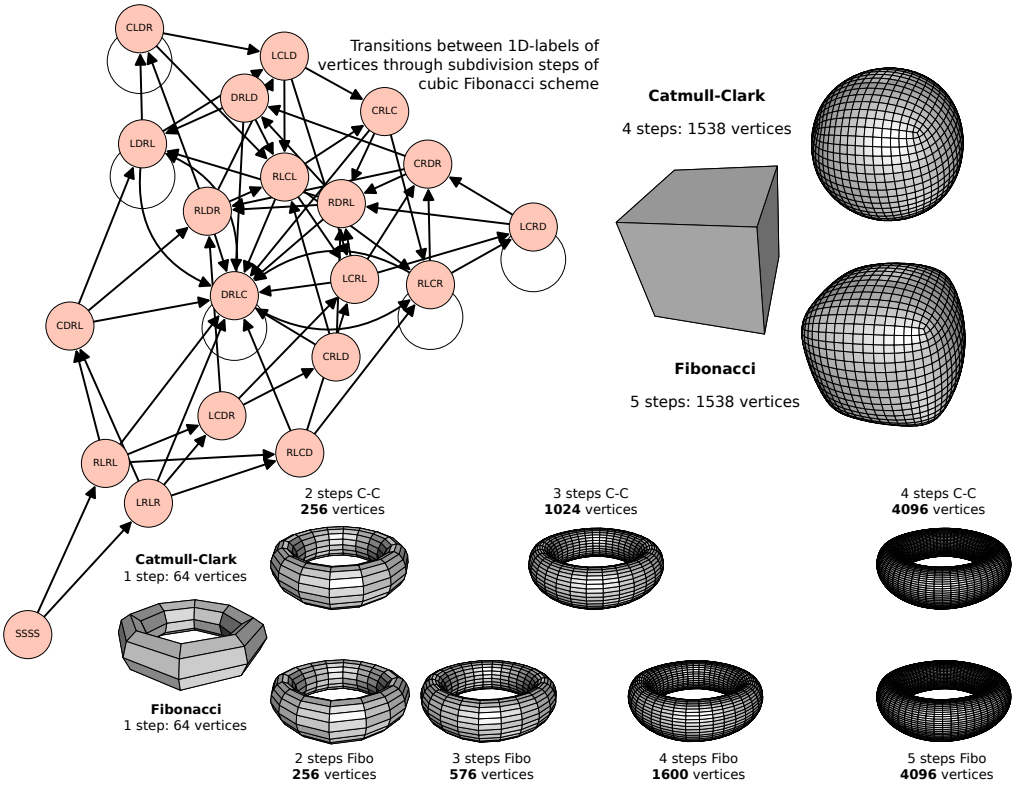
Graphical Abstract

Bivariate non-uniform subdivision schemes based on L-systems

Cédric Gérot, Ioannis Ivrissimtzis



Bivariate non-uniform subdivision schemes based on L-systems

Cédric Gérot[a,*], Ioannis Ivrissimtzis[b]

[a] *Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, Grenoble, 38000, France*
[b] *Department of Computer Science, Durham University, Durham, UK*

Abstract

L–systems have been used to describe non-uniform, univariate, subdivision schemes, which offer more flexible refinement processes than the uniform schemes, while at the same time are easier to analyse than the geometry driven non-uniform schemes. In this paper, we extend L–system based non-uniform subdivision to the bivariate setting. We study the properties that an L–system should have to be the suitable descriptor of a subdivision refinement process. We derive subdivision masks to construct the regular parts of the subdivision surface as cubic B-spline patches. Finally we describe stencils for the extraordinary vertices, which after a few steps become stationary, so that the scheme can be studied through simple eigenanalysis. The proposed method is illustrated through two new subdivision schemes, a Binary-Ternary, and a Fibonacci scheme with average refinement rate below two.

Keywords: bivariate non-uniform subdivision, L–systems, non-uniform B-spline refinement, eigenanalysis around EVs

1. Introduction

In this paper, we generalise the univariate L–system based non-uniform subdivision schemes introduced in [1] to the bivariate setting. While the generalisation of a univariate scheme to a regular quadrilateral mesh by taking tensor product is usually straightforward, its adaptation to extraordinary

*Corresponding author
Email address: cedric.gerot@gipsa-lab.fr (Cédric Gérot)

vertices (EV), that is, vertices with $n \neq 4$ incident edges, is more challenging. Specifically, the L–system based subdivision system should be able to handle connectivities without a natural tensor product structure, and still provide a consistent labelling and processing of the mesh edges, while it should also compute and update the positions of the regular vertices and the EVs with masks and stencils that will ensure nice convergence properties.

A Lindenmayer system, or L–system, is a collection of symbols and a set of rewriting rules, which are applied iteratively on an initial sequence of symbols, a *word*, thus generating a sequence of words. Context-sensitive L–systems have been used to specify uniform subdivision for curves [2] or surfaces [3]. There, the symbols label mesh primitives, vertices, edges, or faces, and each rule describes a substitution through a simple topological or geometric operation, such as face split, edge flip, vertex position update. Their aim was to develop a unified system of subdivision implementation. In contrast, here, in the context of extending [1], L–systems are not used to implement a known scheme, but to describe new hierarchies of parametric spaces for meshes. Each word in the sequence corresponds to a subdivision step, its symbols identifying the ratios in which the edges of the mesh split in the parametric space. Then, outside the EV neighbourhoods, the scheme is defined as the usual non–uniform B–spline subdivision with knots positioned on this grid hierarchy, while for the EV neighbourhoods special rules are derived.

With the use of context–free L–systems to describe non-uniform subdivision, one is able to study in a unified and systematic way a large family of bivariate non-uniform subdivision schemes. Compared to standard uniform stationary schemes, such as the Catmull-Clark [4], or the Doo-Sabin [5] schemes, the non-uniformity of the L–system based subdivision offers more control over the subdivision of the parametric space, which does not have to be uniform and can vary between subdivision steps. Moreover, when they are used as a prediction operator in a multiresolution analysis framework, or directly as a process to create various levels of detail of an object, usual subdivision schemes define a set of scales with an integer ratio between them, usually equal to 2 in each dimension. The design of subdivision schemes based on L–systems widens the set of possible transition speeds between scales to non integer ratios, allowing, in particular, more progressive transitions when the ratio is lower than 2. Furthermore, this control is offered within a rigid framework, which guarantees a small number of different ratios of interval lengths in all subdivision steps, meaning that the implementation requires a

finite, and usually small, number of different masks.

Previous works have generalised non-uniform B-splines subdivision to meshes with EVs [6, 7, 8], but they are based on binary subdivision which splits each parameter interval into two sub-intervals of equal length. Cashman *et al.* [8] introduce some exceptions serving two aims. First, to reduce high disparities in interval lengths, which may yield undesirable shapes. To do so, only the longest knot intervals are split in the first steps, until no interval has more than double the length of any other. Second, to get a uniform vicinity around any EV, allowing to spectral analysis tools on constant subdivision matrices. To do so, intervals incident to the EV may be split into two sub-intervals of unequal lengths. In both cases, the non-uniformity is not introduced in order to widen the set of possible average refinement rates (arities), as we do, but to deal with different knot interval lengths at the initial stage: their aim was to generalise NURBS to meshes with EV. We note that, in their work, a new computation of subdivision coefficients is required at each step, whereas in our case, masks and stencils are pre-computed once and for all. However, we also note that, for simplicity, here we start with the same parametric interval length for all edges of the initial mesh, leaving the generalisation to non-uniform initialisation for future work.

In contrast to other non-uniform or non-stationary schemes, our approach eases the proof of convergence and smoothness properties. In the univariate case, our work shares the same analysis arguments as other schemes based on controlled B–spline knot insertion [9, 10, 11], and the only remaining complication towards the generalisation to meshes is the analysis at the EVs. As in [8], we create a locally stationary subdivision around the EV, adding if necessary some extra requirements on the L–systems to achieve that, and we use known techniques of subdivision matrix eigenanalysis. In contrast, the analysis of univariate geometry-driven, or interpolatory subdivision schemes is challenging, even in the binary case [12, 13, 14, 15, 16, 17, 18]. Moreover, while the complicated analysis of such schemes can be extended to the regular bivariate case [19, 20], the generalisation to meshes with EVs is even more involved [21], and the same holds for non-stationary subdivision [22, 23].

The main contributions of the paper are:

1. We generalise the work in [1] on L–system based univariate non-uniform subdivision to the bivariate setting.
2. We study the relationship between the properties of the L–systems and the properties of the corresponding subdivision schemes.

3. We introduce two new L–system based subdivision schemes, a Fibonacci scheme with sub-binary average refinement rate, and a Binary-Ternary scheme which requires only a small number of masks.

The rest of the paper is organised as follows. In Section 2 we show how we can describe a subdivision scheme corresponding to an L–system. In Section 3 we introduce the Fibonacci and the Binary-Ternary schemes and in Section 4 we prove some of their properties. In Section 5 we briefly discuss some other potentially interesting subdivision schemes, and we then briefly conclude in Section 6.

2. L–systems and subdivision schemes

In Section 2.1, we recall some necessary machinery that has been developed in [1] for the study of univariate L–system based subdivision. Next, in Section 2.2 we extent their notions to the bivariate case.

2.1. L–systems and univariate subdivision

A Lindenmayer system, or L–system, is defined by a collection of *symbols*, or the alphabet, $\Sigma = \{A_i,\ i \in I\}$ where $I = \{1, \dots, |\Sigma|\}$, a set of *rewriting rules*, one for each symbol

$$A_i \to A_{i_1} \dots A_{i_r}, \quad i \in I \tag{1}$$

where $(i_1, \dots, i_r) \in I^r$, and an *axiom* defined as an initial word of symbols.

Starting from the axiom, the L-system produces a sequence of words by rewriting at each step every symbol according to (1), defining this way an interval subdivision descriptor. In particular, at subdivision step $j \in \mathbb{N}$, the *label* A_i corresponds to intervals of length $\ell_{A_i}^{(j)} = \ell_{A_i}/\rho^j$ for some $\rho > 0$, and according to (1), at step $j+1$, an interval labelled A_i splits into intervals labelled $A_{i_1}, \dots, A_{i_r}$ of length $\ell_{A_{i_1}}^{(j+1)}, \dots, \ell_{A_{i_r}}^{(j+1)}$. The ρ is a constant associated to the subdivision scheme, corresponding to its average refinement rate. For example, the L–system with a single symbol alphabet $\{A_1\}$, $\rho = 2$, and rule

$$A_1 \to A_1 A_1$$

describes uniform binary subdivision, as the interval lengths in the parametric space at step j will be $\ell_{A_1}^{(j)} = \ell_{A_1}/2^j$.

2.1.1. From interval subdivision to subdivision masks

Since the L–system describes subdivision in the parametric space, we next need to define explicit masks for the actual subdivision scheme. In the univariate case, a convenient approach is to see the interval ends in the parametric space as the knots of a B–spline of degree d.

Each sequence of $d+1$ consecutive knot intervals is the domain of a B–spline basis function, corresponding to one vertex of the control polygon. The subdivided control polygon corresponds to the same B–spline curve, but with refined knots corresponding to the intervals subdivided according to the L–system rules. When d is odd, each control point is associated with the middle knot of its basis function domain, and each interval label is also an edge label which can be used to lead the topological subdivision of the polygonal line: an edge is split in as many sub-edges as the number of symbols of the word at the right side of the rewriting rule of its label. The case of even d yields *dual* subdivisions, which are not addressed in this paper.

Regarding actual position of new vertices, it can be defined efficiently by *mask diffusion* with pre–recorded weights. Indeed, each new vertex can be written as a convex combination of vertices of the previous control polygonal line and the set of weights giving the contribution of the position of an old vertex to the new ones is called a *mask*. An old vertex is involved in the definition of a new one if their B–spline basis domains are nested. Let w be the $(d+1)$–word of the labels of the intervals that make up the domain associated with an old vertex. From knot-insertion and the blossom formulation, each weight is defined as a sum of products of fractions $f = \ell^{(j)}/\ell^{(j-1)}$ where $\ell^{(j)}$ is the distance between two particular knots at the j-th subdivision step [24]. Each distance $\ell^{(j)}$ writes as a sum of interval lengths $\ell^{(j)}_{A_i} = \ell_{A_i}/\rho^j$ with ℓ_{A_i} deduced from w only. As a consequence, each fraction f does not depend on the subdivision step j since it can be written as ρ multiplied by the ratio between two sums of ℓ_{A_i}. Thus, in contrast to general non–uniform B–spline based subdivision, it is sufficient to define a finite number of masks whatever the number of subdivision steps applied: one mask for each word w of $d+1$ consecutive symbols that can be produced by the rules of the L-system from the axiom. In our implementation of subdivision by mask diffusion, these words w are used as vertex labels, indicating which mask to apply. In contrast, the label w of a given new vertex does not encode enough information to know which stencil to use. Recall here that a *stencil* collects the contributions from all old vertices to compute the position of a new one.

As a consequence, here, implementation by mask diffusion is preferred to implementation by stencils.

2.1.2. Valid L-systems

Not every combination of L–system, ρ, and length ratios ℓ_{A_i}, allow such a subdivision scheme construction. From (1), ℓ_{A_i} should be non-negative real numbers such that

$$\ell_{A_i} = \sum_{l=1}^{r} (\ell_{A_{i_l}}/\rho)$$

and $\rho > 1$, in order to get a dense set of knots when the subdivision is applied repeatedly.

