DAMTP/1999/NA05

# Explorations in 3D integer-based linear geometry

M.A.Sabin

**Abstract:**

Current practice in geometric modelling is to use floating point numbers for coordinates. This leads to logical problems with precision and rounding errors[2]. There is also a technology used in computational geometry, using rational numbers of indefinite precision, which leads relatively quickly to very long word-lengths, with the corresponding problem of sloth[3].

Now in practical geometry we are interested in coordinates with a certain range and only to a certain precision. In mechanical engineering we are unlikely to regard objects of a metre across needing a precision of much better than a micron. But unless the extra precision is kept it appears that we hit problems of lack of robustness in the software, arising from incompatiblities between geometric and topological information.

This document explores the extension into three dimensions of a technique which works extremely well in two[5], of regarding points as having integer coordinates, and redefining what is meant by a line to be a set of such points.

Because it records an exploration, with many dead ends discovered, this document will appear negative to the reader. None the less, there is a lot of mathematical interest, and the fact that certain obvious analogues do not carry forward into 3D does not mean that there is no 3D equivalent.

Department of Applied Mathematics and Theoretical Physics,
University of Cambridge,
Silver Street,
Cambridge CB3 9EW,
England.

June 1999

# Contents

# List of Figures

# 1 Background: the 2D theory

The description here is based on the analysis presented by Luby[5]. his interest was in providing a description of geometry which would support robust area filling in computer graphics. He was therefore interested in domains about $2^{10}$ pixels across. His main problem was that the standard technique for creating lines across a graphics screen were those invented by Bresenham[1] for generating lines on drum plotters. This technique approximated a line by a series of axial and diagonal steps, and seemed a natural way of choosing pixels to illuminate also. However, it is very easy for two line segments making a small angle with each other to intersect in pixels which do not form connected groups. There is then no clear definition of the interior of a polygon, a fatal problem for raster graphics which wishes to fill such areas with different colours.

Luby therefore took as a primary axiom that lines must be defined in such a way that if two line segments intersect, they must do so in a connected sequence of points.

He constructed a scheme in which this property holds as follows.

## 1.1 Luby 4-connected system

### 1.1.1 Construction

Consider just the first quadrant and lines passing through the origin. There are two lines of length 1. Call these $L^1_0$ which is along the $x$-axis, and $L^1_1$ which is along the $y$-axis.

We then construct lines whose length is twice whatever we have considered before, constructing them by the rules

$$L^{2n}_{2i} = L^n_i L^n_i \quad i = 0, n \tag{1}$$

$$L^{2n}_{2i+1} = L^n_i L^n_{i+1} \quad i = 0, n-1 \tag{2}$$

where the concatenation $L^n_i L^n_j$ means that the longer line is formed by applying a translated copy of $L^n_j$ on the end of $L^n_i$.

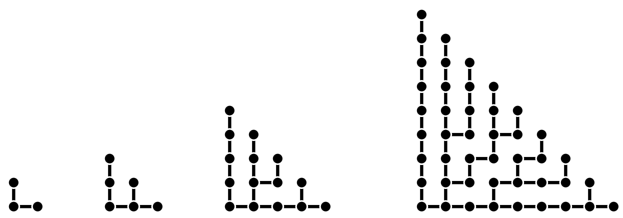

Figure 1: A doubling tree.

He showed that such lines are 4-connected, and if we intersect two such 'lines' with the euclidean lines $x + y = k$ (there is only one intersection point for each 'line') for different values of $k$, the directed distance between the two intersection points varies monotonically with $k$.

### 1.1.2 Minor Variations

Luby explored in some detail the possibility of different doubling rules.

Clearly it is possible to use as a second rule

$$L^{2n}_{2i+1} = L^{n}_{i+1} L^{n}_{i} \;\; i = 0, n-1 \tag{3}$$

and Luby calls this a 'down-tree' instead of an 'up-tree'. He also considered the possibility of letting the choice of rule vary with the order of the current level of doubling, and also with the detail of the value of $i$. For example, one could choose to use one second rule for $2i > n$ and the other for $2i \leq n$. There is some advantage in terms of accuracy which can be gained by exploiting these variations, but it is at best a constant, and not a very large one.

### 1.1.3 Doubling Trees and the Luby ruler

The set of lines through the origin forms a tree, and because these were created by an iterated doubling process, Luby called them **doubling trees**.

The domain is thus built up by doubling stripes between $x + y = 0$ and $x + y = n$. He called lines of equation $x + y = k$ **antidiagonals**, and clearly values of $k$ equal to powers of 2 have a particular significance in this theory.

In particular, if we look at the region of a point on antidiagonal $k = 2^i$, and consider all lines through it, we find that for lines shorter than $k$ the region immediately above is an exact image of the region near the origin.

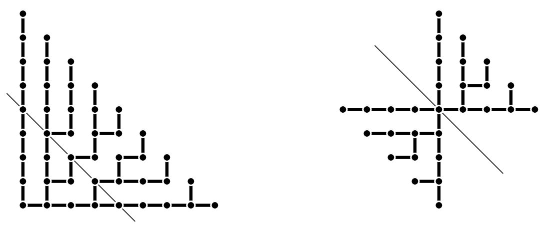

Figure 2: Lines through an intermediate point.

We define the **order** of an antidiagonal $k$ to be $o+1$ in the form $k = a * 2^o$, $a \neq 0$ modulo 2. Thus $k = 1, 3, 5, 7$ are of order 1, $k = 2, 6, 10$ of order 2, $k = 4, 12, 20$ of order 3 and so on. $k = 0$ is considered to be of the order 1 greater than the highest order encountered in the domain.

The **Luby ruler** is the relationship between antidiagonal and order, in particular applied to antidiagonals.

An important property is the fact that any two antidiagonals of the same order are always separated by one of higher order.

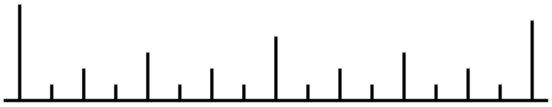

Figure 3: The ruler.

### 1.1.4 Other Quadrants

As is visible from figure 2 the pattern within the 3rd quadrant is a reflection across the antidiagonal direction of the pattern in the first. This can be proved by recursion on the level in the tree.

In the second (and thus fourth) quadrants we have an exactly similar mechanism for creating lines. Note that there is no elegant choice of which way the tree should go, as we have symmetry across either the $x = 0$ axis or across the $y = 0$ axis but not both.

The combination of the doubling mechanism and the definition of origin-based lines in all four quadrants gives a mechanism for extending lines as far as we need in any direction.

### 1.1.5 Notation for choice of rules

Here is some notation which will prove useful in dealing with threedimensional analogues.

If, in the first quadrant, the piece closer to the $x$-axis appears earlier in a 'mixed' line than the piece closer to the $y$-axis, we say that $x$ **has priority over** $y$, and we denote this by drawing an arrow from [1,0] to [0,1]. If the opposite rule is chosen the arrow is drawn in the opposite direction.

If, in the second quadrant, the piece closer to the $y$-axis appears earlier in a mixed line than the piece closer to the $-x$-axis, we say that $y$ has priority over $-x$, and denote this by drawing an arrow from [0,1] to [-1,0].

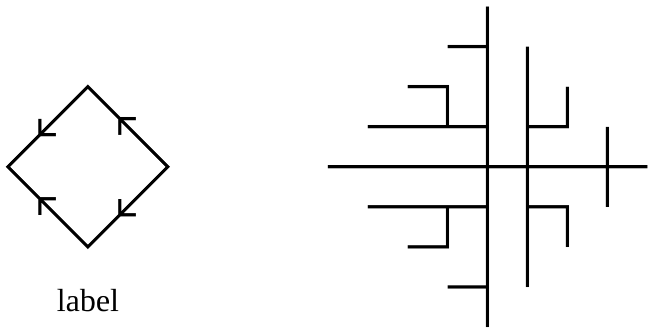


Figure 4: $x$ is dominant in $xy$.

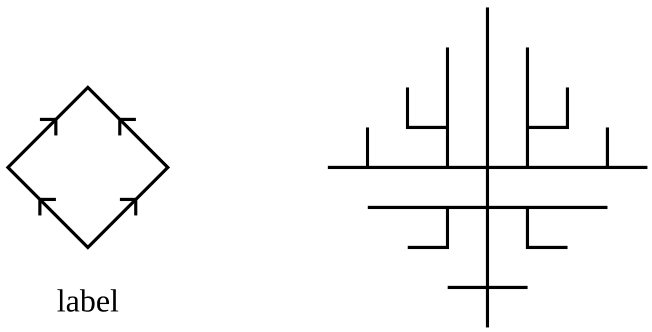


Figure 5: $-y$ is dominant in $xy$.

The remaining two quadrants cannot be chosen independently, and so we complete the diagram by drawing two more arrows, through [0,-1], parallel to the arrows already drawn.

There is just one vertex from which two arrows emerge, and we can say that this vertex '**dominates** in the plane. In figures 4 and 5, for example, $x$ and $-y$ dominate respectively. We comment that there is just one axis of symmetry.

