Fast bi-3 Quadratic-Attraction Subdivision

Kęstutis Karčiauskas[a], Jörg Peters[b,*]

[a] *Institute of Mathematics, Vilnius University, Lithuania*
[b] *CISE, University of Florida. Gainesville, USA*

Abstract

Fast bi-3 Quadratic-Attraction Subdivision (FASt) offers noticeable shape improvement over Catmull-Clark subdivision and its tuned variants, yet is equally easy to implement as repeated multiplication of the control net vertices by an explicit subdivision matrix. FASt achieves maximal smoothness for degree bi-3 (bi-cubic) surfaces: C^2 everywhere except for the extraordinary limit point where the surface curvature is bounded and a unique normal exists. Remarkably, FASt surfaces have highlight line distributions comparable to Quadratic-Attraction Subdivision (QAS), the shape-quality leader among bi-3 subdivision algorithms. But while QAS and recent 'tuned subdivision' algorithms, generate one surface ring consisting of $12n$ bi-3 pieces per refinement step, FASt requires just $5n$ bi-cubic pieces per refinement step. This economy of pieces is possible due to a technical innovation in the derivation of FASt: combining regular but non-uniform refinement with irregular but uniform quadratic-attraction subdivision.

And, whereas Quadratic-Attraction and Catmull-Clark subdivision fill n-sided holes ever more slowly as n increases, FASt fills holes with a fixed contraction speed 0.57 for $n = 6, \ldots, 10$ (and faster for $n < 6$). That is, two steps of FASt shrink the unfilled area by an order of magnitude. Faster contraction not only improves rendering efficiency, but fewer pieces reduce the cost of data exchange and engineering analysis, such as computing integrals up to tolerance.

Keywords: Subdivision algorithm, surface quality, bi-cubic, fast contraction, bounded curvature

1. Introduction

The generalization of uniform knot insertion to refine bi-cubic B-spline control nets, a.k.a. bi-cubic subdivision, is the main geometry representation in Computer Graphics and Animation, and is widely used in shape design for initial ideation and visualization. The ability of quadrilateral meshes to follow a cross field of principal curvature or feature lines, with degree 3 in each direction to allow for curvature continuity between the pieces almost everywhere; and intuitive local averaging rules, where $n \neq 4$ quadrilaterals meet at so-called extraordinary nodes, have led to widespread use of bi-cubic subdivision. Implementations in all major modeling packages and availability of efficient GPU-based open source code [1] has placed Catmull-Clark subdivision [2] at everyone's finger tip. Yet, especially for higher n, i.e. more than 5 quadrilaterals meeting in the control net outline of the shape, various shape artifacts were observed early on, and later formalized in [3, 4, 5]. Shape artifacts led [6] to pioneer manipulation of the eigenstructure of the subdivision matrix that groups together the local averaging rules. Multiplication with this matrix refines the control points that generate one n-sided surface ring, see Fig. 1 . The sequence of rings contract, at the speed of a subdominant eigenvalue $\lambda < 1$, to jointly form the subdivision surface. Sabin et al's and subsequent tuning focuses on the limit point and consists of adjusting one or more averaging rules. Since bi-3 (bi-cubic) subdivision surfaces cannot generically be C^2 at extraordinary points [7, Thm 7.19], and since Catmull-Clark curvatures dramatically diverge near the extraordinary point, an important aim is to bound the curvature.

Key to the superior shape of an alternative class of iterative surface generation algorithms is the initial, separate construction of a guiding surface. Avoiding a separate construction, Quadratic-Attraction Subdivision (QAS) [8] bakes an *evolving guide* into the explicit linear formulas operating on the control nets. The guide surface appears, in terms of an abstract control net, *only virtually during the derivation of the refinement rules*, so that QAS requires only the classical implementation of multiplying the control net with a fixed subdivision matrix per refinement step. Fast bi-3 Quadratic-Attraction Subdivision (FASt) shares this advantage and

- FASt produces surfaces of the similar top quality as the quality-leader QAS, using less than 1/2 the resources: FASt requires only $5n$ bi-cubic pieces per ring vs $12n$ for QAS. Compare Fig. 1 b to Fig. 1 c.

This reduction is possible due to a key technical innovation in the derivation of the refinement formulas:

- combining regular non-uniform refinement with irregular uniform quadratic-attraction subdivision.

