A Paper Presented at the 2008 ISNSCE Foundations of Nanoscience (FNANO) Conference: A Formal Crystal Description System

W. Edwin Clark[†], Mohamed Eddaoudi[‡] & Gregory McColm[§]

Abstract. We outline primary components of an algorithm based on geometric first principles for general crystal net prediction. The algorithm, and its computer implementation as a MAPLE program, is under development, and we present some examples of its operation. This algorithm arose out of an outline for nanostructure design heuristic proposed in FNANO 2007.

[†] Department of Mathematics, University of South Florida, Tampa, FL 33620; eclark@math.usf.edu.

[‡] Department of Chemistry, University of South Florida, Tampa, FL 33620; eddaoudi@cas.usf.edu.

[§] Department of Mathematics, University of South Florida, Tampa, FL 33620; mccolm@cas.usf.edu.

As the demand increases for more complex nanostructures, the demand will increase for mathematical and computational tools for structure design. As structures grow more complex, more systematic methods of design would be helpful. In [13] and [14], a theoretical rationale for such a computer based design method is proposed. The motivation was purely geometric, and presented a theoretical method for designing molecular building blocks (or tiles) that could be assembled in a given form. Here, we follow the initial impetus of that proposal applied to crystal design, and this presentation is a status report on a computer project for designing crystals originating from this approach.

There are at least two good reasons for starting with crystals.

- First, despite considerable extant activity in crystal design (as we shall see in a moment), there is is a feeling that more mathematical support and collaboration would be helpful, a point expressed in O. Yaghi's plenary presentation here last year (abstract at [26]). There is a thread of scientific and engineering op-ed pieces – e.g., [17], [10], and [6] – expressing a great distance travelled but an even greater distance to go. Recently, [12] proposed an essentially geometric and combinatorial approach. This is an important goal even in DNA computing, for Seeman (e.g., [21]) has presented as a major goal the construction of a DNA crystal able to serve as a skeleton or scaffold.
- In addition, crystals are fairly well understood, with a tradition of mathematical analysis going back to the seventeenth century. Indeed, several approaches for crystal prediction and design were proposed in the papers mentioned above, most notably the minimization of entropy or a similar parameter, although the subject retains its initial geometric impetus.

The initial proposal being based on a purely geometric analysis, we began a programme for the geometric synthesis of crystalline structures.

The very first investigation of crystals now regarded as both modern and theoretical – namely Kepler's analysis of the snowflake (see [22]) was geometric. And underlying our approach is the fundamentally geometric perspective of A. F. Wells ([25]), who described crystals as periodic (graphical) *nets* embedded in two- or three-dimensional space. Nevertheless, several extant systems incorporate energy or entropy optimization considerations; here are some examples of projects which design crystals out of pre-assembled building blocks. The design process was dubbed *reticular synthesis* in [27], and there is a preference for designing porous crystals composed of large and rigid blocks like zeolites or metal-organic frameworks.

- For example, C. Mellot-Draznieks and G. Ferey and their group are working on an *Automated Assembly of Secondary Building Units* (AASBU) method, which takes large and complex "secondary" units and using (energy) minimization and stochastic search techniques develop candidate designs for crystals composed of these units (see, e.g., [18] and [19]).
- Meanwhile, M. D. Foster and M. M. J. Treacy maintain a database [9], which relies on stochastic minimalization and group symmetries to search for candidate nets In fact, Treacy and colleagues have developed a program to exhaustively enumerate the 4-connected (essentially the tetrahedral) uninodal nets using the symmetries and cost (energy) function minimalization ([24]).
- Two threads met when theoretical and computational work of J. Conway, D. H. Huson and W. Thurston ([2]), O. Delgado Friedrichs, A. Dress, and others (see [3]; see also [4]) met O. Yaghi and M. O'Keefe (see [20]) in developing an a cluster of projects as part of a Reticular Chemistry Structure Resource (RCSR), in the most overtly geometric approach we've seen, but uses barycentric placement of nodes within cells, which is a (local) minimization approach (see [11]).

This is only a sample.

Here we will, perhaps naïvely, eschew optimization in order to concentrate on the geometry. We describe on of our projects, and this one follows a bottom up approach inspired by [13] and [14] (and developed further in [15] and [16]). There, a completed structure was viewed as a complex of molecular building blocks, and one imagined that a bug placed on a distinguished block could walk from block to block to block, etc., following a set of instructions of how to step from one block to the next. The list of instructions for just one walk would be a string of substrings, each substring incorporating a particular step; if each substring could be interpreted as a particular rigid motion (translation and rotation), then the composition of rigid motions would itself be a rigid motion which would fix the relative placement of the last block with respect to the first. The set of all such instructions would thus fix the entire structure with respect to the initial block.