### 1.1.6 Monotonicity and the Single Intersection Property

This property of monotonicity remains if we extend the set of first-quadrant lines by taking translates in the (-1,+1) direction, and the set of second-quadrant lines by taking translates in the (+1,+1) direction, thus giving a full set of lines. It is simple to deduce from monotonicity that there is only one line segment joining two given points (if there were two different paths, then the distance would be zero at both ends and non-zero somewhere between, which cannot be monotonic), and so the line segment between any two points of the intersection of two lines of the same quadrant must be unique. The two lines must therefore intersect in a connected set.

The intersection of lines taken from different quadrants satisfies the single intersection principle trivially, since travelling along each line has steps in only two directions, and the intersection of two such lines must be in just one of those two directions. Either $x$ or $y$ is monotonic in the other.

### 1.1.7 Approximation Properties: Convergence to Euclidean

With this scheme we have to give up something, and what we give up is the minimal error property of Bresenham. The distance between a line segment and the Euclidean line between the same endpoints is a maximum of half a pixel for Bresenham, but increases to $log_2(n)/2$ for Luby, where $n$ is the length of the line.

This sounds bad, but in fact the variation is tolerable. Suppose that we want to represent lines $10^6$ ($2^{20}$) pixels long. The error is just 10 pixels. So if we start with a grid a factor of 32 finer than we think we need, our lines are now $2^{25}$ long with an error of only 12.5 pixels, less than half our original pixel size. The cost of the extra precision for this purpose can be compared with the cost of the bits used for the exponent in floating point representations, and it turns out less for practical wordlengths.

In fact this representation has an important property, that as the resolution increases, the lines converge towards the Euclidean, and we can therefore reasonably take them as adequate for incidence calculations. We have to remember, of course, that we cannot measure length by counting pixels, but that is true of all discrete schemes.

| word-length (bits) | error (units) | guard-length (bits) | useful bits | resolution | IEE mantissa (bits) |
|---|---|---|---|---|---|
| 8 | 4 | 3 | 5 | 1:32 | - |
| 16 | 8 | 4 | 12 | 1:4096 | - |
| 32 | 16 | 5 | 27 | $1:1.28*10^8$ | 24 |
| 64 | 32 | 6 | 58 | $1:2.56*10^{17}$ | 53 |

### 1.1.8 Incidence Calculations

Luby showed some very fast algorithms which were based on the use of tables, which could be tuned to reflect sophisticated rule choice . These cannot be directly extrapolated to the general geometric context, as a table of length $2^{10}$ is a much more practical affair than one of length $2^{32}$ or $2^{64}$. However, if we limit ourselves to using the same rule all the time there are bit-manipulation algorithms for incidence calculations whose time is proportional to the word-length, and which could be made as fast as the Euclidean algorithms by use of microcoding. A hybrid can use short tables for, for example, reversing the sequence of bits in a byte, together with a small amount of code to apply this to reversing the sequence of bits in a 64-bit word.

### 1.1.9 Duality and Uncertainty

When we are calculating, for example, the line joining two points, we need a representation of lines. A convenient representation uses the intercepts $p$ and $q$ on the lines $x + y = 0$ and $x + y = 2n$. Each line thus has two coordinates, just like the points.

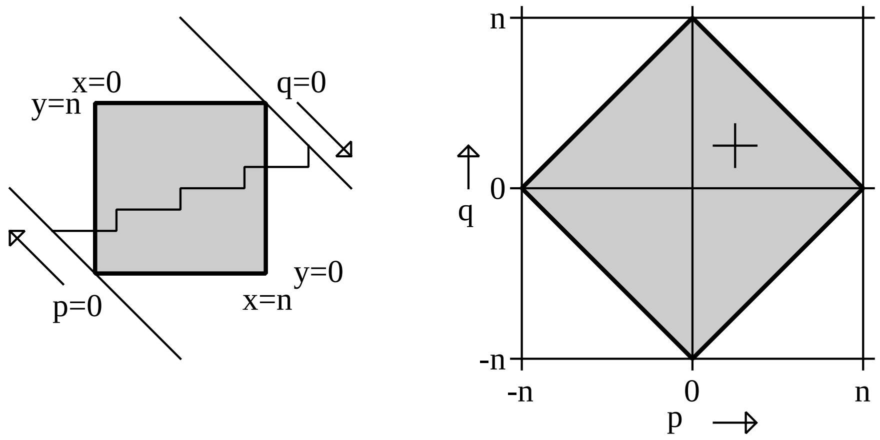


Figure 6: Dual spaces.

This particular choice gives an elegant duality. It turns out that the set of lines passing through a given point $x, y$ is a set of $pq$ pairs which form a 'line' in $pq$-space. The set of such lines forms another tree exactly similar to the tree of lines in $xy$-space.

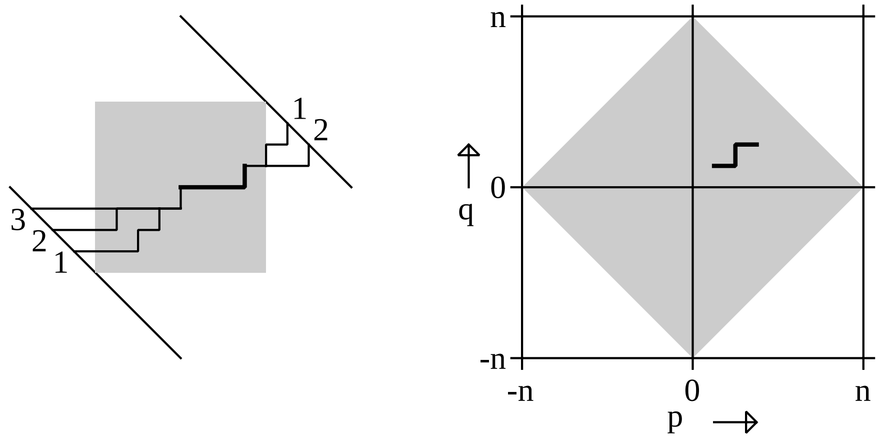


Figure 7: A dual segment.

The algorithm for finding the intersection of two lines is exactly the same as that for finding the join of two points. In both cases the result is a line-segment, which may be regarded as a set of points 4-connected in $xy$-space, or as a set of lines 4-connected in $pq$-space, all of which share the same segment.

Segments which are long in $xy$-space tend to be short in $pq$-space and vice versa. The exact relationship is fairly complicated, but the product of the lengths is about the width of the domain.

### 1.1.10 Preferred Solutions

The facts that the meet of two lines is not a single point, and that the join of two points is not a single line would, of course, mean that the precision of a sequence of incidence calculations would deteriorate with the length of the sequence, particularly if line segments were extrapolated to find further intersections (or if a point in a segment were used as an endpoint of a segment not within its set of lines).

Such deterioration can be hidden by a process of choosing a preferred line among all lines through a segment and a preferred point on each segment.

A good choice for the preferred line is the one given by the intersection of maximum order between the dual segment and the Luby ruler. This line is the straightest among all the possibilities. Similarly, the preferred point of a segment may be taken as the intersection of the primal segment of the maximum order with the Luby ruler.

These preferred choices turn out to be quicker to compute than the complete segment, since recursive subdivision will find them as the first solution found. More important, however, is the fact that this choice is always unique and there is never any arbitrary rounding.

### 1.1.11 How many lines are there ?

From figure 7 we observe that the area of $pq$-space is about twice that of $xy$-space, so that the total number of lines including both quadrants is slightly less than four times the number of points ($4n^2 - 6n$ as against $n^2 + 2n + 1$). This apparent breach of the symmetry that we expect to result from duality is related to the fact that while every pair of points in the domain has a join, not every pair of lines has an intersection within the domain.

## 1.2 8-connected lines

A variant on this theory was independently derived by van Lierop[4], who used the doubling tree idea within not a quadrant, but a 45 degree part of the $xy$-plane. In hindsight it is easy to identify this scheme as being an affine transformation of the set of Luby lines in a quadrant.

The advantage is that the steps can be at 45 degrees as well as at 0 and 90, thus giving smoother lines.

The disadvantage is that a line at an angle of about 30 degrees can cross one at about 100 degrees without sharing a common point, as was illustrated in figure 8.

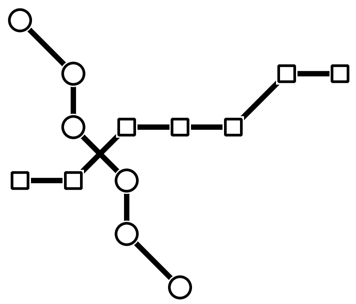

Figure 8: 8-connected lines not intersecting.

## 1.3 Hybrid and the triangular grid

There is a possibility of using the van Lierop scheme in the first (and third) quadrants and the Luby one in the other two. At first sight this looks really kludgy, but in fact it is an affine transform of a scheme on a triangular lattice instead of a rectangular one. It has the elegant property that we can then derive the trees in all the hexants by reflections across the primary axes: in the Luby and van Lierop cases we have to choose which axes to reflect across.

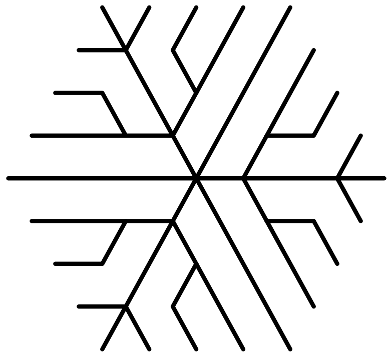

Figure 9: The 6-way tree.

