Lattice Computers for Approximating Euclidean Space

John Case[1]
Computer and Information Sciences Department
103 Smith Hall
University of Delaware
Newark, DE 19716

Dayanand S. Rajan
Knight Securities, LP
525 Washington Blvd.
Jersy City, NJ 07310

Anil M. Shende
Mathematics, Computer Science and Physics Department
Roanoke College
221 College Lane
Salem, VA 24153

[1]The research was supported in part by NSF Grant CCR 8713846. The email address for communication regarding this paper is 'case@cis.udel.edu'.

Abstract

In the context of mesh-like, parallel processing computers for (i) approximating continuous space and (ii) *analog* simulation of the motion of objects and waves in continuous space, the present paper is concerned with *which* mesh-like interconnection of processors might be particularly suitable for the task and why.

Processor interconnection schemes based on nearest neighbor connections in geometric lattices are presented along with motivation. Then two major threads are explored regarding which lattices would be good: the *regular lattices*, for their symmetry and other properties in common with continuous space, and the well-known *root lattices*, for being, in a sense, the lattices required for physically natural basic algorithms for motion.

The main theorem of the present paper implies that the well-known lattice $\mathbf{A}_n$ is *the* regular lattice having the maximum number of nearest neighbors among the n-dimensional regular lattices. It is noted that the only n-dimensional lattices that are both regular and root are $\mathbf{A}_n$ and $\mathbf{Z}^n$ ($\mathbf{Z}^n$ is the lattice of n-cubes). The remainder of the paper specifies other desirable properties of $\mathbf{A}_n$ including other ways it is superior to $\mathbf{Z}^n$ for our purposes.

1 Introduction

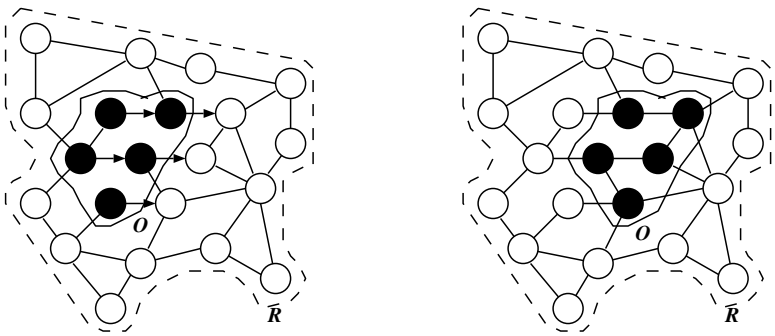
We are interested in multi-processor, mesh-like computers for providing an approximate representation of continuous (euclidean) space and as a medium for the *analogical* (approximate) simulation of the continuous motion of objects and waves in the space so represented [She91, CRS91, CRS94a, CRS94b, CCRS95]. Our *eventual* goals are to provide an analog simulation approach to scientific computing and to have a model of computing which exploits non-verbal representation of spatial/kinematic data for artificial intelligence [CRS94a].

In the present paper we are concerned with *which* mesh-like interconnections of processors are particularly suitable for the task of analogically representing space and motion in it.

Before we overview the results of the present paper, it is useful to provide a simple example illustrating what we have in mind by mesh-like computers providing a representation of continuous space and motion therein.

Consider the problem of simulating the uniform motion of a 2-dimensional object, $\boldsymbol{O}$, within a bounded region, $\boldsymbol{R}$, of the plane (euclidean 2-space).

We first choose a finite set of points from $\boldsymbol{R}$. We place identical, synchronized computer processors centered on each of the points in this set and connect some of these processors by identical bandwidth, bi-directional communication channels having delay time proportional to length. Our choices could produce the mesh-work of computer processors representing $\boldsymbol{R}$ shown in Fig. 1(a). The circles (filled and unfilled) stand for the processors, and the solid lines for direct communication connections.



(a) (b)

Figure 1: Analogical Representation of the Motion of $\boldsymbol{O}$ in $\boldsymbol{R}$.

Imagine that each of the processors in this mesh-work is programmed to "behave" like the region of $\boldsymbol{R}$ around the point on which it it is centered. So, we would have some processors programmed to behave like empty space (the unfilled ones in Fig. 1), and others (the filled ones in Fig. 1) programmed to behave as particles (or chunks) of $\boldsymbol{O}$.

Motion is represented, then, by the synchronized passage of messages, across direct communi-

cation channels, from one processor to another so that a processor that represents a particle of $\boldsymbol{O}$ in its region of space instructs a directly connected processor, in the direction of motion, to start representing that particle in its region of space. The arrows in Fig. 1(a) indicate a particularly simple example direction of motion for $\boldsymbol{O}$. In this example, there is a length ℓ so that the nearest neighbor $\mathbf{P}'$, in the direction of motion, of any processor $\mathbf{P}$ simulating a particle of $\boldsymbol{O}$ happens to be directly connected to $\mathbf{P}$ by a communication channel of length ℓ. Since all the processors in the mesh-work execute in synchrony (or in parallel), after one message passing, $\boldsymbol{O}$ would effectively have moved in the example direction of motion. Fig. 1(b) shows the state of the mesh-work, as far as what each processor represents, after one message passing. Clearly it could be very difficult or impossible to find algorithms to simulate, in the mesh of Fig. 1, some not so judiciously chosen example motions. For this and other reasons, we present, in the present paper, processor interconnection schemes that are more nicely patterned than the higgledy-piggledy scheme in Fig. 1.

For such schemes, our two principal design decisions are that for representing a bounded region $\boldsymbol{R}$ of n-dimensional continuous space (euclidean n-space),

1. we first represent *all* of euclidean n-space by a set of points $\mathcal{D}$ in it and then we place identical, synchronized computer processors centered on each of the points in this set and connect all and only nearest neighbor processors by identical bandwidth, bi-directional communication channels having delay time proportional to the distance between the points on which the processors are centered thus forming the (possibly infinite) *idealized mesh* $\mathcal{M}(\mathcal{D})$;

2. then we cut out for use only the finite portion of the idealized mesh corresponding to $\boldsymbol{R}$.

In the beginning of Section 2 we present four reasonable additional properties we want (Properties 1, 2, 3 and 4) complementing these principal design decisions. These four properties pertain to the collection of points $\mathcal{D}$, from all of euclidean n-space, on which to place processors to form an idealized mesh for n-space. For example, these properties will guarantee that there *is* a minimal distance between points in $\mathcal{D}$ and that every point in n-space is not too far from some point in $\mathcal{D}$. Property 4 says that $\mathcal{D}$ is invariant under translations by vectors in $\mathcal{D}$. This property, equivalent to $\mathcal{D}$ being closed under vector addition, is wonderful for algorithm creation, and makes our approach unlike that of Funt's analog retinal computer [RN74, Fun80] and unlike the finite element method's philosophy of placement of points [CB91]. In the beginning of Section 2 we sketch a proof that satisfaction of the four properties is equivalent to the indicated collection of points being an n-dimensional (geometric) lattice. The formal definition of lattice [GL87, CS93] is presented there, but, informally, an n-dimensional *lattice* is a set of points, from some copy of euclidean n-space, providing a discrete variant of an n-dimensional vector space (see, for example, [Her64]) in euclidean n-space in which the scalar multipliers for magnifying vectors are restricted to being integers. In Fig. 3(a) below in Section 2.1 is depicted a tiny parallelepiped portion of the idealized mesh based

on the important 3-dimensional lattice A_3, where, for dimension $n > 0$,

$$A_n \stackrel{\text{def}}{=} \{(k_0, k_1, k_2, \ldots, k_n) \in \mathbf{Z}^{n+1} \mid \sum_{i=0}^{n} k_i = 0\}, \tag{1}$$

and where $\mathbf{Z}^n$ is the set of all n-tuples of integers (see [CS93] for more about A_n). In Fig. 3 the spheres represent processors and the bars their nearest neighbor connections. Fig. 4 in Section 2.2 below depicts a portion of the idealized mesh based on A_2.

Once we have converged on *lattice computers*, by definition idealized mesh computers based on the points from a lattice, in the rest of the paper we explore two separate threads each having to do with a particular way of considering which lattices and corresponding idealized lattice computers are especially good for our purposes.

The first thread (Section 2.1), the one introduced by the present paper, involves what we call the *regular lattices*, nicely symmetric lattices sharing important properties with euclidean space (e.g., Property 6 in Section 2.1 below). The *regular lattices* are (by definition) those having a generating basis B for which the vectors in B have lattice minimal length and the angle between any two distinct vectors in B is a constant. Intuitively, no direction in B is radically distinguishable from any other; B has a kind of regularity (whence the name) or symmetry. A_n from (1) above is a regular n-dimensional lattice as is its dual A_n^* [CS93]. Fig. 3(b) below in Section 2.1 depicts a tiny parallelepiped portion of the idealized mesh based on A_3^*. A more mundane n-dimensional regular lattice is $\mathbf{Z}^n$ itself. Fig. 2(b) below in Section 2 depicts a portion of the idealized mesh based on $\mathbf{Z}^3$. In euclidean n-space, there are continuum many vectors of any positive length emanating from each point. In $\mathbf{Z}^n$ there are only $2n$ lattice minimal length vectors emanating from each point. Hence, in the idealized mesh based on $\mathbf{Z}^n$, the number of local directions explicitly "wired in" is $2n$. For the idealized mesh based on A_n, the number is $n(n+1)$. In the latter mesh, there is, then, greater resolution of local directions hard-wired in than there is for the idealized mesh based on $\mathbf{Z}^n$. A_n is locally bushier than $\mathbf{Z}^n$. We consider this greater local bushiness indicative of being a finer approximation of euclidean n-space. The major result of the present paper, Theorem 2 in Section 2.1, entails (see Corollary 4 in the same section) that there is a unique (up to geometric similarity) regular lattice with *maximal* local bushiness among the regular lattices, and that regular lattice is A_n.

The second thread we pursue (Section 2.2) involves the lattices for which we have algorithms for finely approximating very basic continuous motions of particles and waves fronts. Interestingly, coincidentally, these lattices are (up to geometric similarity) the well-known *root lattices* [CS93]. In later discussion of root lattices we will omit the tedious qualifier 'up to geometric similarity'. In the next three paragraphs, first is listed which algorithms we have (to date), then a definition of root lattice is presented, and lastly the results of Section 2.2 are briefly indicated.

We have algorithms (so far)

1. for simulating collections of particles with each particle moving at constant speed in a straight line [She91, CRS91, CCRS95] or in a circle [She91] and

2. for simulating constant speed, non-dissipating spherical wave fronts [She91, CRS94b].

We expect that more general particle, fluid, and solid body motion can be piecewise analogically simulated with the algorithms of 1 just above; doing so is work in progress. We also expect that the algorithms of 2 just above can be used in analog simulation of magnification and zooming of figures. Of course in simulating the motion of multiple particles (for example, in fluids and solid bodies) one nicely *and naturally* exploits the inherent parallelism of mesh computers.

In the present paper we will work with a geometric characterization from [RS97] (cited in [CS93]) of the n-dimensional root lattices.[1] First: the *Voronoi cell* around a point P in a lattice embedded in euclidean space of the same dimension is (by definition [CS93, Aur90]) the set of points in that euclidean space at least as close to P as to any other lattice point. (For A_n, the associated copy of n-dimensional euclidean space is the set of $(n+1)$-tuples of *reals* that add to 0. For example, the set of 3-tuples of *reals* that add to 0 is just a plane, a copy of 2-dimensional euclidean space.) Intuitively, the Voronoi cell around a lattice point P is the natural set of points in the corresponding euclidean space that is approximated by the single lattice point P. (Of course the Voronoi cell around P shares boundaries with the each of the Voronoi cells around P's near neighbors in the lattice.) Then: the n-dimensional *root lattices* are the n-dimensional lattices whose Voronoi cells have inscribed n-spheres. A_n and $\mathbf{Z}^n$ are n-dimensional root lattices. Fig. 4 below in Section 2.2 shows, in dashed lines, the hexagonal shaped Voronoi cell around a lattice point in A_2. Fig. 5 below, also in Section 2.2, shows, in dashed lines, the hexagonal Voronoi cells around lattice points in a 2-dimensional lattice which fails to be a root lattice. Note that its Voronoi cells, by contrast with A_2's, are skewed a bit, and, hence, do not admit inscribed 2-spheres (circles). For each $n > 3$, the n-dimensional lattice D_n, defined as the set of n-tuples of integers that add up to an even number [CS93], is root but not regular. The lattice A_n^* is regular, not root.

Section 2.2 presents and motivates properties of lattices clearly desirable for lattice computer algorithms for *analog* simulation of physical motion. In fact our algorithms mentioned and referenced above depend on the (overlapping) conjunction of these properties. The results of Section 2.2 entail that the conjunction of these properties exactly characterizes the root lattices.

Some lattices, e.g., A_n, are both regular and root, and there are important connections between the threads. For example, vector inner product, a crucial operation for our root lattice algorithms, is particularly easy to compute in regular lattices. Theorem 7 of Section 2.3 says that the *only* n-dimensional lattices that are both regular and root are (up to geometric similarity) A_n and $\mathbf{Z}^n$.

[1] See also [RS96]. Any results in the present paper not totally attributed to [RS97] are new to the present paper and appear in neither [RS96] nor [RS97].

Hence, in the light of this theorem and our major result that A_n has maximal local interconnectivity among the regular lattices, Section 2.3 bills A_n as the best of both worlds (threads).

Section 3 then builds on Section 2.3 and presents many other relevant properties of A_n.

Section 3.1 presents properties of A_n pertaining to computational cleanness. For example, in Section 3.1.2 we point out that lattice meshes based on A_n can be elegantly partitioned by dimension and similarly how algorithms for them can be easily parameterized by dimension.

Section 3.2 provides a comparison of A_n and $\mathbf{Z}^n$ further witnessing the superiority of A_n to $\mathbf{Z}^n$. Note that the lattice minimal distance in A_n is $\sqrt{2}$, and in $\mathbf{Z}^n$ it is 1. This sometimes makes meaningfully comparing A_n and $\mathbf{Z}^n$ difficult. We circumvent this difficulty as follows. For each n, let $\mathbf{R}^n$ be the set of all n-tuples of reals, the standard euclidean n-space. As noted above, A_n is contained in the n-dimensional euclidean space

$$\{(x_0, x_1, x_2, \ldots, x_n) \in \mathbf{R}^{n+1} \mid \sum_{i=0}^{n} x_i = 0\}. \tag{2}$$

Clearly, this space together with A_n can, in effect, be rotated about the origin *within* $\mathbf{R}^{n+1}$ *into* $\mathbf{R}^n$ and shrunk by a factor of $\sqrt{2}$. Let $\mathbf{A}_n$ be (by definition) the lattice resulting from restricting this transformation to A_n. Clearly $\mathbf{A}_n$ and $\mathbf{Z}^n$ have unit lattice minimal distance and each is conveniently embedded in the standard n-dimensional euclidean space $\mathbf{R}^n$. Thus, when comparing A_n and $\mathbf{Z}^n$, many times we will work with the geometrically similar variant $\mathbf{A}_n$ of A_n in place of A_n.