In order to know if such interval lengths and ratio compatible with the rules (1) exist, and to compute them, [1] defines the square matrix $\mathbf{M} = [m_{ij}]$ where m_{ij} is the number of A_j on the right side of the rewriting rule of A_i. It follows that any possible ρ and interval length ratios are eigenvalues and entries of the associated eigenvectors of $\mathbf{M}$, respectively. In particular, if $\mathbf{M}$ has a real eigenvalue $\rho > 1$ and an associated real and positive eigenvector, then the L-system can define an interval subdivision descriptor, whatever the axiom, and is called *valid* [1]. In the same paper, they also prove that the sequence of control polygons of a subdivision scheme constructed as above by a valid L-system, converges uniformly to the B–spline curve.

2.2. From univariate to bivariate subdivision

2.2.1. Topological subdivision

We choose the natural generalisation of the univariate subdivision to the bivariate setting by tensor–product. The bivariate scheme is applied on quad meshes with all their faces having their opposite edges labelled with the same symbol. As both pairs of opposite edges in a quad face split identically, the face can be partitioned into quadrilateral faces. While, other hierarchies of lattices are possible, like the hierarchy of Penrose lattices used in [25], they do not allow us to define B–spline based subdivision schemes. The construction of other known types of subdivision schemes, for example the ones based on polynomial interpolation are outside the scope of this paper.

This topological subdivision adapts directly to meshes with *extraordinary vertices (EVs)*, that is, vertices with a number of incident edges different from four. EVs appear on any closed quad mesh of genus other than one.

On the downside, it may make the labelling of the input mesh a difficult task if the axiom is made up of several symbols. Indeed, as the labels of the edges are symbols of words derived from the axiom ω, the initial mesh should be a union of sub-meshes with tensor-product connectivity, whose topologically parallel edges are labelled with the symbols of the axiom in a globally consistent way. We bypass the issue of a consistent initial labelling by choosing the axiom to be a single symbol assigned to all the edges of the initial mesh. In subsequent subdivision steps, since all new vertices are ordinary, the subdivided faces are sub-meshes with a regular tensor-product connectivity with ordinary vertices, except perhaps their corners which can be extraordinary. That is, in this paper, the non-uniformity of the subdivision concerns paths of ordinary vertices, apart perhaps from their ends.

Alternatively, one could allow the assignment of different labels to the edges of the initial mesh, either manually, as a designer choice, or computed from the geometry or topology of the initial mesh, depending, for example on some measures of mesh curvature. However this freedom would still be restrained by the fact that in each face, opposite edges must have the same label. Thus, a label given to an edge would also automatically be given to a complete strip of topologically parallel edges, as in [7, 8].

The use of L–systems to define the subdivision rules yields a particular constraint. The produced word at the right side of each rewriting rule is read from the left to the right and, in general, it is not palindromic, and as a consequence, each symbol actually corresponds to a directed edge. However, assigning a direction to each edge of the input mesh may be impractical, or even impossible if it is non-orientable.

Dealing with this constraint, in Section 4.1 we introduce the notion of *symmetric* L-systems. In particular, we define for each symbol A_i, the symbol $\sigma(A_i)$ which labels the head-to-tail edge such that their rewriting rules define a consistent topological subdivision. Besides, the single letter axiom $\omega = A_i$ chosen to label all initial edges satisfies $\sigma(A_i) = A_i$, i.e., input edges can be considered as non-directed.

Figure 1 illustrates these labellings in the first three steps of the Fibonacci

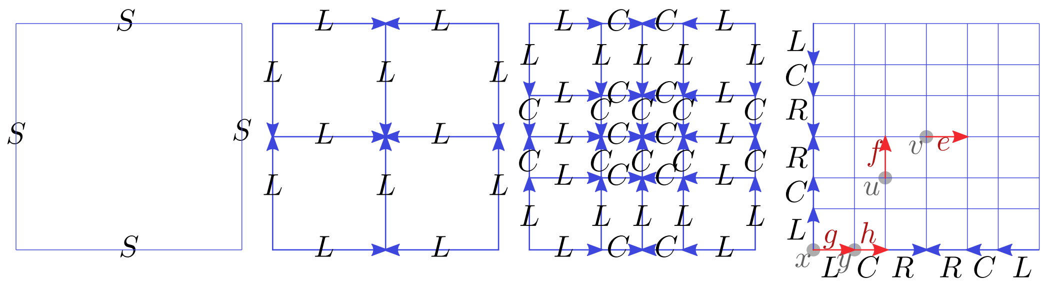


Figure 1: Edge labelling for the first three steps of the Fibonacci scheme. For readability, in the last step, edge directions and labels of are shown for the last line and the first column only. The vertices v and u are labelled $(e, CRLD, CRLD)$ and $(f, LCRL, RLDR)$, respectively. If x is an EV of valency n, it would be labelled $(LC)^n$ and y would be labelled $(h, RLCR, DRLC)$.

scheme of Section 3.2. The L–system is

$$\begin{cases} S & \rightarrow & LR \\ L & \rightarrow & LC \\ C & \rightarrow & R \\ D & \rightarrow & L \\ R & \rightarrow & DR \end{cases} \tag{2}$$

with $\omega = S = \sigma(S)$, $\sigma(L) = R$, $\sigma(R) = L$, $\sigma(C) = D$ and $\sigma(D) = C$.

2.2.2. Computing vertex positions by mask diffusion

After splitting the edges and faces according to the L–system rules, the positions of the new vertices are computed from the old vertices by mask diffusion as follows:

1. Store two meshes: the coarse old one, and the fine new one with the vertex positions initialised to zero. A link is stored from each old vertex to its new version (its *child*) in the fine mesh. The coarse mesh is read, whereas the new one is written.
2. Loop over the old ordinary vertices, applying on the new mesh a tensor–product regular mask associated with the labels of the edges in its old neighbourhood: each new vertex u in the neighbourhood of the child of an old vertex v is updated by

$$u \leftarrow u + \beta v, \tag{3}$$

where the scalar weight β is the mask entry which corresponds to the topological position of u relative to the child of v.

3. Loop over the old extraordinary vertices and apply the single extraordinary mask, with weights depending on the valency of the EV. Then, divide each new vertex in the neighbourhood of the EV by the sum of the contribution weights it has received from old vertices, making the linear combinations convex, and maintaining affine invariance.

Step 2 deserves two remarks. In order to know what tensor–product mask to apply from a given old vertex to its new neighbourhood, we label each vertex v with a triple (e, w, s), where e is an outgoing edge defining a local orientation, and w and s are two $(d+1)$-words made up of the symbols of the edges around v, see as Figure 1 for $d = 3$. The outgoing edge from the child of v which comes from the splitting of e, sets the orientation of the mask applied to the new mesh. Several labels are possible for a given vertex, depending on the choice of e: w and s adapt according to the *symmetry* of the L–system, as already mentioned and further elaborated in Section 4.1. For example, $(-e, \bar{\sigma}(w), \bar{\sigma}(s))$ is another possible label for v, where $\bar{\sigma}(w)$ is obtained by reading the word w from the right to the left, while changing each symbol A_i by its symmetric symbol $\sigma(A_i)$. In particular, it will be shown that the different possible tensor–product masks for a given vertex are consistent.

Furthermore, in order to update the new neighbourhood its topology must be regular: if any EV is involved, then it should belong to the boundary of the neighbourhood. If this condition is satisfied, the ordinary mask is said to be *contained* within the ordinary part of the new mesh. In Section 4.2.2 we propose sufficient conditions on the L–system to get such *contained* ordinary masks.

Step 3 also deserves some extra remarks. In order to use convergence arguments at an EV based on the eigenanalysis of a constant subdivision matrix [26], we ask the scheme to be *stationary* around an EV: the same subdivision rules apply whatever the step. In Section 4.2.1, we discuss properties of the L-system that make the subdivision scheme stationary around the EV, possibly after a few initial steps. In particular, between subdivision steps edge labels remain the same within a given neighbourhood, and an EV of valence n is labeled $(\xi)^n$, where ξ is the $(d+1)/2$-word made up of the edge symbols along any direction radiating from the EV. Since all the edges of the initial mesh are labelled with the same symbol, the ξ is always the same, whatever the EV, while $(\xi)^n$, which encodes the configuration around an EV, only depends on n. This stationary configuration also allows us to

define the label (e, w, s) of a vertex v when the EV is inside the edge paths w or s (which is possible in the old mesh, even if the mask remains *contained* within the ordinary part of the new mesh): the labels of edges beyond the EV are chosen to be the symmetric symbols of the edges radiating from the EV, as described in Figure 1 (caption) for vertex y.

Finally, since for a given valence n, the configuration around an EV is the same, whatever the EV and whatever the step, the sum of the normalising weights for the new vertices around an EV can be pre-computed together with the extraordinary mask entries.

3. Two new subdivision schemes

Before studying which properties of L-systems are required for getting bivariate subdivision schemes that can be applied to any 2-manifold quad mesh, we first present two examples of particular interest: a Binary-Ternary scheme which needs only a small number of different masks, and a Fibonacci scheme which, with an average refinement ratio lower than 2, produces a more progressive multiresolution mesh refinement than the usual binary schemes.

3.1. A Binary-Ternary subdivision scheme

This first proposed scheme corresponds to the L-system on the three symbol alphabet $\{L, C, R\}$ with the axiom $\omega = C$ used as label of all initial edges, and the rules

$$\begin{cases} L & \to & LC \\ C & \to & LCR \\ R & \to & CR \end{cases} \tag{4}$$

It defines a ratio $\rho = 1 + \sqrt{2} \approx 2.414$ and interval lengths $\ell_L = \ell_R = 1$ and $\ell_C = \rho - 1 \approx 1.414$. It is a mix between a binary and a ternary scheme: it splits the intervals into either two or three sub-intervals.

3.1.1. Masks for the ordinary rules

In the second step of the algorithm presented in Section 2.2.2, each old ordinary vertex v updates the neighbourhood of its child u as in Eq.(3). If the old vertex is labelled (e, w, s), then the 2D mask is computed as the tensor product of the 1D masks w and s. For example, if v is labelled $(e, LCRL, CRLC)$ and e, labelled R, splits into e' labelled C and e'' labelled R, then the child u is labelled $(e', CRCR, CRLC)$ and is updated according to the 1D-masks given in Table A.5, with $\beta = (12 - 8\sqrt{2}) \times (5 - 3\sqrt{2})$, where

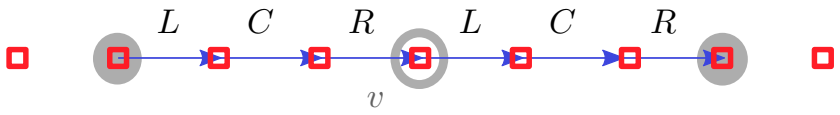


Figure 2: The 1D-mask associated with $CCCC$ applies from v to new vertices (red squares) beyond updated old neighbours (grey disks). Labels of new edges are given from $C \to LCR$.

the weight which applies to the child vertex is shown in bold. Similarly, the first neighbour of u in the direction of e' is labelled $(e'', RCRL, CRLC)$ and it is similarly updated with $\beta = (\sqrt{2}-1) \times (5 - 3\sqrt{2})$.

As a consequence, it is sufficient to store the entries of the 1D-masks as shown in Table A.5 which records all the ten 1D-masks that could be used. Figure A.8 shows the all the transitions between 1D-labels that can be reached from the $CCCC$ of the input mesh.

We further remark that, through symmetry, there are only five essentially different 1D-masks after the first step, following the association:

$$\begin{array}{lll} CLCR = \bar{\sigma}(LCRC) & CRCR = \bar{\sigma}(LCLC) & CRLC = \bar{\sigma}(CRLC) \\ LCRL = \bar{\sigma}(RLCR) & RCRL = \bar{\sigma}(RLCL) & \end{array}$$

3.1.2. Stencil implementation of the first subdivision step

We notice that the 9×9 ordinary mask associated with $(e, CCCC, CCCC)$ cannot be applied if a neighbouring vertex of an old vertex v is EV, as shown in Figure 2 along a 1D path of edges. To address this complication, for the first subdivision step only, we switch from an mask implementation to an implementation with the stencils illustrated in Figure 3 and the weights given below, coming from the 9×9 ordinary mask for valence $n = 4$. We have chosen an *ad hoc* generalisation to extraordinary valences.