Moreover,

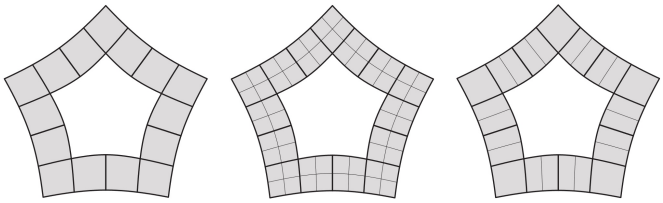
- combining two consecutive rings into one leads to $10n$ bi-3 patches per ring and extremely fast convergence: the subdominant eigenvalue of the combined subdivision is less than 0.325 , i.e. the hole diameter shrinks to 1/3rd in a single step and the unfilled area shrinks by an order of magnitude.

- FASt has the same single-ring structure as Catmull Clark Subdivision (CCS) and is easily implemented by replacing the CCS refinement matrices (collection of rules for extraordinary points) by FASt matrices whose non-zero entries are given in the Appendix.

- FASt surfaces are of higher quality than CCS– despite being faster at filling n-sided holes by subdivision surface rings.

*Corresponding author
Email addresses: kestutis.karciauskas@mif.vu.lt (Kęstutis Karčiauskas), jorg.peters@gmail.com (Jörg Peters)

- Compared to [8], FASt offers the choice from three types of quadratic attraction: attraction towards (i) a C^2 expansion at the extraordinary point, (ii) a C^1 expansion, (iii) an expansion that is C^1 only at the center and C^0 otherwise – and is thereby maximally flexible in the shape without compromising the final smoothness.



(a) $3n$ bi-cubics CCS (b) $12n$ bi-cubics (c) $5n$ bi-cubics FASt

Figure 1: Surface rings with $n = 5$ sectors: each quadrilateral corresponds to one bi-cubic surface patch. (a) 3 patches per sector in the ring layout of CCS. (b) 12 patches per sector for 'tuned subdivision' [9, 10], PAS [11], QAS [8]); (c) 5 patches per sector for FASt.

1.1. Tuned and Guided Subdivision Algorithms

For almost 50 years, bi-cubic subdivision has been synonymous with Catmull-Clark subdivision (CCS) [2]. Improvements have focused on the poor behavior of CCS near extraordinary nodes, by adjusting or tuning the control net averaging rules that characterize the subdivision algorithm near extraordinary nodes, where more or fewer than $n = 4$ quadrilaterals meet. In parallel, Guided Subdivision has been developed to more explicitly control and improve smoothness of the overall limit shape. Neither tuned, nor guided subdivision have been widely adopted since the shape improvement via tuning has been modest and the original guided approach departed from the simple matrix-of-stencils structure of established bi-cubic subdivision.

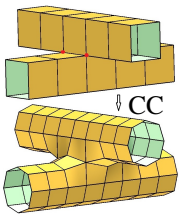
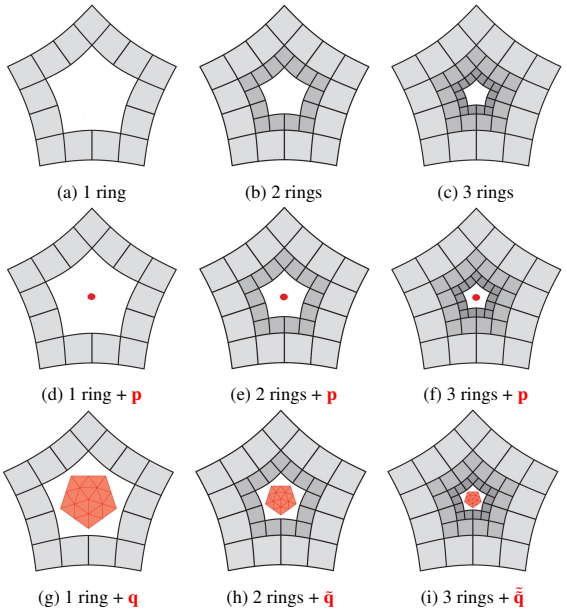


Figure 2: Separating Catmull-Clark step.