In Section 3.2.1 we show that $\mathbf{Z}^n$ can be *rigidly* embedded in some $\mathbf{A}_{n'}$ but not conversely.

In Section 3.2.2 it is noted that the density of points is greater in $\mathbf{A}_n$ than in $\mathbf{Z}^n$.

In Section 3.2.3 it is shown that, in a sense, $\mathbf{A}_n$ provides superior coverage of the points in $\mathbf{R}^n$ than does $\mathbf{Z}^n$. Specifically, it is noted that the maximal n such that every point in $\mathbf{R}^n$ is within the lattice minimal distance of a lattice point of $\mathbf{A}_n$ is larger than that for $\mathbf{Z}^n$.

In Section 3.2.5 it is shown that the interconnection or communication bandwidth across comparable partition boundaries is greater in the meshes based on $\mathbf{A}_n$ than in those based on $\mathbf{Z}^n$.

Section 3.2.6 discusses the automorphism group symmetries of $\mathbf{A}_n$ and $\mathbf{Z}^n$ and the apparent comparative relevance of that for us and for analog modeling of fluid flow in Lattice Gas Automata [FHP86, Has87].

In Section 3.3 we present a scheme, which can be described as simply recursive in dimension n, for actually embedding in real 3-dimensional space, any finite submesh of the idealized mesh based on $\mathbf{A}_n$. For example, the scheme specifies 3-dimensional layouts for constructing in the real world any finite submesh of the idealized mesh based on $\mathbf{A}_4$.

Lastly, in Section 3.4 we play devil's advocate a bit concerning the present paper's focus on lattices which are both regular and root. We discuss, for dimensions $n \geq 4$, root lattices which are

not regular but which are denser than $\mathbf{A}_n$.[2]

2 Lattice Computers to Approximate Continuous Space

Let $\mathbf{N} = \{0, 1, 2, \ldots\}$, the set of natural numbers. Let $\mathbf{Z}$, $\mathbf{Q}$, and $\mathbf{R}$ denote the sets of integers, rationals, and reals, respectively. For $n \in \mathbf{N}$ and $R \in \{\mathbf{N}, \mathbf{Z}, \mathbf{Q}, \mathbf{R}\}$, R^n is the n-fold cartesian product $R \times \cdots \times R$. The (standard) orthonormal basis (see, e.g., [Her64]) for $\mathbf{R}^n$ is denoted by $\{\vec{e}_1, \ldots, \vec{e}_n\}$.[3]

Definition 1 $\mathbf{E}$ *is a* euclidean n-space *iff* $\mathbf{E} \subseteq \mathbf{R}^{n'}$, *for some* $n' \geq n$ *such that there are positive real* m *and an isometry*[4] T *mapping* $\mathbf{R}^n$ *(1-1) into* $\mathbf{R}^{n'}$ *for which* $m \cdot T(\mathbf{R}^n) = \mathbf{E}$.

A euclidean n-space is just a subset of $\mathbf{R}^{n'}$, for some $n' \geq n$, which is *geometrically similar* to $\mathbf{R}^n$ and contains the zero vector of $\mathbf{R}^{n'}$. For example, $\mathbf{R}^2$ and also the plane, $\{(x, y, z) \in \mathbf{R}^3 \mid x + y + z = 0\}$, are copies of euclidean 2-space. More generally, $\mathbf{R}^n$ and also $\{(x_1, x_2, \ldots, x_{n+1}) \in \mathbf{R}^{n+1} \mid x_1 + x_2 + \ldots + x_{n+1} = 0\}$ are copies of euclidean n-space.

Let $\mathcal{D}$ be a set of points in a euclidean n-space. We are interested in the possibility that the points in $\mathcal{D}$ correspond to the processors in an *idealized mesh computer* with nearest neighbor direct communication links as described in Section 1 above. As indicated in Section 1, such an idealized mesh computer will be denoted by $\mathcal{M}(\mathcal{D})$ and will be referred to as the *idealized mesh computer based on* $\mathcal{D}$. $\mathcal{D}$ is called the *underlying set* of $\mathcal{M}(\mathcal{D})$. For a point P in $\mathcal{D}$, we will denote the *processor at point* P by $\mathcal{M}(P)$; P will be called the *point corresponding to processor* $\mathcal{M}(P)$. In this paper, we will not make any distinction between the interconnection scheme for $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D})$ itself. We introduce desirable properties (1 through 10) for $\mathcal{D}$ to have, in sequence.

Definition 2 *Suppose* $\mathbf{E}$ *is a euclidean* n*-space. An* approximation *of* $\mathbf{E}$ *is a countable subset,* $\mathcal{D}$*, of* $\mathbf{E}$ *for which there is a* $p = p(\mathcal{D}) \geq 0$ *so that every open ball* [5] *in* $\mathbf{E}$*, with radius greater than* p*, contains at least one point of* $\mathcal{D}$. $p(\mathcal{D})$ *is called a* measure of proximity of $\mathcal{D}$.

Informally, each point in $\mathbf{E}$ is not very far away from some point in $\mathcal{D}$.

The sets $\mathbf{Q}^n$, $\mathbf{Z}^n$, $\mathcal{D}_1 = \mathbf{Z}^n \setminus (2\mathbf{Z})^n$ and $\mathcal{D}_2 = \mathbf{Z}^n \cup \{(x_1, \ldots, x_n) \mid x_j = (2k + 1/i), k, i \in \mathbf{Z}, i > 0\}$ are all approximations of $\mathbf{R}^n$, while $\{(\pm 2^{i_1}, \ldots, \pm 2^{i_n}) \mid i_1, \ldots, i_n \in \mathbf{Z}\}$ is not.

Let $\mathcal{D}$ be a set of points in a euclidean n-space $\mathbf{E}$. We will, in the rest of Section 2, list various desirable properties.

[2] $\mathbf{A}_2$ and $\mathbf{A}_3$ *are* the densest 2 and 3-dimensional lattices [CS93].

[3] $\vec{e}_i$ is (by definition) the n-tuple of reals with a 1 in the i-th position and 0's elsewhere.

[4] An *isometry* is (by definition) a linear transformation preserving euclidean distances.

[5] The *open ball of radius* r*, centered at* c *in* $\mathbf{E}$ is the set of points in $\mathbf{E}$ whose distance from c is less than r. This is a standard topological notion and the reader is referred to any standard text on topology for further details.

Property 1 $\mathcal{D}$ IS AN APPROXIMATION OF $\mathbf{E}$.

Property 2 NO BOUNDED REGION OF $\mathbf{E}$ CONTAINS AN INFINITE SUBSET OF $\mathcal{D}$.

We require this property because processors cannot be placed arbitrarily close together inside a bounded region.

Definition 3 *An approximation $\mathcal{D}$ of $\mathbf{R}^n$ is a* discretization *iff each point of $\mathcal{D}$ is contained in an open ball, of $\mathbf{R}^n$, containing no other point of $\mathcal{D}$.*

$\mathcal{D}$ satisfying Property 2 is equivalent to $\mathcal{D}$ being a discretization.

Of the four approximations in the examples above, $\mathbf{Q}^n$ and $\mathcal{D}_2$ are not discretizations, while $\mathbf{Z}^n$ and $\mathcal{D}_1$ are.

Physical connections between processors in a possibly infinite idealized mesh cannot be of arbitrarily small length either. So we require $\mathcal{D}$ to have the following property.

Property 3 $\mathcal{D}$ IS A DISCRETE SUBSET OF $\mathbf{E}$; that is, there is a real number, $\delta > 0$, so that the euclidean distance between any two points in $\mathcal{D}$ is at least δ. Informally, points in $\mathcal{D}$ do not cluster too closely. If there are two points in $\mathcal{D}$ so that the distance between these two points *is* such a δ, then we say $\mathcal{D}$ *has minimal distance* δ (or that the *minimal distance of* $\mathcal{D}$ *is* δ). *We will denote the minimal distance of* $\mathcal{D}$ *by* $\mu(\mathcal{D})$.

Obviously, if $\mathcal{D}$ has Property 3, then $\mathcal{D}$ has Property 2. The converse is not true, since, for example, the discretization $\mathcal{D} = \mathbf{Z} \cup \{i + (1/i) \mid i \in \mathbf{Z}, i \neq 0\}$, of $\mathbf{R}$, has Property 2, but does not have Property 3.

The points in euclidean n-space can and should be thought of as vectors. A translation of $\mathcal{D}$ by some vector in $\mathbf{E}$ is the set obtained by adding that vector to each vector in $\mathcal{D}$. It is clearly too much to require that *every* translation of $\mathcal{D}$ by a vector in $\mathbf{E}$ is contained in $\mathcal{D}$ (then $\mathcal{D}$ would be $\mathbf{E}$). However, in the interest of geometric uniformity of $\mathcal{D}$ as an approximation to euclidean n-space and in the corresponding algebraic interest of uniformity and existence of possible algorithms, we require

Property 4 $\mathcal{D}$ IS INVARIANT UNDER TRANSLATIONS BY VECTORS IN $\mathcal{D}$; equivalently, $\mathcal{D}$ is closed under vector addition.

In what follows we will freely use standard notions, such as vector space, basis, linearly independent set, span, inner product, fields, groups, etc., from algebra; the reader is referred to [Her64] or any other standard modern algebra text for a precise treatment of these and other relevant concepts.

Convention 1 IN WHAT FOLLOWS, BY *linear independence* WE WILL ALWAYS MEAN LINEAR INDEPENDENCE WITH RESPECT TO THE *reals* AS SCALARS.

Next we define *lattice* and then show (Theorem 1) that being an n-dimensional lattice is equivalent to satisfying Properties 1, 2, 3 and 4 above.

Definition 4 *$\mathcal{L}$ is a* lattice *iff there is an $n' \in \mathbf{N}$ and a finite, linearly independent subset $S = \{s_1, \ldots, s_n\}$ of the* vector space $\mathbf{R}^{n'}$ *such that*

$$\mathcal{L} = \{x \in \mathbf{R}^{n'} | (\exists \alpha_1, \ldots, \alpha_n \in \mathbf{Z})[x = \alpha_1 \cdot s_1 + \cdots + \alpha_n \cdot s_n]\}.$$

In this case we say that S is a basis *for the lattice $\mathcal{L}$ and that the* dimension *of $\mathcal{L}$ is $card(S) = n$. (N.B. Every basis for a lattice has the same number of elements (see [GL87]) and so the dimension of a lattice is well-defined.) The* origin*, denoted $\vec{0}$, of $\mathcal{L}$, is the origin of $\mathbf{R}^{n'}$, i.e., the n'-tuple of zeroes.*

The notions of vector, origin, basis, linear transformation, isomorphism, etc., are all similar to those in euclidean space (considered as a vector-space over the field of real numbers). The length of a vector $\vec{d}$ (denoted $\| \vec{d} \|$) in a lattice is the same as the length of that vector in euclidean space.

Lemma 1 *A subset $\mathcal{D}$ of $\mathbf{E}$ satisfies Properties 1, 3 and 4 iff $\mathcal{D}$ is a discrete subgroup of the additive group $\mathbf{E}$ and is not contained in a euclidean $(n-1)$-space.*

Proof. Suppose $\mathcal{D}$ satisfies Properties 1, 3 and 4. Property 1 guarantees that $\mathcal{D}$ is not contained in any euclidean $(n-1)$-space. Property 3 guarantees that $\mathcal{D}$ is a discrete subset of $\mathbf{E}$. Property 4 guarantees that $\mathcal{D}$, along with the operation of vector addition in $\mathbf{E}$, and the zero vector as the identity element, satisfies all the group axioms and hence is a subgroup of $\mathbf{E}$.
The reasoning for the converse is quite similar. □

The following theorem is, then, just a combination of Lemma 1 above and Theorems 1 and 2 proved in [GL87, pages 18–19].

Theorem 1 *A subset $\mathcal{D}$ of $\mathbf{E}$ is an n-dimensional lattice iff $\mathcal{D}$ has properties 1, 3 and 4.* □

Note that in the presence of Property 4, Property 2 is actually equivalent to Property 3.

Definition 5 *For lattices, $\mathcal{L}$ and $\mathcal{L}'$, we say $\mathcal{L} \simeq \mathcal{L}'$ iff there are a positive real m and an additive group isomorphism T which preserves euclidean distances such that $m \cdot T(\mathcal{L}) = \mathcal{L}'$.*

Note that $\simeq$ is an equivalence relation. Lattices belonging to the same equivalence class of $\simeq$ are geometrically similar; they have the same "structural layout." For example, $\mathbf{Z}^2 \simeq 2\mathbf{Z}^2$. Recall from Section 1 that A_n is contained in an n-dimensional subspace of $\mathbf{R}^{n+1}$ and that $\mathbf{A}_n$, a subset of $\mathbf{R}^n$, is obtained by rotating A_n about the origin within $\mathbf{R}^{n+1}$ into $\mathbf{R}^n$ and then shrinking this rotated version of A_n by a factor of $\sqrt{2}$. Clearly, then, $\mathbf{A}_n \simeq A_n$.

In what follows, we are mostly interested in the "structural layout" of lattices, not their exact set-theoretic definitions.[6] It is expositionally painful to carry around as parameters a lattice's minimal distance and which euclidean n-space it approximates. Furthermore, in Section 3.2 we need to normalize the minimal distance of lattices to meaningfully compare them. Hence, we adopt the following convention.

Convention 2 FROM NOW ON WE WILL ASSUME THAT, *unless we explicitly say otherwise*, EVERY n-DIMENSIONAL LATTICE WE MENTION IS A SUBSET OF $\mathbf{R}^n$, AND HAS MINIMAL DISTANCE 1. FURTHER, *in the rest of this paper*, $\mathcal{D}$ WILL ALWAYS DENOTE AN n-DIMENSIONAL LATTICE WITH MINIMAL DISTANCE 1.

In all the terminology we define below and in the results we present, we assume Convention 2 *unless we say otherwise*. Nonetheless, it should be noted that most of the terminology and results apply *mutatis mutandis* to lattices not subject to Convention 2. Also, when we speak of a particular lattice such as A_n, which does not have minimal distance = 1, we should be implicitly understood as making an exception to Convention 2.

The distance between points, P and Q, in a lattice is the length of the vector $(Q - P)$, from P to Q. So the minimal length of vectors in a lattice is exactly the minimal distance of that lattice considered as an approximation of euclidean space (of appropriate dimension). Henceforth, we will not distinguish between the points in a lattice and the corresponding vectors.

