related *rooted numbering algorithm* and the example shown in Fig. 2.2).

PROPOSITION 2.2. [17] *Two gems are colour-isomorphic iff their codes coincide.*

Furthermore the code can be used to represent numerically the coloured triangulations in order to manipulate them by computers. Actually it is the most efficient way to represent a coloured graph not only from the point of view of avoiding duplicates of the same triangulation, but also because it contains only the essential information which allow to reconstruct the graph (other kind of representations such as the incidence matrix, in fact, are mainly redundant). Figure 2.2 shows an example of code for the orientable case: the string of the code displays the i-adjacencies of the vertices labelled by small letters for $i \in \{1, \dots, n\}$. In the non-orientable case the string would be longer since it ought to contain also the n-adjacencies of the capital letters.

The problem of recognising non-colour-isomorphic gems representing the same manifold has been solved too, but not algorithmically. A finite set of moves - the so called dipole moves - has been defined such that two gems represent the same manifold if and only if they can be related by a finite sequence of such moves [18].

DEFINITION 2.3. *An h-dipole $\theta = (x, y)$ in a $(n+1)$-coloured graph Γ is a subgraph consisting of two vertices x and y connected by h edges coloured by $c_1, \dots, c_h$, such that x and y belong to different connected components of the graph $\Gamma_{\hat{c}_1 \dots \hat{c}_h}$ obtained by deleting all edges of Γ coloured by $c_1, \dots, c_h$.*

By deleting the vertices of a h-dipole θ from a $(n+1)$-coloured graph Γ and pasting together the hanging edges according to their colours (see Fig. 2.3), we obtain a new $(n+1)$-coloured graph Γ'. The transformation from

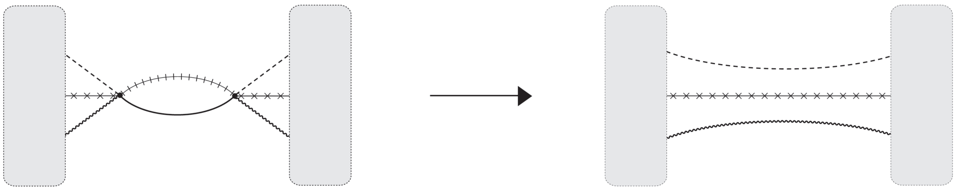


Figure 2.3. *Dipole move*

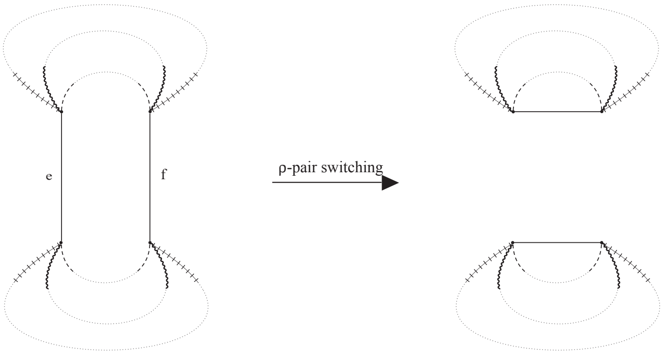


Figure 2.4. *Switching of a ρ-pair.*

Γ to Γ' is called the *cancellation of* θ, its inverse the *addition of* θ. Both are called *dipole moves.* Neither cancellations nor additions of a dipole change the represented manifold [18], so dipole moves are an easy tool for manipulating gems without changing the PL-homeomorphism type of the underlying manifold.

Another useful kind of moves relies on the concept of ρ_h-pairs [5, 15]:

Definition 2.4. *A pair* (e, f) *of distinct i-coloured edges in a* $(n + 1)$*-coloured graph* Γ *is said to form a* ρ_h*-pair iff e and f belong both to exactly h common bi-coloured cycles of* Γ.

ρ-pairs can be eliminated by *switching* (see Fig. 2.4). The following proposition shows the effect of switching on the represented manifold:

Proposition 2.5. [15] *Let* Γ *be a crystallization of a (connected) n-manifold M,* $n \geqslant 3$ *and let* Γ' *be obtained by switching a* ρ_h*-pair in* Γ.

(a) If $h = n - 1$ *then* Γ' *is a crystallization of M.*

(b) If $h = n$ *then* Γ' *is a crystallization of an n-manifold* M' *such that* $M \cong M'\#(\mathbb{S}^{n-1} \otimes \mathbb{S}^1)^1$.