$$\left\{ \begin{array}{rcccl} \gamma_1 & = & \frac{7-2\sqrt{2}}{6} \times \frac{7-2\sqrt{2}}{6} & = & \frac{57-28\sqrt{2}}{36} \\ \delta_1 & = & \frac{2\sqrt{2}-1}{12} \times \frac{7-2\sqrt{2}}{6} & = & \frac{16\sqrt{2}-15}{72} \\ \epsilon_1 & = & \frac{2\sqrt{2}-1}{12} \times \frac{2\sqrt{2}-1}{12} & = & \frac{9-4\sqrt{2}}{144} \end{array} \right. \tag{5}$$

$$\left\{ \begin{array}{rcccl} \gamma_2 & = & \frac{5-\sqrt{2}}{6} \times \frac{7-2\sqrt{2}}{6} & = & \frac{39-17\sqrt{2}}{36} \\ \delta_2 & = & \frac{3+\sqrt{2}}{12} \times \frac{7-2\sqrt{2}}{6} & = & \frac{17+\sqrt{2}}{72} \\ \epsilon_2 & = & \frac{5-\sqrt{2}}{6} \times \frac{2\sqrt{2}-1}{12} & = & \frac{11\sqrt{2}-9}{72} \\ \zeta_2 & = & \frac{3+\sqrt{2}}{12} \times \frac{2\sqrt{2}-1}{12} & = & \frac{11\sqrt{2}-9}{144} \\ \eta_2 & = & \frac{\sqrt{2}-1}{12} \times \frac{7-2\sqrt{2}}{6} & = & \frac{9\sqrt{2}-11}{72} \\ \theta_2 & = & \frac{\sqrt{2}-1}{12} \times \frac{2\sqrt{2}-1}{12} & = & \frac{5-3\sqrt{2}}{144} \end{array} \right. \tag{6}$$

$$
\begin{cases}
\gamma_3 &= \frac{5-\sqrt{2}}{6} \times \frac{5-\sqrt{2}}{6} &= \frac{27-10\sqrt{2}}{36} \\
\delta_3 &= \frac{3+\sqrt{2}}{12} \times \frac{5-\sqrt{2}}{6} &= \frac{13+2\sqrt{2}}{72} \\
\epsilon_3 &= \frac{3+\sqrt{2}}{12} \times \frac{3+\sqrt{2}}{12} &= \frac{11+6\sqrt{2}}{144} \\
\zeta_3 &= \frac{5-\sqrt{2}}{6} \times \frac{\sqrt{2}-1}{12} &= \frac{6\sqrt{2}-7}{72} \\
\eta_3 &= \frac{3+\sqrt{2}}{12} \times \frac{\sqrt{2}-1}{12} &= \frac{2\sqrt{2}-1}{144} \\
\theta_3 &= \frac{\sqrt{2}-1}{12} \times \frac{\sqrt{2}-1}{12} &= \frac{3-2\sqrt{2}}{144}
\end{cases}
\tag{7}
$$

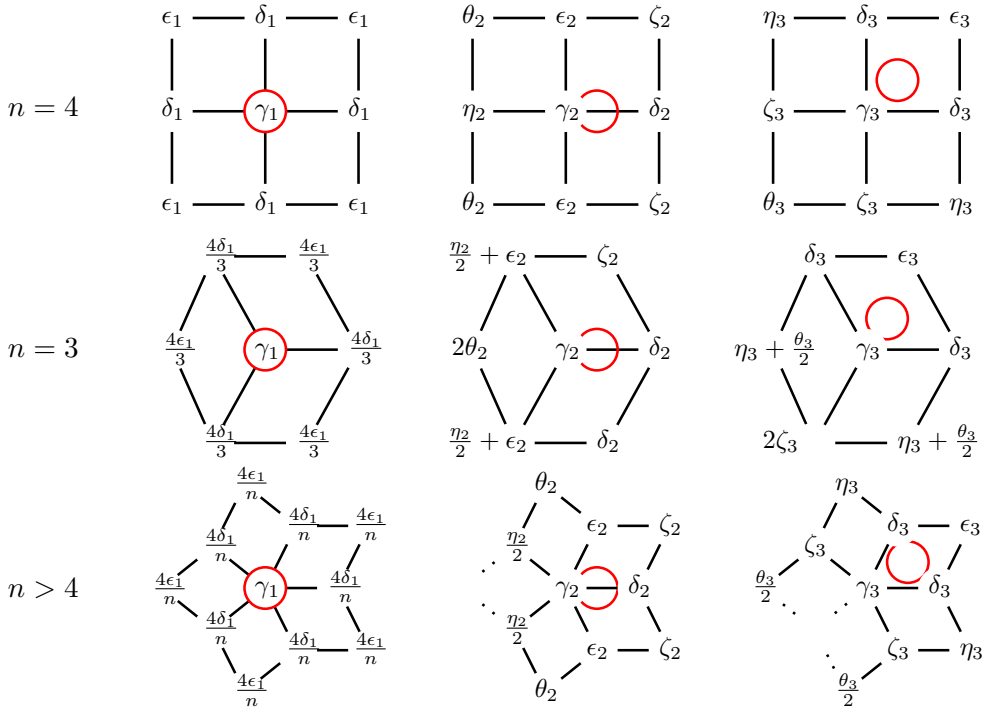


Figure 3: Stencils for the first step of the Binary-Ternary scheme. The red circles locate the new vertices. The weights γ_i apply on old vertices of valence n.

3.1.3. The extraordinary mask

From the second subdivision step on, each EV of valence n is labelled $(LC)^n$. The extraordinary mask used in the third step of the algorithm in Section 2.2.2, is built sector by sector from tensor products of the second half of the 1D ordinary mask associated with $CRLC$ with itself, except for the central entry α corresponding to the update of the child of the EV by

the EV itself. A proposed set of α values for various valences n are shown in Table 1. They were computed through eigenanalysis, see Section 4.2.3.

n	3	4	5	6	7	8	9
α	0.2801	0.5735	0.8964	1.2427	1.5996	1.9598	2.3193
n	10	11	12	13	14	20	25
α	2.6766	3.0309	3.3822	3.7305	4.0762	6.1082	7.7668

Table 1: The values of the central entry α of the extraordinary masks of the Binary-Ternary scheme for various EV valences.

Since the other mask entries have ordinary values, no normalisation is required for the ordinary new vertices around the child of the EV. However, the child of the EV must be normalised by division with the sum of the received contributions, which is equal to

$$\alpha + \frac{15\sqrt{2} - 21}{2}\, n. \tag{8}$$

3.2. A Fibonacci subdivision scheme

The second scheme is built on an L-system with the five-symbol alphabet $\{S, L, C, D, R\}$, the axiom $\omega = S$ used to label all initial edges, and the rules

$$\begin{cases} S & \to & LR \\ L & \to & LC \\ C & \to & R \\ D & \to & L \\ R & \to & DR \end{cases} \tag{9}$$

It defines a ratio equal to the golden ratio $\rho = \frac{1+\sqrt{5}}{2} \approx 1.618$, and interval lengths $\ell_L = \ell_R = 1$, $\ell_C = \ell_D = \rho - 1 \approx 0.618$, and $\ell_S = \frac{2}{\rho} \approx 0.828$. The scheme can be viewed as a generalisation to the bivariate setting of the univariate Fibonacci L-system in [1], defined with the rules $\{L \to CL;\ C \to L\}$, with the two letters of the word produced from L swapped into LC in order get a stationary scheme near the EV. Ratio and interval lengths are the same.

Remark 3.1. *A perhaps more straightforward generalisation of the univariate Fibonacci scheme corresponds to a scheme that splits the initial interval into two sub-intervals, which are then subdivided independently following the*

rules of two independent univariate Fibonacci L-systems, but starting from each end of the initial interval, in a mirror way. The rewriting rules of this other L-system are

$$\begin{cases} S & \to & LR \\ L & \to & LC \\ C & \to & L \\ D & \to & R \\ R & \to & DR \end{cases} \tag{10}$$

and its implementation requires 17 ordinary 1D-masks, that is 2 more than the chosen scheme with the rewriting rules of (9).

3.2.1. Masks for the ordinary rules

As per the Binary-Ternary scheme, since the scheme is based on degree 3 B-splines, we use one ordinary 1D mask for each 4-word that can be produced from $SSSS$. Figure A.9 shows these possible words, some of them being only transitional:

$SSSS$ appears at the initial step only;

$RLRL$ and $LRLR$ appear at the 2nd step only;

$CDRL = \bar{\sigma}(RLCD)$ and $LCDR$ appear at the 3rd step only;

$CRLD$ appears at the 4th step only.

Table A.6 gives the entries of these masks.

Similarly to the Binary-Ternary scheme, some masks are the same up to a symmetry:

$$\begin{array}{lll} CDRL = \bar{\sigma}(RLCD) & CLDR = \bar{\sigma}(LCRD) & CRDR = \bar{\sigma}(LCLD) \\ CRLC = \bar{\sigma}(DRLD) & CRLD = \bar{\sigma}(CRLD) & DRLC = \bar{\sigma}(DRLC) \\ LCDR = \bar{\sigma}(LCDR) & LCRL = \bar{\sigma}(RLDR) & LDRL = \bar{\sigma}(RLCR) \\ LRLR = \bar{\sigma}(LRLR) & RDRL = \bar{\sigma}(RLCL) & RLRL = \bar{\sigma}(RLRL) \end{array}$$

and thus, only 13 different 1D-masks have to be stored.

3.2.2. The extraordinary mask

In contrast to the Binary-Ternary scheme, ordinary masks can be applied on all regular vertices from the very first step. However, an extraordinary mask still has to be defined, again as one tensor-product of half ordinary

1D-masks for each sector. Besides, the label of the EV could change after the first step, but not for many subsequent subdivision steps. Specifically, the labels $(\xi)^n$ of the EV and (e, w, s) of any of its first neighbours satisfy

at the first step, $\xi = SS$ and $w = SSSS$;

at the second step, $\xi = LR$ and $w = RLRL$;

at the third step, $\xi = LC$ and $w = RLCD$;

at the fourth step and beyond, $\xi = LC$ and $w = RLCR$.

As a consequence, each sector-mask is defined by the tensor product of the second half of the $SSSS$ mask with itself for the first step; the $LRLR$ mask for the second step; and the $DRLC$ mask for the third step and beyond. As with the Binary-Ternary scheme, no normalisation is needed for any new vertex in the neighbourhood of the child of the EV, except for the child of the EV itself. Regarding the central entry of the mask to be applied to the child of the EV, we chose to keep the same value α for every step (see Table 2), even if it is computed from the eigenanalysis of the local subdivision matrix corresponding to the rules from the fourth step and beyond only. As a consequence, the child of the EV should be divided by different normalisation values, depending on the subdivision step:

$$\begin{array}{ll} \alpha + \frac{7}{64}\, n & \text{at the first step,} \\ \alpha + \frac{7\sqrt{5}-9}{72}\, n & \text{at the second step,} \\ \alpha + \frac{6\sqrt{5}-13}{4}\, n & \text{at the third step and beyond.} \end{array} \tag{11}$$

n	3	4	5	6	7	8	9
α	0.2084	0.5835	0.9959	1.4440	1.9105	2.3843	2.8589
n	10	11	12	13	14	20	25
α	3.3314	3.8003	4.2651	4.7258	5.1826	7.8604	10.0380

Table 2: The values of the central entry α of the extraordinary masks of the Fibonacci scheme.

3.3. Examples

In this section, we show visualisations from the two proposed schemes, and the Catmull-Clark scheme [4] for comparison. Our aim is not the demonstration of any superior behaviour of the proposed schemes over the original Catmull-Clark, bur rather to highlight their similarities. Indeed, they can be seen as belonging to the same family of schemes, differing in their refinement ratios: ~ 1.618, 2, and ~ 2.414.

We also note that the proposed schemes have been implemented in order to illustrate what is possible within L–system based subdivision. They have not been optimised, regarding the transitional first steps, which could have a significant "low frequency" effect on the final surface [27], or the tuning of the extraordinary mask, which would require to consider more degrees of freedom than just the central entry α in order to get bounded curvature [28]. Such a study, including a formal proof of C^1-continuity of the limit surface at an EV [26, 29], is outside the scope of this paper.

Figure 4 shows control polygons of characteristic maps [30], corresponding to the eigenvector of the local subdivision matrix associated with its subdominant eigenvalue [26]. We notice that while they are not identical, they are very similar, indicating (but not proving) that at the limit surface the two proposed schemes share with Catmull-Clark the property of C^1-continuity around an EV.

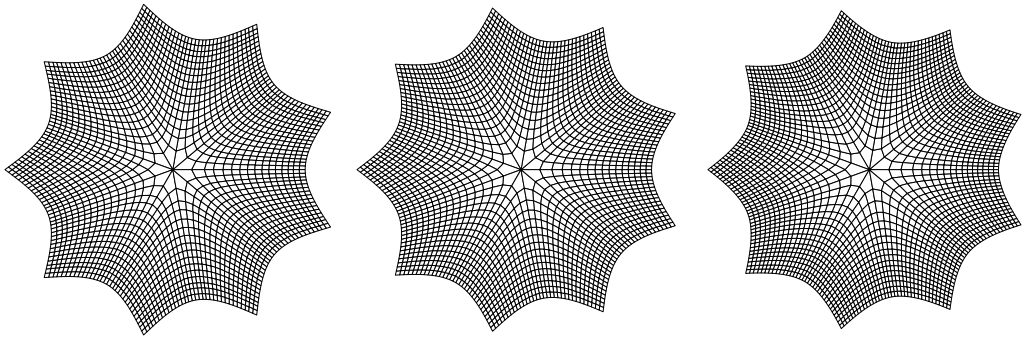


Figure 4: Characteristic map approximations.

Figures 5 and 6 show an initial quad mesh subdivided with: 3 steps of Binary-Ternary scheme; 4 steps of Catmull-Clark scheme; 5 steps of the Fibonacci. The number of steps was chosen in order to create meshes of similar size. Indeed, one initial edge is subdivided into:

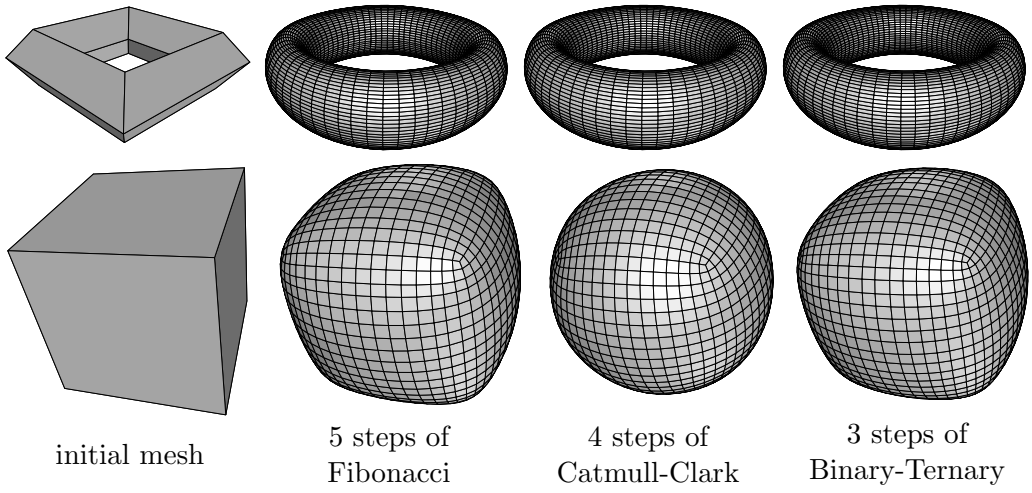


Figure 5: **Top row:** A mesh with only ordinary vertices shows that L–system based subdivision achieves non-rational refinement rates without deviating too much from uniformity. **Bottom row:** The initial mesh is a cube with only 3-valent extraordinary vertices.