# 2 Possible approaches in 3D

The objects we want to deal with are points, lines and connected pieces of them, and planes and connected pieces of them.

We can base a scheme initially by defining either lines or planes in terms of the points which they contain, and then defining the other on which ever is chosen as a primary definition, or we can define lines and planes independently in terms of points and then show that our definitions are compatible.

Independent definitions of lines and planes in terms of points are useful first steps whichever way we go.

We can also use the duality idea to use lines as our base starting point.

## 2.1 From points to lines

In this approach we define the points to lie on a regular grid, identify a discrete space of lines and then find a way to define which points lie on which lines.

Once this is done, the next step would be to define planes in terms of such lines, presumably in terms of sets of lines joining a point to the points of a generator line.

## 2.2 From points to planes

In this approach we define the points to lie on a regular grid, identify a discrete space of planes and then find a way to define which points lie on which planes.

Once this is done, the next step would be to define lines in terms of such planes, presumably in terms of intersections.

## 2.3 From lines to points and planes

It is also possible to leave points until later, starting with a discrete space of lines as our primary entities, and looking for an approach in which the map from lines to points is closely dual to that from lines to points.

Points and planes in such a scheme would be defined in terms of singular linear complexes, and we would have no guarantee that points lay on a regular grid in Euclidean space. However, we have already lost exact invariance under translation, and so this need not be a problem.

# 3 Definitions and requirements

*** This whole section was written early on and is ripe for serious revision. Nothing in it should be regarded as reliable, and it may be best for the reader to skip this section on first reading, going straight to section 4. ***

The objective in this section is to set up some vocabulary and identify requirements for the discussion of different possible schemes for the discretization of points lines and planes in a space of three dimensions, and the calculations with them based on incidence. Even the articulation of the appropriate 3D objects is tentative.

Each paragraph here is labelled either as a definition of whatever term appears in bold within that paragraph, as a requirement that we hope to achieve within a discrete system, or as a theorem which we can prove from the definitions and requirements.

## 3.1 Points

### 3.1.1 Point definition

A **point** is represented by a triple of integers, called the $x$-, $y$- and $z$-coordinates. These integers are coordinates in a lattice of three dimensions.

Geometric objects are viewed in terms of the sets of points that they contain.

A geometric construction may be posed in terms of computations on points with coordinates taken from some continuous domain. These have to be approximated by discrete points before the equivalent discrete computation, but no further approximation will happen during that computation.

### 3.1.2 Region of Interest definition

We assume that all points encountered within a computation, either as input or during the course of the computation, have finite coordinates. We can define a finite **region of interest**, convex and large enough that all points lie within it.

### 3.1.3 Resolution definition

The **resolution** of the representation of a point set is the ratio of the unit to some measure of the linear size of the region of interest. The concept is that by multiplying all coordinates by some large factor, the grain of the discrete representation can be made as small as required.

### 3.1.4 Convergence to Euclidean requirement

We require that by improving the resolution, the result of any discrete calculation can be made to approximate the corresponding continuous (Euclidean) one as closely as required.

We do not expect that the result at any given resolution will always be that achieved by approximating the continuous result.

## 3.2 Lines

### 3.2.1 Adjacency definition

Two points are **adjacent** if the vector from one to the other is of magnitude unity. Equivalent definitions are

$$\begin{aligned} |\delta x| + |\delta y| + |\delta z| &= 1 \\ \delta x^2 + \delta y^2 + \delta z^2 &= 1 \end{aligned} \tag{4}$$

### 3.2.2 Segment definition

A **segment** is defined by two points, $A$ and $B$.

We denote the segment by $Seg(A, B)$

For most purposes we require that $A$ and $B$ are distinct, but it is convenient to permit the concept of a segment to include the case where the same point is used twice in the definition.

### 3.2.3 Pointset of a Segment definition

A segment $Seg(A, B)$ **contains** points $P_i, i = 0 \ldots n$ such that

$$\begin{gathered} P_0 = A \\ P_n = B \\ P_{i-1} \text{ and } P_i \text{ are adjacent, } i = 1 \ldots n \\ P_i \text{ and } P_j \text{ are not adjacent}, i \neq j \pm 1 \end{gathered} \tag{5}$$

We use the notation $P \in Seg(A, B)$ to denote that $\exists i | P = P_i$

If $A = B$ the segment contains only one point.

### 3.2.4 Length of a Segment definition

The **length** of a segment is $n-1$, one less than the number of points it contains. Note that this is not an approximation to Euclidean length, but a useful technical term in dealing with segments.

The value of $n$ is typically determined directly from the endpoints as the sum of the differences in the three coordinates. This property is probably not going to be used explicitly, and we shall not bother to prove it. What is important is that $n$ is finite, which emerges quickly from any particular scheme, typically via the property that for each point in a segment, each coordinate is within the closed interval bounded by the corresponding coordinates of the endpoints.

If $A = B$ the length is zero.

### 3.2.5 Endpoints definition

The points used to define a segment are, by definition members of its pointset. They are called the **endpoints** of the segment.

### 3.2.6 Symmetry requirement

We require the pointset of a segment to depend only on its unordered endpoints.

$$P \in Seg(A, B) \equiv P \in Seg(B, A)$$

### 3.2.7 Intersection requirement

The intersection of two pointsets is well defined as containing those points which are members of both sets.

We require that two segments shall have as their intersection either the empty set or exactly the points of some segment.

The computation of the endpoints of such an intersection segment is an important computation.

The way in which the intersection could fail to be a segment is that its pointset might contain more than one connected component. This failure occurs for the 2D scheme in which segments are generated by the Bresenham algorithm.

### 3.2.8 Preferred intersection definition

For some purposes we require that a point can be calculated as the intersection of two segments. We require that there be some mechanism within a scheme for choosing a unique **preferred intersection point** of the intersection segment.

This choice mechanism should be applicable to any segment, not just one that is defined by an intersection, because we can always define the intersection of a segment with itself.

### 3.2.9 Line definition

A **line** is a segment whose endpoints are too far way for the finiteness of the segment to influence the geometrical properties we are interested in. We can treat it as the discrete analogue of an infinite line.

The intersection of two lines is therefore a segment.

There is a set of lines containing any given segment.

If we define two lines as being the same if their point sets within the region of interest are the same, this set is finite.

### 3.2.10 Preferred Line definition

For some purposes we require that a line can be calculated as that joining two points. We require that there be some mechanism within a scheme for choosing a unique **preferred line** of the segment defined by the two points.

This choice mechanism should be applicable to any segment, because the endpoints of a segment are determinate from its pointset as the only points which do not have two adjacent neighbours within the set.

### 3.2.11 Distance between two Lines definition

If two lines have a non-zero intersection, the distance between them is zero. If they do not, the distance is the length $(n-1)$ of the shortest segment joining a point of one line to a point of the other within the region of interest. The sign of the distance needs to be defined in some way analogous to continuous winding number, but defined over lines of finite length.

## 3.3 Planes

### 3.3.1 Triangle definition

A **triangle** is defined by three points, $A$, $B$ and $C$.

We denote it by $Tri(A, B, C)$

For most purposes we expect these to be distinct points, but it is convenient to permit the term to be used when the set $\{A, B, C\}$ has only two members or only one member.

### 3.3.2 Pointset of a triangle definition

If the three points are coincident, the triangle contains that point.

If there are only two distinct points, the triangle is the segment defined by them.

If there are three distinct points, the triangle is the union of the pointsets of all segments from any point on any one of the segments $seg(A, B)$, $seg(B, C)$, $seg(C, A)$ to any other such point.

### 3.3.3 Segment/Triangle Intersection requirement

We require that the intersection of a segment and a triangle shall contain the points of some segment.

### 3.3.4 Thinness of a triangle requirement

For a triangle to be the analogue of a continuous triangle, we want to identify that in some sense it is only 'one point thick'.

This might be posed in the form that for any given point of any given triangle, there must exist a segment which intersects the triangle in that point only, but that does not even work for segments (we used the non-adjacency condition for the equivalent segment condition).

In the 2D theory, anti-diagonals are used. These are line-like collections of points which are not 4-connected. Maybe the same construction is what we want.

A third alternative definition is that the points of a triangle can be arranged in level sets by the length of a segment from a given point. We can ask that the points of any given level set should form an adjacency chain or loop, where each point is adjacent to at most two others.

The difficulty in posing this intuitive property is only a precursor to actually achieving the property, which looks hard.

Once the property has been posed, we shall probably define an object which has both it and the property of being connected, and give it a name (**plate ?**) so that we can show that the intersection of two such objects is again such an object, as a bivariate analogue of the segment intersection property.

The level-set approach might lead to a definition of the other intuitive property that there are no 'holes' in the triangle. This could be expressed by saying that the number of connected components in a level set never decreases as the level increases.

### 3.3.5 Plane definition

A plane is a triangle whose defining points are well enough distributed and far enough away that the boundedness of the triangle does not influence the properties that we are interested in.