A $(n + 1)$-coloured graph without ρ_{n-1}- and ρ_n-pairs is called *rigid.* The restriction to the class of rigid crystallizations does not affect the set of represented PL-manifolds, as the following result proves.

Proposition 2.6. [15] *Each closed connected PL n-manifold M admits a rigid crystallization. Moreover, if M is handle-free (i.e. there is no* M' *such that* $M \cong M'\#(\mathbb{S}^{n-1} \otimes \mathbb{S}^1)$*), it admits a rigid crystallization of minimal order.*

3. Generation algorithms.

By Proposition 2.1, generation of catalogues of all PL n-manifolds represented by edge-coloured graphs with a fixed number of vertices requires:

- to proceed inductively on dimension n;
- to perform sphere recognition at each step.

In fact, the input data of the generation algorithm in dimension n are the codes of all gems (not necessarily crystallizations) representing $(n - 1)$-spheres, which have to be generated previously. To each of these graphs, n-coloured edges are added in all possible ways so as to obtain a crystallization of a closed n-manifold (possible attachments are therefore limited by topological constraints). This algorithm becomes computationally very intensive as the number of vertices of the graphs grows, so it has been necessary to improve its efficiency. The

[1] $\mathbb{S}^{n-1} \otimes \mathbb{S}^1$ denotes the orientable or the non-orientable (according to the orientability of M and M') $\mathbb{S}^{n-1}$-bundle over $\mathbb{S}^1$.

theoretical results presented in Sect. 2 allow to exclude "a priori" a large number of possible configurations. In particular, by Proposition 2.6 and the cited results about dipole moves, we can restrict the catalogues to rigid crystallizations with no dipoles[2].

Let p be a positive integer, we will denote by $\mathcal{C}_n^{(2p)}$ (resp. $\tilde{\mathcal{C}}_n^{(2p)}$) the catalogue of all non-isomorphic rigid bipartite (resp. non-bipartite) crystallizations of PL n-manifolds with $2p$ vertices and lacking in dipoles. Because of contractedness and rigidity, the starting set of the generation algorithm will take into account only connected n-coloured graphs representing the $(n-1)$-sphere with no ρ_{n-1}-pairs. We will denote this set by $\mathcal{S}_n^{(2p)}$. Let $\bar{\Gamma}$ be an $(n+1)$-coloured graph with boundary obtained from an element of $\mathcal{S}_n^{(2p)}$ by addition of n-coloured edges, then $\bar{\Gamma}$ will be kept for further additions if and only if:

(i) it contains no $n-1$ edges with the same endpoints (otherwise there will be ρ-pairs in the resulting completed graph);

(ii) for each $i \in \{0, \ldots, n-1\}$, $\bar{\Gamma}_{\hat{i}}$ represents an $(n-1)$-sphere with holes.

The above described restrictions allow to prune the generation tree and succeed in reducing considerably both the computation time and the size of the resulting catalogues, by keeping only essential triangulations. The outline of the algorithm is presented below.

```
Input: $\mathcal{S}_n^{(2p)}$
new $\mathcal{C}_n^{(2p)}$, $\tilde{\mathcal{C}}_n^{(2p)} = \emptyset$;
for each $\Sigma \in \mathcal{S}_n^{(2p)}$ do
  new pair $(\Gamma, \partial\Gamma) = (\Sigma, \Sigma)$;
  set $queue = \emptyset$ ;
  $(\Gamma, \partial\Gamma) \to queue$;
  while $queue \neq \emptyset$ do
    get $queue.firstElement(\bar{\Gamma}, \partial\bar{\Gamma})$;
    if $\partial\bar{\Gamma} = \emptyset$ then
      if $\bar{\Gamma}$ is rigid and contracted then
        if $\bar{\Gamma}$ is bipartite then $\bar{\Gamma} \to \mathcal{C}_n^{(2p)}$;
        else $\bar{\Gamma} \to \tilde{\mathcal{C}}_n^{(2p)}$;
        end if
      end if
    else $v =$ random vertex of $\bar{\Gamma}$;
      new $ver = (w_0, w_1, \ldots, w_k)$ array of vertices which can be joined to $v$;
      for each $w_i \in ver$ do
        if $v,\ w_i$ have not $n-1$ common edges then
          new edge $e = n$-coloured edge joining $v$ and $w_i$;
          new pair $(\Gamma', \partial\Gamma') = (\bar{\Gamma} \cup \{e\}, \partial\bar{\Gamma} \setminus \{v, w_i\})$;
          if $\Gamma'$ satisfies condition (ii) then
            $(\Gamma', \partial\Gamma') \to queue$;
          end if
        end if
      end for
    end if
  end while
end for
Output: $\mathcal{C}_n^{(2p)}$, $\tilde{\mathcal{C}}_n^{(2p)}$
```

The algorithms for generation of catalogues of triangulations in dimension 3 and 4 have been sufficiently developed from the theoretical point of view and have been implemented in C++ programs. Although they share the approach described above, different combinatorial conditions have to be implemented in order to realise the conditions on Γ' and $\bar{\Gamma}$.