Tuning addresses shape problems near the limit extraordinary point by introducing new, additional refinement rules for the extraordinary node i.e. nodes with $n \neq 4$ neighbors. The larger footprint of the rules than CCS require wider separation of extraordinary nodes by regular 4-valent nodes. Quad-meshing algorithms [12, 13, 14, 15] typically do not separate sufficiently when the output has to have a low number of quads to be suitable for design. Therefore one or more initial CCS refinement steps need to be applied, cf. Fig. 2. However CCS is prone to artifacts introduced by the first refinement steps [16]. Fundamentally, the large scale 'first step artifacts' are not mended by algorithms focused on limit tuning [9, 10] as demonstrated, e.g. in [11, Fig 1d] for [9]; see also the eigen-polyhedron approach of [17] to address shape problems of NURCCS [18] in the difficult context of irregular knot sequences. While the repeated averaging in [10] improves the shape, this comes at the cost of increasing polynomial degree.

Guided Subdivision [19] harnesses a larger number of degrees of freedom by first computing, from the control net, a fixed prototype shape called the guide. A refinement step can then be viewed as smooth prolongation of an existing spline surface towards the guide shape with any remaining degrees of freedom set by shape information from smartly sampling of the guide.



(a) 1 ring (b) 2 rings (c) 3 rings
(d) 1 ring + **p** (e) 2 rings + **p** (f) 3 rings + **p**
(g) 1 ring + **q** (h) 2 rings + $\tilde{\mathbf{q}}$ (i) 3 rings + $\tilde{\tilde{\mathbf{q}}}$

Figure 3: Evolution of subdivision algorithms. (Note: these are structure diagrams not flattened surfaces.) *Row 1*: Catmull-Clark (CCS) surface as a sequence of contracting rings. *Row 2*: [11]: adds a carefully-chosen *fixed* central point **p**, marked as •. This point becomes a limit point of the subdivision surface. *Row 3*: [8] introduces a quadratic central expansion **q**, marked as ⬡ that evolves and attracts the surface.

Fig. 4. This approach can be traced back, for polynomial guides to [20, 21] and for piecewise polynomial guides to [19]. While Guided Subdivision improves the shape, it consists of two distinct stages: the guide construction followed by the construction of contracting rings. In more detail, (a) a **c**-net, see Fig. 5 d, pro-

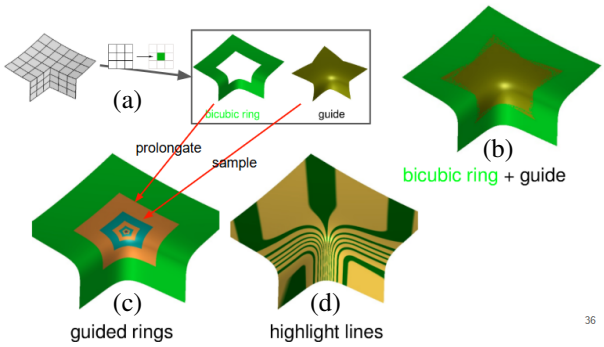
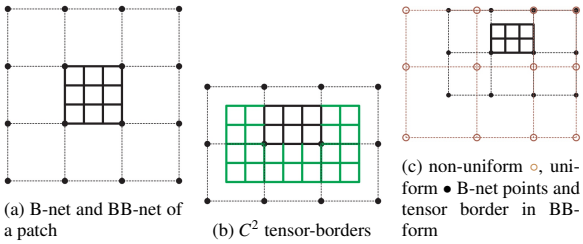


Figure 4: Guided subdivision separates shape-finding from the mathematical constraints of the final output surface.

vides both a surface ring and a guide shape. These pieces of geometry are close but do not join consistently as the pixel-fighting in (b) illustrates. Instead Guided Subdivision smoothly prolongs the existing surface and supplements it with data from the guide to complete (c) a new (bronze) ring that smoothly joins the existing (green) ring and follows the shape of the guide by sampling the guide for data needed to complete the new ring. This approach generically yields excellent highlight lines, see (d).

[11] fold the guide and prolongation into a single refinement step. This refined control net is interpreted, where possible, as a bi-cubic spline control net. Repeating this process defines a sequence of surface rings that contract towards the limit extraordinary point, see Fig. 3. That is, the algorithm behaves exactly like classical subdivision, but yields better shape without a

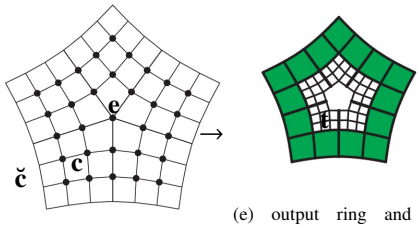
separately-constructed explicit guide: Replacing the augmenting limit point **p** in row 2 of Fig. 3 by a quadratic expansion **q** in row 3, the algorithm in [8] improves smoothness and increases design flexibility at the extraordinary point. [22] additionally achieves a uniform contraction speed. FASt combines these advances into an efficient, bounded curvature subdivision algorithm at less than half the cost in spline ring pieces.