This bottom up approach will motivate a MAPLE program one of us (W. E. Clark) has composed to *exhaustively* enumerate prospective crystals (as defined within tight criteria, thus eliminating paracrystals and similar variants). This particular project is one of an array of interconnected programs, which we envision will eventually make up a functioning ensemble that will assist crystallographers and others in designing such structures.

1 The Situation

In this abstract, we will construct a crystal out of a small number of types of blocks. We will make the following simplifying assumptions:

- Each block is a polygon or polyhedron, which will be positioned so that it's center will be at a particular point in space, and the block will be oriented in a particular way.
- Each block will bond to other blocks by joining them at their vertices. The bond will be rigid, and the junction of the two bonded vertices lies on the line segment joining the two centers of the two bonded blocks.
- All bonds will be the same, so if there is one kind of block, each bond will consist of two identical blocks, facing each other across the pair of adjoined vertices, with one block rotated about the bond with respect to one another by some *articulating angle* θ in accordance with some convention (to be described below). If there are two kinds of blocks, each bond will run from one kind to another, and will require a convention for determining the relative orientations of the blocks.

Notice that under these assumptions, if we fixed a single block in space (position and orientation), it's neighbors would then be fixed, and so would their neighbors, and their neighbors, and so on. We could call such a structure *singly generated* as the placement of a single block will fix the entire structure.[1] Assuming that the structure extends into all three dimensions (i.e., it does not lie on a plane) and is a complete (no blocks unbonded) discrete structure, Bieberbach's Theorem [1] says that it will be a crystal, whose symmetries will be given by one of the crystallographic groups.

Our goal is a computer program with various desiderata that will design for us block type(s) and fix an articulating angle for them such that they may assemble into a singly generated (crystalline) structure satisfying the the criteria of the previous paragraph. We would specify block shapes qualitatively (e.g., equilateral triangles and rectangles, with some kind of θ), and the computer would then produce an *exhaustive* list of all possibilities, where each possibility is given, in the equilateral-triangles-and-rectangles case, by a pairs (r, θ), where r is the height-to-length ratio of the block and θ is the bonding angle.

Notice that by restricting our attention to singly generated structures, we (hopefully) finesse most assembly, stability and entropy issues: if there is only one way that these blocks can assemble into a crystal, and if they are somehow persuaded to assemble into something crystalline, then that crystal will hopefully be what they will assemble into.

1.1 Some Geometry

The algorithm will rely on three dimensional geometry.

It is common practice in this area of theoretical crystallography to concentrate on the *net* generated by the crystal (see [5]): the center of an block – which is a point in space – becomes a "node" of the net, and a bond between two adjacent blocks becomes a line segment from the center of one of the blocks to the center of the other. In this abstract, we will refer to the "block" centered at a node, and with ports at its corners for forming bonds to other blocks; a bond is formed by having two blocks touch at their corners, and we will prevent bent bonds by requiring that each bond runs from the center (node) of a block to center of the adjacent block, and so we can represent bonds by straight line segments. Here are some details.

We will find it useful to prevent ambiguity by defining *block types*. Informally, a block type is a block, in standard position, centered at the origin. Formally, a *block type* is a set of unit vectors $\mathbf{p}_1, \dots \mathbf{p}_d$, which will be the vectors from the center of the block (in standard orientation) to its articulating corners. Then a *block* is merely the result of rotating and translating (i.e., moving by rigid motions) a block type to some position in space. By rotating and translating block types, we can get an array of many blocks of a few types (see Figure 1). These blocks articulate at their corners to form bonds, and again, we require that all bonds run from center of block to center of block.

[1] These correspond to *uniform tilings* and *uninodal nets*.