### 3.3.6 Separation requirement

We should then be able to show that a plane separates the region of interest into three parts. One part is the pointset of the plane itself; the other two correspond to the regions on the two sides of the plane. Call these three regions ON, IN and OUT respectively. Then a segment joining two points in ON lies entirely in the plane (from R18). A segment joining two points in IN or two points in OUT has no intersection with the plane, and a segment joining a point in IN to a point in OUT will have a non-zero intersection.

### 3.3.7 Triangle/triangle intersection requirement

Two triangles will in general intersect in some plate. We would like to be able to choose a preferred maximal segment lying in the intersection. Similarly we would like to be able to choose a preferred plane for any triangle and a preferred line for the intersection of two planes.

# 4 From Points to Lines

The Luby 2D theory works by first defining origin-based lines (lines through the origin) using the concept of a doubling tree. The set of all lines are then translates in an antidiagonal direction of the lines through the origin.

We can map this into 3D in at least two ways.

One is to use the concept of a doubling tree in 3D, followed by translation over an antidiagonal plane; the other is to require that projections of 3D lines into each of the coordinate planes shall result in 2D lines as described by the Luby scheme.

We deal with the second first, as it is easy to show that it doesn't work.

## 4.1 Definition of Lines by Intersection of Extrusions

### 4.1.1 Projections and Extrusions

The **projection** of a point set in 3D into a coordinate plane is the set obtained by ignoring one of the coordinates.

Let $A = \{P_i = [x_i, y_i, z_i]\}$. Then the projection of $A$ into the $xy$-plane is $\{Q_i = [x_i, y_i]\}$.

It is easy to show that the projection of a segment is a segment.

(Not quite: connectedness is easy, but the non-adjacency of non-consecutive points in the projection probably depends on monotonicity of coordinates in the original, which we didn't actually state. However, monotonicity follows from the property that the segment between two points differing only in one coordinate should vary only in that coordinate, together with the segment intersection property.)

The **extrusion** $E$, of a 2D segment $S$, is the set of points which project into points of $S$.

It is clearly a bivariate entity.

The intersection of two of these is not in general a segment, but the intersection of three might be.

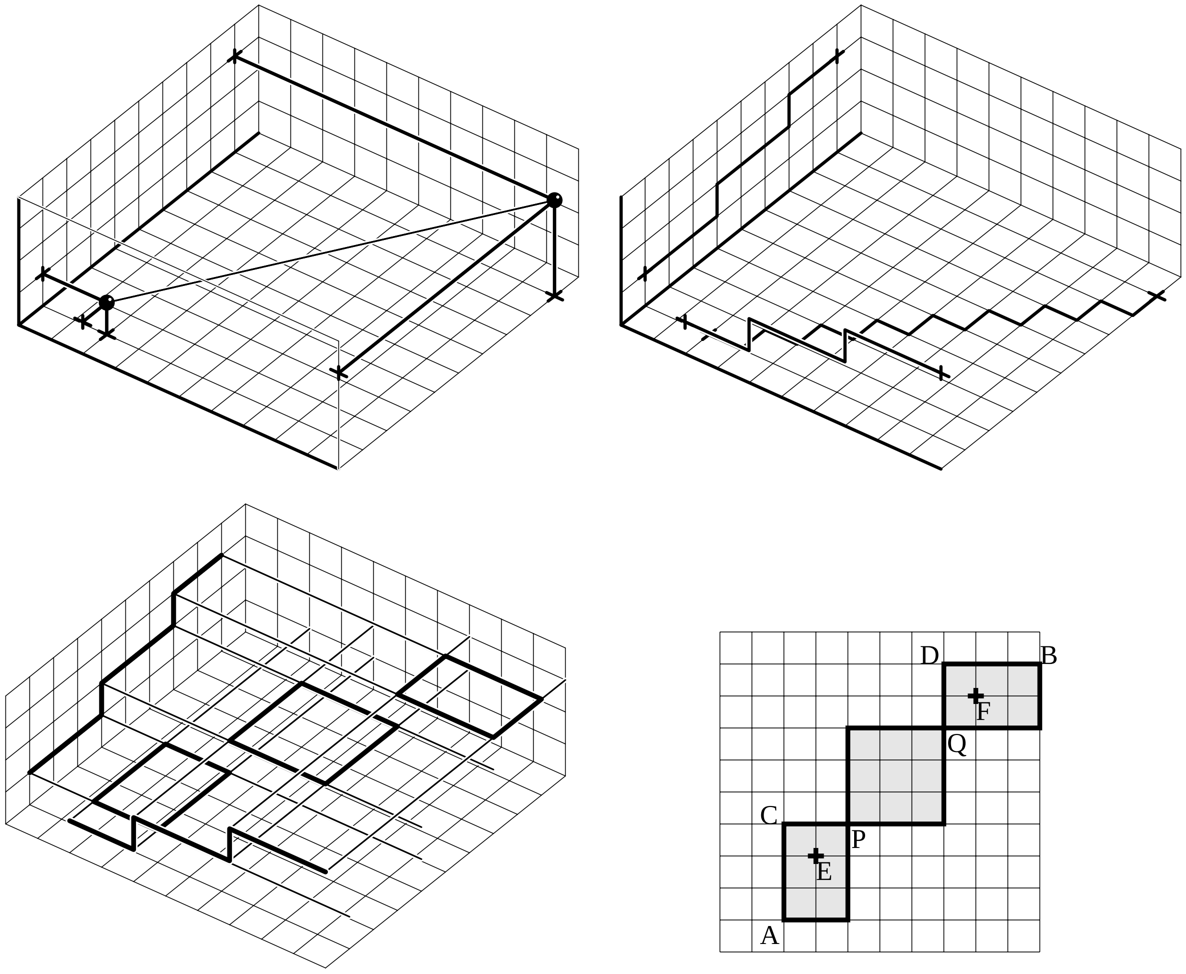


Figure 10: Incompatibility of extrusions.

### 4.1.2 Theorem: Inconsistency of projection definition

We shall show by counterexample that in general the three extrusions of the 2D segments defined by projecting the endpoints of a potential 3D segment do not intersect in a 3D segment.

Consider the case of two 3D endpoints where the change in $z$ is non-zero, but considerably less than the changes in $x$ and $y$, as illustrated in figure 10.

The segments in the $xz$- and $yz$- planes consist therefore of wide shallow steps.

The intersection of the extrusions of these steps takes the form of a sequence of rectangular plateaux meeting at diagonally opposite corners.

The potential 3D segment must step from one $z$-level to the next at these corners.

However, the position of these corners does not change for small changes in $x$ and $y$ of the endpoints (within one plateau of the system). It is possible to move both endpoints in the antidiagonal direction in the $xy$-plane. The projection of the endpoints into the $xy$-plane should therefore give parallel segments whose extrusions cannot all pass through the requred corners.

For example, while the segment from A to B may well pass through E, P, Q and F, the segment from C to D, which is displaced from EF in the antidiagonal direction and cannot therefore share any points with EF, does share the same projections and must therefore also pass through P and Q.

We deduce not just that the intersection of three extrusions is inconsistent, but that the intersection of two extrusions is incapable of guaranteeing the intersection property.

Note that this theorem holds for a wide range of possible 2D line schemes; probably all that satisfy the segment intersection property. It is not limited to the specific cases treated by Luby.

#### 4.1.3 Theorem: Inconsistency with projections

There is a worse result. Because the extrusions cannot be consistent among themselves, we cannot expect a 3D segment defined by any other scheme to be consistent with its projections.

This in turn means that the extrusion of a 2D line cannot be defined to be an example of a 3D plane.

### 4.2 Definition of Lines by Doubling trees

The direct analogue of the Luby theory can be set up straightforwardly, with only a few possible variants appearing.

Consider the segments in the positive octant rooted at the origin.

There are exactly three segments of length 1. The segments of length 2 are constructed by appending segments of length 1, with a choice of which way to go diagonally. The permissible combinations are only $xx, yy, zz, xy, xz, yz$. The choice of $xy$ rather than $yx$ is made on the basis of lexicographic ordering.

#### 4.2.1 Pure doubling

It is then possible to define the segments of length 4 by combining two segments of length 2. Then the segments of length 8, and so on, creating an exact analogue of the two dimensional construction described above.

The origin-lines at the double size are represented by the nodes of a regular triangular lattice of their endpoints. Underlying this is another lattice whose triangles have sides twice as long, being the lines which are created by repetition of precursor lines.

All the others lie on the mid-edges of these large triangles, and are created by taking first the line corresponding to one end of the triangle edge, then the line corresponding to the other end.

We define them by choosing a direction along each edge, and stating that the line at the tail of the arrow appears before the line at the head of the arrow.

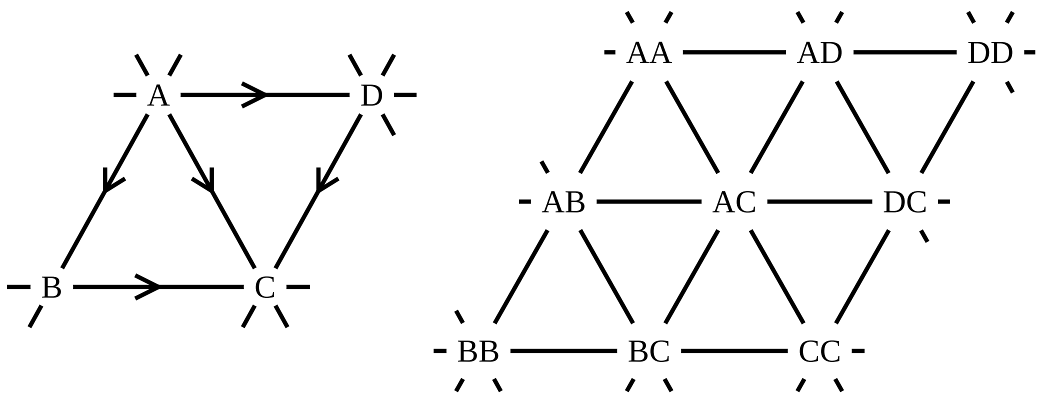


Figure 11: Choice of priorities.