(a) B-net and BB-net of a patch

(b) C^2 tensor-borders

(c) non-uniform ○, uniform • B-net points and tensor border in BB-form



(d) **c**-net ⊂ **č**-net

(e) output ring and tensor-border **t**

Figure 5: B-to-BB conversion. (a) A 4×4 B-net with control points (in $\mathbb{R}^3$ indicated as • defines one bi-cubic patch whose BB-net is shown as the solid inner 4×4 grid. (b) A partial B-net still defines a C^2 expansion at the central edge (black solid BB-net) as well as its green counterpart below. Expansions at the endpoints of the edge are 3×3 green BB-nets. (c) The *non-uniform* knot spacing B-spline net ○ is transformed to the uniform B-spline net with nodes marked •. Both nets define the same tensor-border. (d) The central irregular node of valence n is labeled **e**. The quad-framed enlargement **č** of a (Catmull-Clark) **c**-net contains $3n$ overlapping 4×4 B-spline control nets. A **č**-net therefore defines (e) a surface ring of $3n$ C^2 bi-3 patches and the tensor-border **t**.

2. Control nets and Piecewise Polynomials

This section reviews basic control structures for subdivision surfaces, summarized in Fig. 5 , and so introduces the notation for the derivation of FASt in Section 3. Fig. 5 a illustrates the well-established B-to-BB conversion, that changes the basis from the control net of uniform bi-cubic B-splines (short B-net) [23]

$$\mathbf{f}(u,v) := \sum_{i=0}^{3}\sum_{j=0}^{3} \mathbf{d}_{ij} N^3(u-i) N^3(v-j), \quad 0 \le u,v \le 1,$$

to the Bernstein-Bézier form (short BB-net) [24, 25]):

$$\mathbf{f}(u,v) := \sum_{i=0}^{3}\sum_{j=0}^{3} \mathbf{f}_{ij} B_i^3(u) B_j^3(v), \quad 0 \le u,v \le 1.$$

Here $N^3(t)$ is the uniform degree 3 B-spline, the $\mathbf{d}_{ij} \in \mathbb{R}^3$ (marked • in Fig. 5) are *B-spline control points*, $B_k^d(t) := \binom{d}{k}(1-t)^{d-k}t^k$ are Bernstein polynomials of degree d, and $\mathbf{f}_{ij} \in \mathbb{R}^3$ are called *BB-coefficients*. Connecting $\mathbf{d}_{ij}$ to $\mathbf{d}_{i+1,j}$ and $\mathbf{d}_{i,j+1}$ wherever possible yields the *B-net* of $\mathbf{f}$, displayed as thin grid in Fig. 5 a,b,c,d; and connecting $\mathbf{f}_{ij}$ to $\mathbf{f}_{i+1,j}$ and $\mathbf{f}_{i,j+1}$ wherever possible yields the *BB-net* of $\mathbf{f}$, displayed as solid black or green grids in Fig. 5 a,b,c,e. The B-to-BB conversion formulas are the tensor-product of the univariate formulas that set $\mathbf{f}_{3i} := (\mathbf{d}_{i-1} + 4\mathbf{d}_i + \mathbf{d}_{i+1})/6$ and $\mathbf{f}_{3i+1} := (2\mathbf{d}_i + \mathbf{d}_{i+1})/3$ (all other formulas follow by shift and rotational symmetry).