[2] In dimension three, contractedness and rigidity assure that dipoles do not appear. In higher dimensions the absence of dipoles must be checked.

3.1. Dimension three. Dimension three has great advantages from the computational point of view, because:

- the generation of the set $\mathcal{S}_3^{(2p)}$ can be performed by a very efficient recursive algorithm based on a result by Lins [5]: all elements of $\mathcal{S}_3^{(2p)}$ are obtained from those of $\mathcal{S}_3^{(2p-2)}$ by means of the *antifusion of an edge*[3] except for the 1-skeleton of a prism, which appears only when p is even;
- 2-spheres with and without holes can be easily and efficiently recognised by computing their Euler characteristic directly on the representing graphs;
- the zero Euler characteristic identifies closed 3-manifolds.

In particular, the conditions to be checked in the generation algorithm become as follows: a 4-coloured graph Γ *without* boundary is a rigid crystallization of a closed 3-manifold if and only if it has no ρ-pairs and

$$\sum_{i,j\in\Delta_3} g_{ij} - 2p - 4 = 0, \tag{3.1}$$

where g_{ij} is the number of $\{i,j\}$-coloured cycles of Γ; while a 4-coloured graph Γ' *with* boundary satisfies condition (ii) if and only if for each $r \in \{0,1,2\}$, we have

$$2g_{\hat{r}} - {}^{\partial}g_{\hat{r}} = \sum_{i,j\in\Delta_3-\{r\}} \dot{g}_{ij} - m, \tag{3.2}$$

where $2g_{\hat{r}}$ (resp. ${}^{\partial}g_{\hat{r}}$) is the number of connected components (resp. not regular connected components) of $\Gamma'_{\hat{r}}$, m is the number of 3-coloured edges of Γ' and $\dot{g}_{ij}$ is the number of closed $\{i,j\}$-coloured paths of Γ'.

3.2. Dimension four. The recognition of the 3-sphere, which is involved both in the generation of the set $\mathcal{S}_4^{(2p)}$ and in Proposition 2.1, cannot be performed through easy computations. Nevertheless it can be solved for graphs with a low number of vertices by dipole eliminations since, by the 3-dimensional classification results [13], it is known that no rigid crystallization of $\mathbb{S}^3$ exists (different from the "trivial" one of order two) with less than 24 vertices. Furthermore condition (ii) would be very heavy to check, since it implies recognition of 3-spheres with holes. Instead, we use a weaker condition, which is equivalent to require $\Gamma_{\hat{i}}$ to be a manifold (with boundary), i.e. each connected component of $\Gamma_{\hat{i}\hat{j}}$ must represent $\mathbb{S}^2$ possibly with holes for each pair of colours $i, j \in \Delta_3$. This is equivalent to require the following equality to hold:

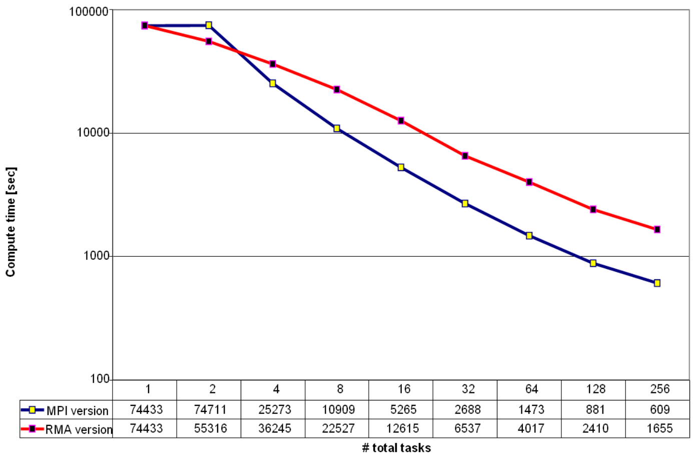
$$\sum_{k,t\in\Delta_4\setminus\{i,j\}} g_{kt} - \frac{\bar{p}}{2} = 2g_{\hat{i}\hat{j}} - \bar{g}_{\hat{i}\hat{j}} \tag{3.3}$$

where g_{kt} is the number of $\{k,t\}$-coloured cycles of Γ, $\bar{p}$ is the number of vertices of Γ lacking in 4-coloured edges, $g_{\hat{i}\hat{j}}$ (resp. $\bar{g}_{\hat{i}\hat{j}}$) is the number of connected components of $\Gamma_{\hat{i}\hat{j}}$ (resp. $\partial\Gamma_{\hat{i}\hat{j}}$).