Experiment shows that we cannot choose a cyclic sequence of arrows: we have to chose a sequence $abc$ which denotes that the arrows go from $a$ to $b$ and to $c$ and from $b$ to $c$. If we try to make a tree using cyclic arrows, we very rapidly get paths colliding where tree paths should not. Without loss of generality we can choose the sequence $xyz$, as any other sequence is just a different positioning of the same pattern in space.

Exhaustive numerical checking shows that with such a convention for choosing the first and second halves of a mixed line the behaviour is indeed tree-like and lines in the same octant have the single intersection property. We cannot use the monotonic variation of the antidiagonal coordinate for a rigorous proof in the 3D case, but it appears that the magnitude of distance in the antidiagonal plane is indeed monotonic on either side of the intersection region.

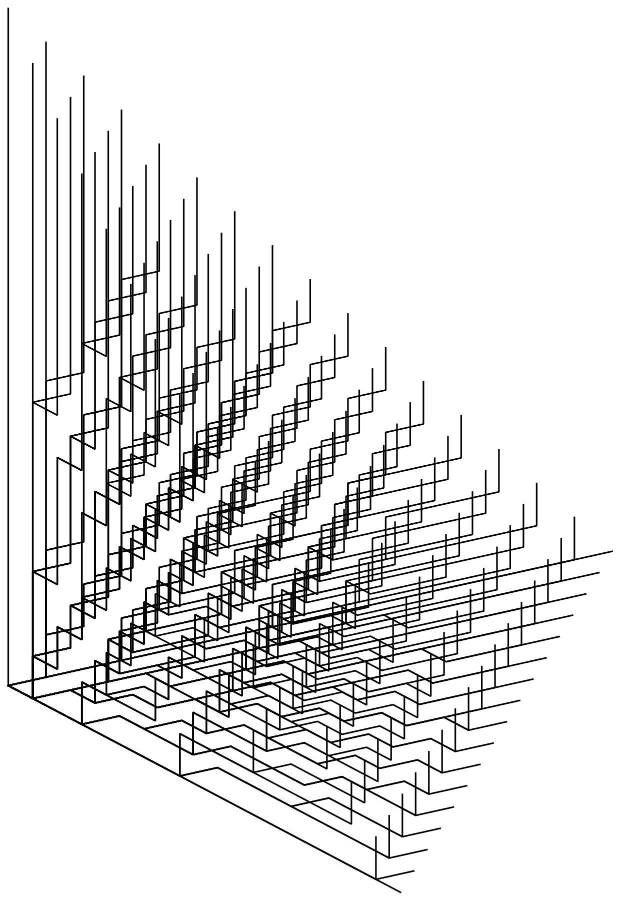

Figure 12: A doubling tree in 3 dimensions.

### 4.2.2 Minor variations within an octant

In the 2D case we have just two possibilities at each level of the doubling, the up choice or the down choice. In 3D there are more possibilities, but they are all equivalent to each other.

The priority scheme in one octant can be expressed as a labelling with orientations of the edges of the triangle seen at any given antidiagonal doubling plane.

Each level in the doubling could have its own labelling, but there was no advantage in the 2D case in this extra complication and so it has not been explored.

We could also possibly have a triangular lattice where different subtriangles are differently oriented. This complication also was not explored. We just assume that all similarly oriented edges have the same sense.

### 4.2.3 3-tree at first level

A slight problem with the pure doubling tree is that it does not ever extend exactly to the far vertex of a cube at the origin.

If this were a problem then we could define a variant in which the first layer is three steps thick, chosen lexicographically to be

$$xxx, xxy, xxz, xyy, xyz, xzz, yyy, yyz, yzz, zzz$$

From this point onwards the same doubling procedure can be applied. This option has not been explored in detail.

### 4.2.4 Trebling trees

The same principle could be applied at every level, rather than just at the first. There is no obvious advantage in doing so.

### 4.2.5 Compatibility of Octants

A problem which does not appear in the 2D theory is how to make adjacent octants compatible. This is necessary for two reasons.

First, there has to be a clear definition of segments whose endpoints share the value of one of their coordinates. This could possibly be overcome by some kludge giving priority to one of the octants it could be deemed to lie in.

However, there is a deeper underlying problem that we need to maintain the single intersection property for lines whose directions are in adjacent octants. This is not going to be as easy as in the 2D case, and a first start is by getting the priority rules for the two lines as consistent as we can.

This is necessary, in order to ensure that two segments whose endpoints are symmetric about a coordinate plane have a single intersection. The intersection will lie in the plane of symmetry, and will lie in a line in that plane of symmetry. If the octants had different patterns on the plane of symmetry there would be no proof of compatibility, and indeed a counterexample would probably not be hard to find.

Now when we consider more than one octant at a time we can imagine the triangles as the faces of an octahedron. The edges have to be labelled in such a way that every triangle face has the pattern that we saw to be necessary.

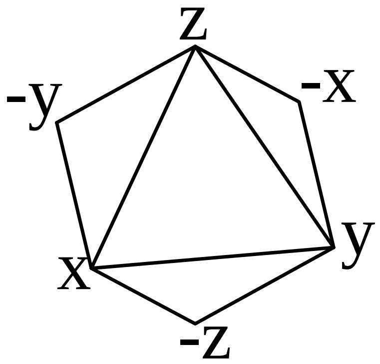


Figure 13: Choices for all octants.

We also have the constraint that opposite octants are compatible, since they actually describe the same orientation of line. If the sequence for the first octant is $xyz$, the sequence for the opposite one must be $ZYX$ where $X$ denotes the opposite direction to $x$.

We can express this compatibility as selecting edge-orientations on an octahedron and it turns out that there is essentially only one pattern which satisfies these requirements.

Each of the three planes of symmetry of the octahedron has a pattern of the kind shown in figures 4 and 5, with one of its vertices dominant. If the three dominant vertices were all distinct then one of the faces would have cyclic arrows.

Take any vertex of the octahedron (wlog we can choose $X$) and draw arrows to its antipode along the four possible direct routes. Then choose one of the four equatorial vertices and draw arrows to its antipode along the two remaining unarrowed routes. The result satisfies all the necessary constraints.

Note that the complete tree has exactly one plane of symmetry.

### 4.2.6 Theorem: lines in adjacent octants do not have connected intersections.

If the common intersection cannot be shown to lie in a coordinate plane, the situation becomes harder. Consider the line from 0,0,0 to 11,2,3 assuming that $y = 0$ is the plane of symmetry. We can elaborate its pointset as follows

| recursion | | | | | step | point |
|---|---|---|---|---|---|---|
| 11,2,3 | | | | | | |
| | 6,1,1 | | | | | |
| | | 3,1,0 | | | | |
| | | | 2,0,0 | | | |
| | | | | 1,0,0 | 1,0,0 | 1,0,0 |
| | | | | 1,0,0 | 1,0,0 | 2,0,0 |
| | | | 1,1,0 | | | |
| | | | | 1,0,0 | 1,0,0 | 3,0,0 |
| | | | | 0,1,0 | 0,1,0 | 3,1,0 |
| | | 3,0,1 | | | | |
| | | | 2,0,0 | | | |
| | | | | 1,0,0 | 1,0,0 | 4,1,0 |
| | | | | 1,0,0 | 1,0,0 | 5,1,0 |
| | | | 1,0,1 | | | |
| | | | | 1,0,0 | 1,0,0 | 6,1,0 |
| | | | | 0,0,1 | 0,0,1 | 6,1,1 |
| | 5,1,2 | | | | | |
| | | 3,0,1 | | | | |
| | | | 2,0,0 | | | |
| | | | | 1,0,0 | 1,0,0 | 7,1,1 |
| | | | | 1,0,0 | 1,0,0 | 8,1,1 |
| | | | 1,0,1 | | | |
| | | | | 1,0,0 | 1,0,0 | 9,1,1 |
| | | | | 0,0,1 | 0,0,1 | 9,1,2 |
| | | 2,1,1 | | | | |
| | | | 1,1,0 | | | |
| | | | | 1,0,0 | 1,0,0 | 10,1,2 |
| | | | | 0,1,0 | 0,1,0 | 10,2,2 |
| | | | 1,0,1 | | | |
| | | | | 1,0,0 | 1,0,0 | 11,2,2 |
| | | | | 0,0,1 | 0,0,1 | 11,2,3 |

We take the part of this line from 2,0,0 to 8,1,1
The part which lies in $y = 1$ is that from 3,1,0 to 8,1,1

Now consider the reflection of this line in the plane $y = 1$. It runs from 2,2,0 to 8,1,1, which is a displacement of the origin-based line 0,0,0 to 6,-1,1.