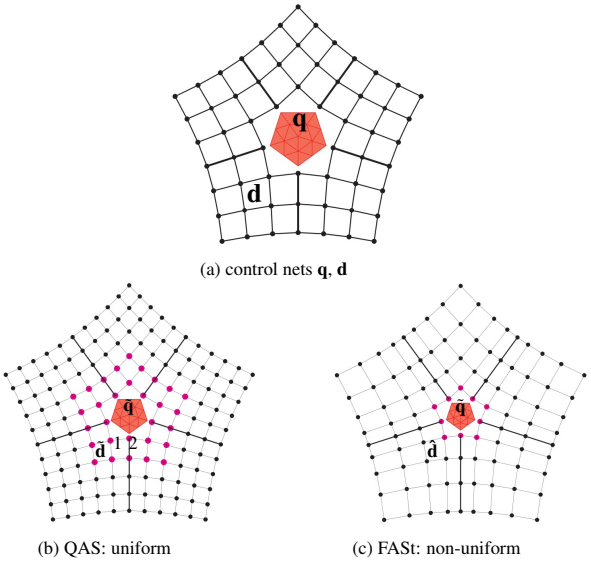
Fig. 5 b illustrates how a partial B-net still determines a BB-net that defines a C^2 expansion at the central edge (black solid BB-net). Other expansions fully defined by the B-net are indicated as green BB-nets. Fig. 5 c shows the same BB-net constructed both from a non-uniform and a uniform B-net.

The **č**-net in Fig. 5 d enlarges the classic Catmull-Clark **c**-net by one ring of quadrilaterals so that B-to-BB conversion defines $3n$ pieces of a C^2 uniform bi-cubic B-spline surface ring and their C^2 prolongation across edges, the tensor-border **t**.

3. The FASt Construction

This section contains the *technical derivation* of the FASt algorithm. An end user can skip this section since the design control-net remains the familiar Catmull-Clark control net **c**. Moreover, the derivation is not needed for implementation since FASt refinement can, for each valence n, be expressed as a linear transformation, namely a matrix $\mathbf{M}_n$ of size $\bar{n} \times \bar{n}$, where $\bar{n} := 1 + (12+5)n$, All non-zero $\mathbf{M}_n$ entries (other than (1)) for implementation up to $n = 10$ are provided in Tables DQ^n and D^n of the Appendix. The FASt control net is refined by multiplying its control net by $\mathbf{M}_n$. An initial step, also a matrix multiplication, converts the Catmull-Clark control net **c** to the FASt control net.

For the specialist, this section offers some, possibly surprising, technical innovations – such as the non-smooth choice of the central guide and an intermediate non-uniform refinement step.



(a) control nets **q**, **d**

(b) QAS: uniform

(c) FASt: non-uniform

Figure 6: Subdivision control nets. (a) $12n$ nodes of the FASt control net **d** and the n sectors of the quadratic **q**. (b) Refinement of [8] (resulting in $12n$ patches per refinement step). (c) FASt: intermittent non-uniform refinement (resulting in only $5n$ patches per step).

The (internal) FASt control net consists of two sub-nets, **d** and **q**, see Fig. 6 a. The two sub-nets collaborate to yield the control nets $\tilde{\mathbf{d}}$ and $\tilde{\mathbf{q}}$ of the next nested surface ring. This two-net structure was introduced in [8] to separate prolongation of the surface and its convergence and attraction towards the limit shape. Keeping the two nets (**q**, **d**) separate also simplifies the limit analysis of the resulting surface in Section 4.

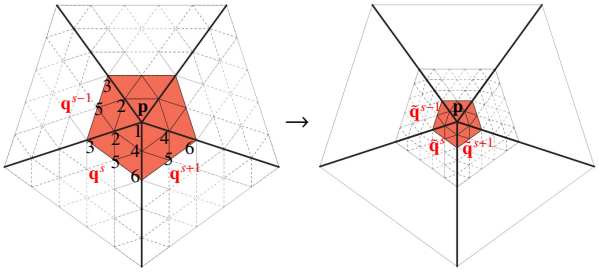
The design control-net is still the familiar Catmull-Clark **c**-net. That is, shape manipulation is at the level of the **c**-net and this

automatically initializes (**d**, **q**), see Section 3.2. That is, the (**d**, **q**) pair is typically not exposed for modeling. In each subdivision step, (**d**, **q**) is refined and generates one surface ring.

3.1. *Overview of the FASt derivation*

The d-net refinement. The **d**-*net* refinement is *non-uniform*, see Fig. 6 c. This non-uniformity distinguishes FASt from earlier quadratic-attraction subdivision algorithms where, cf. Fig. 6 b, $24n$ refined nodes • are obtained by uniform refinement rules, i.e. new knots are inserted as midpoints into each interval, and $6n$ new nodes • are calculated by 6 special rules. The $6n$ • and the $6n$ innermost • form a net denoted by $\tilde{\mathbf{d}}$ that is the refined net and serves as the input for next refinement step.