Definition 6 *The* set of basic directions *in $\mathcal{D}$, denoted $Dir(\mathcal{D})$, is the set of all vectors, in $\mathcal{D}$, of length $\mu(\mathcal{D})$ $(= 1$ as per our Convention 2 above).*

Recall that $\mathcal{M}(\mathcal{D})$ is the lattice computer (idealized mesh) based on the lattice $\mathcal{D}$. Figure 2 shows a piece of $\mathcal{M}(\mathbf{Z}^3)$. The filled-in circles denote processors, and the thick lines denote identical, bi-directional communication channels between processors at points the $\mathbf{Z}^3$ lattice minimal distance apart.

The following proposition is an immediate consequence of our definition of lattice computers, especially the design decision that only nearest neighboring processors are directly connected.

Proposition 1 *A lattice computer, $\mathcal{M}(\mathcal{D})$, is connected iff there exists a basis, B, for $\mathcal{D}$ such that $B \subseteq Dir(\mathcal{D})$.* □

2.1 Regular Lattice Computers and a Maximality Theorem

Lattices exist in abundance, and are quite varied in their properties [GL87, CS93]. Although the study of lattices is quite incomplete [GL87, CS93], it is known that many facts, clearly true about

[6] An important exception is in Property 8 in Section 2.2 where we are concerned with being able to use integer coordinates to describe the vectors of lattices.

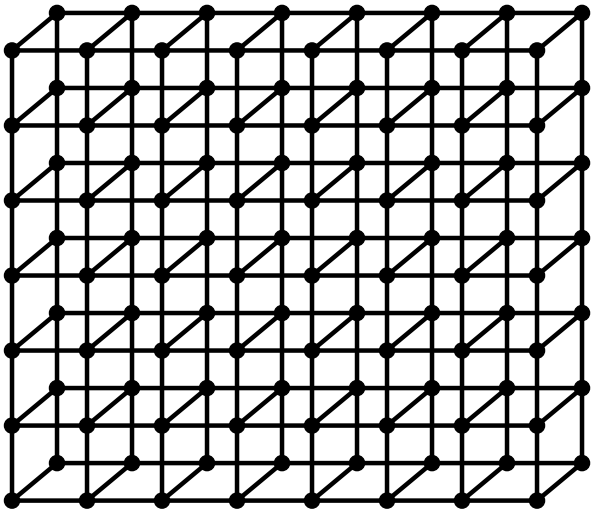


Figure 2: A piece of $\mathcal{M}(\mathbf{Z}^3)$

$\mathbf{R}^n$, are surprisingly not true in all lattices [GL87, CS93]. For example, there is *only one* (up to $\simeq$) n-dimensional lattice, $\mathbf{Z}^n$, with an orthonormal basis[7]. The existence of an orthonormal basis for $\mathbf{Z}^n$ facilitates clean computation within $\mathbf{Z}^n$. Cleanness in computation is important for the efficient realization of our methodology in a mesh. Yet, considering the way we expect to use a mesh to simulate physical phenomena, it would be to our advantage if there were many connections per processor. We wish to exactly represent as many local directions as possible.

Among n-dimensional lattices, $\mathbf{Z}^n$ does not have the maximum number of basic direction vectors. Hence, we will forgo the convenience of a lattice with an orthonormal basis in the hope that we will find a lattice with more basic direction vectors (i.e., more local directions exactly represented), and in which computations are nearly as clean.

Definition 7 *A basis, $B = \{\vec{b}_1, \ldots, \vec{b}_n\}$, for $\mathcal{D}$, is* regular *iff there is a constant, $K(\mathcal{D}, B)$, so that the inner product*

$$(\vec{b}_i, \vec{b}_j) = \begin{cases} K(\mathcal{D}, B), & \text{if } i \neq j; \\ 1, & \text{if } i = j. \end{cases}$$

$\mathcal{D}$ itself is said to be regular *iff $\mathcal{D}$ has a regular basis.*

Hence, a regular basis B is one in which the vectors in B have unit length and the angle between any two distinct vectors in B is a constant.

Having a regular basis is a powerful symmetry condition to the effect that with respect to some basis one dimension is not to be preferred to another. The existence of a regular basis is important for both computational cleanness and having a lattice that *nicely* represents the corresponding

[7] An *orthonormal basis* is a basis of minimal length, pairwise perpendicular vectors. Hence, the inner product between any distinct two of them is 0.

euclidean space of the same dimension. Our methodology dictates that our algorithms [She91, CRS94b, CRS91] be local and independent of any fixed co-ordinatization of euclidean space. Vector inner product is an *extremely* basic operation in our algorithms; having its calculation simplified is guaranteed by the existence of a regular basis.

The following characterization of $\mathbf{Z}^n$ shows that it is regular in the strongest possible sense.

Proposition 2 *$\mathbf{Z}^n$ is the unique (up to $\simeq$) n-dimensional lattice, $\mathcal{D}$, in which* every *basis contained in $Dir(\mathcal{D})$ is regular.* □

On the other hand, there are numerous lattices, $\mathcal{D}$, satisfying

Property 5 $\mathcal{D}$ IS REGULAR.

The best one can hope for is that there is a unique (up to $\simeq$) regular n-dimensional lattice (on which to base an idealized mesh) for which the number of basic direction vectors is maximum. We now state and prove a theorem (Theorem 2) which, by explicitly giving expressions for the number of basic direction vectors in regular lattices, shows us that our hope can be fulfilled (see Corollary 4). Corollaries 1, 2, and 3 below isolate some other useful consequences.

The following lemma implies that, in the proof below of Theorem 2 below, we consider all possible values for the inner product between two basic direction vectors in a lattice (with minimal distance 1 as per Convention 2). Equation (6) of Theorem 2 strengthens this lemma for the case of regular lattices.

Recall that, by our Convention 2 above, that $\mathcal{D}$ is an n-dimensional lattice with minimal distance 1. Then:

Lemma 2 *The inner product of any two non-collinear vectors $\vec{d}_1, \vec{d}_2$ in $Dir(\mathcal{D})$ lies in the interval $[-1/2, 1/2] \subset \mathbf{R}$.* □

PROOF. Since $\vec{d}_1$ and $\vec{d}_2$ are unit length, the inner product

$$(\vec{d}_1, \vec{d}_2) = \cos(\theta), \tag{3}$$

where θ is the angle between $\vec{d}_1$ and $\vec{d}_2$. Suppose by way of contradiction, then, that θ satisfies.

$$0 < \theta < \pi/3 \tag{4}$$

or

$$2\pi/3 < \theta < \pi. \tag{5}$$

Then, if (4), $0 < \| \vec{d}_1 - \vec{d}_2 \| < 1$, a contradiction. If (5), then $0 < \| \vec{d}_1 + \vec{d}_2 \| < 1$, a contradiction.

□

Theorem 2 *Let $n > 1$, let $\mathcal{D}$ be* regular with regular basis B, *and let $k = K(\mathcal{D}, B)$. Then:*

$$k \in [-1/n, +1/2], \tag{6}$$

and

$$card(Dir(\mathcal{D})) = \begin{cases} 2(n+1), & \text{if } k = -\frac{1}{n}; \\ 2n, & \text{if } -\frac{1}{n} < k < \frac{1}{2}; \\ n(n+1), & \text{if } k = \frac{1}{2}. \end{cases}$$

Proof. By Lemma 2 above we can restrict our attention to $k \in [-1/2, +1/2]$. In this proof, when we speak of $\vec{d}$, we will assume $\vec{d} = \sum_{i=1}^{n} p_i \vec{b}_i \in Dir(\mathcal{D})$, $\vec{b}_i \in B, p_i \in \mathbf{Z}$. Then,

$$1 = \| \vec{d} \|^2 = \sum_{i=1}^{n} p_i^2 + 2k \sum_{1 \le i < j \le n} p_i p_j. \tag{7}$$

Clearly, also,

$$(\exists i \mid 1 \le i \le n)[p_i \neq 0]. \tag{8}$$

Therefore, since each p_i is an integer,

$$\sum_{i=1}^{n} p_i^2 \ge 1. \tag{9}$$

Case 0: $(\forall \vec{d} \in Dir(\mathcal{D}))[2k \sum_{1 \le i < j \le n} p_i p_j = 0]$.

Then $1 = \sum_{i=1}^{n} p_i^2$. So $\vec{d} = \pm \vec{b}_i$ for some $\vec{b}_i$, giving us $card(Dir(\mathcal{D})) = 2n$ in this case.

Case 1: $0 \le k < 1/2$.

This case reduces to Case 0: If $k = 0$ or $(\forall \vec{d} \in Dir(\mathcal{D}))[\sum_{1 \le i < j \le n} p_i p_j = 0]$, this is obvious.

So suppose, by way of contradiction, $k > 0$ and $\vec{d} \in Dir(\mathcal{D})$ is such that $\sum_{1 \le i < j \le n} p_i p_j \neq 0$. Then, using Equations (7) and (9),

$$\sum_{1 \le i < j \le n} p_i p_j < 0. \tag{10}$$

Also, since $0 \le k < 1/2$ in this case, using Equations (7) and (10), we have

$$\sum_{i=1}^{n} p_i^2 + \sum_{1 \le i < j \le n} p_i p_j < 1. \tag{11}$$

Now, using Equations (7) and (10), and our hypothesis that $0 < k < 1/2$, we have

$$0 \leq \left(\sum_{i=1}^{n} p_i\right)^2 = \sum_{i=1}^{n} p_i^2 + 2 \sum_{1 \leq i < j \leq n} p_i p_j < \sum_{i=1}^{n} p_i^2 + 2k \sum_{1 \leq i < j \leq n} p_i p_j = 1.$$

Since each p_i is an integer, we have

$$\sum_{i=1}^{n} p_i^2 + 2 \sum_{1 \leq i < j \leq n} p_i p_j = 0. \tag{12}$$

Also, by Equations (12), (10), and (11), we have

$$0 = \sum_{i=1}^{n} p_i^2 + 2 \sum_{1 \leq i < j \leq n} p_i p_j < \sum_{i=1}^{n} p_i^2 + \sum_{1 \leq i < j \leq n} p_i p_j < 1. \tag{13}$$

Since each p_i is an integer, this is a contradiction.

Hence, $k = 0$ or $(\forall \vec{d} \in Dir(\mathcal{D}))[\sum_{1 \leq i < j \leq n} p_i p_j = 0]$. Therefore, as in the analysis of CASE 0, $card(Dir(\mathcal{D})) = 2n$.

CASE 2: $k = 1/2$. For each $\vec{d} \in Dir(\mathcal{D})$, using Equation (7), we have

$$\sum_{i=1}^{n} p_i^2 + \sum_{1 \leq i < j \leq n} p_i p_j = 1, \tag{14}$$

and, then, using Equation (8) and the fact that each p_i is an integer, we have

$$\sum_{1 \leq i < j \leq n} p_i p_j \leq 0. \tag{15}$$

Now, using Equations (15) and (14), we have

$$1 \geq 1 + \sum_{1 \leq i < j \leq n} p_i p_j = \sum_{i=1}^{n} p_i^2 + 2 \sum_{1 \leq i < j \leq n} p_i p_j = \left(\sum_{i=1}^{n} p_i\right)^2 \geq 0, \tag{16}$$

and, therefore,

$$\sum_{1 \leq i < j \leq n} p_i p_j = 0 \text{ or } -1. \tag{17}$$

We count the number of $\vec{d} \in Dir(\mathcal{D})$ satisfying *each* of the possible outcomes in Equation (17). For $\vec{d} \in Dir(\mathcal{D})$ satisfying $\sum_{1 \leq i < j \leq n} p_i p_j = 0$, CASE 0 applies, and there are $2n$ ways this can happen: $\vec{d} = \pm \vec{b}_i$.

For $\vec{d} \in Dir(\mathcal{D})$ satisfying $\sum_{1 \le i < j \le n} p_i p_j = -1$, using Equation (14), we have

$$\sum_{i=1}^{n} p_i^2 = 2, \tag{18}$$

and then, using Equation (16), we have

$$\sum_{i=1}^{n} p_i = 0. \tag{19}$$

Thus, this happens iff: exactly two of the p_i's are non-zero; one of the non-zero p_i's is a 1 and the other is a -1. This gives us $n(n-1)$ choices for $\vec{d}$.

Therefore, in all, we have $card(Dir(\mathcal{D})) = 2n + n(n-1) = n(n+1)$.

CASE 3: $-1/2 \le k < 0$.

CLAIM 1: For each $\vec{d} \in Dir(\mathcal{D})$, either each $p_i \ge 0$ or each $p_i \le 0$.

If, for each $\vec{d} \in Dir(\mathcal{D})$, there is only one non-zero p_i, then the claim is trivially true.
Suppose $\vec{d} \in Dir(\mathcal{D})$ is such that there are more than one non-zero associated p_i's.
Pick a $p_{i_0} \neq 0$.
Let $\vec{w} = \vec{d} - p_{i_0}\vec{b}_{i_0}$. Since, then, $\vec{w} \neq 0$, the minimal distance in the lattice is 1, and $\vec{w}$ is a lattice vector,

$$\|\vec{w}\|^2 \ge 1. \tag{20}$$

Then,

$$\begin{aligned} 1 = \|\vec{d}\|^2 &= \|\vec{w} + p_{i_0}\vec{b_{i_0}}\|^2 \\ &= \|\vec{w}\|^2 + \|p_{i_0}\vec{b_{i_0}}\|^2 + 2(\vec{w}, p_{i_0}\vec{b_{i_0}}) \\ &= \|\vec{w}\|^2 + \|p_{i_0}\vec{b_{i_0}}\|^2 + 2(\vec{d}, p_{i_0}\vec{b_{i_0}}) - 2(p_{i_0}\vec{b_{i_0}}, p_{i_0}\vec{b_{i_0}}) \\ &= \|\vec{w}\|^2 + {p_{i_0}}^2 + 2kp_{i_0}(\sum_{i=1,\ i\neq i_0}^{n} p_i). \end{aligned}$$

Therefore,

$$-2kp_{i_0}(\sum_{i=1,\ i\neq i_0}^{n} p_i) = \|\vec{w}\|^2 + {p_{i_0}}^2 - 1. \tag{21}$$

In this case, since $k < 0$, $-2k > 0$. Then, since $p_{i_0} \neq 0$ and Equation (20), Equation (21) implies

$$p_{i_0}(\sum_{i=1,\ i\neq i_0}^{n} p_i) > 0,$$

and, therefore, adding ${p_{i_0}}^2$ to the left hand side, we have

$$p_{i_0}(\sum_{i=1}^{n} p_i) > 0. \tag{22}$$

Equation (22) applies to *each* i_0 such that $p_{i_0} \neq 0$. Clearly $(\sum_{i=1}^{n} p_i)$ is either positive or negative. In each case, then, the parity of *each* $p_{i_0} \neq 0$ is fixed one way or the other, thus proving Claim 1.