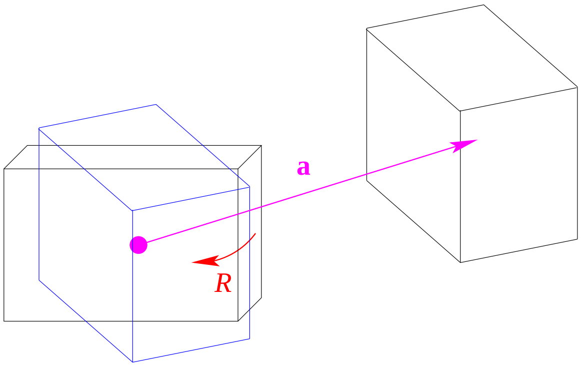

Fig. 1. The node type (representing a block type) can be rotated by a rotation R and translated by a translation to serve as a mode placed in space. As any translation can be represented as a vector function $\mathbf{x} \mapsto \mathbf{x} + \mathbf{a}$ (for $\mathbf{a}$ the fixed displacement vector of the translation), the composed isometry can be represented by the formula $\mathbf{x} \mapsto R(\mathbf{x}) + \mathbf{a}$: any direct isometry in three dimensions can be represented by such a formula.

Then a *block* is merely the result of rotating and translating (i.e., moving by rigid motions) a block type to some position in space. By rotating and translating block types, we can can get an array of many blocks of a few types (see Figure 1). These blocks articulate at their corners to form bonds, and again, we require that all bonds run from center of block to center of block. The centers of the blocks are points in space (which we call *nodes*), joined by bonds that are line segments from node to node, and the resulting structure is a *net* (see Figure 2).

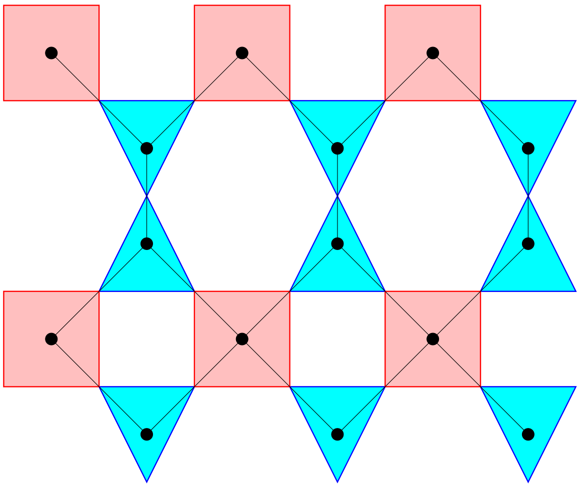

Fig. 2. A two-dimensional net induced by rectangles and triangles. This net is "binodal" in the sense that some nodes are centers of rectangles and some centers of triangles. Bent bonds in this system would be represented by 2-port blocks representing the bend.

We will look at several kinds of blocks in this abstract. In particular, our blocks will be highly symmetric. A *symmetry* of the block type is a rotation mapping the block type onto itself, so the symmetry restriction will be: for each $i, j \in [d]$, there is a symmetry R of the block sending $\mathbf{p}_i$ to $\mathbf{p}_j$: $R(\mathbf{p}_i) = \mathbf{p}_j$.[2] Informally, the vectors run to the (bonding) corners of the block, and the requirement that all bonds be the same (i.e.,

[2] Technically, reflections are also symmetries, but we will not deal with reflections in this abstract.

that they be symmetric) induces the requirement that we can rotate any corner of a block type to any other corner of that block type. Our favorite examples will be:

- Equilateral triangles, where $d = 3$ – and to be more precise, $\mathbf{p}_1 = (1, 0, 0)$, $\mathbf{p}_2 = (-1/2, \sqrt{3}/2, 0)$, and $\mathbf{p}_3 = (-1/2, -\sqrt{3}/2, 0)$. Notice that the symmetries of the equilateral triangles are generated by the rotations of 120° around the z-axis, and by rotations of 180° about the three medial axes.
- Rectangles, where $d = 4$ – and to be more precise, for some r, $0 < r \leq 1$, $\mathbf{p}_1 = (a, ra, 0)$, $\mathbf{p}_2 = (a, -ra, 0)$, $\mathbf{p}_3 = (-a, ra, 0)$, and $\mathbf{p}_4 = (-a, -ra, 0)$ where $a = 1/\sqrt{1 + r^2}$. Notice that the symmetries of the rectangle (for $r \neq 1$) are generated by the rotations by 180° about the three axes; if $r = 1$, we get additional symmetries generated by rotations by 90° about the z axis.

Since the structure will be singly generated, we can imagine that is is annealed as follows. Start with a single block in standard position at the origin. Attach to it its neighbors. Attach to them their neighbors. Continue ad infinitum until the crystal is generated. We need a detail on how neighbors are attached.