17 sub-intervals after 3 steps of Binary-Ternary;

16 sub-intervals after 4 steps of original Catmull-Clark scheme;

16 sub-intervals after 5 steps of bivariate Fibonacci scheme.

In addition, the two bottom rows in Figure 6 show meshes after more subdivision steps, consisting of between two and three millions vertices. Their behaviour is illustrated with reflection lines renderings, and mean curvature colourmaps. All meshes were rendered with Geomview except these two renderings which were produced with Meshlab [31]. The mean curvature was computed with Algebraic Point Set Surfaces [32], with the colours going from blue for negative curvature, to red for positive, and the green corresponding to zero curvature. We notice that as with similar tunings of the Catmull-Clark scheme based on the eigenanalysis of its subdivision matrix, the reflection lines show an improvement of the surface near the EV while the curvature indicates that is has been flattened [28].

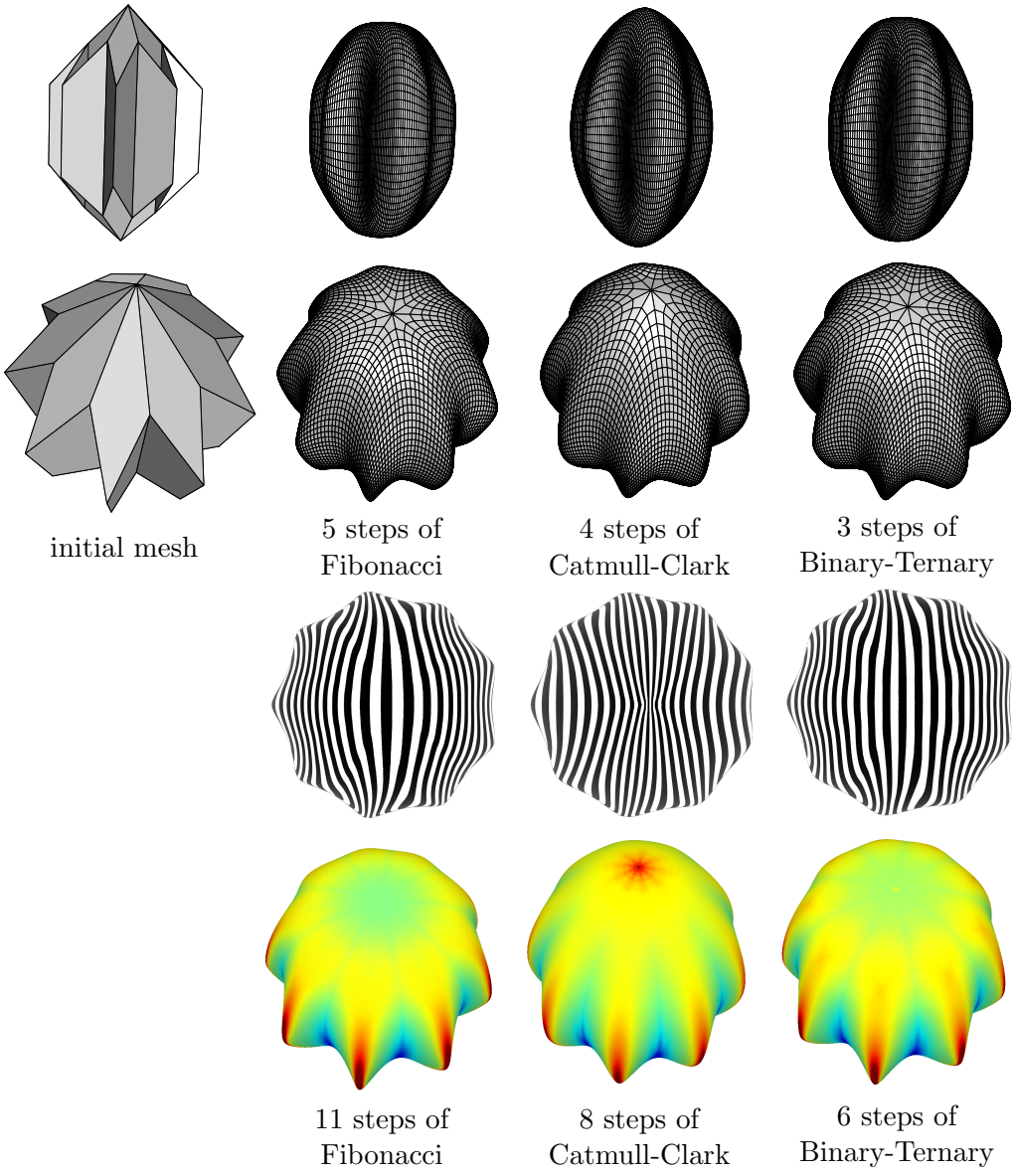


Figure 6: A mesh with 9-valent and 3-valent extraordinary vertices surrounded by ordinary ones. The top two rows show different views of the same meshes. The meshes in the two bottom rows have been further subdivided. The third row shows reflection lines from a top view. The fourth row shows mean curvature colourmaps: blue negative, red positive, and green zero.

4. Theoretical analysis of the proposed schemes

In this section we derive the properties an L-system should satisfy to give subdivision schemes that are suitable for general 2-manifold meshes, and verify that both proposed schemes are defined from such L-systems.

4.1. Symmetric L-systems

Since we aim at constructing a subdivision scheme which applies uniformly on any 2-manifold mesh, a consistent subdivision should be applied whatever the direction of reading of consecutive edge labels. To do so, we introduce the notion of *symmetric* L-system.

4.1.1. Definitions

To motivate our definitions, assume that the symbols of the L-system are labels of either directed or non–directed edges. A directed edge can be either onward or counter to the direction of reading the word, however, the initial edge and its subdivided sequences may be read in any direction. As an example, consider the following sequence of directed and non–directed edges:

$$\rightarrow\rightarrow - \leftarrow$$

labelled with the symbols of the Binary-Ternary scheme: C for a non–directed edge, L if it is directed onward with the direction of reading, R otherwise. Reading from left to right, the word of labels is $w = LLCR$; reading from right to left, it is $\bar{\sigma}(w) := LCRR$.

An involution (a bijection that is its own inverse) $\sigma()$ over the alphabet of the L-system partitions it into two subsets:

- $\mathcal{S}$, on which $\sigma()$ is the identity, is the set of *symmetric symbols* labelling non–directed edges: $\{C\}$ in the above example;
- $\mathcal{U}$, on which $\sigma()$ is of order 2, is the set of *unsymmetric symbols* labelling directed edges: the two unsymmetric symbols of a 2–cycle are called *twins*: $\{L, R\}$ in the above example.

An involution σ over the alphabet Σ is recursively extended into a *mirror function* $\bar{\sigma}$ defined over the set W of all words, by

$$\begin{cases} \bar{\sigma}(A) = \sigma(A) & \text{for any letter } A \in \Sigma, \\ \bar{\sigma}(ww') = \bar{\sigma}(w')\bar{\sigma}(w) & \text{for any words } w, w' \in W. \end{cases} \tag{12}$$

The mirror function $\bar{\sigma}$ is an involution over W, transforming each word w into $\bar{\sigma}(w)$ by rewriting it from right to the left, and changing each unsymmetric symbol by its twin. A word w is said to be *symmetric* if $w = \bar{\sigma}(w)$.

Definition 4.1. *Let $\pi(w)$ denote the word w rewritten with the rules of the L–system. An L-system is symmetric if its axiom ω is symmetric, and if there is an involution σ over its alphabet which, extended to $\bar{\sigma}$, satisfies:*

$$\pi(\bar{\sigma}(w)) = \bar{\sigma}(\pi(w)), \quad \forall w \in W. \tag{13}$$

4.1.2. Properties

The following proposition states that it suffices establishing symmetry on the axiom and the rewriting rules over the alphabet.

Proposition 4.2. *A necessary and sufficient condition for an L-system to be symmetric is that its axiom ω is symmetric and there is an involution σ over its alphabet Σ such that,*

$$\pi(\sigma(A)) = \bar{\sigma}(\pi(A)), \quad \forall A \in \Sigma. \tag{14}$$

Proof. ($\Rightarrow$) Let us consider a symmetric L-system with an involution σ which, extended to $\bar{\sigma}$, satisfies (13). In particular, if w is any single letter of the alphabet Σ, then we get Eq (14).

($\Leftarrow$) Let us consider an L-system with an involution σ over its alphabet which satisfies (13). For any word $w = A_1 A_2 \dots A_n \in W$,

$$\begin{aligned}
\bar{\sigma}(w) &= \sigma(A_n) \dots \sigma(A_2)\ \sigma(A_1) && (15)\\
\pi(\bar{\sigma}(w)) &= \pi(\sigma(A_n)) \dots \pi(\sigma(A_2))\ \pi(\sigma(A_1)) && (16)\\
&= \bar{\sigma}(\pi(A_n)) \dots \bar{\sigma}(\pi(A_2))\ \bar{\sigma}(\pi(A_1)) && (17)\\
&= \bar{\sigma}(\pi(A_1)\ \pi(A_2) \dots \pi(A_n)) && (18)\\
&= \bar{\sigma}(\pi(w)). && (19)
\end{aligned}$$

□

The Binary-Ternary and the Fibonacci L-systems are symmetric. For example, for the Binary-Ternary scheme the axiom C is symmetric, and

$$\begin{cases}
\pi(\sigma(L)) = \pi(R) = CR = \bar{\sigma}(LC) = \bar{\sigma}(\pi(L)) \\
\pi(\sigma(C)) = \pi(C) = LCR = \bar{\sigma}(LCR) = \bar{\sigma}(\pi(C)) \\
\pi(\sigma(R)) = \pi(L) = LC = \bar{\sigma}(CR) = \bar{\sigma}(\pi(R))
\end{cases} \tag{20}$$

Corollary 4.3. *The rewriting rules of a symmetric L-system can be split into two subsets:*

- *rewriting rules of symmetric symbols C, which have the form*

$$C \to w \tag{21}$$

  with $\bar{\sigma}(w) = w$;

- *rewriting rules of pairs of twin symbols (L, R), which have the form*

$$\begin{cases} L & \to & w \\ R & \to & \bar{\sigma}(w) \end{cases} \tag{22}$$

Corollary 4.4. *If the L–system is symmetric, then the axiom and any subsequent word produced by the rewriting rules are symmetric.*

Proof. By definition the axiom $\omega = \pi^0(\omega)$ is symmetric. To complete the inductive proof, let us assume that $\pi^n(\omega) = A_1 A_2 \dots A_p$ is symmetric:

$$\begin{aligned} \pi^n(\omega) &= \bar{\sigma}(\pi^n(\omega)) && (23)\\ \pi^n(\omega) &= \sigma(A_p) \dots \sigma(A_2)\ \sigma(A_1) && (24)\\ \pi^{n+1}(\omega) &= \pi(\sigma(A_p)) \dots \pi(\sigma(A_2))\ \pi(\sigma(A_1)). && (25) \end{aligned}$$

Since the L-system is symmetric,

$$\begin{aligned} \pi^{n+1}(\omega) &= \bar{\sigma}(\pi(A_p)) \dots \bar{\sigma}(\pi(A_2))\ \bar{\sigma}(\pi(A_1)) && (26)\\ \bar{\sigma}(\pi^{n+1}(\omega)) &= \bar{\sigma}(\bar{\sigma}(\pi(A_1)))\ \bar{\sigma}(\bar{\sigma}(\pi(A_2))) \dots \bar{\sigma}(\bar{\sigma}(\pi(A_p))) && (27)\\ \bar{\sigma}(\pi^{n+1}(\omega)) &= \pi(A_1)\ \pi(A_2) \dots \pi(A_n) = \pi^{n+1}(\omega). && (28) \end{aligned}$$

□

Remark 4.5. *The involution σ associated with a symmetric L-system is not necessarily unique.*

For example, the L-system

$$\begin{cases} L & \to & ACB \\ R & \to & BCA \\ A & \to & LR \\ B & \to & RL \\ C & \to & C \end{cases} \tag{29}$$

is symmetric either with $\mathcal{S} = \{A, B, C\}$ and $\mathcal{U} = \{L, R\}$ (with $R = \sigma(L)$), or with $\mathcal{S} = \{L, R, C\}$ and $\mathcal{U} = \{A, B\}$ (with $B = \sigma(A)$). However, the L-system

$$\begin{cases} C & \to & C \\ L & \to & LR \\ R & \to & LR \end{cases} \tag{30}$$

is symmetric if and only if $\mathcal{S} = \{C\}$ and $\mathcal{U} = \{L, R\}$.