CLAIM 2: For each $\vec{d} \in Dir(\mathcal{D}), (\sum_{1 \leq i < j \leq n} p_i p_j) \geq 0$.

This is a direct consequence of Claim 1.

CLAIM 3: $k \geq -1/n$.

Let $\vec{v} = \sum_{i=1}^{n} \vec{b}_i$. Since the minimal distance in the lattice is 1, $\|v\|^2 \geq 1$. That is,

$$\| \vec{v} \|^2 = \sum_{i=1}^{n} 1^2 + 2k \sum_{1 \leq i < j \leq n} 1 \cdot 1 = n + 2k\frac{n(n-1)}{2} = n + kn(n-1) \geq 1.$$

Therefore, subtracting n from both sides of the last inequality just above and then dividing both resultant sides by $n(n-1)$, using the hypothesis that $n > 1$, we have that $k \geq -1/n$.

CLAIM 4: If $k > -1/n$, then $(\forall \vec{d} \in Dir(\mathcal{D}))[\sum_{1 \leq i < j \leq n} p_i p_j = 0]$.

Suppose, by way of contradiction, $k > -1/n$ and $\vec{d} \in Dir(\mathcal{D})$ is such that $\sum_{1 \leq i < j \leq n} p_i p_j \neq 0$. Then, by Claim 2,

$$\sum_{1 \leq i < j \leq n} p_i p_j > 0. \tag{23}$$

By a straightforward counting argument,

$$\sum_{1 \leq i < j \leq n} (p_i^2 + p_j^2) = (n-1)\sum_{i=1}^{n} p_i^2. \tag{24}$$

Hence, $n((\sum_{i=1}^{n} p_i^2) - 1) - 2\sum_{1 \leq i < j \leq n} p_i p_j = \sum_{1 \leq i < j \leq n} (p_i^2 - 2p_i p_j + p_j^2) + (\sum_{i=1}^{n} p_i^2) - n = \sum_{1 \leq i < j \leq n} (p_i - p_j)^2 + \sum_{i=1}^{n} p_i^2 - n$.

Therefore,

$$n(\sum_{i=1}^{n} p_i^2 - 1) - 2 \sum_{1 \leq i < j \leq n} p_i p_j = \sum_{1 \leq i < j \leq n} (p_i - p_j)^2 + \sum_{i=1}^{n} p_i^2 - n. \tag{25}$$

Now, if exactly $m \geq 1$ of the p_i's are non-zero, then $\sum_{i=1}^{n} p_i^2 \geq m$ and, by a straightforward

counting argument,

$$\sum_{1\leq i<j\leq n}(p_i^2-p_j^2)\geq \sum_{\substack{[1\leq i<j\leq n\\ \text{and}\\ [[p_i=0 \text{ and } p_j\neq 0]\\ \text{or}\\ [p_i\neq 0 \text{ and } p_j=0]]]}}(p_i^2-p_j^2), \tag{26}$$

which

$$=\sum_{\substack{[1\leq i,j\leq n\\ \text{and}\\ [p_i=0 \text{ and } p_j\neq 0]]}}(p_i^2-p_j^2)\geq (n-m)m. \tag{27}$$

So, we have

$$\sum_{1\leq i<j\leq n}(p_i-p_j)^2+(\sum_{i=1}^{n}p_i^2)-n\geq (n-m)m+m-n=(n-m)(m-1)\geq 0. \tag{28}$$

Hence, combining Equations, (25) and (28), we have

$$n((\sum_{i=1}^{n}p_i^2)-1)\geq 2\sum_{1\leq i<j\leq n}p_ip_j. \tag{29}$$

Then, from Equations (7), (23) ,and (29), we have

$$k\leq -1/n,$$

giving us a contradiction, and, hence, proving this claim.

CLAIM 5: Suppose $k=-1/n$. Suppose, for $\vec{d}\in Dir(\mathcal{D})$, that exactly $m\geq 1$ of the p_i's are non-zero. Then, $m=n$ or $m=1$ and each non-zero p_i is 1 or each non-zero p_i is -1.

By Claim 1, each non-zero p_i is is positive or each non-zero p_i is negative. We prove Claim 5 for the case where each non-zero p_i is positive, the other case being similar. Suppose, then, without loss of generality, $\vec{d}\in Dir(\mathcal{D})$ is such that, $p_i>0$ for $1\leq i\leq m\leq n$ and $p_i=0$ otherwise. Let $p_i=1+q_i,\ q_i\geq 0$ for $1\leq i\leq m$ and let $\vec{w}=\sum_{i=1}^{m}q_i\vec{b}_i$. Then, by a straightforward counting argument,

$$\sum_{1\leq i<j\leq n}(q_i+q_j)=(m-1)\sum_{i=1}^{n}q_i, \tag{30}$$

and, hence, by Equations (7) and (30) and some additional algebraic manipulation,

$$\begin{aligned} 1 = \| \vec{d} \|^2 &= \sum_{i=1}^{m} p_i^2 - \frac{2}{n} \sum_{1 \le i < j \le m} p_i p_j = \sum_{i=1}^{m} (1+q_i)^2 - \frac{2}{n} \sum_{1 \le i < j \le m} (1+q_i)(1+q_j) \\ &= m - \frac{m(m-1)}{n} + \| \vec{w} \|^2 + \big(2 - \frac{2(m-1)}{n}\big)\big(\sum_{i=1}^{m} q_i\big). \end{aligned}$$

Now, $m - (m(m-1)/n) \ge 1$ since $1 \le m \le n$. Furthermore, $m - (m(m-1)/n) = 1$ iff $m = 1$ or $m = n$. Also, the last two terms of the displayed equation just above are positive unless every q_i is 0, i.e., unless every $p_i = 1$. Thus, Claim 5 is true.

Claim 3 shows that the case $-1/2 \le k < 0$ reduces to $-1/n \le k < 0$.
Claim 4 shows that within this case, if $k > -1/n$, then this case reduces to CASE 0, and so, $card(Dir(\mathcal{D})) = 2n$.
Finally, Claim 5 shows that, if $k = -1/n$, then $\vec{d} \in Dir(\mathcal{D})$ iff either $\vec{d} = \pm\vec{b}_i$ or $\vec{d} = \pm\sum_{i=1}^{n} \vec{b}_i$.
Therefore, $card(Dir(\mathcal{D})) = 2n + 2 = 2(n+1)$.

□

Corollary 1 just below is immediate from the proof of Theorem 2 above.

Corollary 1 *Let $\mathcal{D}$ be regular with regular basis B.*
Then, $Dir(\mathcal{D}) = P(B) \cup Q(B)$, where,

$$\begin{aligned} P(B) &= \{\ \pm\vec{b} \mid \vec{b} \in B\ \} \quad \textit{and} \\ Q(B) &= \begin{cases} \{\ \pm(\sum_{\vec{b} \in B} \vec{b})\ \}, & \textit{if } K(\mathcal{D}, B) = -1/n; \\ \emptyset, & \textit{if } -1/n < K(\mathcal{D}, B) < 1/2; \\ \{\ (\vec{b} - \vec{b}') \mid \vec{b}, \vec{b}' \in B,\ \vec{b} \ne \vec{b}'\ \}, & \textit{if } K(\mathcal{D}, B) = 1/2. \end{cases} \end{aligned}$$

□

Here is an interesting property curiously not satisfied by every lattice.

Property 6 *Every* SET OF n LINEARLY INDEPENDENT, MINIMAL LENGTH VECTORS IN $\mathcal{D}$ IS A BASIS FOR $\mathcal{D}$.

The lattices $\mathbf{Z}^n$ and $\mathbf{A}_n$ satisfy Property 6, but, for each $n > 3$, any lattice $\simeq$ to D_n, defined as the set of n-tuples of integers that add up to an even number [CS93], does not [GL87]. Suppose B is a set of $n > 3$ linearly independent, minimal length vectors in D_n such that B does not span that lattice. Intuitively, if one tries to walk about in that lattice using *only* the vectors in B as basic steps, there are points, actually in the lattice, that one will not be able to reach. Of course the lattice that *is* spanned by B will be n-dimensional too.

Trivially, in a n-dimensional vector space, every set of n linearly independent vectors spans the space. n-dimensional lattices not having Property 6 are, then, quite unlike n-dimensional euclidean space.

The next corollary shows that, if $\mathcal{D}$ is a regular lattice, then $\mathcal{D}$ has Property 6.

Corollary 2 *If $\mathcal{D}$ is a regular lattice, then every subset, S, of $Dir(\mathcal{D})$, such that (i) $card(S) = n$ and (ii) S is a linearly independent set, is a basis for $\mathcal{D}$.*

PROOF. Suppose $\mathcal{D}$ is a regular lattice with regular basis B. Suppose S is a subset of $Dir(\mathcal{D})$ satisfying clauses (i) and (ii) in the statement of this corollary. Then, from the proof of Theorem 2, every vector in S can be obtained as a linear combination of vectors in B such that the coefficients in the linear combination are elements of the set $\{0, \pm 1\}$. Thus each vector in B can be expressed as a linear combination, with integer coefficients, of vectors in S, and, hence, S is a basis for $\mathcal{D}$. □

Corollary 3 *Suppose $\mathcal{D}$ is a regular lattice and $n \geq 2$ (n is the dimension of $\mathcal{D}$ by Convention 2). Then, for all regular bases B and C of $\mathcal{D}$,*

$$K(\mathcal{D}, C) = \begin{cases} \pm K(\mathcal{D}, B), & \text{if } n = 2; \\ K(\mathcal{D}, B), & \text{otherwise.} \end{cases}$$

PROOF. We first show that (*i*) $K(\mathcal{D}, C) = \pm K(\mathcal{D}, B)$ for every $n \geq 2$, and, then, that (*ii*) $K(\mathcal{D}, C) \neq -K(\mathcal{D}, B)$ if $n > 2$.

To prove (*i*), we consider the following three cases.

CASE 1: $-1/n \leq K(\mathcal{D}, B) < 1/2$. Then, from Corollary 1), $C \subseteq \{\pm \vec{b} \mid \vec{b} \in B\}$; hence, $K(\mathcal{D}, C) = \pm K(\mathcal{D}, B)$.

CASE 2: $K(\mathcal{D}, B) = 1/2$. If $K(\mathcal{D}, C) = 1/2$, we are done.

Suppose $K(\mathcal{D}, C) \neq 1/2$. Then, from Theorem 2),

$$card(Dir(\mathcal{D})) = n(n+1), \tag{31}$$

and yet $-1/n \leq K(\mathcal{D}, C) < 1/2$.

If $-1/n < K(\mathcal{D}, C) < 1/2$, then, from Theorem 2), $card(Dir(\mathcal{D}))$ also $= 2n$, contradicting Equation 31 and the hypothesis that $n \geq 2$.

If $K(\mathcal{D}, C) = -1/n$, then, from Theorem 2), $card(Dir(\mathcal{D})) = 2(n+1)$, which from Equation 31, must also $= n(n+1)$. But $2(n+1) = n(n+1)$ iff $n = 2$.

Hence, in this case, $K(\mathcal{D}, C) = -1/n = -1/2 = -K(\mathcal{D}, B)$.

CASE 3: $K(\mathcal{D}, B) = -1/n$. Then, by Corollary 1),

$$C \subset \{\pm\vec{b} \mid \vec{b} \in B\} \cup \{\pm(\sum_{\vec{b}\in B} \vec{b})\}. \tag{32}$$

If $C \subset \{\pm\vec{b} \mid \vec{b} \in B\}$, then (i) follows from the argument in CASE 1 above.

If $C \subset \{\pm(\sum_{\vec{b}\in B} \vec{b})\}$, then, since $n \geq 2$, C is not linearly independent, a contradiction.

If $\vec{c}_1, \vec{c}_2$ are elements of C such that $\vec{c}_1 = \pm \in \{\pm\vec{b} \mid \vec{b} \in B\}$ and $\vec{c}_2 \in \{\pm(\sum_{\vec{b}\in B} \vec{b})\}$, then, $(\vec{c}_1, \vec{c}_2) = (\pm\vec{b}_1, \pm(\sum_{\vec{b}\in B} \vec{b})) = \pm(1 - (n-1)/n) = \pm 1/n = \pm K(\mathcal{D}, B)$.

To prove (ii), we consider the same three cases as above in this proof but additionally supposing $n \geq 3$.

CASE 1: Then, as above, $C \subseteq \{\pm\vec{b} \mid \vec{b} \in B\}$. In *this* case we let the SIGN of $c \in C \stackrel{\text{def}}{=} +$ if $c \in B$; else, $\stackrel{\text{def}}{=} -$. Suppose by way of contradiction $K(\mathcal{D}, C) = -K(\mathcal{D}, B)$. Then, for any three distinct elements of C (and there *are* at least three elements in C), each distinct two of them must have opposite SIGN, a combinatorial impossibility. Hence, in this case, (ii) holds.

CASE 2: Then from our analysis of this case above, $K(\mathcal{D}, C) = -K(\mathcal{D}, B)$ is possible iff $n = 2$. Hence, since $n \geq 3$, in this case also (ii) holds.

CASE 3: Suppose by way of contradiction $K(\mathcal{D}, B) = -K(\mathcal{D}, C)$, which then, for this case, $= -1/n$. Then by Theorem 2, since $n \geq 3$, $card(Dir(\mathcal{D}))$ can be expressed simultaneously as $2(n+1)$ and also as $2n$, a contradiction. Hence, in this (third and last) case, (ii) holds.

□

As mentioned earlier, our algorithms, in certain instances, depend, in effect, on calculations of $K(\mathcal{D}, B)$. By the last corollary, the quantity $K(\mathcal{D}, B)$ does not depend on the choice of regular basis B, except possibly in dimension 2. Hence, the concept in the definition just below is well-defined; furthermore, the lack of dependence on any particular regular basis is good, computationally, for our purposes.

Definition 8 *Suppose $\mathcal{D}$ is a regular lattice (and n-dimensional by Convention 2). Suppose B is a regular basis for $\mathcal{D}$. The* regular inner product*, denoted $K(\mathcal{D})$, of $\mathcal{D}$ is defined by*

$$K(\mathcal{D}) = \begin{cases} |K(\mathcal{D}, B)|, & \text{if } n = 2; \\ K(\mathcal{D}, B), & \text{otherwise.} \end{cases}$$

It is not difficult to construct, for each positive integer, $n > 2$, and for each k in the interval $[-1/n, 1/2]$ the unique (up to $\simeq$) n-dimensional regular lattice with regular inner product equal to k. Recall the definition of $\mathbf{A}_n$ from Section 1. It follows from this definition that $\mathbf{A}_n$, a subset of

$\mathbf{R}^n$, is an n-dimensional regular lattice whose minimal length is 1 and whose regular inner product is $1/2$.