For simplicity, suppose that there is only one block type: the resulting net will be *uninodal*. We describe how a new block of the net is generated by an extant block. For simplicity, let's suppose that the extant block is the original block placed at the origin. Say that the major axis of the extant block is the z-axis, while the minor axis is the y-axis, and the extant block connects to its neighbors out of the bonding vectors $\mathbf{p}_1, \ldots \mathbf{p}_d$; suppose that we want to generate the neighboring block connected via $\mathbf{p}_i$.

We want to rotate a copy of the extant block from the extant block's position to its new position, so that it's bonding vector $\mathbf{p}'_i$ will be $-\mathbf{p}_i$, and its center will be at $2\mathbf{p}_i$. To do this, let P_i be the plane through the target of the vector $\mathbf{p}_i$ and perpendicular to $\mathbf{p}_i$, and let P'_i be the plane containing the major axis of the extant block and $\mathbf{p}_i$; these two planes are not parallel and both contain the target point of $\mathbf{p}_i$, so their intersection is a line ℓ perpendicular to $\mathbf{p}_i$ containing the target point of $\mathbf{p}_i$ (see Figure 3).

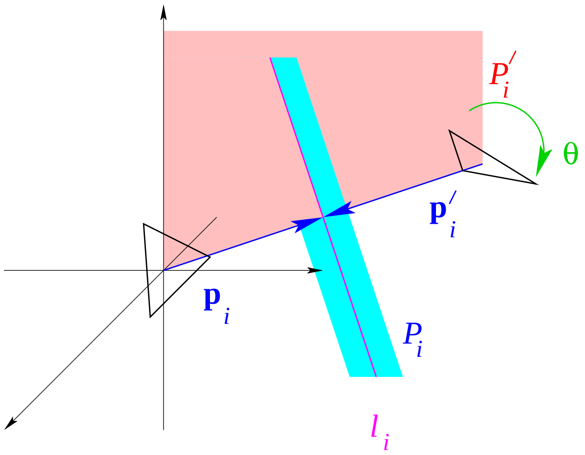

Fig. 3. The line ℓ_i is the intersection of the planes P_i and P'_i, and a copy of the extant block is (a) rotated about ℓ_i by 180° to get its center at the new block, and then (b) rotated by θ to get into orientation.

We then rotate a copy of the extant block 180° about ℓ_i, and the center of the copy will now be at $2\mathbf{p}_i$ while it's ith bond vector will be $\mathbf{p}'_i = -\mathbf{p}_i$. All that remains is to get the new block oriented. As $\mathbf{p}_i$ and $\mathbf{p}'_i$ are anti-parallel, the bond is already straight, so all we need do is rotate the new block about the bond axes by the angle θ.

This is how we can imagine the crystal being constructed: the reverse problem is, supposing that the crystal was constructed in this way, what does that tell us about the vectors $\mathbf{p}_i$ and the rotation angle θ? In order to do this, we reformulate the above operations:

- The translation (from block to block) was $\mathbf{x} \mapsto \mathbf{x} + 2\mathbf{p}_i$.
- Let ℓ'_i be the line parallel to ℓ_i, but through the origin, and R'_i the rotation by 180° about ℓ'_i. Then let $R_{i,\theta}$ be the rotation by θ about the line through $\mathbf{p}_i$. The net rotation is then $R_i = R_{i,\theta} \circ R'_i$.

And the *step* function that moves us from the first block to the next is $\mathbf{x} \mapsto R_i(\mathbf{x}) + 2\mathbf{p}_i = M_i\mathbf{x} + 2\mathbf{p}_i$ if M_i is the matrix of R_i.

2 The Algorithm

This algorithm is a filter designed to reduce the (uncountably infinite!) number of candidates to a manageable number that we can check. This first part of the algorithm works by setting up a system of equations based on a sequence of steps – a *walk* – through the net. These equations are based on the "crystallographic restriction" of Rénè-Just Haüy ([22]): any direct symmetry sending one node of the net to another node of the same type in the net must have, as its rotational component, a rotation by one of the angles $0°, \pm 60°, \pm 90°, \pm 120°, \pm 180°$. Since the step function itself entails a net rotation of 180° (the θ merely determines the axis of rotation), the equations are determined by composing multiple step functions.

We need need to use two old facts.