4. Parallelization strategy. Although theoretical optimizations have been introduced in order to reduce the computational cost for generating a catalogue of crystallizations with a fixed number of vertices, the computation becomes more and more intensive as the number of vertices increases, because of the combinatorial procedure on which the algorithm relies and the large number of input spheres that are processed. Therefore, a parallel version of both the 3- and the 4-manifold generation algorithms described in Sect. 3 have been implemented by exploiting the Message Passing Interface (MPI) paradigm [21].

Parallelization is quite straightforward, as the data can be distributed among several tasks that can work independently from each other. Since the number and the computation time of each crystallization generated by a single sphere can vary significantly depending on the sphere processed, the workload may not be well balanced if each task processes the same number of input spheres. For this reason the work has been distributed among tasks according to a master-slave structure, where the master reads all the sphere codes from the input file and distributes a sphere at a time to slaves. Those slave processes will generate all the possible crystallizations associated to the sphere they received; when a slave has ended the processing of a sphere, it saves the results on its own memory and asks the master for a new one. Master process either provides a new sphere, if some of them have not been processed yet, or collects all the crystallizations produced by the slave task. Note that starting

[3] This operation is the inverse of the cancellation of an edge, which occurs in the same way as for a 1-dipole.



	1	2	4	8	16	32	64	128	256
MPI version	74433	74711	25273	10909	5265	2688	1473	881	609
RMA version	74433	55316	36245	22527	12615	6537	4017	2410	1655

FIGURE 4.1. *Computing times on a Linux cluster to generate 3-dimensional PL manifolds represented by crystallizations with 32 vertices: balanced distribution of spheres by the master versus a RMA approach. The RMA version is less performant than MPI because of the poor implementation of remote memory access on this platform.*

from different spheres we can obtain colour-isomorphic crystallizations, i.e. with the same code, therefore each slave stores crystallization codes in two *set* containers (for orientable and non-orientable manifolds respectively) of the STL library [23] so that results are not replicated.

This strategy implies that the master process does not contribute to the computation, because it must be ready to provide new data as soon as it is required by a slave. We also investigated an MPI-2 Remote Memory Access (RMA) approach which allows the master to process some spheres after having read and stored the input data in a memory window. In fact, each process can get from that window its input data by itself (*one-sided communication*, where memory windows access is regulated by a synchronisation command). In this case another window is required to store a counter keeping track of the already distributed spheres. Such a solution, tested for dimension 3, turned out to be not very efficient, because its performance depends on the system used. In particular its efficiency is guaranteed only on systems with hardware support for remote-memory access, such as SGI Altix or Sun Fire [25]. Whereas it shows poor performance on IBM SMP systems like the one we used for our tests, because the MPI-2 implementation is not optimized for RMA operations.

See Fig. 4.1 for a comparison of the computing times between the original, referred to as MPI, and the RMA approach for generating 3-manifold crystallizations with 32 vertices. Moreover, the contribution of the master to the computation has a negligible impact on the performance as the number of processors increases. Therefore we decided to keep the original strategy and implement it also for the 4-dimensional case.

At the end of the overall computation, the master will have collected all the results produced by each slave process. It will then write the results in two output files: one containing a list of triangulations of orientable manifolds (i.e. bipartite crystallizations), the other for non-orientable ones (as of now, this is true only for the case of the 3-dimensional manifold generation program: in dimension four, because of the very small number of non-orientable manifolds discovered, all the manifolds are still written in a single catalogue). Output produced by each slave is collected only at the end of the overall computation, in order to reduce communications with the master. However, when the number of crystallizations produced by each slave is very large (e.g. in dimension 4), the collection of results by the master can become a serious bottleneck. This can be avoided by letting each slave process write its own output and eventually post-processing all the files in a second step in order to merge them into only two. Moreover, as the number of input spheres increases, each slave may have to deal with a total number of output crystallizations that is too large for its local memory. Notice also that some processes can store isomorphic, and therefore redundant, codes on their local memory, but they cannot find it out since they do not communicate. This translates into a troublesome waste of memory space. Therefore a further small change, required for generating crystallizations of PL 4-manifolds with more than 18 vertices, has been implemented: allow the generation program to process a subset of the input data, so that the list of input spheres can be split in several chunks and each of them can be processed one at a time. The last two modifications discussed (removing the final communication bottleneck and splitting the input processing) have