4.2. L-system adapted to extraordinary vertices

When a mesh contains extraordinary vertices, one would wish to handle them with as few extraordinary subdivision rules as possible, which translates into two properties that are studied here:

- a constant *local subdivision matrix* at the EV, Section 4.2.1;
- ordinary masks that are *contained* within the ordinary part of the new mesh, such that if we consider the set of new vertices updated by an old ordinary vertex, they are all ordinary except possibly for the mask's boundary which could contain some EVs, Section 4.2.2.

4.2.1. A stationary scheme around any EV

As already mentioned in Section 2, the set of old control points involved in the definition of a new one corresponds to the set of $(d+1)$ consecutive old intervals that include the $(d+1)$ intervals of the B-spline domain associated with the given new control point.

We have already introduced the *mask*, as the collection of all the non-zero contributions of one old vertex to the positions of the new vertices, and the *stencil* which collects the contributions from all old vertices to the computation of the position of a new one. The subdivision matrix encodes the convex combinations defining the new vertices around an EV from the old ones: each column contains the entries of the mask associated with an old vertex, and each row contains the entries of one stencil.

Let us organise the vertices into *topological rings*. The 0-ring contains the EV only, and the r-ring $r > 0$ is the set of vertices that are neighbours of a vertex of the $(r-1)$-ring without belonging to any s-ring, $s < r$. In the univariate case, each r-ring, $r > 0$ is a pair of vertices, one on each side of the EV.

If we sort vertices by topological rings, then, because the combinations are local and at least one interval splits, there is a minimal column index N such that all columns which contains at least one non-zero entry on or up the diagonal entry have an index less or equal to N. The upper-left $N \times N$ block is the *local subdivision matrix* and shares the non-zero eigenvalues with the infinite subdivision matrix. For simplicity, this upper-left block will be called *subdivision matrix* from now on. If this matrix remains unchanged through subdivisions, then the scheme is *stationary*. We are looking for that property, at least after a few steps.

Lemma 4.6. *If $L \to Lw$, with $(L, w) \in \Sigma \times W$, is one of the rewriting rules of the L-system then, for all $n > 0$, $\pi^n(L) = L\pi^0(w)\pi(w)\dots\pi^{n-1}(w)$, where $\pi^0(w) := w, \forall w \in W$.*

Proposition 4.7. *If the label L of the edges incident to the EV has rewriting rule $L \to Lw$, with w an m-word, $m \geq 1$, then, after a few steps, the local subdivision matrix remains unchanged through subdivision steps.*

Proof. The local subdivision matrix entries depend on the relative location of old and new knots around the central knot associated with the EV, which are defined, on each side, by the P first symbols of $\pi^n(L)$ where P is a finite number independent from n. The result is then a direct consequence of Lemma 4.6. □

As a consequence, a sufficient condition to get a stationary subdivision scheme near the EV is that, possibly after a few steps, the word produced by the rewriting rules of the L-system from the axiom starts with a symbol L satisfying Lemma 4.6. For both Binary-Ternary and Fibonacci schemes, this is true after one step since, in the first case, $\pi(C) = LCR$ and $L \to LC$ while in the second case, $\pi(S) = LR$ and $L \to LC$.

4.2.2. Contained ordinary masks

In implementation in Section 2, each ordinary mask should not contain EVs, apart perhaps from its boundary: it should remain *contained* within the ordinary part of the mesh. From a stencil perspective, that property translates into the similar requirement that each EV stays on the border of any stencil which might include it, except of course of its own stencil.

We study containment in the 1D setting, obtaining a necessary and sufficient condition in the form of Proposition 4.8, which translates to a sufficient conditions in the 2D setting.

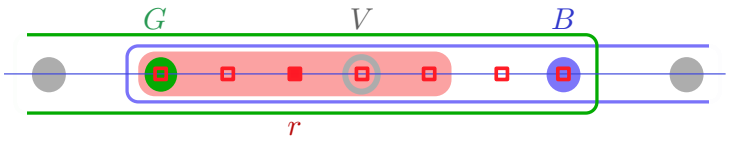


Figure 7: Vicinity of the central vertex V after one subdivision step, in the 1D setting. The domains of the degree 3 B-spline basis functions associated with the new vertex r (red) and the old vertices G (green) and B (blue), show that r is defined as a combination of three old vertices, of which, G and B which are in different sides of V. This happens because the intervals $[VB]$ split into 3 rather than at most 2 sub-intervals.

Thus, we consider a 1D bi-infinite sequence of intervals, starting from a vertex and spreading on both sides, all labelled with the 1-word symmetric axiom of the L-system. Each $d+1$ consecutive intervals are the domain of a degree d B–spline basis function associated with one control vertex of the polygonal line. The control vertex associated with the domain centred on the central knot of this 1D bi-infinite sequence corresponds to the EV of interest in the 2D setting. Figure 7 shows an 1D example where the containment property does not hold.

Proposition 4.8. *If the L-system is valid, then the stencil for the central vertex is the only one that involves old vertices from both sides of the central vertex if and only if among the* $\frac{d+1}{2} - 1$ *first intervals (at each side of the central vertex), none splits except one which splits into at most two sub-intervals.*

In particular, if $d = 3$*, then the label* L *of the edges incident to the central vertex should follow the rewriting rule* $L \to X$ *or* $L \to YZ$ *with* X*,* Y *and* Z *being symbols, but only* Y *and* Z *can possibly be equal to* L*.*

Proof. If the system is valid, no entry of the valid eigenvector $\mathbf{a}$ which defines the lengths of knot interval (up to a multiplication by $1/\rho(\mathbf{M})^n$) is equal to zero, and thus there is no multiple knots. Then, since an old vertex is involved in the stencil of a new one if the domain of the degree d basis function which is associated with it, contains the domain associated with the new vertex, $\frac{d+1}{2} - 1$ first old intervals at each side of the central vertex must not contain $\frac{d+1}{2} + 1$ new intervals, as illustrated in Figure 7 for $d = 3$.

Finally, if $L \to L$ then the interval never splits, which is not compatible with a valid L-system since then either 1 would be an eigenvalue of $\mathbf{M}$ or the entry of the valid eigenvector associated with L would be zero. □

By having the rule $L \to LC$ in both schemes, we satisfy both Proposition 4.7 and 4.8, getting a constant *local subdivision matrix* at the EV, and *contained* ordinary masks, but only after one subdivision step as L is not the axiom, but the first symbol of the word produced from it. More precisely, since the axiom of the Binary-Ternary L–system rewrites as $C \to LCR$, the ordinary masks are not contained at the first step and special stencils have to be defined. From the second step and onward however, the chosen rules result to a constant subdivision matrix. In contrast, the axiom of the Fibonacci L-system rewrites as $S \to LR$, yielding contained ordinary masks from the very first step. But, on the downside, we have to wait for the fourth subdivision step to get a constant subdivision matrix.

These extraordinary first steps would deserve a proper study as they could have a significant "low frequency" effect on the final surface [27], but such as study is outside the scope of this paper.

4.2.3. Eigenanalysis of the Binary-Ternary and the Fibonacci scheme

In both schemes, after perhaps a small number of steps, ordinary masks can be applied from every old ordinary vertex, while the extraordinary mask is built sector by sector from tensor-products of half 1D ordinary masks, leaving just the central entry α to be computed. Note that the new EV is then divided by a normalising value to become a convex combination of old vertices.

As a consequence, after perhaps a few steps the local subdivision matrix becomes stationary, with all lines corresponding to ordinary stencils, except the first one whose entries depend on α. The value of this parameter can be optimised by analysing the main eigenvalues and eigenvectors of the local subdivision matrix [26], aiming at achieving the best possible convergence behaviour, ideally towards a C^2-surface. However with a single free variable, this might not be possible.

Let λ_i and μ_i be the main and the second main eigenvalues from Fourier index i, respectively. As all stencils represent convex combinations, we have that always $\lambda_0 = 1$. As α can only affect eigenvalues related to Fourier index 0, thus, it can be chosen to satisfy $\mu_0 < \lambda_1$, or other relations involving μ_0.

In order to get a convergent subdivision scheme at the EV, λ_1 should be the global subdominant eigenvalue of the local subdivision matrix, and as a consequence, α must be chosen such that $\mu_0 < \lambda_1$. Then, two further properties could be targeted:

- $\mu_0 = \lambda_1^2$, and then, if $\lambda_2 \leq \lambda_1^2$ we get bounded curvature at the EV;

- $\mu_0 = \lambda_2$, and then we get the so-called *flexibility* property [33].

Ideally, bounded curvature and flexibility could be satisfied simultaneously, but not with the proposed schemes since $\lambda_2 > \lambda_1^2$ (except for $n = 4$) and thus, even bounded curvature alone is not possible. Of course, more variables could be added to the EV-mask to try and get that property, but this is outside the scope of this paper.

The values of α given in Section 3 for the proposed new schemes have been computed to achieve the first condition $\mu_0 = \lambda_1^2$, hoping for good surface behaviour at least when the shape is cup-like. Tables 3 and 4 give the computed subdominant eigenvalues for various valences.

n	λ_1	$\mu_0 = \lambda_1^2$	λ_2	n	λ_1	$\mu_0 = \lambda_1^2$	λ_2
3	0.3231	0.1043	0.0911	10	0.5463	0.2985	0.4660
4	0.4142	0.1715	0.1715	11	0.5514	0.3041	0.4834
5	0.4660	0.2171	0.2547	12	0.5553	0.3084	0.4971
6	0.4971	0.2471	0.3231	13	0.5584	0.3118	0.5080
7	0.5169	0.2672	0.3751	14	0.5608	0.3145	0.5169
8	0.5302	0.2811	0.4142	20	0.5686	0.3234	0.5463
9	0.5396	0.2911	0.4436	25	0.5714	0.3265	0.5569

Table 3: Subdominant eigenvalues of the Binary-Ternary scheme for various valences.

n	λ_1	$\mu_0 = \lambda_1^2$	λ_2	n	λ_1	$\mu_0 = \lambda_1^2$	λ_2
3	0.5376	0.2890	0.2713	10	0.7261	0.5272	0.6613
4	0.6180	0.3819	0.3819	11	0.7301	0.5330	0.6756
5	0.6613	0.4374	0.4721	12	0.7332	0.5375	0.6867
6	0.6867	0.4716	0.5376	13	0.7332	0.5411	0.6955
7	0.7026	0.4937	0.5843	14	0.7375	0.5439	0.7026
8	0.7133	0.5088	0.6180	20	0.7436	0.5530	0.7261
9	0.7207	0.5194	0.6427	25	0.7458	0.5562	0.7344

Table 4: Subdominant eigenvalues of the Fibonacci scheme for various valences.

5. Other L–system based schemes

Here we study other L–system based 1D schemes, which may be good candidates for extension to quad subdivision on arbitrary meshes. In Section 5.1 we derive properties that a 1D scheme should have, should one wishes to extend to a nice bivariate scheme for arbitrary quad meshes with the methods of the previous sections. Next, having restricted our search space, in Section 5.2 we give a list of such 1D subdivision schemes on small alphabets, aiming at demonstrating the versatility of the method and illustrating some salient points.

We note that the implementation of the corresponding bivariate schemes is out of the scope of the paper. To the best of our knowledge some of these 1D schemes have not been proposed before, and their topological refinement rules, i.e. the way by which each edge splits in each subdivision step, could be interesting. However, the focus in this paper, and its main contribution, is the extension of 1D L–system subdivision to quad meshes of arbitrary topology, rather than novel 1D schemes.

5.1. Possible valid L–systems for bivariate subdivision with nice properties

First, we show that some desirable properties for the L–system–based subdivision scheme require the symbol L assigned to the edges emanating from the EVs to be unsymmetric, while the axiom should be symmetric.

5.1.1. Twin unsymmetric symbols

From Proposition 4.7 and 4.8 in Section 4 we know that for a scheme based on cubic B-splines and a valid L–system, a sufficient condition to get *contained* ordinary masks and an eventually stationary local subdivision matrix, is that the label L of the edges incident to the EV has rewriting rule $L \to LC$ (where possibly $C = L$). Moreover, if the rewriting rule is $L \to LL$, then, up to a permutation of rows and columns, we get

$$\mathbf{M} = \begin{bmatrix} 2 & 0 \\ * & N \end{bmatrix}. \tag{31}$$

If ρ is an eigenvalue of $\mathbf{M}$ and l_A the first entry of the associated eigenvector, then $2l_A = \rho l_A$, that is, either $\rho = 2$ or $l_A = 0$. In the first case we do not get a non–integer ratio, while in the second case we do not get a positive eigenvector.

Furthermore, if $L \in \mathcal{S}$, then

$$\begin{aligned} \pi(L) &= \bar{\sigma}(\pi(L)) && (32)\\ LC &= \sigma(C)\sigma(L) && (33)\\ C &= \sigma(L) && (34)\\ C &= L && (35) \end{aligned}$$

Thus, all L–systems considered here include twin unsymmetric symbols L and R whose rules are:

$$\begin{cases} L \rightarrow LC \\ R \rightarrow \sigma(C)R \end{cases} \tag{36}$$

with $C \neq L$, L the label of edges going out from any EV, and R the label of edges with opposite orientation.

Proposition 5.1. *If an L-system is symmetric then equal lengths can be associated with any pair of twin unsymmetric symbols.*

Proof. Let $\mathbf{P}$ be the permutation matrix of the involution $\sigma()$ in Proposition 4.2, that is, if $\Sigma = \{A_1, A_2, \ldots, A_n\}$ is the alphabet, and $\mathbf{e}_i$ is the $n \times 1$ vector with 1 in the i-th entry and 0 otherwise, then $\mathbf{P}\mathbf{e}_i = \mathbf{e}_j$ if $\sigma(A_i) = A_j$.