- If M is the matrix of a rotation, then the angle θ of that rotation is given by: the sum of the diagonal entries (the *trace*) of that matrix is $1 + 2\cos\theta$, so in order to assure that the structure we are generating satisfies the crystallographic restriction (that $\theta \in \{0°, \pm 60°, \pm 90°, \pm 120°, \pm 180°\}$) we need so make sure that the trace is in $\{-1, 0, 1, 2, 3\}$.
- When composing rotations, we find that the second step in a walk was taken from a block *that was already oriented by the first step*, so that after a little algebra, the net rotation of steps out the i_1st, i_2nd, i_3rd, ..., i_jth corners of successive blocks, in that order, is the matrix product $M_{i_1}M_{i_2}M_{i_3}\cdots M_{i_j}$, where each M_i is the matrix of the step out of original block type out the ith corner.

Each matrix M_i represents a rotation by 180°, but these rotations are not necessarily around the same axis, so if we impose the condition, *for any* $j > 1$,

$$\mathbf{trace}(M_{i_1}M_{i_2}M_{i_3}\cdots M_{i_j}) \in \{-1, 0, 1, 2, 3\},$$

we obtain a set of polynomials of two variables, $u = \cos\theta$ and $v = \sin\theta$, and solving for u and v we then get a (small) set of possible values θ to check. A computer search shows that we need to check only for $j = 2, 3, \ldots, 6$.

The point is that any list of candidates obtained this way is exhaustive: there will be no other singly generated crystals using the given blocks. So, for example, if we made a crystal out of squares, using as our block type a square with a port vectors being basis vectors $\pm\mathbf{e}_1$, $\pm\mathbf{e}_2$, we could compute the matrix M_1 for the step out of the first corner of the first square then the matrix M_2 for the step out of the second corner of the second square, we get

$$\mathbf{trace}(M_1M_2) = 2u + u^2 \in \{-1, 0, 1, 2, 3\},$$

which gives us four quadratic polynomials, of solution sets (for u) $\{-1\}$, $\{0, 2\}$, $\{-1 \pm \sqrt{5}/2\}$, $\{-1 \pm \sqrt{3}\}$, and $\{-3, 1\}$, respectively. Similarly, the net rotation for a step out of first corner of the first square and then a step out of the *third* corner of the second square produces the polynomial equations $3 - 4u^2 \in \{-1, 0, 1, 2, 3\}$, with the solutions (for u) being 0, $\pm 1/2$, $\pm 1/\sqrt{2}$, $\pm\sqrt{3}/2$, and ± 1. The only solutions (for u) appearing on both lists are $-1, 0, 1$. As $u = \cos\theta$, we get our list of candidate values of θ: 180°, 90°, and 0°. All three of these generate crystalline structures (although $\theta = 0°, 180°$ generate the same structures); see Figure 4.

The bugwalk algorithm has several problems. For example, they may generate false positives. We have found a rather unviewable example of a non-crystal that is not eliminated by the bugwalk algorithm, but for a viewable example, consider the regular tetrahedra, for which we can use the first step of the bugwalk algorithm ($j = 2$) to get the candidates $\theta = 60°$ and $\theta = \arccos(1/4) \approx 75.52°$. The first generates the diamond (cubic) lattice, while the latter generates the non-crystal in Figure 5.[3] Additional iterations, or other packages under development, will eliminate the latter candidate as a crystal.

[3] Other crystals of regular tetrahedra *must* admit more than one type of bond.