$2p$	2	8	12	14	16	18	20	22	24	26	28	30	32
$\mathcal{S}_3^{(2p)}$	1	1	1	1	2	2	8	8	32	57	185	466	1543
$\mathcal{C}_3^{(2p)}$	1	1	1	1	3	4	23	44	262	1252	7760	56912	444102
$\tilde{\mathcal{C}}_3^{(2p)}$	0	0	0	1	1	1	9	12	88	480	2790	21804	170367

Table 5.1

Number of input gems and corresponding crystallizations generated in dimension 3 up to 32 vertices. Some values, i.e. $2p = 4, 6, 10$, *are missing, since in these cases there are no input 2-spheres and consequently no crystallizations.*

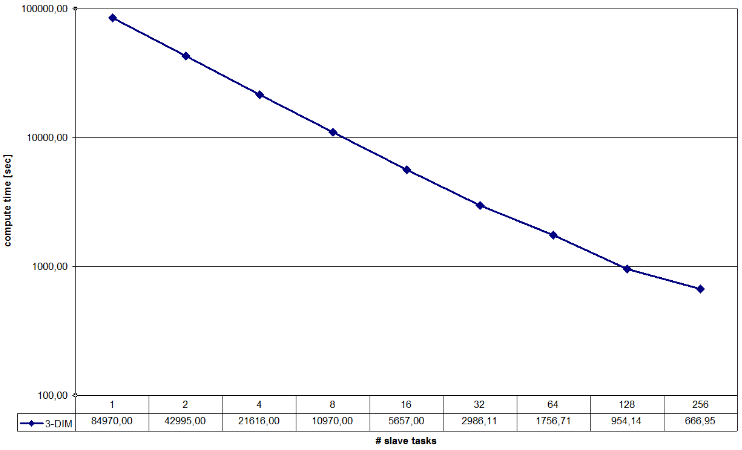


Figure 5.1. *Scaling of the computing time with the number of slaves on PLX for the generation of 3-manifold crystallizations with 30 vertices.*

been implemented only on the 4-dimensional generation algorithm: as for dimension three, the number of input and output codes is still small enough to be handled without such changes.

5. Performance results. Some benchmark tests have been made in order to evaluate, for both programs, the scalability with the number of processors on two different systems. The first one is a 3288 cores Linux Infiniband cluster IBM iDataPlex DX360M3 (referred to as PLX), made of 274 IBM X360M2 12-way computing nodes with 2 Intel Xeon Westmere 6-core E5645 2.4GHz processors and a memory of 48GB per node. The second one is an IBM BlueGene/Q system (referred to as FERMI) made of 10240 computing nodes with a chip of 16 IBM PowerA2 1.6 GHz cores and 16GB of memory each.

Performance of the code implementing the generation of triangulations of 3-dimensional PL manifolds has been tested using as an input the catalogue of gems with 30 vertices representing the 2-sphere. In this case we have 466 input 3-coloured graphs producing 78716 crystallizations (see Table 5.1 for a list of the number of input gems representing the 2-sphere and crystallizations produced for increasing numbers $2p$ of vertices). The computation for our test is intensive enough to allow us to complete a scalability test on the PLX system, exploring the range between 1 and 256 slave processes. Computing times plotted in Fig. 5.1 show a linear scalability until the number of slaves becomes greater than 128, because of the increasing communication overhead with the master.

In dimension 4, as the number of input spheres and crystallizations generated is very large even for a low number of vertices (see Table 5.2), the code is suitable to run on massively parallel architectures such as the BlueGene/Q. We considered the case of the 16 vertices catalogue in order to compare performances observed on this platform with the ones on PLX. Figure 5.2 shows scalability with the number of slaves ranging from 64 to 512, both on the PLX and the FERMI systems. As we expected, FERMI computing times are worst than PLX ones because the BlueGene cores are less powerful. For the same reason scalability on FERMI is better than on PLX because slaves' requests for new input data should be more distributed over the time as the processing time of each input sphere is longer, thus avoiding the communication bottleneck.