Its elaboration is, with its displaced version

| recursion | | | | step | point | displ.point |
|---|---|---|---|---|---|---|
| 6,-1,1 | | | | | | |
| | 3,-1,0 | | | | | |
| | | 2,0,0 | | | | |
| | | | 1,0,0 | 1,0,0 | 1,0,0 | 3,2,0 |
| | | | 1,0,0 | 1,0,0 | 2,0,0 | 4,2,0 |
| | | 1,-1,0 | | | | |
| | | | 1,0,0 | 1,0,0 | 3,2,0 | 5,2,0 |
| | | | 0,-1,0 | 0,-1,0 | 3,-1,0 | 5,1,0 |
| | 3,0,1 | | | | | |
| | | 2,0,0 | | | | |
| | | | 1,0,0 | 1,0,0 | 4,-1,0 | 6,1,0 |
| | | | 1,0,0 | 1,0,0 | 5,-1,0 | 7,1,0 |
| | | 1,0,1 | | | | |
| | | | 1,0,0 | 1,0,0 | 6,-1,0 | 8,1,0 |
| | | | 0,0,1 | 0,0,1 | 6,-1,1 | 8,1,1 |

whose section in the plane $y = 1$ runs from 5,1,0 to 8,1,1

Note that the step in $z$ is at $x = 8$ whereas that in the original segment was at $x = 6$. We thus have a counterexample, which shows that merely ensuring compatibility of labelling between octants is not enough.

# 5 From Points to Planes

If defining lines first has its problems, then an alternative approach is to define first discrete planes, and then look to see how lines might emerge from them.

We have already seen that we cannot just extrude 2D Luby lines along the coordinate directions to create examples of planes, since the pairwise intersections will not be compatible.

Four alternatives are considered in this section. We first introduce the concept of a vector sum plane, and apply it is a straightforward way. This is not very symmetric and so a symmetric variant is produced, introducing the concept of the Gauss figure. A variation on the basic vector sum is considered, and then a vector sum of a different basis.

## 5.1 Vector Sum plane definitions

The vector sum of two point sets $A$ and $B$ is the set of points which can be constructed by taking one member each of $A$ and $B$ and adding their coordinates. In order to make this invariant under changes of coordinates system, we also need to subtract a datum point also a member of at least one of the sets.

The particular relevance here is that we can make a bivariate set by taking the vector sum of two univariate sets.

For example, we can make what we shall call an origin-based plane by taking the vector sum of one 2D origin-based line in the XZ coordinate plane and another in the YZ coordinate plane. The datum is the common point at the origin.

General planes are then defined as translates in the $-X - Y + Z$-direction of origin-based planes in this octant, and along corresponding directions in the other octants. These directions are chosen to take the role of antidiagonals because a ray in this direction cuts each vector sum plane of the octant in only one point.

Note that the planes defined in this way include the extrusions and so we are likely to have problems difining a linear system of planes to include a given line.

### 5.1.1 Normals

We define the normal to such a plane to be the cross product of the vectors from the origin to the ends of the generating lines. If the generating lines are [0,0,0] to $[2n - z_1, 0, z_1]$ and to $[0, 2n - z_2, z_2]$, the normal vector has components $[-z_1(2n-z_2), -z_2(2n-z_1), (2n-z_1)(2n-z_2)]$. A more convenient integer labelling scheme is just $[-z_1, -z_2, 1]$

There is good symmetry between the X and Y directions here, but Z is treated differently. The distribution of planes is not even, in the sense that the surface normals cover the Gauss sphere very unevenly. There are no planes defined which include the $Z$-axis.

## 5.2 More symmetric variant and the Gauss Figure

This lack of symmetry can be addressed by using this formulation only with lines close to the X and Y axes, but also creating planes from lines in the other pairs of coordinate planes.

The normals are now labelled with points of a lattice not on just a plane, but on the faces of a cube. (Opposite faces are identified because planes form a projective space.)

This cube is a discretisation of the Gauss-Sphere. Is surface, with the obvious grid on each face, gives a discrete space for origin-based planes. The ability to define some lattice of planes like this is likely to be an important feature of any 3D formulation. As we shall see in subsequent sections, different polyhedra can provide the basis for this, and we shall call each of them the **Gauss figure** in its own context.

We also note that an origin-based line is going to have to correspond to a well-behaved path around the space of origin-based planes.

### 5.2.1 Compatibility of planes near the frontier

In a piecewise definition such as this, we have to be careful about compatibility where different parts of the definition meet. In this case the critical case is that the plane whose normal is labelled $[\pm1, \pm1, \pm1]$ can be constructed in three ways. The detail of the point-set of the plane depends on the priority schemes of the 2D lines in the coordinate planes.

The totality of the priority choices can be summed up in the labelling of the edges of an octahedron with arrows. The requirements are rather different here from the labelling used for constructing doubling-tree lines in section 4.2.5 above.

We have essentially two possible configurations. In the first the origin has six adjacent points in the plane $z = x + y$

### 5.2.2 Theorem: 6-adjacent origin does not give compatible planes

We can make such a plane in two symmetric ways. Wlog we can choose the priorities such that $x \to z \to y$.

When we look to see how the same plane is expressed as $x = z - y$, we find that the priorities have to be $-y \to x \to z$.

When looking for $y = z - x$, however, we find that the priorities have to be $z \to y \to -x$, and so we require incompatible priorities in the $xy$-plane.

This configuration does not work.

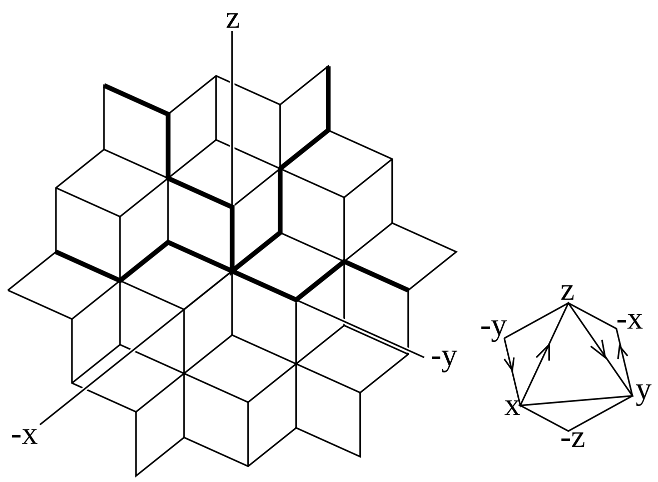


Figure 14: 6-connected origin.

### 5.2.3 Theorem: 3-adjacent origin does not give compatible planes

The second possible configuration has the origin adjacent to three other points of the plane.

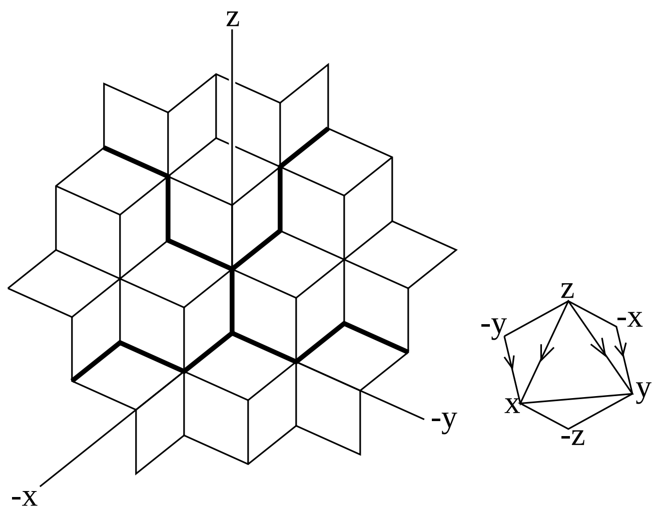


Figure 15: 3-connected origin.

We now find that it is possible to make the three representations of this plane compatible, using the labellings

$$\begin{array}{rcl} z & \rightarrow & x \\ z & \rightarrow & y \\ -y & \rightarrow & x \\ -x & \rightarrow & y \end{array}$$

Unfortunately it is not possible to label the entire octahedron in such a way that this pattern of alternating arrows on a zigzag path applies to all of the four critical planes. We can only achieve it for two at a time.

So this configuration does not give us a good set of planes either.

It must be understood that this incompatibility is not one which can be overcome by merely deciding on a convention for which one should be used. Planes which are close to the corner in plane-space share many points in the neighbourhood of the origin with the true corner planes and we require true continuity at the junction. Otherwise the intersection will fragment of two close planes taken from different faces of the Gauss figure.

## 5.3 Extrusion along antidiagonals

The idea here is that the $xz$ 2D line system in the first quadrant should be used to make planes by shifting the origin so that it remains in the plane $x+y+z=0$, and so that the generator lines intersect successive points in a line in the $yz$ 2D line system.

This does not have symmetry between $x$ and $y$, because we accept that we cannot have extrusion in more than one direction. As described above, the line $y=z=0$ taken as a generator creates an extrusion from any $yz$ line used as a directrix, but sliding a different generator along the directrix $x=z=0$ does not give an extrusion, because the steps slide at 45 degrees in the $xy$ plane.