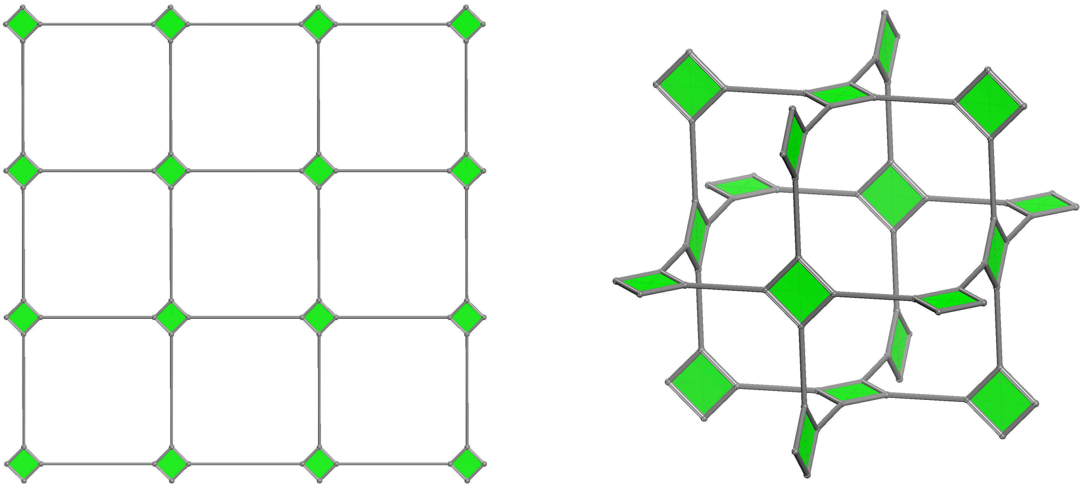

Fig. 4. The two possible singly generated crystals composed of identical square blocks connect by identical straight bonds. Both are well-known – see [7].

3 Excelsior

The proximate goal was to develop a program that could design crystals of many kinds of blocks, but for the time being keeping to the simplifying assumption of single generation so that geometric restrictions would (hopefully) force the structure to assemble as desired. The algorithm described here generates exhaustive lists, but with false positives, so another component of the project is to develop programs using other methods as secondary filters, checks, or alternatives.

Finally, as we mentioned in the beginning of this abstract, the original impetus was to develop a system for nanostructure design. One of the original motivations for dealing with crystals first was their very familiarity; dealing with non-crystals will certainly require alternatives to familiar tools like the crystallographic restriction. As V. Y. Shevchenko et al have observed ([23]), "The structure of many nanoparticles synthesized over the last decade is so unusual from the standpoint of classical crystallography that many authors describe it in literary rather than in rigorous scientific terms." Shevchenko et al then propose that by enlarging the toolbox, we will find useful tools for analyzing and synthesizing these non-crystals, and this seems to be a reasonable if optimistic prognostication.

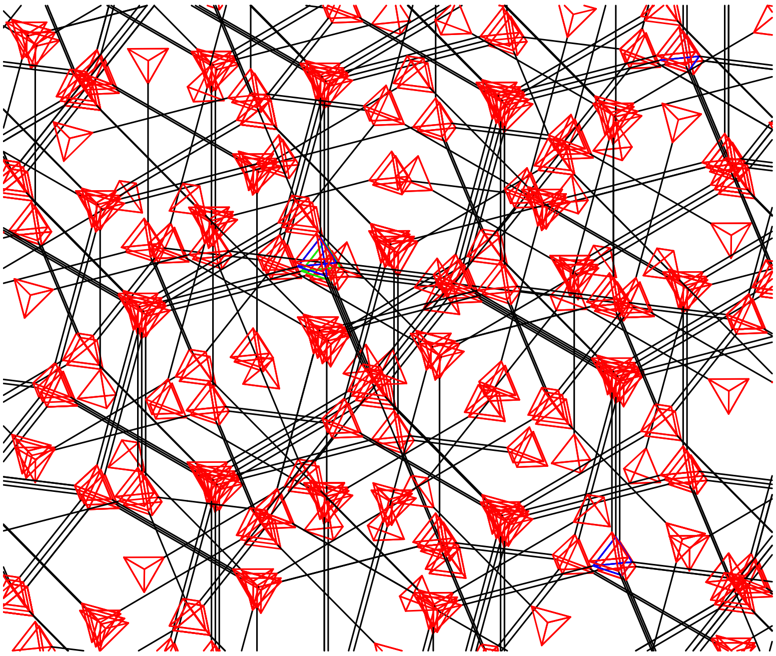

Fig. 5. The MAPLE implementation of this algorithm gave the above picture for about 900 tetrahedra of the arccos(1/4) net (this is a zoomed in picture, with the tetrahedra in red and the bonds in black). Permitted infinitely many iterations, the resulting net is not discrete in three dimensional space, and hence is not a crystal.