By contrast, FASt, cf. Fig. 6 c, generates $18n$ new points • via *non-uniform* refinement rules. New knots are inserted as midpoints only in some intervals and $2n$ new • are calculated by 2 special rules. The $2n$ • and the $10n$ innermost • form a net denoted $\hat{\mathbf{d}}$. Unlike the uniform case (b), $\hat{\mathbf{d}}$ has a non-uniform structure and therefore can neither be the input $\tilde{\mathbf{d}}$-net for next refinement step nor define the surface rings of *uniform* bi-cubic B-splines. Section 3.3 will explain how to convert $\hat{\mathbf{d}}$ to $\tilde{\mathbf{d}}$.



(a) labeling (b) refinement

Figure 7: A quadratic expansion **q** with central point $\mathbf{p} = \mathbf{q}^s{}_1$, $s = 1, \ldots, 5$. (a) Indexing. (b) Refinement $\mathbf{q} \to \tilde{\mathbf{q}}$ according to (1),

The q-net refinement. The **q**-*net* of the *quadratic expansion* connects the BB-coefficients of the inner brick-red underlaid 2-link of the polynomial pieces in Fig. 7 a. Fig. 7 a displays the BB-nets of $n = 5$ polynomials of degree $d = 5$ in *total degree* Bernstein-Bézier form [25],

$$\mathbf{b}(u,v) := \sum_{\substack{i+j+k=d \\ i,j,k\geq 0}} \mathbf{b}_{ijk} \frac{d!}{i!j!k!}(1-u-v)^k u^i v^j, \quad 0 \leq u+v \leq 1.$$

(Degree 5 stems from bi-5 guide construction in [8] that, when baked into the rules, yields high-quality subdivision surfaces.)

Connecting the quadratic pieces C^1 only at the central point **p** and C^0 otherwise is permissible and adds flexibility over joining the pieces C^2. Notably, this lack of smoothness in **q** does not harm the final surface quality since **q** only guides the shape whereas the smoothness of the surface is determined by other algorithmic steps that encode C^2 prolongation of the existing surface. Fig. 7 illustrates how to obtain the BB-coefficients $\tilde{\mathbf{q}}_i^s$ of the refined quadratic expansion in terms of **q** by scaling the domain of **b** towards the origin by a factor λ

$$\begin{pmatrix} \tilde{\mathbf{q}}_1^s \\ \tilde{\mathbf{q}}_2^s \\ \tilde{\mathbf{q}}_3^s \\ \tilde{\mathbf{q}}_4^s \\ \tilde{\mathbf{q}}_5^s \\ \tilde{\mathbf{q}}_6^s \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1-\lambda & \lambda & 0 & 0 & 0 & 0 \\ (1-\lambda)^2 & 2(1-\lambda)\lambda & \lambda^2 & 0 & 0 & 0 \\ 1-\lambda & 0 & 0 & \lambda & 0 & 0 \\ (1-\lambda)^2 & (1-\lambda)\lambda & 0 & (1-\lambda)\lambda & \lambda^2 & 0 \\ (1-\lambda)^2 & 0 & 0 & 2(1-\lambda)\lambda & 0 & \lambda^2 \end{pmatrix} \begin{pmatrix} \mathbf{q}_1^s \\ \mathbf{q}_2^s \\ \mathbf{q}_3^s \\ \mathbf{q}_4^s \\ \mathbf{q}_5^s \\ \mathbf{q}_6^s \end{pmatrix}. \qquad (1)$$

This recalculation of **q** on the scaled domain leaves unchanged the quadratic expansion *jet* at the central point. Therefore the subdivision algorithm is stationary, and hence easy to implement and analyze, see Section 4.

3.2. *Initialization of* **q** *and* **d** *from a* **c**-*net*

While the subdivision algorithm is perfectly well-defined in terms of the contracting **q** and **d** nets, quad-faceted models typically have the structure of a **c**-net as in Fig. 5 d and only this **c**-net will be exposed to the designer.