From (14) we get

$$\begin{aligned} \mathbf{M} &= \mathbf{P}^\top \mathbf{M}\mathbf{P} && (37)\\ \mathbf{P}\mathbf{M} &= \mathbf{M}\mathbf{P}. && (38) \end{aligned}$$

Indeed, multiplying $\mathbf{M}$ by $\mathbf{P}$ from the right swaps columns of twin unsymmetric symbols, while multiplying from the left by $\mathbf{P}^\top$ swaps associated rows.

If $\mathbf{a}$ is an eigenvector of $\mathbf{M}$ associated with the eigenvalue ρ, then

$$\begin{aligned} \mathbf{M}\mathbf{P}\mathbf{a} &= \mathbf{P}\mathbf{M}\mathbf{a} && (39)\\ \mathbf{M}\mathbf{P}\mathbf{a} &= \rho\mathbf{P}\mathbf{a} && (40)\\ \mathbf{M}\left(\mathbf{a} + \mathbf{P}\mathbf{a}\right) &= \rho\left(\mathbf{a} + \mathbf{P}\mathbf{a}\right). && (41) \end{aligned}$$

Thus, $\mathbf{b} := \mathbf{a}+\mathbf{P}\mathbf{a}$ is an eigenvector of $\mathbf{M}$ associated with ρ and if $\sigma(A_i) = A_j$, then $b_i = a_i + a_j = b_j$. □

5.1.2. Symmetric axiom

From Section 4.1, in the rules (36), if we have $C = \sigma(C)$, and if C, possibly after a few steps, produces a word beginning with L and ending with R, then it can be used as the label of the initial edges. Otherwise, C is an unsymmetric symbol, with let say D being its twin, and a symmetric symbol has to be introduced for the initialisation. A simple choice reducing the transitional steps to the minimum, is:

$$S \to LR \tag{42}$$

Proposition 5.2. *If an L-system with ratio ρ includes (42) in its rewriting rules, and $R = \sigma(L)$, then the lengths associated with S and L satisfy $\ell_S = \frac{2}{\rho}\ell_L$.*

Proof. Rule (42) yields $\ell_L + \ell_R = \rho\ \ell_S$ and the result comes from Prop. 5.1. □

Remark 5.3. *As an alternative to the starting rule (42), the following L-system gives a nice example:*

$$\begin{cases} S & \to & CD \\ L & \to & LS \\ C & \to & L \\ D & \to & R \\ R & \to & SR \end{cases} \tag{43}$$

with

- *ratio* $\rho = \frac{1}{3}\left(1 + \sqrt[3]{28 + \sqrt{783}} + \sqrt[3]{28 - \sqrt{783}}\right) \approx 1.695.$
- *interval lengths* $\ell_S = \rho^2 - \rho$, $\ell_C = \ell_D = 1$, $\ell_L = \ell_R = \rho$.

and 23 masks in total.

5.2. A list of L–systems with a small number of symbols

In our list of subdivision schemes based on L–systems on small alphabet we consider the following four cases:

- three symbols $\{L, C, R\}$, C as the axiom, lengths $\ell_L = \ell_R = 1$ and $\ell_C = \rho - 1$, and rules

$$\begin{cases} L & \to & LC \\ C & \to & w \\ R & \to & CR \end{cases} \tag{44}$$

where w is a symmetric word which contains at most three letters, beginning with L and ending with R.

- four symbols $\{S, L, C, R\}$, S as the axiom, lengths $\ell_L = \ell_R = 1$, $\ell_C = \rho - 1$, and $\ell_S = \frac{2}{\rho}$, and rules

$$\begin{cases} S & \to & LR \\ L & \to & LC \\ C & \to & w \\ R & \to & CR \end{cases} \tag{45}$$

where w is a symmetric word which contains at most three letters.

- four symbols $\{L, C, D, R\}$, C as the axiom, lengths $\ell_L = \ell_R = 1$ and $\ell_C = \ell_D = \rho - 1$, and rules

$$\begin{cases} L & \to & LC \\ C & \to & w \\ D & \to & \bar{\sigma}(w) \\ R & \to & DR \end{cases} \tag{46}$$

where w is a word which contains at most three letters, beginning with L and ending with R.

- five symbols $\{S, L, C, D, R\}$, S as the axiom, lengths $\ell_L = \ell_R = 1$, $\ell_C = \ell_D = \rho - 1$, and $\ell_S = \frac{2}{\rho}$, and rules

$$\begin{cases} S & \to & LR \\ L & \to & LC \\ C & \to & w \\ D & \to & \bar{\sigma}(w) \\ R & \to & DR \end{cases} \tag{47}$$

where w is a word which contains at most three letters.

For each possible $w = \pi(C)$ we compute two quantities:

- the number of 1D masks needed. Each mask corresponds to a word of $d+1 = 4$ symbols that can be produced from the initial word made up of the axiom repeated 4 times. We count as one each pair of symmetric words, as their weights are the same, in reverse order.
- the ratio ρ.

The lists are shown in Appendix B.

Remark 5.4. *Different $\pi(C)$ would give rise to the same matrix $\mathbf{M}$, and so to the same ratio ρ, if they comprise the same symbols in different order. Moreover, two different matrices may correspond to the same ratio. Here, we present the results first by increasing ratio, and then by increasing number of masks. We note that for nice subdivision schemes, both quantities should be as small as possible, thus, the nicest schemes are at the top of the tables. Finally, as an extra information, we also give $\mathbf{M}[C;\Sigma]$, the line of $\mathbf{M}$ corresponding to C, where Σ is the alphabet with its symbols written in the same order as the corresponding columns of $\mathbf{M}$.*

Remark 5.5. *Also note that in the lists we do not include the L–systems with integer ratios, that is, all the L-systems corresponding to a 2-word $\pi(C)$, and almost all of those corresponding to a single letter $\pi(C)$. As a consequence, apart from the first two L–systems with five symbols, which are related to the Fibonacci L–system, all the other are associated with a 3-word $\pi(C)$.*

5.3. Discussion

Here, we collect some final remarks about L–systems based subdivision, the deduced ratio and the number of masks, some of which could be somewhat counter-intuitive.

The ratio of an L–system with $\pi(C)$ comprising more than three symbols may be smaller than that of an L–system with 3-letter $\pi(C)$. The two L–systems below, on five symbol alphabets, are such an example.

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(S,L,C,D,R)]$
$\frac{3+\sqrt{5}}{2} \approx 2.618$	20	CLD	$0\ \ 1\ \ 1\ \ 1\ \ 0$
$\frac{1+\sqrt{17}}{2} \approx 2.561$	17	$LLLL$	$0\ \ 4\ \ 0\ \ 0\ \ 0$

Having $\pi(C)$ with more symbols does not necessarily yield more masks, as not every possible word from a given alphabet is produced by a given set of rewriting rules, and even more so from a given axiom. The two L–systems below, on five symbol alphabets, are such an example.

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(S,L,C,D,R)]$
$1+\sqrt{2}\approx 2.414$	7	DRL	0 1 0 1 1
$\frac{1+\sqrt{5}}{2}\approx 1.618$	15	L	0 1 0 0 0

Choosing $D=\sigma(C)$ rather than $C=\sigma(C)$ does not necessarily yield more masks. The two L–systems below, with S as axiom and five and four symbols, respectively, is such an example.

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(S,L,C,D,R)]$
$1+\sqrt{2}\approx 2.414$	7	DRL	0 1 0 1 1
			$\mathbf{M}[C;(S,L,C,R)]$
	17	RCL	0 1 1 1

A larger number of distinct symbols in $\pi(C)$ does not necessarily yield more masks. The two L–systems below, on five symbol alphabets, give an example.

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(S,L,C,D,R)]$
$1+\sqrt{2}\approx 2.414$	7	DRL	0 1 0 1 1
	30	RRD	0 0 0 1 2

By just permuting the symbols in $\pi(C)$, we do not necessarily get the same number of masks. Below is such an example with two such L–systems on five symbol alphabets.

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(S,L,C,D,R)]$
$1+\sqrt{2}\approx 2.414$	7	DRL	0 1 0 1 1
	21	RDL	0 1 0 1 1

Different L–systems, yielding different numbers of masks, may still have identical interval subdivision patterns. Below is an example of three such L–systems, with axiom S, and four, five and five symbols, respectively.

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(S,L,C,R)]$
$1+\sqrt{2}\approx 2.414$	10	LCR	0 1 1 1
			$\mathbf{M}[C;(S,L,C,D,R)]$
	12	LDR	0 1 0 1 1
	14	LCR	0 1 1 0 1

The two L–systems below with axiom C and alphabets of three and four symbols, respectively, is another example.

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(L,C,R)]$
$1+\sqrt{2}\approx 2.414$	6	LCR	1 1 1
			$\mathbf{M}[C;(L,C,D,R)]$
	9	LCR	1 1 0 1

In these cases, it is most likely that C and D are equivalent and can be switched at anytime. As we have counted one mask per each pair of twin unsymmetric words, we could also count one only mask for each set of words that become identical if we do not differentiate between C and D, and then find the minimum number of masks, which would correspond to $C=\sigma(C)$.

Finally, we note that there are L–systems with ratio lower than the golden ratio. For example, we could combine the bivariate Fibonacci L–system and the delay queue example introduced in [1], to get the L–system :

$$\begin{cases} S & \to & LR \\ L & \to & LC \\ C & \to & A \\ A & \to & R \end{cases} \qquad \begin{cases} D & \to & B \\ B & \to & L \\ R & \to & DR \end{cases} \tag{48}$$

with

- ratio $\rho = \frac{1}{3}\left(1 + \sqrt[3]{\frac{29+\sqrt{837}}{2}} + \sqrt[3]{\frac{29-\sqrt{837}}{2}}\right) \approx 1.465 < 1.618$
- interval lengths $\ell_S = 2\rho$, $\ell_C = \ell_D = 1$, $\ell_A = \ell_B = \rho$, $\ell_L = \ell_R = \rho^2$

which, however, has a large number of masks, namely 31.

6. Conclusions

We extended the L–system based, non-uniform, univariate subdivision proposed in [1], to the bivariate setting. The focus was on subdivision schemes that generalise the cubic B-spline to arbitrary meshes, meaning that the masks and stencils we dealt with had manageable size. We studied how properties of an L–system relate to the properties of the subdivision scheme, and we proposed a method for computing eventually stationary EV stencils for subdivision schemes corresponding to L–systems with nice properties. We illustrated the proposed method by implementing and studying two new subdivision schemes with interesting properties, the Binary-Ternary and the Fibonacci.

We believe that L–system based subdivision is an interesting compromise between the classic uniform subdivision, with its very basic and inflexible control mesh refinement processes, and geometry driven non-uniform subdivision, where the convergence results are very difficult to obtain. As a future research direction, we propose to formulate the computation of EV stencils as a multi-variable optimisation problem, instead of the single free variable we currently use, and devise non-uniform subdivision schemes with better smoothness properties. The study of dual subdivision schemes, corresponding to B-splines of even degree, is another promising research direction.

Acknowledgements

The authors thank the anonymous referees for their diligence and helpful comments.

References

[1] V. Nivoliers, C. Gérot, V. Ostromoukhov, N. F. Stewart, L–system specification of knot-insertion rules for non–uniform B–spline subdivision, Computer Aided Geometric Design 29 (2012) 150–161.

[2] P. Prusinkiewicz, F. Samavati, C. Smith, R. Karwowski, L-system description of subdivision curves, International Journal of Shape Modeling 9 (2002) 41–59.

[3] L. Velho, Stellar subdivision grammars, in: SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Eurographics Association, 2003, pp. 188–199.

[4] E. Catmull, J. Clark, Recursively generated B-spline surfaces on arbitrary topological meshes, Computer Aided Geometric Design 6 (1978) 350–355.

[5] D. Doo, M. Sabin, Behaviour of recursive division surfaces near extraordinary points, Computer-Aided Design 10 (1978) 356–360.

[6] T. W. Sederberg, J. Zheng, D. Sewell, M. Sabin, Non-uniform recursive subdivision surfaces, in: Proc. SIGGRAPH '98, ACM, 1998, p. 387–394. doi:10.1145/280814.280942.

[7] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and t-nurccs, ACM Trans. Graph. 22 (2003) 477–484.

[8] T. J. Cashman, U. H. Augsdörfer, N. A. Dodgson, M. A. Sabin, NURBS with Extraordinary Points: High-degree, Non-uniform, Rational Subdivision Schemes, ACM Transactions on Graphics 28 (2009) #46, 1–9.

[9] R. Goldman, J. Warren, An extension of Chaiken's algorithm to B-spline curves with knots in geometric progression, CVGIP: Graphical Models and Image Processing 55 (1993) 58–62.

[10] T. J. Cashman, N. A. Dodgson, M. A. Sabin, Non-uniform B-spline subdivision using refine and smooth, in: R. Martin, M. Sabin, J. Winkler (Eds.), Mathematics of Surfaces XII, 2007, pp. 121–137.

[11] S. Schaefer, R. Goldman, Non-uniform subdivision for B-splines of arbitrary degree, Computer Aided Geometric Design 26 (2009) 75–81.

[12] I. Daubechies, I. Guskov, W. Sweldens, Regularity of irregular subdivision, Constr. Approx. 15 (1999) 381–426. doi:https://doi.org/10.1007/s003659900114.

[13] M. Marinov, N. Dyn, D. Levin, Geometrically controlled 4-point interpolatory schemes, in: N. A. Dodgson, M. S. Floater, M. A. Sabin (Eds.), Advances in Multiresolution for Geometric Modelling, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 301–315.