What happens when this idea is elaborated is that the rectangular facets which we saw in the extrusion case above still occur, but become sheared. This shearing gives the result that the intersection with a plane of constant $z$ is no longer a connected group of points.

## 5.4 Face centred cubic system

We saw in an earlier section a 2D system based on a triangular lattice instead of the rectangular one. This system has rather better symmetry properties than the rectangular ones.

We can put this pattern on each of the planes

$$\begin{aligned} +x+y+z &= 0 \\ -x+y+z &= 0 \\ +x-y+z &= 0 \\ +x+y-z &= 0 \end{aligned}$$

with the principal directions aligned with those lines of the set

$$\begin{aligned} \pm x \pm y &= z = 0 \\ \pm y \pm z &= x = 0 \\ \pm z \pm x &= y = 0 \end{aligned}$$

which the plane cuts.

Now consider the planes

$$\begin{aligned} +x+y+z &= 0 \\ +x+y-z &= 0 \end{aligned}$$

which intersect in the line

$$x + y = z = 0$$

and look at the 2D lines in the hexants not touching that line.

We can create a set of origin-based planes as the tensor product of these two sets of lines. If we then extend this set of planes by sliding the origin along the intersection line $x+y = z = 0$, we can see that the intersection of any two planes in this extended set is exactly the tensor product of two 2D segments, thus trivially proving the single intersection property.

This set of planes does not cover the entire Gauss-sphere. We need to take a total of six such sets to cover all possible normal directions. Each set covers a four sided region and its antipode, and these regions fit together into a partitioning of the sphere into a semi-regular polyhedron with 12 faces, which meet either 3 or 4 at each vertex.

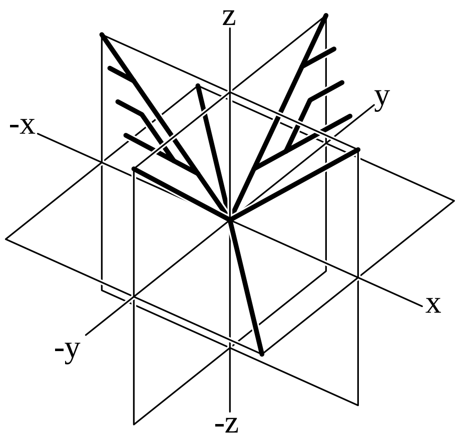


Figure 16: FCC vector sums.

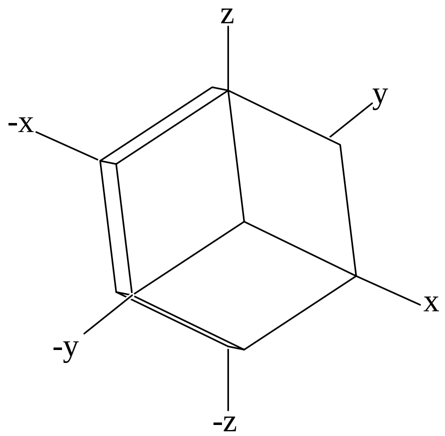


Figure 17: Gauss-figure for FCC system.

However, a problem with this elegant system is that the points contained in the planes defined do not fill a regular 3D cubic lattice in space. Only points of even parity ($x + y + z$ is even) are constructed by the vector sum construction. The points lie on a face-centred cubic (FCC) semiregular lattice rather than one a cubic one.

I did consider the possibility of grouping points by fours, deeming the four points in the group to be a single logical point, but this destroys the single connected intersection property.

I now suggest that it may be relatively easy to live with a face-centred cubic lattice of points. When new points are defined from external data they have to be rounded to a discrete system point anyway, and this process can perfectly well round them to an FCC point. From that stage onwards, all constructions are closed within the FCC point universe.

## 5.5 Compatibility of planes between different regions.

We need to show the single intersection property for pairs of planes from regions touching along an edge and for those from regions touching at a vertex. Also those of different orientation within one region.

### 5.5.1 Compatibility at region vertices

The Gauss-figure is divided into regions which meet either three or four at each vertex. Compatibility is perfect at each of these, because the corresponding planes are those which are exactly flat.

### 5.5.2 Theorem: Compatibility across region edges of the Gauss sphere

The interior of the edges is a different matter.

The patterns on the generating hexants can be identified by labelling the edges of the solid whose vertices are the bisectors of the coordinate axes. for example (1,1,0), (0,-1,1) etc. The faces of this solid are alternatiely squares and triangles.

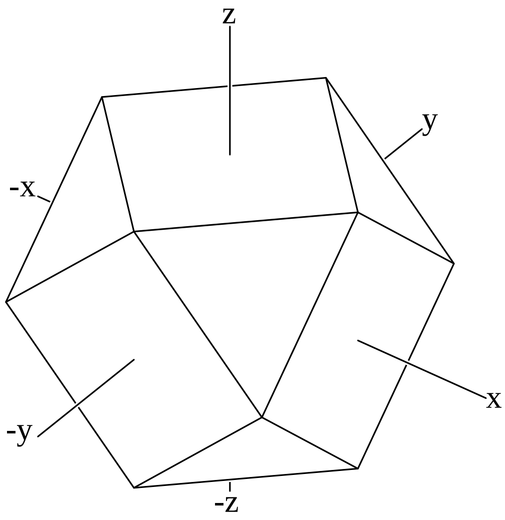


Figure 18: Labels for FCC system.

The only constraint is that antipodal labels must be in the same direction. This is not a problem.

Consider the plane whose normal points in the direction $[-\alpha, +\alpha, +\beta], \alpha < \beta$. It is generated by the vector sum of the line $[-1, -1, 0]$ and a line in the pattern between $[+1, -1, 0]$ and $[0, -1, +1]$.

It is also generated by the vector sum of the line $[+1, +1, 0]$ and a line in the pattern between $[+1, -1, 0]$ and $[+1, 0, +1]$.

The line $[+1, -1, 0]$ is straight and is common between the two planes. Because it is straight, the plane will be an extrusion. Compatibility between the two definitions of the plane thus demands simply that the arrows on the edges $[+1, -1, 0]$ to $[0, -1, +1]$ and $[+1, -1, 0]$ to $[+1, 0, +1]$ should either both point out of $[+1, -1, 0]$, or both point in.

This places a constraint on the labelling of the edges of a triangular face, which cannot be satisfied for all vertices of the triangle simultaneously. At least one vertex must have one arrow pointing out and another pointing in.

Thus this scheme cannot be made self-consistent either.

# 6 From Lines to Planes

The two previous approaches have started with points and then constructed lines and planes respectively. It is possible that there might be a better way, based on starting with lines, and constructing both points and planes in some dual way.

The full approach here would take the four-dimensional space of lines in 3-dimensions, and derive the three-dimensional spaces of points and of planes, probably as special linear complexes.

This looks complicated, and so an intermediate step is to look at the origin-based lines and planes, to see if we can set up some kind of system of lower dimensionality.

## 6.1 Origin-Based Lines and Planes

Origin-based lines and origin-based planes are both projective spaces of two dimensions, which we can visualise on a discretisation of the Gauss sphere.

We can either use the same polyhedron for both planes and lines (during this subsection we will omit the words 'origin-based' in the interest of readability), or else use dual polyhedra. The previous section gave hints of this, and dual polyhedra emerged from the two vector-sum methods considered.

However, we do not have to identify the point sets straight away, and this section avoids the question of what points lie on a line or on a plane until the relationships between lines and planes are clear.

If we use the same Gauss-figure for both lines and planes, each discrete point on it represents both a line and a plane, and the correspondence is clearly a perpendicularity relationship.

We can reasonably require that perpendicularity between two lines is a symmetric relationship.

### 6.1.1 The cube as a common Gauss-figure

The lowest order cube we can consider has points only at its vertices. Each line/plane is represented by a vertex and its antipode, and the set of perpendicular lines/planes is the set of all other vertices. This appears clean.

However, we wish to have more than four lines in our universe, and so we have to consider what happens when the discretisation is fairly dense.

Consider the case when the grid on each face is the obvious one (in the style of Figure 20(a)).

The lines perpendicular to a vertex line now form a zigzag belt around the cube, crossing every face from the midpoint of one edge to the midpoint of an adjacent edge. These are shown as hollow squares and triangles in figure 19, representing lines perpendicular to the lines represented by the solid square and solid triangle vertices.

Consider the belts of two adjacent vertices, $A$ (squares) and $B$ (triangles), on a face perpendicular to the edge between them. These belts, shown by the open squares and triangles are parallel and a distance apart $1/\sqrt{2}$ of the edge length. The number of points on the perpendicular diagonal (shown by the open circles) is half the number of points on an edge.