References

1. That crystallographic book with bieberbach in the appendix – or bieberbach's original paper.
2. J. H. Conway & O. Delgado-Friedrichs, D. H. Huson & W. P. Thurston, *On Three-Dimensional Space Groups* **Beiträge Algebra Geom. 42:2** (2001), 475 - 507.
3. O. Delgado Friedrichs, A. W. M. Dress, D. H. Huson, J. Klinowski & A. L. Mackayk, *Systematic enumeration of crystalline networks*, **Nature 400** (1999), 644 – 647.
4. O. Delgado Friedrichs, M. D. Foster, M. O'Keefe, D. M Proserpio, M. M. J. Treacy & O. M. Yaghi, *What do we know about three-periodic nets?* **Journal of Solid State Chemistry 178** (2005), 2533 - 2554.
5. O. Delgado Friedrichs, M. O'Keefe, *Crystal nets as graphs: Terminology and definitions* **J. Solid State Chemistry 178** (2005), 2480 - 2485.
6. J. D. Dunitz, *Are crystal structures predictable?*, **Chem. Comm. 2003** (2003), 545 – 548.
7. M. Eddaoudi, J. Kim, D. Vodak, A. Sudik, J.Wachter, M. OKeeffe & O. Yaghi, *Geometric requirements and examples of important structures in the assembly of square building blocks*, **Proc. Natl Acad. Sci. 99:8** (4/16/2002), pp. 4900-4904.
8. M. D. Foster, M. M. J. Treacy, J. B. Higgins, I. Rivin, E. Balkovsky & K. H. Randall, *A systematic topological search for the framework of ZSM-10*, **J. Appl. Crystallography 38** (2005), 1028 - 1030.
9. M. D. Foster, M. M. J. Treacy, **A Database of Hypothetical Zeolite Structures**, <http://www.hypotheticalzeolites.net/>.
10. A. Gavezzotti, *Are Crystal Structures Predictable?* **Acc. Chem. Res. 27** (1994), 309 – 314.
11. **Generation, Analysis and Visualization of Reticular Ornaments using Gavrog**, http://www.gavrog.org.
12. G. L. Hart, *Where are nature's missing structures?* **Nature Materials 6** (2007), 941 – 945.
13. N. Jonoska & G. L. McColm, *Flexible versus Rigid Tile Assembly*, in: C. S. Calude, et al, **Proc. 5th Internat'l Conf. Unconv. Comp.** (Springer LNCS 4135, 2006), 139 - 151.
14. N. Jonoska & G. L. McColm, *Formalizing Nanostructure Description*, **Proc. 4th ann. conf. Found. Nanoscience (FNANO)** (2007).
15. N. Jonoska & G. L. McColm, *Describing Complex Structures I: Rigid to Flexible*, in preparation.
16. N. Jonoska & G. L. McColm, *Describing Complex Structures II: Flexible to Rigid*, in preparation.
17. J. Maddox, *Crystals from First Principles*, **Nature 335** (1988), 201.

18. C. Mellot-Draznieks, J. M. Newsam, A. M. Gorman, C. M. Freeman, and G. Férey, *De Novo Prediction of Inorganic Structures Developed Through Automated Assembly of Secondary Building Units (AASBU Method)*, **Angew. Chem. Int. Ed. Engl. 39:13** (2000), 2270 – 2275.
19. C. Mellot-Draznieks and G. Férey, *Assembling molecular species into 3D frameworks: Computational design and structure solution of hybrid materials*, **Progress in Solid State Chemistry 33** (2005), 187 – 197.
20. M. O'Keefe & B. G. Hyde, *Crystal Structures: I. Patterns and Symmetry* (Mineralogical Society of America, 1996).
21. N. Seeman, *Nucleic acid junctions and lattices*, **J. Theoretical Biology 99** (1999), 237 – 247.
22. M. Senechal, *Brief history of geometrical crystallography*, in: J. Lima-De-Faria et al, eds., **Historical Atlas of Crystallography** (Kluwer, 1990), 43 – 59.
23. V. Ya. Shevchenko, M. I. Samoilovich, A. L. Talis & A. E. Madison, *Structure of Icosahedral Nanoobjects* **Glass Physics & Chemistry 31:6** (2005), 823 – 828.
24. M. M. J. Treacy, I. Rivin, E. Balkovsky & K. H. Randall, *Enumeration of periodic tetrahedral frameworks II. Polynodal graphs*, **Microporous and Mesoporous Materials 74** (2004), 121 - 132.
25. A. F. Wells, **Three-dimensional nets and polyhedra** (Wiley, 1977).
26. O. M. Yaghi, *Reticular chemistry and the design of new materials*, **Proc. 4th ann. conf. Found. Nanoscience (FNANO)** (2007), 12.
27. O. M. Yaghi, M. O'Keefe, N. W. Ockwig, H. K. Chae, M. Eddaoudi & J. Kim, *Reticular synthesis and the design of new materials*, **Nature 423** (2003), 705 – 714.