$\mathbf{A}_n$ is $\simeq$ the well known (see [GL87, CS93]) lattice, A_n (a subset of $\mathbf{R}^{n+1}$ with minimal distance $\sqrt{2}$), defined in Section 1. For each $n > 1$, the dual lattice A_n^* (also a subset of $\mathbf{R}^{n+1}$ with minimal distance $\sqrt{\frac{n}{n+1}}$; see [GL87, CS93]), of A_n, is regular and is $\simeq$ the unique (up to $\simeq$) n-dimensional regular lattice (with minimal distance 1) with regular inner product equal to $-1/n$.

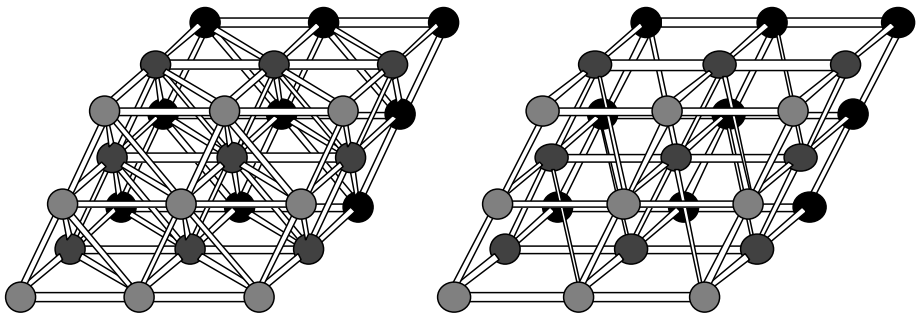


Figure 3: A piece of (a) $\mathcal{M}(\mathbf{A}_3)$ and (b) $\mathcal{M}(A_3^*)$

We conclude this section with the following important corollary (a direct consequence of Theorem 2) which asserts the existence and uniqueness of a mesh computer, whose underlying set satisfies Properties 1 through 5 (i.e., is a regular lattice), *and which has maximum local connectivity.*

Corollary 4 *The idealized mesh computer $\mathcal{M}(\mathbf{A}_n)$ alone has maximum local connectivity among all idealized meshes based on n-dimensional* regular *lattices.* □

2.2 Root Lattice Computers

In Section 2.1 we approached the problem of representing euclidean n-space by those lattices that retain the symmetry properties of euclidean n-space, and that lend themselves to efficiency in the kind of computations (e.g., inner product) we anticipate performing often. For our methodology, the point of representing euclidean n-space by a discrete set of points is to be able to build a (idealized) mesh and then to analogically simulate physical phenomena in this discretized universe. For an accurate simulation of a physical phenomenon, it is clear, then, that each processor in the mesh used for the simulation must be "responsible" for a region of euclidean n-space in such a way that each point in the domain of the physical phenomenon, a piece of euclidean n-space, is "covered" by some processor in the mesh. In this section, we consider lattices that are suitable, from this perspective, for use in our methodology.

For a processor $\mathcal{M}(P)$ in a lattice computer $\mathcal{M}(\mathcal{D})$, an obvious choice for the region of euclidean space that $\mathcal{M}(P)$ represents (is "responsible" for) is some suitable region including and surrounding the point P. Furthermore, for uniformity, we would like the shape and size of regions represented by different processors in $\mathcal{M}(\mathcal{D})$ to be identical. *Voronoi cells around lattice points*, introduced informally in Section 1 and formally defined below, are a natural choice for such regions around lattice points, and a fairly good choice for our purposes as witnessed by Propositions 3 and 4 below.

Once again, recall that, by Convention 2 above, $\mathcal{D}$ is an n-dimensional lattice which is a subset of $\mathbf{R}^n$ and which has minimal distance 1.

Definition 9 *For each point P in $\mathcal{D}$, the* Voronoi cell around P in $\mathcal{D}$, *denoted* Voronoi$(P, \mathcal{D})$, *is the set of points in $\mathbf{R}^n$ which are at least as close to P than to any other lattice point. We will denote the* interior of the Voronoi cell around P *in $\mathcal{D}$ by $V(P, \mathcal{D})$.*

The following two propositions note some useful facts about Voronoi cells around lattice points. (See [CS93] and [RS97] for more details on Voronoi cells around points in lattices.)

Proposition 3 *Voronoi cells around points in $\mathcal{D}$ tessellate $\mathbf{R}^n$, i.e.,*

1. *for any two distinct points, P and Q, in $\mathcal{D}$, $V(P, \mathcal{D}) \cap V(Q, \mathcal{D}) = \emptyset$, and*
2. $\bigcup_{P \in \mathcal{D}}$ Voronoi$(P, \mathcal{D}) = \mathbf{R}^n$.

□

Proposition 4 *For all P, Q in $\mathcal{D}$,* Voronoi$(P, \mathcal{D})$ *is congruent to* Voronoi$(Q, \mathcal{D})$. □

Fig. 4 shows a piece of $\mathbf{A}_2$ and the Voronoi cell (dashed lines) around the origin, $\vec{0}$, of $\mathbf{A}_2$.

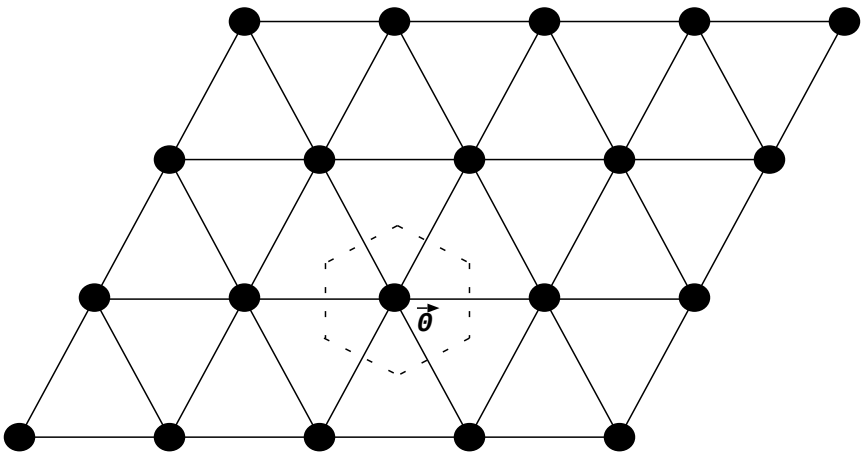


Figure 4: A piece of $\mathcal{M}(\mathbf{A}_2)$ with the Voronoi cell around $\mathcal{M}(\vec{0})$

A Voronoi cell is a *polytope* [Cox73], i.e., the n-dimensional analog of a polyhedron in 3-dimensions or a polygon in 2 dimensions. Thus, Voronoi cells have "faces" and "vertices".

It is useful to consider the following fact about Voronoi cells around lattice points. Since all Voronoi cells in a lattice are congruent, it is sufficient to discuss properties of the Voronoi cell around the origin, $\vec{0}$, of a lattice.

Proposition 5 *For each ($n-1$ dimensional) face F of* Voronoi$(\vec{0}, \mathcal{D})$*, there is a unique point Q in $\mathcal{D}$ such that F perpendicularly bisects the line segment from $\vec{0}$ to Q (denoted $F \pm Q$).* □

This motivates the following definition.

Definition 10 *The set of* relevant vectors in $\mathcal{D}$*, denoted* Rel$(\mathcal{D})$*, is given by*

$$\text{Rel}(\mathcal{D}) = \{Q \in \mathcal{D} \mid (\exists F \in \text{Voronoi}(\vec{0}, \mathcal{D}))[F \pm Q\}].$$

The following two propositions are easy to show (also see [RS97]).

Proposition 6 *There is a basis B for $\mathcal{D}$ that is a subset of* Rel$(\mathcal{D})$. □

Proposition 7 $Dir(\mathcal{D}) \subseteq \text{Rel}(\mathcal{D})$. □

Nonetheless, there are lattices in which some relevant vectors are not minimal length. Fig. 5 shows a piece of a 2-dimensional lattice, $\mathcal{L}$, around the origin of the lattice along with the Voronoi cell around the origin. As usual, line segments are drawn between points that are a minimal distance apart. Points P_1 P_2, P_3 and P_4 correspond to relevant vectors that are minimal length; points Q_1 and Q_2 correspond to relevant vectors that are not minimal length.

Consider the simulation, in $\mathcal{M}(\mathcal{L})$, of the motion of a particle along the curve $\boldsymbol{C}$ (see Fig 5). What we would like is to have processor $\mathcal{M}(\vec{0})$ represent the particle as long as it is traveling in Voronoi$(\vec{0}, \mathcal{L})$, and then for processor $\mathcal{M}(Q_2)$ to represent the particle while it is traveling in Voronoi$(Q_2, \mathcal{L})$. So, as the particle exits Voronoi$(\vec{0}, \mathcal{L})$ and enters Voronoi$(Q_2, \mathcal{L})$, $\mathcal{M}(\vec{0}, \mathcal{L})$ must send an appropriate message to $\mathcal{M}(Q_2)$. Given the interconnection structure in $\mathcal{M}(\mathcal{L})$, $\mathcal{M}(\vec{0})$ can send this message either via $\mathcal{M}(P_2)$ or $\mathcal{M}(P_3)$, thereby suggesting (due to the path taken by the message) that one of $\mathcal{M}(P_2)$ or $\mathcal{M}(P_3)$ represents the particle as it moves along its trajectory. But this is misleading and obviously introduces extra book-keeping on the part of processors during a simulation, and we would like to avoid that. Notice that this problem arose because the Voronoi cells around $\vec{0}$ and Q_2 share a face, i.e., Q_2 is a relevant vector, but Q_2 is not a minimal length vector and so $\mathcal{M}(\vec{0}, \mathcal{L})$ is not directly connected to $\mathcal{M}(Q_2)$. So, we are interested in lattices, $\mathcal{D}$, which have the following property as well.

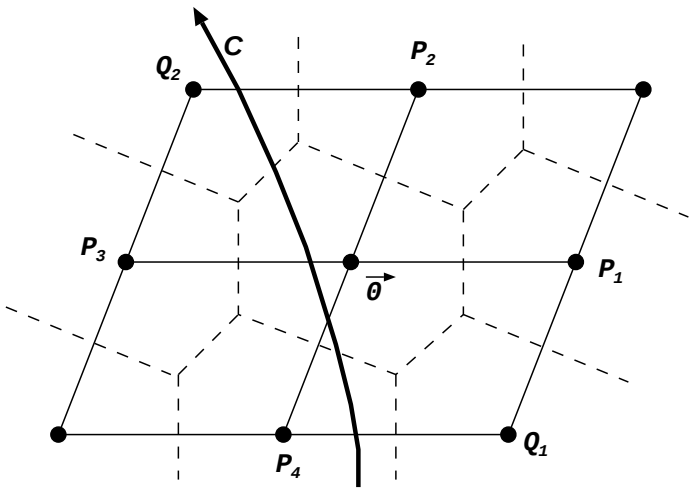


Figure 5: A piece of $\mathcal{M}(\mathcal{L})$ where $Rel(\mathcal{L}) \neq Dir(\mathcal{L})$ and trajectory $\boldsymbol{C}$ of a moving particle

Property 7 EACH RELEVANT VECTOR OF $\mathcal{D}$ IS ALSO A MINIMAL LENGTH VECTOR OF $\mathcal{D}$.

For the standard definition of root lattices [CS93], we have, from [RS97],

Theorem 3 *$\mathcal{D}$ satisfies Property 7 iff $\mathcal{D}$ is a root lattice.*

Property 7 is equivalently described geometrically in [CS93] as requiring $\mathcal{D}$'s Voronoi cells to have inscribed n-spheres. Hence, Theorem 3, provides the characterization of root lattice we give in Section 1 above.

Note that, in light of Propositions 1 and 6, all lattice computers with root lattices as underlying sets are connected.

The lattices $\mathbf{Z}^n$ and $\mathbf{A}_n$ (and lattices $\simeq$ them) are root lattices. So also is every lattice $\simeq$ the lattice D_n defined as the set of n-tuples of integers that add up to an even number (see [CS93]). For $n \leq 3$, $\mathbf{A}_n \simeq D_n$. So, Figure 3 above also shows a piece of a lattice computer with D_3 as the underlying set. For purposes of computational efficiency and precision, it would be ideal if processors in a lattice computer could perform all computations using only integers and integer vectors. In particular, it is desirable to base a lattice computer on a lattice $\mathcal{D}$ which has the following property.

Property 8 $\mathcal{D}$ IS INTEGRALLY REPRESENTABLE; that is, there is a lattice $\mathcal{D}' \simeq \mathcal{D}$ (possibly with $\mu(\mathcal{D}') > 1$), called *an integral representation of $\mathcal{D}$,* and an integer $n' \geq n$ such that $\mathcal{D}' \subseteq \mathbf{Z}^{n'}$.

If $\mathcal{D}$ is integrally representable and if $\mathcal{D}'$ is an integral representation of $\mathcal{D}$, then, inner products of pairs of vectors in $\mathcal{D}'$ are integers. Moreover, for a pair of vectors P and Q in $\mathcal{D}$, the inner product of a corresponding pair of vectors in $\mathcal{D}'$ is just the inner product of P and Q scaled up by a factor of $\mu(\mathcal{D}')^2$. It is very convenient that this factor, being just the inner product of a minimal length vector in $\mathcal{D}'$ with itself, is guaranteed to be an *integer*. Thus, since the lattice computer based on

$\mathcal{D}$ is unique up to $\simeq$ for lattices, it would be preferable for computational purposes if we considered this lattice computer to be based on $\mathcal{D}'$ instead. Not all lattices are integrally representable. The 2-dimensional lattice with basis $\{(0,1),(\pi,0)\}$ is obviously an example. On the other hand, the following theorem, a consequence of the summarizing theorem of [RS97], shows that a lattice having Property 7 also has Property 8.

Theorem 4 *Every root lattice* is *integrally representable.* □

Physical motion, of a particle, say, traces trajectories in euclidean space, the simplest such trajectories being straight line segments. Analogical simulation of such physical motion in $\mathcal{M}(\mathcal{D})$, though, traces trajectories (sequences of connections between processors) in $\mathcal{M}(\mathcal{D})$. In general, these two trajectories (one in euclidean space and one in $\mathcal{M}(\mathcal{D})$) differ; nonetheless, for the simulation to be faithful to the physical phenomenon, the two trajectories must be closely related. At the very least, it is reasonable to want a simple and uniform relationship between the straight line segments (shortest paths in euclidean n-space) between pairs of points in $\mathcal{D}$ and shortest paths in $\mathcal{M}(\mathcal{D})$ between processors centered at each of the points. The properties following the definition below specify such relationships.