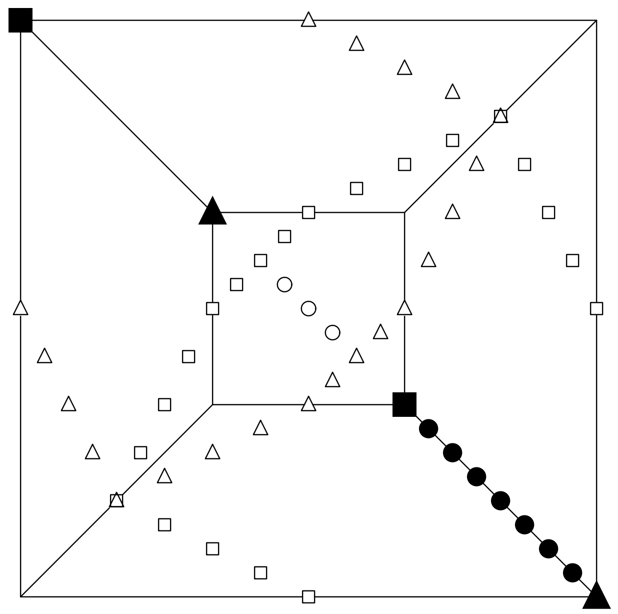

Figure 19: Not enough diagonal points.

Now consider what the belts corresponding to points (represented by solid circles) along the edge between $A$ and $B$ must do. They clearly ought to be parallel to those of $A$ and $B$. Near the diagonal they should also be disjoint, because if two belts hold a common point, then the belt of that common point will pass through both of the owners of those belts (symmetry of perpendicularity). Thus the belt of that common point will be pointing in exactly the wrong direction as it crosses the edge $AB$.

The final stage in the proof is that the points where the belts cross the diagonal cannot be disjoint, because there are only half as many diagonal points as there are edge points.

### 6.1.2 Other grids on a Gauss-cube

There is no prima facie reason why the grid of origin-based lines has to be the grid of figure 20(a). We can consider other possible grids on each face.

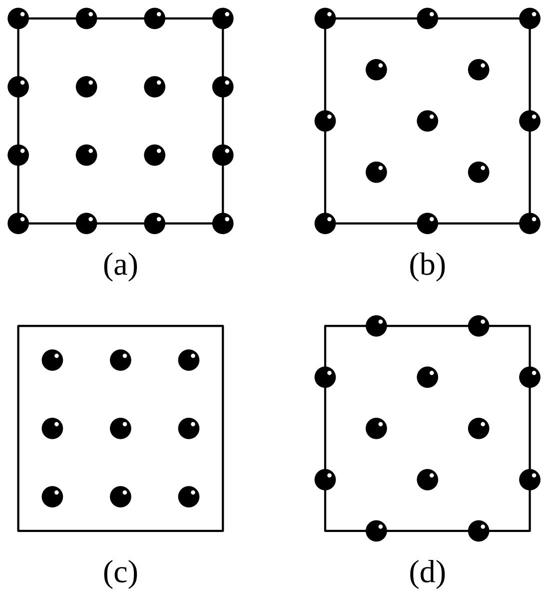


Figure 20: Possible grids on a square face.

However, using the styles of figure 20(c) and (d) heads to the question of how points closer to an edge are going to be constructed. If a doubling approach is used, they will need to be created as means of lines on different faces, thus opening the question of which antidiagonal direction to use for translation of origin-based lines into general lines.

The style of figure 20(b) does not have this problem. It does have a problem, probably not insuperable, that the set of lines perpendicular to a mid-face point has to zigzag. This breaks the symmetry, and so it is not possible to use symmetry of the cube to reduce the scale of the construction. Every line of every face has to have its perpendicular lines identified. This takes this possibility out of the scope of this paper.

#### 6.1.3 Layering problems

Once we consider these possibilities we can identify a further problem. The generation of points on the origin-based lines represented by points on the Gauss figure will involve the creation of nested figures, like onion skins, starting from the origin.

With a square grid we have the question of how the density is to be increased with increasing radius. The doubling procedure, which we would like to use, requires each point at the $i^{\mathrm{th}}$ doubling to be derived from just two points at the $i-1^{\mathrm{th}}$. Going direct from any one of these patterns to another in the same style has a problem with the definition of new points at the centroid of four old ones. Which diagonal to use ? This also effectively turns a rectangular based system into two triangles.

A way of avoiding this is to let the patterns alternate as the layers increase. If we create new points corresponding to the means of adjacent old points, the new pattern forms a grid at 45 degrees to the old. The patterns follow the transition diagram

$$a \rightarrow d \leftrightarrow c \leftarrow b$$

Each new point now has exactly two precursors. There are no 'straight' lines constructed, but this does not matter. The stable pair of patterns are $c$ and $d$ and so we do have the problem mentioned above of defining what happens at the edges of the face.

Unfortunately the linear density at each doubling then multiplies by a factor of only $\sqrt{2}$ instead of 2, and so there is severe reduction of resolution away from the origin. It also makes the idea of getting from origin-based lines to general lines by antidiagonal translation impossible.

#### 6.1.4 The octahedron as a common Gauss-figure

Using an octahedron as Gauss-figure appears to be more promising.

Lines perpendicular to the line of a vertex of the octahedron lie along those edges of the octahedron which do not meet the original vertex or its antipode.

Lines perpendicular to the line of a point on the octahedron edge correspond to a line across four triangular faces. The number of such lines is exactly right, and they can probably be constructed as a 2D Luby pattern, using some labelling of the priorities of the vertices of the octahedron. This is not the labelling described in section 4.2.5, but uses the same ideas.

Lines perpendicular to the line of a point in the interior of an octahedron face require the merging of lines in more than one orientation, and this document does not explore that question. There may be a small problem of how many planes cross a face and how many interior vertices an octahedron face contains, but it is not obvious that this is going to be serious.

### 6.1.5 Other Gauss-figures

We can consider a common Gauss-figure for lines and planes as being the Gauss-figure for lines with a perpendicularity relationship.

The discrete Gauss-figure does not need to be a regular polyhedron. It appears that life will be simpler if it is composed of triangular faces, but this is not necessarily essential.

The only requirement is that it should have central symmetry and be capable of having an antisymmetric labelling of its edges. This rules out only the tetrahedron of the regular polyhedra. The dodecahedron is probably ruled out by the difficulty of defining a useful discretisation of the pentagon, but there might be a very creative construction there.

## 6.2 General Lines and Planes

Looking at origin-based lines and planes only is really playing on a nursery slope. It may be possible that a complete solution can be constructed by taking the outer product of such definitions and translation of lines over an antidiagonal plane and planes in an antidiagonal direction. If such a solution can be found, then it will have the advantage of simplicity.

However, it seems likely that we need to consider a more general situation, in which the space of 'all' lines in 3D is discretised. This is a projective space of four dimensions, but for which we typically use in continuous geometry a representation (Plucker coordinates) as a (non-linear) subspace of a space of 6 dimensions.

The use of a polyhedral Gauss-figure in the discrete case might enable us to deal with coordinates in the faces of that figure, thus reducing our space to five or even four dimensions. Plucker coordinates can be regarded as being an outer product of a direction term and a displacement term - a factorisation which is equivalent to working with origin-based lines and antidiagonal translation. If we need to use discretisation of the full line space, it will be because that does not work, and so we probably need to find either some other factorisation or else work in the full dimension.

In the continuous case, we can define points and planes in terms of singular linear complexes (sets of lines satisfying a linear equation). The discrete equivalent does not have a linear equation, but we can set up sets of lines using some analogue of the doubling tree, thus giving simultaneously definitions of points and of planes by duality.

This duality probably means that the points will form a projective space, and so points will not turn out to lie on a finite regular grid. This does not matter, as we lost invariance under translation at the first move in this game. Indeed, it may have some advantages in that very distant points may still have a representation albeit of coarse resolution.

Exploring this direction of attack is clearly a major exercise in its own right, and is not covered further here.

# 7 Conclusions

This paper has shown that the extension of the elegant ideas for discrete geometry of 2D due to Luby is not a simple matter. Several obvious ideas and a few less obvious ones fail to fulfil the natural requirements.

Certain principles have emerged, however, which we expect to figure in any successful scheme.

– there should be a duality between points and planes, broken quantitatively by the fact that every pair of points has a join, but not every pair of planes meets within the region of interest.

Under this duality, line segments map into intersections of planes. These two kinds of object are not the same. Only 'complete' lines are self-dual.

– there is also a duality between origin-based lines and origin-based planes. Each of these needs to be discretised over some polyhedral analogue of the Gauss sphere.

Every origin-based line corresponds to a path around the Gauss figure for origin-based planes, and vice versa.

A further possibility, not explored here, would therefore be to start by defining a discrete self-dual space of lines and treat both points and planes as degenerate linear complexes in that space.

## References


[1] Jack E.Bresenham *An incremental algorithm for digital plotting* ACM National conference, Denver, Colorado. August 1963.

[2] Steven Fortune *Polyhedral modelling with multiprecision integer arithmetic.* CAD vol 29 no 2 pp123–133, 1997

[3] Daniel H.Greene and F.Frances Yao *Finite Resolution Computational Geometry* IEEE 1986

[4] M.L.P. van Lierop, C.W.A.M. van Overfeld and H.M.M.van de Wetering *Line Rasterization Algorithms That Satisfy the Subset Line Property* CVGIP, vol 41 pp210–228, 1988

[5] Michael G.Luby, *Grid geometries which preserve Properties of Euclidean Geometry: A Study of Graphics Line Drawing Algorithms.* pp397–432 in *Theoretical Foundations of Computer Graphcs and CAD* ed R A Earnshaw. NATO ASI series F vol 40. Springer Verlag 1988. ISBN 3-540-19506-8