In order to explore scalability on FERMI over a very large range of processors (from 1024 to 16384), we considered the generation of the 18 vertices catalogue and measured the computing time with respect to the

$2p$	2	8	10	12	14	16	18	20
$\mathcal{S}_4^{(2p)}$	1	9	39	400	5255	95870	1994952	45654630
$\mathcal{C}_4^{(2p)}$	1	1	0	0	1109	4512	44803	47623129

Table 5.2

Number of input gems and corresponding crystallizations generated in dimension 4 up to 20 vertices. Some values, i.e. $2p = 4, 6$, are missing, since in these cases there are no input 3-spheres and consequently no crystallizations.

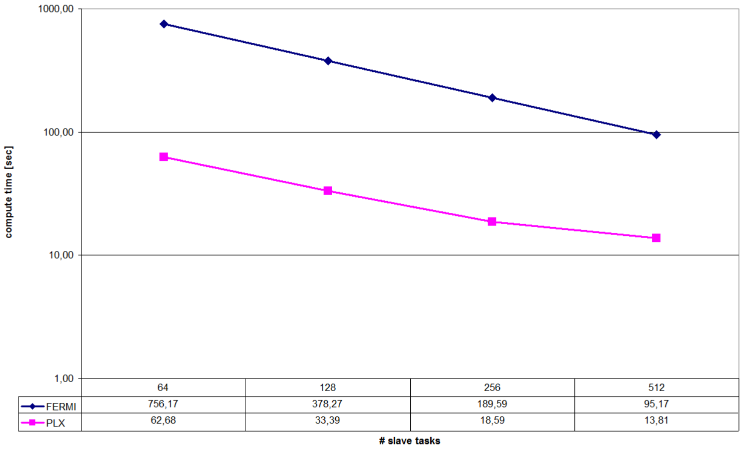
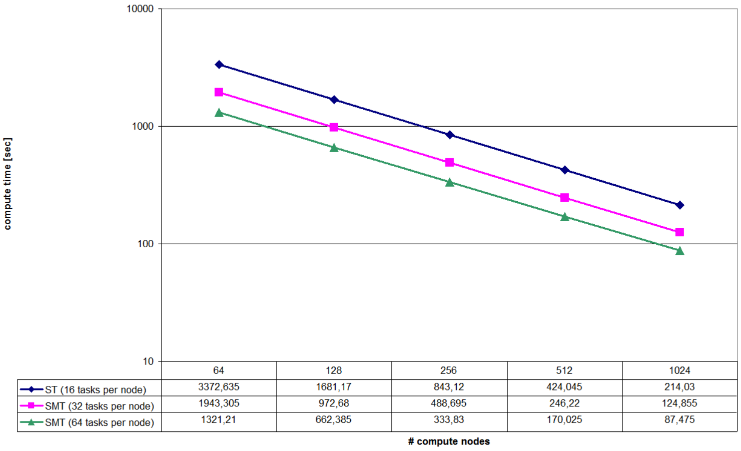


Figure 5.2. *Scaling of the computing time with the number of slaves for the generation of 4-manifold crystallizations with 16 vertices. Comparison between FERMI and PLX timings.*

number of total MPI tasks, instead of slave processes. This has been decided due to the peculiar configuration of the system [24], where access to the compute nodes is mediated by the access to specific nodes with the task of dealing with I/O operations ("I/O nodes"). More specifically, in order to let your simulation run on FERMI's compute nodes, an I/O node needs also to be allocated, and since there is one of such nodes every 64 or 128 compute nodes, each resource allocation needs to require (a multiple of) such number of compute nodes. FERMI system is made of 10 racks (each containing 16K cores): 2 of them have 16 I/O nodes each, implying a minimum job allocation of 64 nodes (1024 cores); the other racks have 8 I/O nodes each, implying a minimum job allocation of 128 nodes (2048 cores). PowerA2 cores can schedule for execution two or four threads in the same clock cycle, therefore it is possible to use a single core as two (resp. four) virtual CPUs. This technique is called Simultaneous Multi-Threading (SMT), to distinguish it from the standard Single Thread (ST). Figure 5.3 shows a linear scalability both for the ST and the SMT modes. Deploying SMT improves the performance by reducing the computing time about 42% and 60% by using two and four virtual CPUs respectively. Communication does not affect scalability even for a large number of cores, not only because processors are less powerful and the processing of each input sphere is more intensive, but also because in dimension 4 the output results are not collected by the master. On the other hand, I/O performance may degrade (see Fig. 5.4 as an example) since the I/O nodes are the only ones able to interact with the file systems and each slave writes its own file.