Definition 11

1. *A* path *in $\mathcal{M}(\mathcal{D})$ is a nonempty sequence $\mathcal{P} = (\mathcal{M}(P_0), \ldots, \mathcal{M}(P_n))$ of processors, so that for each $0 \leq i < n$, $\mathcal{M}(P_i)$ is directly connected to $\mathcal{M}(P_{i+1})$, i.e., for each $0 \leq i < n$, there is a basic direction D_i so that $P_{i+1} = P_i + B_i$.*

2. *The set of* basic directions occurring in a path $\mathcal{P} = (\mathcal{M}(P_0), \ldots, \mathcal{M}(P_n))$ *in $\mathcal{M}(\mathcal{D})$ is the set*

$$Dir(\mathcal{P}) = \{B \in Dirs(\mathcal{D}) \mid (\exists i \mid 0 \leq i < n)[P_{i+1} = P_i + B]\}.$$

3. *The* path length *of a path is the integer equal to one less that the number of processors occurring in that path.*

4. *The* origin *of a path is the first processor occurring in that path, and the* endpoint *of a path is the last processor occurring in that path.*

5. *A path, $\mathcal{P}$, is a* shortest path *iff no path with lesser path length than that of $\mathcal{P}$ has the same origin and the same endpoint as $\mathcal{P}$ has.*

6. *The* graphical distance $d_{\mathcal{M}(\mathcal{D})}(P,Q)$ between processors $\mathcal{M}(P)$ and $\mathcal{M}(Q)$ *in $\mathcal{M}(\mathcal{D})$ is the length of a shortest path between $\mathcal{M}(P)$ and $\mathcal{M}(Q)$, if there is such a shortest path, and is ∞ otherwise.*

7. *For processors* $\mathcal{M}(P)$, $\mathcal{M}(Q)$, $\mathcal{M}(O)$ *in* $\mathcal{M}(\mathcal{D})$, *a processor* $\mathcal{M}(Q)$ *is an* inner neighbor of $\mathcal{M}(P)$ with respect to $\mathcal{M}(O)$, *and simultaneously,* $\mathcal{M}(P)$ *is an* outer neighbor of $\mathcal{M}(Q)$ with respect to $\mathcal{M}(O)$ *iff* $\mathcal{M}(Q)$ *is directly connected to* $\mathcal{M}(P)$ *in* $\mathcal{M}(\mathcal{D})$ *and* $d_{\mathcal{M}(\mathcal{D})}(P,O) = d_{\mathcal{M}(\mathcal{D})}(Q,O) + 1$.

8. d_E *denotes the euclidean distance function.*

The following property ensures that, if we measure distance between pairs of processors to be the euclidean distance between the lattice points upon which the processors are centered, the sequence of euclidean distances of successive processors in a shortest path in $\mathcal{M}(\mathcal{D})$ from the origin of that path is an increasing sequence.

Property 9 The graphical distance function $d_{\mathcal{M}(\mathcal{D})}$ is locally compatible with the euclidean distance function on $\mathcal{D}$, *i.e., for any pair of neighboring processors* $\mathcal{M}(P)$, *and* $\mathcal{M}(Q)$, *and any other processor* $\mathcal{M}(O)$, *if* $d_{\mathcal{M}(\mathcal{D})}(P,O) > d_{\mathcal{M}(\mathcal{D})}(Q,O)$ *then,* $d_E(P,O) > d_E(Q,O)$.

Note that, if the graphical distance function $d_{\mathcal{M}(\mathcal{D})}$ is locally compatible with the euclidean distance function on $\mathcal{D}$, then, for any pair of neighboring processors, $\mathcal{M}(P)$ and $\mathcal{M}(Q)$, and any other processor $\mathcal{M}(O)$, if $d_E(P,O) > d_E(Q,O)$, then $d_{\mathcal{M}(\mathcal{D})}(P,O) \geq d_{\mathcal{M}(\mathcal{D})}(Q,O)$. If $\mathcal{D}$ has dimension greater than 1, there are neighboring processors $\mathcal{M}(P)$, and $\mathcal{M}(Q)$, another processor $\mathcal{M}(O)$, so that $d_E(P,O) > d_E(Q,O)$, and $d_{\mathcal{M}(\mathcal{D})}(P,O) = d_{\mathcal{M}(\mathcal{D})}(Q,O)$.

The following desirable property says each basic direction occurring in a shortest path in $\mathcal{M}(\mathcal{D})$ is at most at 90° to any other. Not all $\mathcal{M}(\mathcal{D})$ have this property. For example, consider the piece of $\mathcal{M}(\mathcal{L})$ in Figure 5. Every shortest path in $\mathcal{M}(\mathcal{L})$ between $\mathcal{M}(\vec{0})$ and $\mathcal{M}(Q_2)$ contains two basic directions and the angle between the two directions is more than 90°. Physically, having this property could be interpreted as: if a particle is moving from one processor in $\mathcal{M}(\mathcal{D})$ to another, following the shortest path in $\mathcal{M}(\mathcal{D})$ between the two processors, then every move it makes (from one processor to a neighboring processor) is positive, i.e., does not have a negative projection on any previous move.

Property 10 The inner product of each pair of basic directions occurring in any shortest path in $\mathcal{M}(\mathcal{D})$ is non-negative.

Again, as a consequence of the summarizing result in [RS97], we have that any lattice computer $\mathcal{M}(\mathcal{D})$ which has Property 7 must have Properties 9 and 10 as well. In fact the results in [RS97] imply the following stronger

Theorem 5 *A lattice computer* $\mathcal{M}(\mathcal{D})$ *is based on a root lattice* $\mathcal{D}$ *iff* $\mathcal{D}$ *is a minimally generated lattice and* $\mathcal{M}(\mathcal{D})$ *has either of Properties 9 or 10.* □

Finally, another consequence of the summarizing result in [RS97] is that lattice computers based on root lattices are the only lattice computers for which Theorem 6, below, holds. We use the conclusion of Theorem 6 below in [She91, CRS94b] to design a physically natural algorithm to simulate the propagation of a constant speed spherical wavefront, useful in simulating an idealized model for radiation from a point source. Theorem 6 is a simple consequence of the following observation about any lattice computer and the fact that root lattices are minimally generated and satisfy Property10.

Proposition 8 *For processors $\mathcal{M}(P)$, $\mathcal{M}(O)$ in $\mathcal{M}(\mathcal{D})$ a basic direction $B \in Dirs(\mathcal{D})$ occurs in some path $\mathcal{P}$ from $\mathcal{M}(O)$ to $\mathcal{M}(P)$ iff there is an inner neighbor $\mathcal{M}(Q)$ of $\mathcal{M}(P)$ with respect to $\mathcal{M}(O)$, so that $P = Q + B$.* □

Theorem 6 *In a root lattice computer $\mathcal{M}(\mathcal{D})$, for processors $\mathcal{M}(P), \mathcal{M}(O)$ in $\mathcal{M}(\mathcal{D})$ and a basic direction $B \in Dirs(\mathcal{D})$, $\mathcal{M}(P+B)$ is an outer neighbor of $\mathcal{M}(P)$ iff for every B' so that $\mathcal{M}(P-B')$ is an inner neighbor of $\mathcal{M}(P)$, the inner product $(B, B') \geq 0$.*

There are properties of regular lattices the root lattices need not have. For example, some root lattices that are not regular do not have Property 6. D_4 is one such lattice, and, hence, as an approximation of euclidean 4-space is somewhat pathological. However, D_n is a root lattice, and, for dimensions $n > 3$, denser than $\mathbf{A}_n$ [CS93] (see Section 3.2.2 for a discussion of density). Nonetheless, we have not chosen to emphasize, in the present paper, meshes based on D_n (for $n > 3$). See Section 3.4 below.

In the next section, we will see that the class of n-dimensional lattices that are both regular and root is small.

2.3 $\mathbf{A}_n$: The Best of Both Worlds

Having considered two different perspectives on representing euclidean n-space by lattices, we have identified two classes of lattices: the regular lattices, and the root lattices. From the examples given in Sections 2.1 and 2.2, it is also clear that $\mathbf{Z}^n$ and $\mathbf{A}_n$ (and all lattices related to them by $\simeq$) are both regular as well as root lattices. The following theorem, once again a direct consequence of the summarizing theorem of [RS97], characterizes the intersection of the classes of regular and root lattices.

Theorem 7 *For each n, (up to $\simeq$) $\mathbf{Z}^n$ and $\mathbf{A}_n$ are the* only *n-dimensional lattices that are both root and regular.* □

Hence, Theorem 7 and Corollary 4 imply that $\mathcal{M}(\mathbf{A}_n)$ provides maximum local connectivity among all idealized meshes based on n-dimensional lattices that are both regular and root.

3 Additional Merits of $\mathbf{A}_n$

This section specifies other desirable properties of $\mathbf{A}_n$ including other ways it is superior to $\mathbf{Z}^n$ for our purposes.

3.1 Properties of $\mathbf{A}_n$ Related to Computational Cleanness

3.1.1 Ease of Computations

Fact 1 *As noted earlier, A_n is an integral representation of $\mathbf{A}_n$.*

Fact 2 *Each basic direction vector in A_n is the difference, $(\vec{e}_i - \vec{e}_j)$, of two distinct standard basis vectors of $\mathbf{Z}^{n+1}$.*

Fact 3 *The inner product of any pair of vectors in A_n is an integer. Thus, although the euclidean distances between points in A_n need not be integral, the squares of these distances are. It turns out this is good enough for our algorithms referenced in Section 1 above.*

Fact 4 *Since A_n is a subset of $\mathbf{Z}^{n+1}$, the inner product of any two vectors in A_n is simply the sum of the component-wise product of the corresponding two $(n+1)$-tuples of integers. So, a hardware implementation of the computation of the inner product of two vectors in A_n can be easily parallelized.*

Fact 5 *The test for membership in A_n, of a point in $\mathbf{Z}^{n+1}$, useful in algorithms, is computationally very easy.*

Fact 6 *The path length, in $\mathcal{M}(A_n)$, between processors $\mathcal{M}(P)$ and $\mathcal{M}(Q)$ in $\mathcal{M}(A_n)$ is the sum of all the positive co-ordinates in the vector $(P-Q)$. This, in turn, facilitates estimating the minimum time required for the propagation of a message from a processor to another in $\mathcal{M}(\mathbf{A}_n)$.*

Fact 7 *It is easy to compute the expression*

$$\sum_{r=0}^{R}\sum_{i=0}^{n+1}(-1)^i\binom{n+1}{i}\binom{n+r-i}{n}^2 \tag{33}$$

for the number of processors at most R connections away from a given processor in $\mathcal{M}(\mathbf{A}_n)$.

3.1.2 Predecessor Containment

Proposition 9 *Every $k \leq n$ dimensional sublattice of $\mathbf{A}_n$, spanned by k vectors from a regular basis B for $\mathbf{A}_n$, is isometric with $\mathbf{A}_k$.*

PROOF. Suppose $\mathcal{L}$ is a $k \leq n$ dimensional sublattice of $\mathbf{A}_n$ spanned by L, a k-subset of a regular basis, B, for $\mathbf{A}_n$. Then, clearly, $\mathcal{L}$ is a regular lattice with regular basis L, minimal length 1, and $K(\mathcal{L}) = \frac{1}{2}$, and hence isometric with $\mathbf{A}_k$. □

A similar property is true for $\mathbf{Z}^n$. $\mathcal{M}(\mathbf{A}_n)$ contains several copies of every predecessor, $\mathcal{M}(\mathbf{A}_k), k < n$, of $\mathcal{M}(\mathbf{A}_n)$ in the family of idealized meshes $\mathcal{M}(\mathbf{A}_n)$. This allows our algorithms to be nicely parameterized by dimension. It also allows our meshes to be easily partitioned by dimension (see also Section 3.2.5 below).

3.2 Comparison of $\mathbf{A}_n$ with $\mathbf{Z}^n$

Currently, most meshes in use [Uhr72, DL81, PU82, AS84, PD84, Lev85, PF+85, TM87, Dye] are essentially two-dimensional square meshes[8], i.e., essentially finite submeshes of $\mathcal{M}(\mathbf{Z}^2)$. However, for our purposes, $\mathcal{M}(\mathbf{A}_n)$ is preferable to $\mathcal{M}(\mathbf{Z}^n)$ for several reasons enumerated in this section, reasons essentially in addition to $\mathcal{M}(\mathbf{A}_n)$ having greater local bushiness than $\mathcal{M}(\mathbf{Z}^n)$.

3.2.1 Rigidly Embedding a Lattice Computer into Another

Definition 12 *Suppose $\mathcal{M}(\mathcal{D}_1)$ and $\mathcal{M}(\mathcal{D}_2)$ are two lattice computers. We say that $\mathcal{M}(\mathcal{D}_1)$ is* rigidly embeddable *in $\mathcal{M}(\mathcal{D}_2)$ iff (i) $\mu(\mathcal{D}_1) = \mu(\mathcal{D}_2)$ and (ii) $\mathcal{D}_1$ is isometric to some sublattice, $\mathcal{D}'_1$, of $\mathcal{D}_2$. Note that clause (ii) is equivalent to $Dir(\mathcal{D}'_1) \subseteq Dir(\mathcal{D}_2)$.*

Processors and connections in one lattice computer must match up *exactly* in the other. Intuitively, if $\mathcal{M}(\mathcal{D}_1)$ is rigidly embeddable in $\mathcal{M}(\mathcal{D}_2)$, it is possible to remove some processors and some connections from $\mathcal{M}(\mathcal{D}_2)$ so that the remaining structure is graphically isometric to $\mathcal{M}(\mathcal{D}_1)$.

The importance of rigid embeddability is that, if one lattice computer, $\mathcal{M}(\mathcal{D}_1)$, is rigidly embeddable in another, $\mathcal{M}(\mathcal{D}_2)$, then algorithms written for $\mathcal{M}(\mathcal{D}_1)$ could be run, with minimal change, on $\mathcal{M}(\mathcal{D}_2)$. The following proposition then implies $\mathcal{M}(\mathbf{A}_{2n-1})$ can handle all of $\mathcal{M}(\mathbf{Z}^n)$'s algorithms, but not vice-versa.