[14] N. Dyn, M. S. Floater, K. Hormann, Four-point curve subdivision based on iterated chordal and centripetal parameterizations, Computer Aided Geometric Design 26 (2009) 279–286. doi:https://doi.org/10.1016/j.cagd.2008.09.006.

[15] U. Augsdörfer, N. Dodgson, M. Sabin, Variations on the four-point subdivision scheme, Computer Aided Geometric Design 27 (2010) 78–95. doi:https://doi.org/10.1016/j.cagd.2009.09.002.

[16] N. Dyn, D. Levin, J. Yoon, A new method for the analysis of univariate nonuniform subdivision schemes, Constr. Approx. 40 (2014) 173–188.

[17] M. Fang, B. Jeong, J. Yoon, A family of non-uniform subdivision schemes with variable parameters for curve design, Applied Mathematics and Computation 313 (2017) 1–11. doi:https://doi.org/10.1016/j.amc.2017.05.063.

[18] N. Dyn, K. Hormann, C. Mancinelli, Non-uniform interpolatory subdivision schemes with improved smoothness, Computer Aided Geometric Design 94 (2022) 102083. doi:https://doi.org/10.1016/j.cagd.2022.102083.

[19] C. V. Beccari, G. Casciola, L. Romani, Non-uniform non-tensor product local interpolatory subdivision surfaces, Computer Aided Geometric Design 30 (2013) 357–373. doi:https://doi.org/10.1016/j.cagd.2013.02.002.

[20] C. Conti, N. Dyn, Convergence and smoothness of tensor-product of two non-uniform linear subdivision schemes, Computer Aided Geometric Design 66 (2018) 16–18. doi:https://doi.org/10.1016/j.cagd.2018.08.001.

[21] A. Lamnii, M.-Y. Nour, D. Sbibih, A. Zidna, Non-uniform subdivision schemes of ωB-spline curves and surfaces with variable parameters, Computer-Aided Design 154 (2023) 103420. doi:https://doi.org/10.1016/j.cad.2022.103420.

[22] C. Conti, N. Dyn, Non-stationary subdivision schemes: State of the art and perspectives, in: G. E. Fasshauer, M. Neamtu, L. L. Schumaker (Eds.), Approximation Theory XVI, 2021, pp. 39–71.

[23] C. Conti, M. Donatelli, L. Romani, P. Novara, Convergence and normal continuity analysis of nonstationary subdivision schemes near extraordinary vertices and faces, Constr. Approx. 50 (2019) 457–496. doi:https://doi.org/10.1007/s00365-019-09477-y.

[24] R. N. Goldman, Blossoming and knot insertion algorithms for B-spline curves, Computer Aided Geometric Design 7 (1990) 69 – 81. doi:DOI: 10.1016/0167-8396(90)90022-J.

[25] V. Nivoliers, Subdivision Surfaces on a Penrose Tiling, Master's thesis, IVR, UJF INPG, 2008.

[26] J. Peters, U. Reif, Subdivision Surfaces, volume 3 of *Geometry and Computing*, Springer-Verlag Berlin Heidelberg, 2008. doi:10.1007/978-3-540-76406-9.

[27] K. Karčiauskas, J. Peters, An improved refinement rule for multi-sided faces, Computers & Graphics 102 (2022) 370–379. doi:https://doi.org/10.1016/j.cag.2021.10.008.

[28] U. Augsdörfer, N. Dodgson, M. Sabin, Tuning Subdivision by Minimising Gaussian Curvature Variation Near Extraordinary Vertices, Computer Graphics Forum (2006). doi:10.1111/j.1467-8659.2006.00945.x.

[29] U. H. Augsdörfer, T. J. Cashman, N. A. Dodgson, M. A. Sabin, Numerical checking of C^1 for arbitrary degree quadrilateral subdivision schemes, in: E. R. Hancock, R. R. Martin, M. A. Sabin (Eds.), Mathematics of Surfaces XIII, Springer Berlin Heidelberg, 2009, pp. 45–54.

[30] U. Reif, A unified approach to subdivision algorithms near extraordinary vertices, Computer Aided Geometric Design 12 (1995) 153–174.

[31] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, G. Ranzuglia, MeshLab: an Open-Source Mesh Processing Tool, in: V. Scarano, R. D. Chiara, U. Erra (Eds.), Eurographics Italian Chapter Conference, The Eurographics Association, 2008, pp. 129–136.

doi:10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.

[32] G. Guennebaud, M. Germann, M. Gross, Dynamic Sampling and Rendering of Algebraic Point Set Surfaces, Computer Graphics Forum 27 (2008) 653–662.

[33] K. Karciauskas, J. Peters, U. Reif, Shape characterization of subdivision surfaces–case studies, Computer Aided Geometric Design 21 (2004) 601–614. doi:10.1016/j.cagd.2004.04.005.

Appendix A. Transition between labels and associated masks

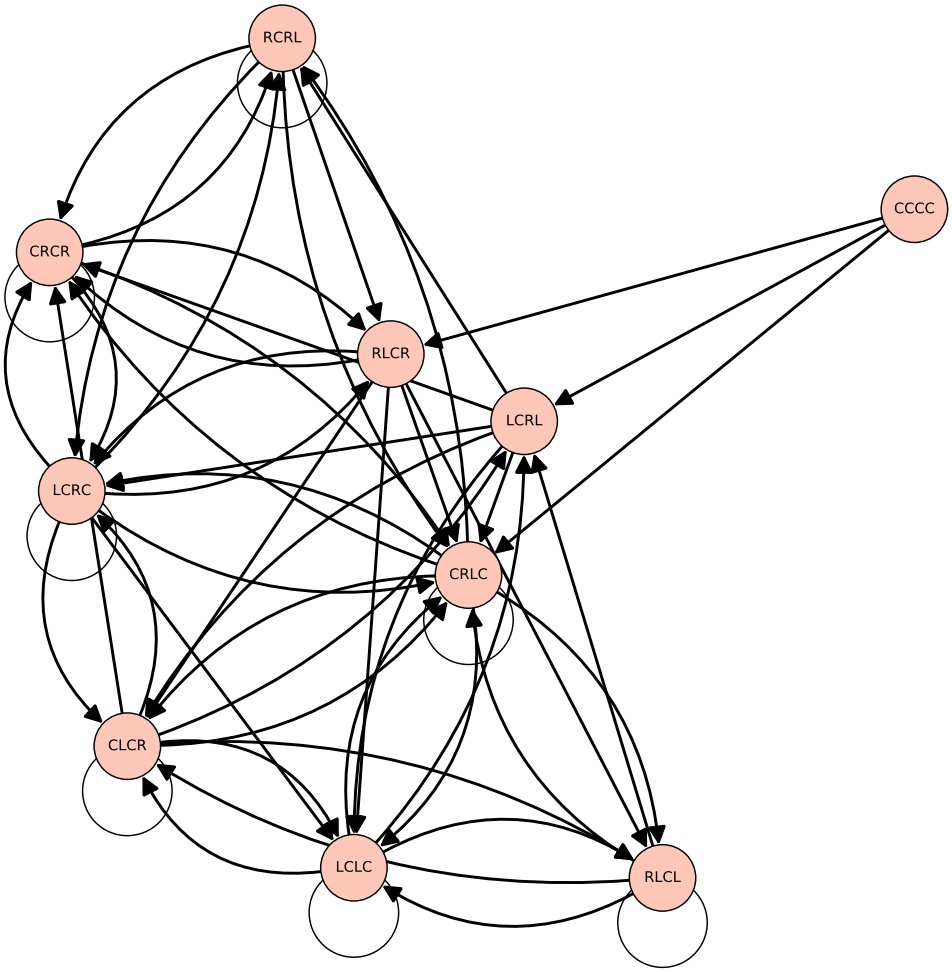


Figure A.8: Transition graph between the vertex 1D-labels of the binary-ternary scheme.

CLCR				LCRC			
LCRL	0.04481	=	$(8\sqrt{2}-11)/7$	LCLC	0.07106	=	$5\sqrt{2}-7$
CRLC	0.19782	=	$(13\sqrt{2}-17)/7$	CLCR	0.34314	=	$6-4\sqrt{2}$
RLCL	0.47759	=	$(9-4\sqrt{2})/7$	LCRC	0.61203	=	$(20\sqrt{2}-24)/7$
LCLC	0.71254	=	$(46-29\sqrt{2})/7$	**CRCR**	0.71254	=	$(46-29\sqrt{2})/7$
CLCR	0.61203	=	$(20\sqrt{2}-24)/7$	RCRL	0.47759	=	$(9-4\sqrt{2})/7$
LCRC	0.34314	=	$6-4\sqrt{2}$	CRLC	0.19782	=	$(13\sqrt{2}-17)/7$
CRCR	0.07106	=	$5\sqrt{2}-7$	RLCR	0.04481	=	$(8\sqrt{2}-11)/7$

CRCR				LCLC			
LCRC	0.04481	=	$(8\sqrt{2}-11)/7$	LCLC	0.07106	=	$5\sqrt{2}-7$
CRCR	0.21638	=	$(10-6\sqrt{2})/7$	CLCR	0.34314	=	$6-4\sqrt{2}$
RCRL	0.52240	=	$(4\sqrt{2}-2)/7$	LCRL	0.61203	=	$(20\sqrt{2}-24)/7$
CRLC	0.70166	=	$(36\sqrt{2}-46)/7$	**CRLC**	0.70166	=	$(36\sqrt{2}-46)/7$
RLCR	0.61203	=	$(20\sqrt{2}-24)/7$	RLCL	0.52240	=	$(4\sqrt{2}-2)/7$
LCRC	0.34314	=	$6-4\sqrt{2}$	LCLC	0.21638	=	$(10-6\sqrt{2})/7$
CRCR	0.07106	=	$5\sqrt{2}-7$	CLCR	0.04481	=	$(8\sqrt{2}-11)/7$

CRLC				(CCCC)			
				LCRL	0.03451	=	$(\sqrt{2}-1)/12$
LCRC	0.05025	=	$(10-7\sqrt{2})/2$	CRLC	0.15236	=	$(2\sqrt{2}-1)/12$
CRCR	0.24264	=	$3\sqrt{2}-4$	RLCR	0.36785	=	$(3+\sqrt{2})/12$
RCRL	0.58578	=	$2-\sqrt{2}$	LCRL	0.59763	=	$(5-\sqrt{2})/6$
CRLC	0.75735	=	$5-3\sqrt{2}$	**CRLC**	0.69526	=	$(7-2\sqrt{2})/6$
RLCL	0.58578	=	$2-\sqrt{2}$	RLCR	0.59763	=	$(5-\sqrt{2})/6$
LCLC	0.24264	=	$3\sqrt{2}-4$	LCRL	0.36785	=	$(3+\sqrt{2})/12$
CLCR	0.05025	=	$(10-7\sqrt{2})/2$	CRLC	0.15236	=	$(2\sqrt{2}-1)/12$
				RLCR	0.03451	=	$(\sqrt{2}-1)/12$

LCRL				RLCR			
LCLC	0.07106	=	$5\sqrt{2}-7$	CRLC	0.12132	=	$(3\sqrt{2}-4)/2$
CLCR	0.34314	=	$6-4\sqrt{2}$	RLCL	0.41421	=	$\sqrt{2}-1$
LCRC	0.60660	=	$(15\sqrt{2}-20)/2$	**LCLC**	0.68629	=	$12-8\sqrt{2}$
CRCR	0.68629	=	$12-8\sqrt{2}$	CLCR	0.60660	=	$(15\sqrt{2}-20)/2$
RCRL	0.41421	=	$\sqrt{2}-1$	LCRC	0.34314	=	$6-4\sqrt{2}$
CRLC	0.12132	=	$(3\sqrt{2}-4)/2$	CRCR	0.07106	=	$5\sqrt{2}-7$

RCRL				RLCL			
CRLC	0.10050	=	$10-7\sqrt{2}$	CRLC	0.12132	=	$(3\sqrt{2}-4)/2$
RLCR	0.34314	=	$6-4\sqrt{2}$	RLCL	0.41421	=	$\sqrt{2}-1$
LCRC	0.60660	=	$(15\sqrt{2}-20)/2$	**LCLC**	0.68629	=	$12-8\sqrt{2}$
CRCR	0.68629	=	$12-8\sqrt{2}$	CLCR	0.60660	=	$(15\sqrt{2}-20)/2$
RCRL	0.41421	=	$\sqrt{2}-1$	LCRL	0.34314	=	$6-4\sqrt{2}$
CRLC	0.12132	=	$(3\sqrt{2}-4)/2$	CRLC	0.10050	=	$10-7\sqrt{2}$

Table A.5: 1D-masks for the Binary-Ternary scheme. An old vertex contributes, with different weights, to a sequence of new vertices. For each 1D-label of an old vertex, the list of weights is given from top to bottom alongside the 1D-label on the new vertex. The label of the child of the old vertex is written in bold.