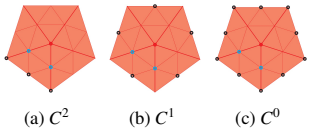
The **initial d- net** is obtained from **c** by uniform bi-3 C^2 B-spline refinement (knot insertion at the midpoint) applied wherever possible. The **d**-net labels are shown in Fig. 9 b.

Given a **c**-net with central irregular node **e**, we compute the **central point** $\mathbf{q}_1^s := \mathbf{p}$, $s = 0, \ldots, n-1$ as the limit point of Catmull-Clark subdivision [26], slightly improved only for $n = 3$ as in [11], and the **initial quadratic expansion** as an average of **e** and the surrounding **c**-net nodes $\mathbf{c}_i^{s+j}$ where the superscript is modulo n:

$$\mathbf{q}_k^s := (1 - \sum_{i=1}^{6}\sum_{j=0}^{n-1} \mu_{ij}^k)\,\mathbf{e} \;+\; \sum_{i=1}^{6}\sum_{j=0}^{n-1} \mu_{ij}^k\,\mathbf{c}_i^{s+j}, \; k = 2,3,5, \qquad (2)$$

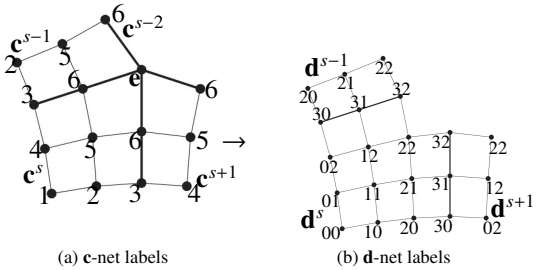
$\mathbf{q}_4^{s-1} := \mathbf{q}_2^s$, $\mathbf{q}_6^{s-1} := \mathbf{q}_3^s$, $s = 0, \ldots, n-1$, and μ listed in Appendix 9.2. This explicit formula is derived just as in [8, Steps S1,S2,S3], and hence is equally able to produce high-quality limit surfaces. However, (2) is more succinct. This **q** is the default for FASt. We refer to it as C^2, even though, formally, this **q** is C^1 and C^2 only to 5 decimal accuracy. Since **q** serves only as a guide, this slight deviation does not affect the surface quality or smoothness of the FASt surface.

By removing constraints on **q**, FASt offers three choices, see Fig. 8 :
(a) **q** is C^2.
(b) **q** is C^1.
(c) **q** is C^1 only at the center and C^0 otherwise.
The first option (a) is that of (2), the second (b) frees up the mixed derivatives and the last option (c) offers the designer the most freedom, see Fig. 23 . The non-smooth choice of **q** is permissible since the subdivision surface construction takes care of smoothness within and between the rings and **q** only provides samples that guide the overall shape. Notably **q** can even be discontinuous, but discontinuity does not benefit design and the construction becomes more cumbersome.



(a) C^2 (b) C^1 (c) C^0

Figure 8: Types of quadratic expansions **q** at the central extraordinary point **p**. Bullets •, ∘ and ⚬ mark unconstrained BB-coefficients; • and ∘ define a tangent plane at **p**.



(a) **c**-net labels (b) **d**-net labels

Figure 9: Initialization **c**-net → **d**-net.

3.3. *Derivation of the FASt rules*

This sub-section motivates and derives the FASt rules. It is intended for the expert. Implementation of the rules is much simpler, see the gray underlaid FASt Algorithm at the end of this sub-section. Usage is identical to (or more flexible if modifying **q** than) Catmull-Clark subdivision.