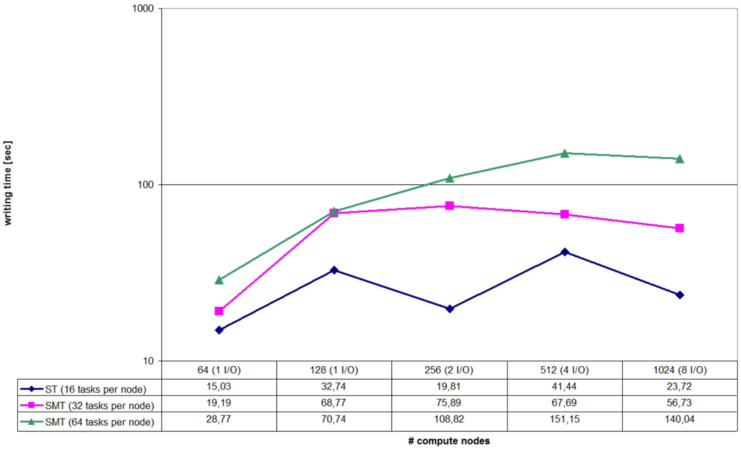
As the number N of input data will increase (with the number of vertices) or scalability will start to degrade (for an increasing number of slaves), a further improvement for the program will be letting the master distribute chunks of a fixed number of input spheres, instead of only one at a time: this should reduce communications with the master and avoid a possible bottleneck. This solution will preserve the workload balance by taking the chunk size very low with respect to N.

6. Conclusions and future developments. In this paper we have described an algorithm for generating 3- and 4-dimensional PL manifolds represented by edge-coloured graphs with a fixed number of vertices. The combinatorial nature of the algorithm has allowed to parallelize it according to a simple master-slave procedure by using the MPI paradigm. However, since the computing time of each slave is dependent on the input data, a suitable data distribution strategy has been adopted in order to have a balanced workload and let the code be able to scale over a very large number of processors. We have discussed scalability results on different high



	64	128	256	512	1024
ST (16 tasks per node)	3372,635	1681,17	843,12	424,045	214,03
SMT (32 tasks per node)	1943,305	972,68	488,695	246,22	124,855
SMT (64 tasks per node)	1321,21	662,385	333,83	170,025	87,475

Figure 5.3. *Scaling of the computing time with the number of cores on FERMI for the generation of 4-manifold crystallizations with 18 vertices.*



	64 (1 I/O)	128 (1 I/O)	256 (2 I/O)	512 (4 I/O)	1024 (8 I/O)
ST (16 tasks per node)	15,03	32,74	19,81	41,44	23,72
SMT (32 tasks per node)	19,19	68,77	75,89	67,69	56,73
SMT (64 tasks per node)	28,77	70,74	108,82	151,15	140,04

Figure 5.4. *Output times on FERMI for writing all the 4-manifold crystallizations generated with 18 vertices. All MPI tasks write concurrently their own results on separate files. The data size may increase with the number of tasks because some crystallizations may be replicated (in this case we reached a maximum size of 1.1GBytes). There is one I/O node available every 128 compute nodes. The first test with 64 compute nodes has been executed in a special rack where an I/O node is available every 64 compute nodes.*

performance computing architectures: iDataPlex Linux Cluster and BlueGene/Q.

Our future work will investigate further improvements of the code performance, both from a theoretical point of view by searching for other topological results that simplify the computation and from a HPC point of view by introducing a second level of parallelization based on a shared memory paradigm. In the latter case we can accelerate time-consuming parts, such as the computation of the crystallization code, by exploiting the OpenMP [22] Application Program Interface where each slave (MPI task) can run either on a single core in a SMT mode or in a single node if a large number of OpenMP threads is required. Finally we will further explore I/O behaviour when large sized output data is produced concurrently by a large number of processors in order to understand its impact on the total wallclock time, in particular on BlueGene like systems.

We envisage that improvements of the code performances will enable us to extend the existing crystallization catalogues of 3- and 4-manifolds to a higher number of vertices of the representing graphs. In dimension three, we could thus make comparisons with other censuses generated by means of different combinatorial methods. With regard to PL 4-manifolds, we expect that our catalogues would be useful to investigate minimal combinatorial structures and test conjectures about known invariants, as well as giving examples of non-equivalent triangulations of some "interesting" topological manifolds.