Proposition 10 *For all n, $\mathbf{Z}^n$ is rigidly embeddable in $\mathbf{A}_{2n-1}$, but for all $n \geq 2$, $\mathbf{A}_n$ is not rigidly embeddable in $\mathbf{Z}^m$ for any m.*

Proof. As stated in Fact 2 above, the set of vectors $\{(e_i - e_j) \mid 0 \leq i \neq j \leq (2n-1)\}$ is the set $Dir(A_{2n-1})$. Then, for some lattice $\mathbf{A'}_{2n-1}$, isometric to $\mathbf{A}_{2n-1}$,

$$\{(\frac{e_i}{\sqrt{2}} - \frac{e_j}{\sqrt{2}}) \mid 0 \leq i \neq j \leq (2n-1)\} = Dir(\mathbf{A'}_{2n-1}).$$

Consider the set

$$S = \{(\frac{e_{2i}}{\sqrt{2}} - \frac{e_{2i+1}}{\sqrt{2}}) \mid 0 \leq i \leq (n-1)\}.$$

Then, for any two distinct vectors, s_1 and s_2, in S, $(s_1, s_2) = 0$. Also the length of any vector in S is 1. Clearly, then, S is a basis for some lattice $\mathbf{Z'}^n$, isometric to $\mathbf{Z}^n$, and $\mathbf{Z'}^n \subseteq \mathbf{A'}_{2n-1}$. Hence, $\mathcal{M}(\mathbf{Z}^n)$ is rigidly embeddable in $\mathcal{M}(\mathbf{A}_{2n-1})$.

[8] N.B. Some of these meshes additionally make toroidal connections.

Clearly, for all $n \geq 2$ and for all m, there is no lattice $\mathbf{A'}_n$ isometric to $\mathbf{A}_n$ such that $Dir(\mathbf{A}'_n) \subseteq Dir(\mathbf{Z}^m)$. □

3.2.2 Density of Points and Virtual Directions

One measure of how well, quantitatively, a lattice approximates euclidean space (of corresponding dimension) is its *resolution*:

Definition 13

1. *For all reals $r > 0$, $N(\mathcal{D}, r)$ is defined by*

$$N(\mathcal{D}, r) = card(\{P \in \mathcal{D} \mid d(P, \vec{0}) \leq r\}).$$

2. *The* resolution*, denoted $Res(\mathcal{D})$, of $\mathcal{D}$ is defined by*

$$Res(\mathcal{D}) = \lim_{r \to \infty} \frac{N(\mathcal{D}, r)}{V_n(r)},$$

where $V_n(r)$ is the volume of an n-dimensional sphere of radius r.

$Res(\mathcal{D})$ is just the number of points of $\mathcal{D}$ per unit volume over all of euclidean n-space. Note, though, that $Res(\mathcal{D})$ *does* depend on the minimal distance of $\mathcal{D}$; in particular, two lattices related by $\simeq$ that differ in their minimal distance do *not* have the same resolution. The following proposition is easy to prove.

Proposition 11 *For an n-dimensional lattice, $\mathcal{D}_1$, $\simeq \mathcal{D}$,*

$$Res(\mathcal{D}_1) = \frac{1}{\mu(\mathcal{D}_1)^n} \; Res(\mathcal{D}).$$

□

It does make sense to compare the resolutions of lattices with equal minimal distances, and we would clearly prefer the one with a better (i.e., larger in value) resolution.

Consider an alternative characteristic quantity, the density of $\mathcal{D}$, $\Delta(\mathcal{D})$, defined (informally) below (see [CS93, Chapter 1] for a formal discussion of this and other related concepts). Note that one could place n-dimensional spheres of radius $\mu(\mathcal{D})/2$ $(= 1/2$ by Convention 2) centered at points in $\mathcal{D}$ without having any of the spheres overlap, except at their boundaries. Then, $\Delta(\mathcal{D})$ is defined ([CS93, page 6]) to be the *proportion of euclidean n-space occupied by the spheres.*[9] Clearly,

[9] These spheres are a lattice sphere packing. For each $n \in \{2, 3\}$, $\mathbf{A}_n$ provides in this way a *densest* lattice packing of n-spheres. N.B. D_4, not $\mathbf{A}_4$, provides a densest lattice packing of 4-spheres [CS93].

then, $\Delta(\mathcal{D})$ is just the number of such spheres (centered at points in $\mathcal{D}$) per unit volume of euclidean n-space.

Then we have

Proposition 12 *Suppose $\mathcal{L}_1$ and $\mathcal{L}_2$ are arbitrary n-dimensional lattices not subject to the restriction of Convention 2. If $\mathcal{L}_1$ and $\mathcal{L}_2$ have the same minimal distance, then the ratio of their resolutions equals the ratio of their densities.* □

PROOF. For any n-dimensional lattice $\mathcal{L}$ with minimal distance m,

$$Res(\mathcal{L}) = \Delta(\mathcal{L})/(\text{volume of one } n\text{-dimensional sphere of radius } m/2). \tag{34}$$

Therefore, if $\mathcal{L}_1$ and $\mathcal{L}_2$ have the same minimal distance, m, then

$$Res(\mathcal{L}_1)/Res(\mathcal{L}_2) = \Delta(\mathcal{L}_1)/\Delta(\mathcal{L}_2). \tag{35}$$

□

[CS93] gives expressions for the density of $\mathbf{Z}^n$ (page 106) and A_n (page 109) The ratio of the densities of points in the two lattices, $\mathbf{Z}^n$ and $\mathbf{A}_n$, is essentially the ratio of the resolutions offered by the two lattices (as seen from Proposition 12 above). Using the expressions for density from [CS93] and Proposition 12, we have, where V_n is the volume of an n-dimensional sphere with radius 1,

Proposition 13

$$\frac{\Delta(\mathbf{A}_n)}{\Delta(\mathbf{Z}^n)} = \frac{\Delta(A_n)}{\Delta(\mathbf{Z}^n)} = \frac{V_n 2^{-\frac{n}{2}}(n+1)^{-\frac{1}{2}}}{V_n 2^{-n}} = \frac{2^{\frac{n}{2}}}{(n+1)^{\frac{1}{2}}}.$$

□

Thus, $\mathcal{M}(\mathbf{A}_n)$ offers *exponentially* better resolution than $\mathcal{M}(\mathbf{Z}^n)$.

Our algorithm for simulating a particle moving at constant speed in a straight line [She91, CRS91, CCRS95] enables us to send particles in *any lattice-representable (virtual) direction* at (approximately) constant speed. In a large spherical finite submesh, from the center, we can, in effect, send a particle in the direction of any other lattice point in that finite submesh. We expect that this software trick, to get around the fact that a much smaller number of local directions are explicitly hard-wired in, will be important for fluid flow simulations in (root) lattice computers (see also Section 3.2.6 below). We don't have, at this point, for $\mathcal{M}(\mathbf{A}_n)$ and $\mathcal{M}(\mathbf{Z}^n)$, formulas, analogous to the density and resolution formulas, for the number of virtual directions emanating from the origin and available within a large sphere centered at the origin. Note that co-linear vectors

of different lengths inside a large sphere contribute to the number of points in that sphere but not to the number of virtual directions. Nonetheless, Proposition 13 just above effectively gives us a *very rough* comparison of the relative number of virtual directions available in $\mathcal{M}(\mathbf{A}_n)$ versus $\mathcal{M}(\mathbf{Z}^n)$.

3.2.3 Well Covering of $\mathbf{R}^n$

For each P in $\mathcal{D}$, informally we define the *sphere of influence* of P as the set of all points in euclidean n-space of distance $< \mu(\mathcal{D})$ (the minimal length in $\mathcal{D}$) from P. It is desirable that each point in euclidean n-space reside in the sphere of influence of some point in $\mathcal{D}$, because, then, in a sense, euclidean n-space is *well-covered* by the points in $\mathcal{D}$. Recall our definition of the measure of proximity, $p(\mathcal{D})$, of an approximation $\mathcal{D}$ (see Definition 2; also note that lattices are approximations). In a lattice there is a minimum such measure of proximity. From now on we will take $p(\mathcal{D})$ to be the minimum measure of proximity in $\mathcal{D}$.

Clearly, then, $\mathcal{D}$ well-covers euclidean n-space iff $\mu(\mathcal{D}) > p(\mathcal{D})$. Our notion of measure of proximity for approximations is identical to the notion of *covering radius* defined in [CS93, page 6] for lattices. [CS93] give expressions for the covering radii of $\mathbf{Z}^n$ (page 106) and A_n (page 109). Using these expressions, after appropriately scaling the one for A_n, the following proposition is easy to prove and shows that $\mathbf{A}_n$ well-covers euclidean n-space for more dimensions than does $\mathbf{Z}^n$.

Proposition 14

1. $\mu(\mathbf{A}_n) > p(\mathbf{A}_n)$ *iff* $n \leq 6$, *and*
2. $\mu(\mathbf{Z}^n) > p(\mathbf{Z}^n)$ *iff* $n \leq 3$.

□

Further, for each n, $p(\mathbf{A}_n) < p(\mathbf{Z}^n)$. Therefore, the family $\{ \mathcal{M}(\mathbf{A}_n) \}$ is superior to the family, $\{ \mathcal{M}(\mathbf{Z}^n) \}$ of square meshes, also in regard to well-covering.

3.2.4 Clustering of Processors

The number of processors at most R connections away from a given processor in $\mathcal{M}(\mathbf{Z}^n)$ is given by

$$\sum_{r=0}^{R} \sum_{i=0}^{n} \binom{n}{i} \binom{n+r-i-1}{n-1}. \tag{36}$$

Comparing this with the corresponding expression for $\mathcal{M}(\mathbf{A}_n)$ from Equation (33) in Section 3.1, we find that the number of processors at most R connections away is less in $\mathcal{M}(\mathbf{Z}^n)$ than in $\mathcal{M}(\mathbf{A}_n)$, for each $n > 1$. In particular, plugging in 1 for R in each of the expressions, the number of near neighbor connections for $\mathcal{M}(\mathbf{A}_n)$ is $n(n+1)$, a feasible quadratic in n, *but*, for $\mathcal{M}(\mathbf{Z}^n)$, the number is only $2n$, a small linear expression in n. In effect, in a given time span, a message broadcast from a

processor in $\mathcal{M}(\mathbf{A}_n)$ reaches more processors in the lattice computer than does a message broadcast from a processor in $\mathcal{M}(\mathbf{Z}^n)$.

3.2.5 Interconnection Bandwidth Across Partitions

The predecessor containment property of $\mathbf{A}_n$ (see Section 3.1.2 above) ensures that $\mathbf{A}_n$ can be partitioned cleanly and naturally.

If $B = \{ \vec{b}_1, \ldots, \vec{b}_n \}$ is a regular basis for $\mathcal{D}$, $P_{\mathcal{D}}(t_1, \ldots, t_n) = \{\sum_{i=1}^{n} p_i \vec{b}_i \mid (\exists p_i \in \mathbf{Z})[0 \leq p_i < t_i]\}$ is a *standard n-dimensional parallelepiped*, in $\mathcal{D}$, with thickness t_i in direction $\vec{b}_i$. The sub-mesh, $\mathcal{M}(P_{\mathcal{D}}(t_1, \ldots, t_n))$, of $\mathcal{M}(\mathcal{D})$ is called a *parallelepiped mesh*, or *p-mesh, (with thickness t_i in direction $\vec{b}_i$)*. The predecessor containment property of $\mathcal{D}$ ensures that each face of a standard n-dimensional parallelepiped is the translate of a *standard* $(n-1)$ dimensional parallelepiped. Fig. 3(a) above in Section 2.1 pictures a p-mesh with thickness 3 in each direction.

Let t be a positive integer, and let $P_{\mathcal{D}} = P_{\mathcal{D}}(t, \ldots, t) \subset \mathcal{D}$. An $(n-1)$-dimensional plane, C, parallel to one of the $(n-1)$-dimensional faces of $P_{\mathcal{D}}$, which plane does not pass through any processors of $\mathcal{M}(P_{\mathcal{D}})$, partitions $\mathcal{M}(P_{\mathcal{D}})$ into two components, each of which is (a translate of) a parallelepiped mesh.

The number, $B_{\mathcal{D}}(t)$, of processor *interconnections*, of $\mathcal{M}(P_{\mathcal{D}})$, that pass through C is the *bandwidth of the partition (by C of $\mathcal{M}(P_{\mathcal{D}})$)*. We can show that $B_{\mathbf{A}_n}(t) = t^{n-2}(nt - n + 1)$ and $B_{\mathbf{Z}^n}(t) = t^{n-1}$, so that $\lim_{t\to\infty} B_{\mathbf{A}_n}(t)/B_{\mathbf{Z}^n}(t) = n$. Hence, in general, our reasonably thick parallelepiped meshes of dimension n will have about n times as many "parallel communication channels" from one partition to the other compared to an analogous n-dimensional square mesh. For our purposes, this is a good thing.

3.2.6 Automorphism Groups

The order of the automorphism group of $\mathbf{A}_n$ is $2 \cdot (n+1)!$ while that of $\mathbf{Z}^n$ is $2^n \cdot n!$ [CS93]. In spite of the large number of resultant symmetries of $\mathbf{A}_n$, the Lattice Gas Automata approach to fluid flow [FHP86, Has87] has problems with a combinatorial clash between the actual symmetries of $\mathbf{A}_3$ and Navier-Stokes [FHP86, Has87]. Their trick to handle 3-dimensional fluid flow is to work with projections from flow in $\mathbf{A}_4$. However, the LGA approach treats lattices *as* space rather than as approximations to space. They are restricted to local directions explicitly hard-wired in, for example, for fluid flow simulations; whereas, as we noted in Section 3.2.2 above, we have available many more (virtual) directions along which we can send the particles of a fluid (and more directions for $\mathbf{A}_n$ than for $\mathbf{Z}^n$).[10] We expect (work in progress) that, with our approach, we will be less bothered by

[10] The LGA are a computational model of physics based on cellular automata [Fey82, Min82, FT82, Tof84, TM87, Sny47, Ung58, Mar84, SW86, Mac86, Tof77b, Tof77a, Vic84, Wol83, Zus69, Hil55, Svo86].

For this CA approach in general, emphasis is placed on the *simplicity* of the local transition rules for each cellular automaton [FHP86]. The processors in our methodology are not restricted to simple cellular automaton-like rules.

the symmetries of $\mathbf{A}_3$. It remains to be seen.

3.2.7 Tessellating Euclidean Space

For each $n > 2$, one common reason that $\mathcal{M}(\mathbf{Z}^n)$, is interesting is that the n-dimensional cube is the only single regular polytope that tessellates $\mathbf{R}^n$ [CS93]. However, we have no need to tessellate space with copies of a single *regular* body. Of course, the set of standard n-dimensional parallelepipeds (see Section 3.2.5), each with thickness 2 in each direction (in a regular basis) in $\mathbf{A}_n$, does, in fact, tessellate euclidean n-space. Of course, these parallelepipeds are not regular bodies.

Conceptually more interesting for approximating euclidean n-space well is, for example, having a low measure of proximity (see Definition 2). As seen in Section 3.2.3, by this criterion, $\mathcal{M}(\mathbf{A}_n)$ is superior to $\mathcal{M}(\mathbf{Z}^n)$.