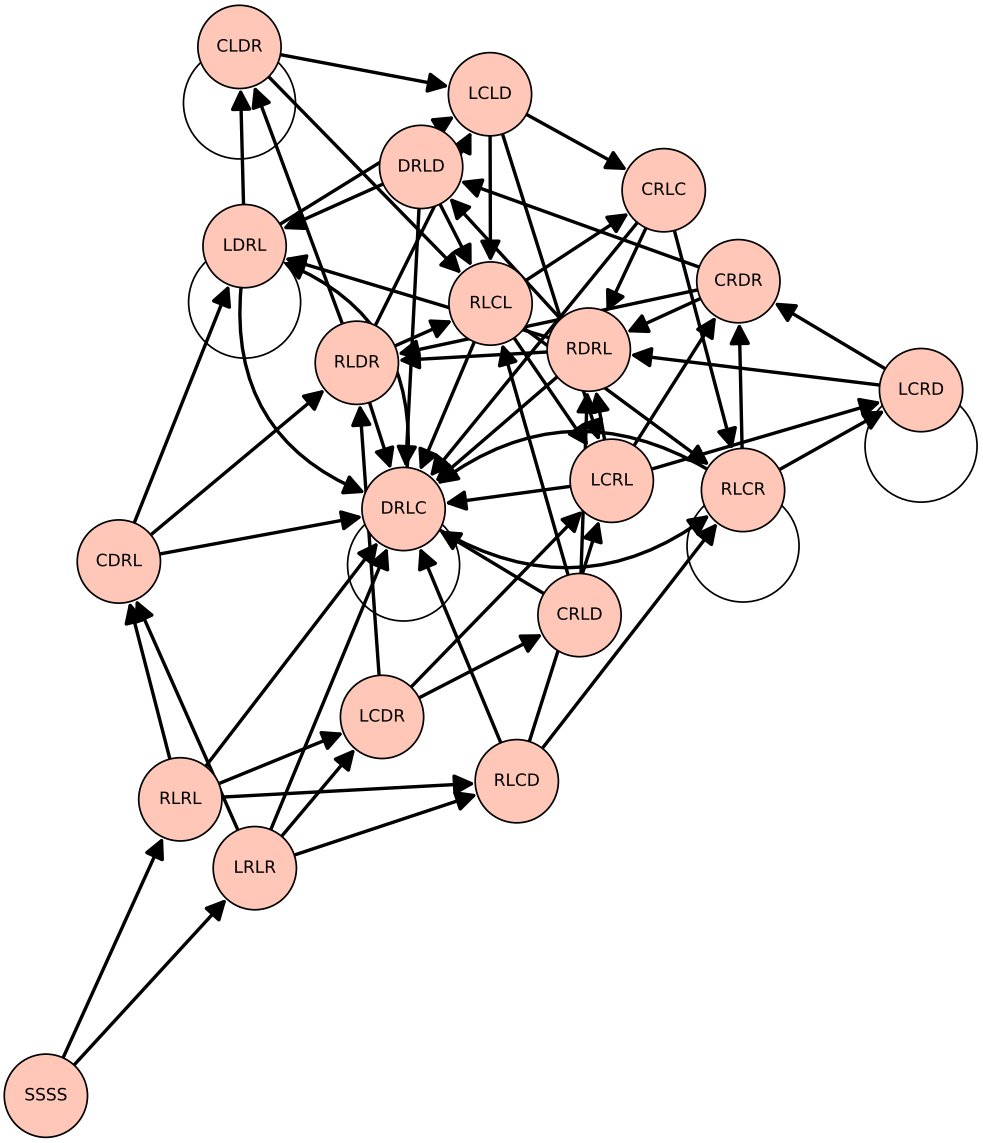


Figure A.9: Transition graph between the vertex 1D-labels of the Fibonacci scheme.

(CDRL)				(RLCD)			
RLDR	0.72361	=	$(\sqrt{5}+5)/10$	DRLC	0.11803	=	$(\sqrt{5}-2)/2$
LDRL	0.61803	=	$(\sqrt{5}-1)/2$	RLCR	0.61803	=	$(\sqrt{5}-1)/2$
DRLC	0.11803	=	$(\sqrt{5}-2)/2$	**LCRL**	0.72361	=	$(\sqrt{5}+5)/10$
CLDR				LCRD			
RLCL	0.55279	=	$(5-\sqrt{5})/5$	LCRD	0.23607	=	$\sqrt{5}-2$
LCLD	0.76393	=	$3-\sqrt{5}$	**CRDR**	0.76393	=	$3-\sqrt{5}$
CLDR	0.23607	=	$\sqrt{5}-2$	RDRL	0.55279	=	$(5-\sqrt{5})/5$
CRDR				LCLD			
RDRL	0.44721	=	$\sqrt{5}/5$	LCRL	0.23607	=	$\sqrt{5}-2$
DRLD	0.85410	=	$(3\sqrt{5}-5)/2$	**CRLC**	0.85410	=	$(3\sqrt{5}-5)/2$
RLDR	0.23607	=	$\sqrt{5}-2$	RLCL	0.44721	=	$\sqrt{5}/5$
CRLC				DRLD			
RDRL	0.38197	=	$(3-\sqrt{5})/2$	LDRL	0.38197	=	$(3-\sqrt{5})/2$
DRLC	0.76393	=	$3-\sqrt{5}$	**DRLC**	0.76393	=	$3-\sqrt{5}$
RLCR	0.38197	=	$(3-\sqrt{5})/2$	RLCL	0.38197	=	$(3-\sqrt{5})/2$
(CRLD)				DRLC			
RDRL	0.38197	=	$(3-\sqrt{5})/2$	LDRL	0.38197	=	$(3-\sqrt{5})/2$
DRLC	0.76393	=	$3-\sqrt{5}$	**DRLC**	0.76393	=	$3-\sqrt{5}$
RLCL	0.38197	=	$(3-\sqrt{5})/2$	RLCR	0.38197	=	$(3-\sqrt{5})/2$
(LCDR)				(SSSS)			
				(LRLR)	0.12500	=	1/8
LCRL	0.27639	=	$(5-\sqrt{5})/10$	(RLRL)	0.50000	=	1/2
(CRLD)	1	=	1	**(LRLR)**	0.75000	=	3/4
RLDR	0.27639	=	$(5-\sqrt{5})/10$	(RLRL)	0.50000	=	1/2
				(LRLR)	0.12500	=	1/8
LCRL				RLDR			
LCRD	0.23607	=	$\sqrt{5}-2$	DRLC	0.11803	=	$(\sqrt{5}-2)/2$
CRDR	0.76393	=	$3-\sqrt{5}$	RLCL	0.61803	=	$(\sqrt{5}-1)/2$
RDRL	0.61803	=	$(\sqrt{5}-1)/2$	**LCLD**	0.76393	=	$3-\sqrt{5}$
DRLC	0.11803	=	$(\sqrt{5}-2)/2$	CLDR	0.23607	=	$\sqrt{5}-2$
LDRL				RLCR			
LCLD	0.23607	=	$\sqrt{5}-2$	DRLC	0.11803	=	$(\sqrt{5}-2)/2$
CLDR	0.76393	=	$3-\sqrt{5}$	RLCR	0.61803	=	$(\sqrt{5}-1)/2$
LDRL	0.61803	=	$(\sqrt{5}-1)/2$	**LCRD**	0.76393	=	$3-\sqrt{5}$
DRLC	0.11803	=	$(\sqrt{5}-2)/2$	CRDR	0.23607	=	$\sqrt{5}-2$
(LRLR)				(RLRL)			
(LCDR)	0.14235	=	$(3\sqrt{5}-5)/12$	DRLC	0.10301	=	$(\sqrt{5}-1)/12$
(CDRL)	0.46066	=	$(5-\sqrt{5})/6$	(RLCD)	0.53934	=	$(\sqrt{5}+1)/6$
DRLC	0.79399	=	$(7-\sqrt{5})/6$	**(LCDR)**	0.71530	=	$(11-3\sqrt{5})/6$
(RLCD)	0.46066	=	$(5-\sqrt{5})/6$	(CDRL)	0.53934	=	$(\sqrt{5}+1)/6$
(LCDR)	0.14235	=	$(3\sqrt{5}-5)/12$	DRLC	0.10301	=	$(\sqrt{5}-1)/12$
RDRL				RLCL			
DRLD	0.14590	=	$(7-3\sqrt{5})/2$	DRLC	0.11803	=	$(\sqrt{5}-2)/2$
RLDR	0.76393	=	$3-\sqrt{5}$	RLCR	0.61803	=	$(\sqrt{5}-1)/2$
LDRL	0.61803	=	$(\sqrt{5}-1)/2$	**LCRL**	0.76393	=	$3-\sqrt{5}$
DRLC	0.11803	=	$(\sqrt{5}-2)/2$	CRLC	0.14590	=	$(7-3\sqrt{5})/2$

Table A.6: 1D-masks for the Fibonacci scheme with the same organisation as in Table A.5. Transitional 1D-labels for early subdivision steps only, are written between parentheses.

Appendix B. Features of L–system based subdivision schemes

In this appendix we list the features of all subdivision schemes based on L–systems on small alphabets, as described in Section 5.2.

Three symbols $\{L, C, R\}$

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(L,C,R)]$
$1+\sqrt{2} \approx 2.414$	6	LCR	1 1 1

This is a minimal number of masks, given all the other constraints. If, for example, we relax the constraint of at most two sub-intervals for the labels of the edges incident to the EV, adopted to ease both the implementation and the eigenanalysis, we could consider the following L-system:

$$\begin{cases} L & \to & LCL \\ C & \to & L \end{cases} \tag{B.1}$$

which uses only two symbols $\{L, C\}$, both symmetric, with L as the axiom (we could also use C but that would require to introduce another transitional mask corresponding to $CCCC$), ratio $\rho = 1+\sqrt{2} \approx 2.414$, and lengths $\ell_L = 1$ and $\ell_C = \rho - 2$, which requires 4 only masks:

$$\begin{aligned} &LLLL \\ &CLLC = \bar{\sigma}(CLLC) \\ &CLLL = \bar{\sigma}(LLLC) \\ &LCLL = \bar{\sigma}(LLCL) \end{aligned}$$

Four symbols $\{S, L, C, R\}$

ρ	# masks	$\pi(C)$	$\mathbf{M}[C; (S, L, C, R)]$
$\frac{1+\sqrt{13}}{2} \approx 2.302$	13	RLR	$0 \quad 1 \quad 0 \quad 2$
	14	LRR	$0 \quad 1 \quad 0 \quad 2$
	14	LLR	$0 \quad 2 \quad 0 \quad 1$
	15	LRL	$0 \quad 2 \quad 0 \quad 1$
	17	LLL	$0 \quad 3 \quad 0 \quad 0$
	17	RRR	$0 \quad 0 \quad 0 \quad 3$
	18	RRL	$0 \quad 1 \quad 0 \quad 2$
	21	RLL	$0 \quad 2 \quad 0 \quad 1$

ρ	# masks	$\pi(C)$	$\mathbf{M}[C; (S, L, C, R)]$
$1 + \sqrt{2} \approx 2.414$	10	LCR	$0 \quad 1 \quad 1 \quad 1$
	17	RCL	$0 \quad 1 \quad 1 \quad 1$

Four symbols $\{L, C, D, R\}$

ρ	# masks	$\pi(C)$	$\mathbf{M}[C; (L, C, D, R)]$
$1 + \sqrt{2} \approx 2.414$	9	LCR	$1 \quad 1 \quad 0 \quad 1$
	9	LDR	$1 \quad 0 \quad 1 \quad 1$

Five symbols $\{S, L, C, D, R\}$

ρ	# masks	$\pi(C)$	$\mathbf{M}[C; (S, L, C, D, R)]$
$\frac{1+\sqrt{5}}{2} \approx 1.618$	13	R	$0 \quad 0 \quad 0 \quad 0 \quad 1$
	15	L	$0 \quad 1 \quad 0 \quad 0 \quad 0$

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(S,L,C,D,R)]$
$1+\sqrt{2}\approx 2.414$	7	DRL	0 1 0 1 1
	9	RLC	0 1 1 0 1
	12	RDR	0 0 0 1 2
	12	LDR	0 1 0 1 1
	13	LLC	0 2 1 0 0
	13	DRR	0 0 0 1 2
	13	DLR	0 1 0 1 1
	14	LCR	0 1 1 0 1
	14	CRL	0 1 1 0 1
	15	LCL	0 2 1 0 0
	15	RLD	0 1 0 1 1
	16	LRD	0 1 0 1 1
	17	LRC	0 1 1 0 1
	17	CLR	0 1 1 0 1
	18	DLL	0 2 0 1 0
	20	RCR	0 0 1 0 2
	21	CLL	0 2 1 0 0
	21	RRC	0 0 1 0 2
	21	RDL	0 1 0 1 1
	22	LDL	0 2 0 1 0
	22	RCL	0 1 1 0 1
	23	CRR	0 0 1 0 2
	24	LLD	0 2 0 1 0
	30	RRD	0 0 0 1 2

ρ	# masks	$\pi(C)$	$\mathbf{M}[C;(S,L,C,D,R)]$
$\frac{3+\sqrt{5}}{2} \approx 2.618$	12	DDR	0 0 0 2 1
	13	CLC	0 1 2 0 0
	13	CDR	0 0 1 1 1
	14	LCC	0 1 2 0 0
	14	DRD	0 0 0 2 1
	14	LCD	0 1 1 1 0
	14	DLC	0 1 1 1 0
	14	DRC	0 0 1 1 1
	15	DCR	0 0 1 1 1
	16	CDL	0 1 1 1 0
	17	DCL	0 1 1 1 0
	18	CRC	0 0 2 0 1
	18	LDC	0 1 1 1 0
	18	RDC	0 0 1 1 1
	20	CCL	0 1 2 0 0
	20	CLD	0 1 1 1 0
	21	CCR	0 0 2 0 1
	21	CRD	0 0 1 1 1
	22	LDD	0 1 0 2 0
	22	DLD	0 1 0 2 0
	22	RCC	0 0 2 0 1
	23	DDL	0 1 0 2 0
	28	RCD	0 0 1 1 1
	29	RDD	0 0 0 2 1