Acknowledgments. This work is performed under the auspices of G.N.S.A.G.A. of I.N.d.A.M. (Italy) and financially supported by M.I.U.R. of Italy, University of Modena and Reggio Emilia, funds for selected research topics. We acknowledge the CINECA award under the Italian Supercomputing Resource Allocation (ISCRA) initiative, for the availability of high performance computing resources.

REFERENCES

[1] P. J. Hilton and S. Wylie, *An introduction to algebraic topology - Homology theory*, Cambridge Univ. Press, 1960.

[2] C.Rourke and B. Sanderson, *Introduction to piecewise-linear topology*, Springer Verlag, New York - Heidelberg, 1972.

[3] A. T. White, *Graphs, groups and surfaces*, North Holland, 1973.

[4] J. Hempel, *3-manifolds*, Annals of Math. Studies, 86, Princeton Univ. Press, 1976.

[5] S. Lins, *Gems, computers and attractors for 3-manifolds*, Knots and Everything 5, World Scientific, 1995.

[6] R. Mandelbaum, *Four-dimensional topology: an introduction*, Bull. Amer. Math. Soc. 2(1) (1980), pp. 1–159.

[7] M.H. Freedman, *The topology of four-dimensional manifolds*, J. Differential Geom. 17 (1982), pp. 357–453

[8] B. A. Burton, *Enumeration of non-orientable 3-manifolds using face-paring graphs and union-find*, Discrete Comput. Geom. 38 (2007), pp. 527–571.

[9] A. Ahmedov and B. Doug-Park, *Exotic smooth structures on small* 4*-manifolds with odd signatures*, Invent. Math. 181(3)(2010), pp. 577–603.

[10] M. Ferri, C. Gagliardi and L. Grasselli, *A graph-theoretical representation of PL-manifolds. A survey on crystallizations*, Aequationes Math. 31 (1986), pp. 121–141.

[11] P. Bandieri, M. R. Casali, and C. Gagliardi, *Representing manifolds by crystallization theory: foundations, improvements and related results*, Atti Sem. Mat. Fis. Univ. Modena Suppl. 49 (2001), pp. 283–337.

[12] P. Bandieri, M. R. Casali, P. Cristofori, L. Grasselli, and M. Mulazzani, *Computational aspects of crystallization theory: complexity, catalogues and classification of 3-manifolds*, Atti Sem. Mat. Fis. Univ. Modena, 58 (2011), pp. 11–45.

[13] M. R. Casali and P. Cristofori, *A catalogue of orientable 3-manifolds triangulated by* 30 *coloured tetrahedra*, J. Knot Th. Ram., 17 (2008), pp. 579–599.

[14] P. Bandieri, P. Cristofori, and C. Gagliardi, *Nonorientable 3-manifolds admitting coloured triangulations with at most 30 tetrahedra*, J. Knot Th. Ram., 18 (2009), pp. 381–395.

[15] P. Bandieri and C. Gagliardi, *Rigid gems in dimension n*, Bol. Soc. Mat. Mexicana (3) 18 (2012), pp. 55–67.

[16] M. R. Casali, *Catalogues of PL-manifolds and complexity estimations via crystallization theory*, Oberwolfach Report, 24 (2012), pp. 58–61(DOI: 10.4171/OWR/2012/24).

[17] M. R. Casali and C. Gagliardi, *A code for m-bipartite edge-coloured graphs*, Rend. Ist. Mat. Univ. Trieste 32 suppl.1, (2001), pp. 55–76.

[18] M. Ferri and C. Gagliardi, *Crystallization moves*, Pacific J. Math. 100 (1982), pp. 85–103.

[19] B. Martelli and C. Petronio, *Census 7, Census 8, Census 9, Census 10, Tables of closed orientable irreducible 3-manifolds having complexity c* , $7 \leq c \leq 10$,
http://www.dm.unipi.it/pages/petronio/public_html/files/3D/c9/c9_census.html

[20] S. Matveev, *Recognition and tabulation of three-dimensional manifolds*, Doklady RAS 400(1) (2005) pp. 26–28 (Russian; English trans. in Doklady Mathematics 71 (2005) pp. 20–22).

[21] http://www.mpi-forum.org

[22] http://openmp.org

[23] http://www.sgi.com/tech/stl

[24] http://www.hpc.cineca.it/content/ibm-fermi-user-guide

[25] W. Gropp and R. Thakur, *Revealing the performance of MPI RMA implementations*, Proc. of the 14th European PVM/MPI Users' Group, September 2007, pp. 272–280.

Edited by: Marc Eduard Frîncu
Received: Feb 28, 2013
Accepted: Mar 29, 2013