3.3 Embedding $\mathcal{M}(\mathbf{A}_n)$ In Physical 3-Space

Here, we indicate how to physically embed each of our meshes in 3-dimensional euclidean space in such a way that the physical length of any communication link between adjacent processors is the same throughout that mesh. At the end of this Section (Section 3.3) we discuss the *solved* feasibility issues regarding the size of these physical lengths.

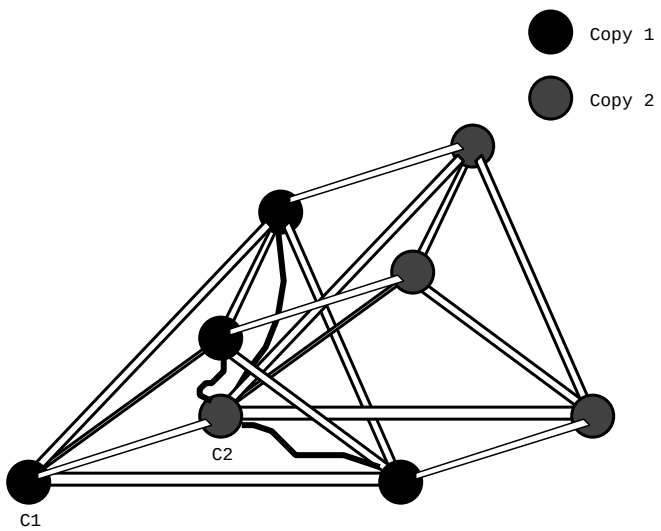


Figure 6: A piece of $\mathcal{M}(\mathbf{A}_4)$ embedded in euclidean 3-space

It is useful to convey the concept with an example. We indicate, with the aid of Fig. 6, how to so embed our 4-dimensional p-meshes. Consider the 3-dimensional p-mesh of Figure 3(a), three

The individual processors in our meshes are, at least in principle, general purpose computers computationally more powerful than finite state automata. Of course, any physical instance of one of our individual processors must have finite local memory and, hence, could be modeled as a finite state automaton — but with transition rules more complex than those of LGA.

processors thick in each dimension. A 4-dimensional p-mesh, three processors thick in three dimensions and two processors thick in the fourth dimension, can be constructed out of two copies of this 3-dimensional p-mesh, by making appropriate interconnections between processors in the two 3-dimensional p-mesh copies (much like constructing a 4-dimensional hypercube from a 3-dimensional cube). We will illustrate our embedding scheme for the two lower left corner tetrahedral corners (see Figure 3(a)) of the two 3-dimensional p-mesh copies. It will be clear this process can be uniformly iterated to handle larger thicknesses and larger dimensions. Here is the process, partially depicted in Fig. 6. Imagine the two copies (of the tetrahedral piece of a 3-dimensional p-mesh) as initially completely coinciding. Then imagine sliding (without rotation) the outer corner processor (C2 in Fig. 6) of one copy *inside* the other copy along a line equidistant from all three tetrahedral connections emanating from C1 in copy 1. Imagine C2 being slid in this fashion unit distance. From the so imagined resultant positioning of copy 1 relative to copy 2, connect each processor in copy 1 with its analog in copy 2 that was imagined as moving one unit away from initial coincidence. Now make the other connections along the curved lines shown in Fig. 6 for the tetrahedral corners of copies 1 and 2 and all corresponding connections in all other similarly corresponding pairs of tetrahedra, one from each of the two copies. To make all of the connections of unit length, these *latter* connections depicted with curved lines will have to be physically curved in 3-space. This is because the *straight-line* distance *in* 3-*space* between the processors so linked will be less than one. The resultant mesh *is* a 4-dimensional corner of one of our 4-dimensional p-meshes.

Now then, it should be clear that there is considerable flexibility in the embedding process described above. For example, in Fig. 6, C2 could be moved physically closer to C1, allowing the unit length connection to curve so as to maintain it at unit length. It is clear, in any case, that the processor labeled C2 needs to occupy some space and that the processors in copy 1 opposite C1, for example would have to be far enough apart to make room for C2. Of course, for our embedding scheme(s), for each dimension, adding more processors needn't increase the distance between immediately connected processors, but, for the schemes for embedding we have just been describing it would appear that the distance between immediately connected processors would have to increase with dimension. Paul Vitanyi's Corollary 4 in [Vit88] applies to the n-dimensional tetrahedral corners of our meshes and yields a *lower bound*, of $\Omega(n^{1/3})$, on the growth, with dimension n, of the distance between immediately connected processors. If we pack processors such as C2 by a known densest 3-dimensional sphere-packing scheme *and neglect interconnect "wire" size*, we can reach this lower bound! A *practical* variant of this scheme, in which we merely pack spherical processors essentially in a 2-dimensional plane (e.g., a plane parallel to the plane described by the three processors other than C1 in copy 1 in Figure 6 and passing through the copy 1 tetrahedron) inside each smallest parallelepiped of our 3-dimensional p-meshes, both allows room for the interconnections and yields a *sub-polynomial* growth of $\mathcal{O}(n^{1/2})$, with dimension n, of the distance between immediately connected processors! We expect this could be improved further although there may not be any

practical reason to do so.

For actually building any finite sub-mesh of $\mathcal{M}(\mathbf{A}_n)$, of possibly high dimension, we are constrained by physical reality—a physical instance of any one of these finite sub-meshes, ostensibly, must be physically embeddable in euclidean 3-space. Furthermore, for our approach, the embedding in euclidean 3-space must be such that the physical length of the connection between adjacent processors is a feasible constant. Constant length interconnections are clearly important to help achieve synchronous motion of the "particles" making up a simulated object. For a given size of processor (of course > 0), we can embed $\mathcal{M}(\mathbf{A}_n)$ in physical 3-space with constant length interconnections. In our embedding, the minimal (constant) length of the interconnection between immediately connected processors does depend, but only *sub*polynomially, on the dimension of the mesh; furthermore, for each fixed dimension, it does *not* depend on the number of processors in the mesh (see the paragraph just above).

3.4 Why Not Meshes Based on Denser Lattices?

Consider the classical problems of packing unit sized n-spheres into euclidean n-space (i) as densely as possible, and (ii) so that each sphere touches the maximum number of neighbors [GL87, CS93]. Consider a solution to one (or both) of these problems. One could define a corresponding mesh (for approximating euclidean n-space) with processors at the centers of each sphere and its near neighbor connections to all and only those processors at the centers of the spheres touching it. In general, the set $\mathcal{S}$ of points corresponding to the centers of the spheres need not even be a lattice [GL87, CS93]. Hence, such $\mathcal{S}$ would likely be too hard to deal with computationally. A more severe consideration is that, for most cases of interest, the solutions to problems (i) and (ii) above are just unknown.

While more is known about problems (i) and (ii) *modified* to further require the solutions to be lattices, only a few cases of interest are solved [CS93], and the solutions, for dimensions ≥ 4, all fail to be regular. N.B. Known solutions to the densest lattice sphere packing problem are root lattices, and, hence, our algorithms for basic motions apply to them (see Section 2.2 above). But their lack of the symmetry offered by regular lattices made them less attractive than $\mathbf{A}_n$ and $\mathbf{Z}^n$ to us. However, this paper does not support any final decisions about use of such lattices for analogical simulations of motion. We do note that the (few) known solutions apparently come from totally different *families* of lattices [CS93], and, hence, algorithms for use in such lattices would likely not be *simply* parameterizable by dimension.

References

[AS84] N. Ahuja and S. Swamy. Multiprocessor pyramid architecture for bottom-up image analysis. *IEEE Trans. on PAMI*, PAMI-6:463–474, 1984.

[Aur90] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. Technical Report B-90-9, Institute for Computer Science, Dept. of Mathematics, Freie Universität Berlin, November 1990.

[BL85] R. Brachman and H. J. Levesque, editors. *Readings in Knowledge Representation*. Morgan Kaufmann Publishers, Inc., 1985.

[CB91] T. Chandrupatla and A. Belegundu. *Introduction to Finite Elements in Engineering*. Prentice-Hall, NJ, 1991.

[CCRS95] J. Case, S. Chitoor, D. Rajan, and A. Shende. Multi-particle motion in lattice computers. Writeup in preparation, 1995.

[Cox73] H. S. M. Coxeter. *Regular Polytopes*. Dover Publications, 1973.

[CRS91] J. Case, D. Rajan, and A. Shende. Simulating uniform motion in lattice computers I: Constant speed particle translation. Technical Report 91-17, University of Delaware, June 1991.

[CRS94a] J. Case, D. Rajan, and A. Shende. Representing the spatial/kinematic domain and lattice computers. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:17–40, 1994.

[CRS94b] J. Case, D. Rajan, and A. Shende. Spherical wave front generation in lattice computers, May 1994. To appear in the *Journal of Computing and Information*, 1(1), Special Issue: Proceedings of the 6th International Conference on Computing and Information, Peterborough, Ontario, Canada.

[CS93] J. Conway and N. Sloane. *Sphere Packings, Lattices and Groups*. Springer Verlag, second edition, 1993.

[DL81] M. J. B. Duff and S. Levialdi, editors. *Languages and Architectures for Image Processing*. Academic Press, 1981.

[Dye] C. R. Dyer. A quadtree machine for parallel image processing. Technical Report KSL 51, Univ. of Illinois at Chicago Circle.

[Fey82] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 1982.

[FHP86] U. Frish, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier Stokes equation. *Physical Review Letters*, 56(14):1505–1508, April 1986.

[FT82] Edward Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3/4), 1982.

[Fun80] B. V. Funt. Problem–solving with diagrammatic representations. *Artificial Intelligence*, 13:201–230, 1980. Reprinted in [BL85].

[GL87] Peter M. Gruber and Gerrit Lekkerkerker. *Geometry of Numbers*. North-Holland Mathematical Library, 1987.

[Has87] Brosl Hasslacher. Discrete fluids. *Los Alamos Science*, (15):175–217, 1987. Special Issue.

[Her64] I. N. Herstein. *Topics in Algebra*. Blaisdell Publishing Co., 1964.

[Hil55] E. L. Hill. Relativistic theory of discrete momentum space and discrete space–time. *Physical Review*, 100(6), December 1955.

[Lev85] S. Levialdi, editor. *Integrated Technology for Parallel Image Processing*. Academic Press, Inc., 1985.

[Mac86] Thinking Machines. Introduction to data level parallelism. Technical Report 86.14, Thinking Machines, April 1986.

[Mar84] N. Margolus. Physics–like models of computation. *Physica 10D*, pages 81–95, 1984.

[Min82] M. Minsky. Cellular vacuum. *International Journal of Theoretical Physics*, 21(6/7), 1982.

[PD84] K. Preston and M. J. B. Duff. *Modern Cellular Automata: Theory and Applications*. Plenum Publishers, 1984.

[PF+85] J. Poulton, H. Fuchs, et al. PIXEL–PLANES: Building a VLSI–based graphic system. In *Chapel Hill Conference on VLSI*, 1985.

[PU82] K. Preston and L. Uhr, editors. *Multicomputers and Image Processing: Algorithms and Programs*. Academic Press, 1982.

[RN74] D. Rumelhart and D. Norman. Representation in memory. In R. Atkinson, R. Herrnstein, G. Lindzey, and D. Luce, editors, *Stevens' Handbook of Experimental Psychology*, volume 2: Learning and Cognition. John Wiley and Sons, NY, second edition, 1974.

[RS96] Dayanand S. Rajan and Anil M. Shende. A characterization of root lattices. *Discrete Mathematics*, (161):309–314, 1996.

[RS97] Dayanand S. Rajan and Anil M. Shende. Root lattices are efficiently generated. *International Journal of Algebra and Computation*, 7(1):33–50, 1997.

[She91] A. Shende. *Digital Analog Simulation of Uniform Motion in Representations of Physical n-Space by Lattice-Work MIMD Computer Architectures*. PhD thesis, SUNY at Buffalo, Buffalo, N. Y., 1991. TR 91-14, Department of Computer Science, and http://www.roanoke.edu/ shende/Thales/Papers/thesis.ps.

[Sny47] Hartland S. Snyder. Quantized space–time. *Physical Review*, 71(1):38–41, 1947.

[Svo86] K. Svozil. Are quantum fields cellular automata? *Physics Letters A*, 119(4), December 1986.

[SW86] J. B. Salem and S. Wolfram. Thermodynamics and hydrodynamics with cellular automata. In S. Wolfram, editor, *Theory and Applications of Cellular Automata*. World Scientific, 1986.

[TM87] Tommaso Toffoli and Norman Margolus. *Cellular Automata Machines*. MIT Press, 1987.

[Tof77a] Tommaso Toffoli. Cellular automata machines. Technical Report 208, Comp. Comm. Sci. Dept., University of Michigan, 1977.

[Tof77b] Tommaso Toffoli. Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences*, 15:213–231, 1977.

[Tof84] T. Toffoli. CAM: A high–performance cellular–automaton machine. *Physica 10D*, pages 195–204, 1984.

[Uhr72] L. Uhr. Layered 'recognition cone' networks that preprocess, classify, and describe. *IEEE Trans. on Computers*, C-21:758–768, 1972.

[Ung58] S. H. Unger. A computer oriented towards spatial problems. In *Proceedings of the IRE*, volume 46, pages 1744–1750, 1958.

[Vic84] G. Vichniac. Simulating physics with cellular automata. *Physica 10D*, pages 96–116, 1984.

[Vit88] P. M. B. Vitányi. Locality, communication and interconnect length in multicomputers. *SIAM Journal on Computing*, 17(4):659–672, 1988.

[Wol83] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644, July 1983.

[Zus69] K. Zuse. *Rechnender Raum*. Vieweg, Braunshweig, 1969. Translated as Calculating Space, Tech. Transl. AZT–70–164–GEMIT, MIT Project MAC, 1970.

Contents

1 Introduction **1**

2 Lattice Computers to Approximate Continuous Space **6**

2.1 Regular Lattice Computers and a Maximality Theorem 9

2.2 Root Lattice Computers 20

2.3 $\mathbf{A}_n$: The Best of Both Worlds 26

3 Additional Merits of $\mathbf{A}_n$ **27**

3.1 Properties of $\mathbf{A}_n$ Related to Computational Cleanness 27

3.1.1 Ease of Computations 27

3.1.2 Predecessor Containment 27

3.2 Comparison of $\mathbf{A}_n$ with $\mathbf{Z}^n$ 28

3.2.1 Rigidly Embedding a Lattice Computer into Another 28

3.2.2 Density of Points and Virtual Directions 29

3.2.3 Well Covering of $\mathbf{R}^n$ 31

3.2.4 Clustering of Processors 31

3.2.5 Interconnection Bandwidth Across Partitions 32

3.2.6 Automorphism Groups 32

3.2.7 Tessellating Euclidean Space 33

3.3 Embedding $\mathcal{M}(\mathbf{A}_n)$ In Physical 3-Space 33

3.4 Why Not Meshes Based on Denser Lattices? 35