Fig. 10 a shows the initial **d**-net with the length-ratios '1' of neighboring knot intervals (in directions marked → and ↑ in Fig. 10 b). Here ratios '1' indicate uniformly-spaced knots. This *uniform* **d**-net is *non-uniformly* refined to the net $\mathbf{h}\!\!\!-$. The intent of halving the knot intervals where the next, contracted ring is to be attached, is to anticipate the split of the next ring (see | and – in Fig. 10 a) and distribute the structural change between the rings. The new nodes of $\mathbf{h}\!\!\!-$ are displayed as • in Fig. 10 b. The explicit formulas of this non-uniform refinement are presented in Appendix 9.3. (Only) the nodes marked as $\bullet_1$ and $\bullet_2$ in Fig. 10 b are defined by special rules in terms of **d** and **q** with the best outcome when set as in Appendix 9.1, essentially identical to [8]. For the larger picture, markers 1,2 are also displayed in Fig. 6 b. Together, $\mathbf{h}\!\!\!-$ and • form one sector's net **h**. We see that **h** contains five overlapping 4×4 sub-nets of non-uniform bi-cubic B-splines with knot sequences:
$\{(1,1,\frac{1}{2},1),(1,1,\frac{1}{2},1)\}$, {(in → direction), (in ↑ direction)}
$\{(1,\frac{1}{2},1,1),(1,1,\frac{1}{2},1)\}$, $\{(\frac{1}{2},1,1,1),(1,1,\frac{1}{2},1)\}$,
$\{(1,1,\frac{1}{2},1),(1,\frac{1}{2},1,1)\}$, $\{(1,1,\frac{1}{2},1),(\frac{1}{2},1,1,1)\}$.
Each 4×4 sub-net defines a bi-cubic patch (as in Fig. 5). Considering all n sectors this yields $5n$ bi-3 patches of one surface ring. By B-to-BB conversion for non-uniform B-splines [27] these patches have the BB-nets illustrated as thin grids in Fig. 10 c.

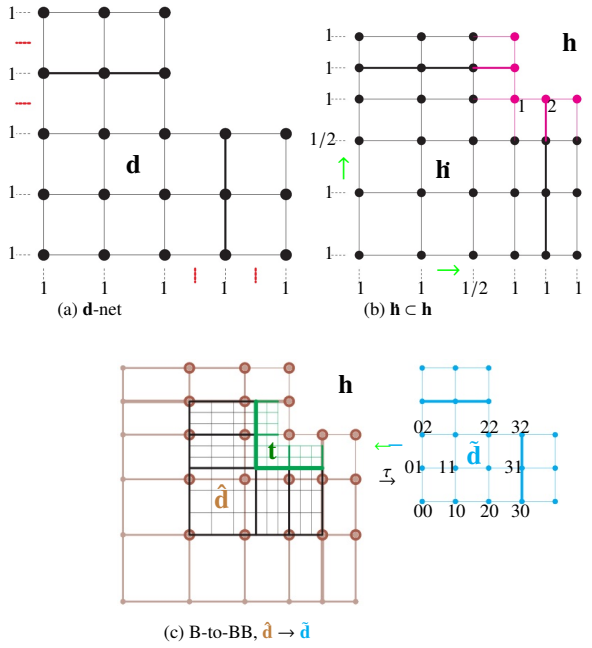


Figure 10: Derivation of FASt refinement step rules $\mathbf{d} \to \check{\mathbf{d}}$. (a) Uniform **d**-net where | and – indicate knot insertion at interval midpoints to obtain (b) the non-uniform B-spline net $\mathbf{h}\!\!\!-$. The extension of $\mathbf{h}\!\!\!-$ by • – defined by the special rules of Appendix 9.1 – yields **h**. (c, *left*) Non-uniform B-to-BB conversion of the sub-net $\hat{\mathbf{d}}$ (with nodes •) to the BB-nets (light gray lines) of BB-patches (outlined in solid black). The green BB-net is the tensor-border **t**. (c,*right*) The transformation to the uniform net $\tau : \hat{\mathbf{d}} \to \check{\mathbf{d}}$. The green-cyan back-arrow indicates the extraction of a tensor-border **t** from $\check{\mathbf{d}}$ by uniform B-to-BB conversion (whose agreement with the tensor-border **t** obtained from $\hat{\mathbf{d}}$ provides the constraints that determine $\check{\mathbf{d}}$).

The second step prepares for the next refinement by removing the non-uniformity. The non-uniform subnet $\hat{\mathbf{d}}$ of **h**, formed by the nodes marked as ○ in Fig. 10 c, is transformed, as detailed in Eqs. (3), into the uniform net $\check{\mathbf{d}}$ *preserving* the tensor border **t** defined by $\hat{\mathbf{d}}$ and colored green in Fig. 10 c. The labels shown in Fig. 10 (c,*right*) apply to both $\hat{\mathbf{d}}$ and $\check{\mathbf{d}}$. Enforcing that the tensor-border (obtained from $\check{\mathbf{d}}$ by uniform B-to-BB rules) coincides with the tensor-border **t** (obtained from $\hat{\mathbf{d}}$ via B-to-BB rules for non-uniform B-splines) uniquely defines the transformation τ of $\hat{\mathbf{d}}$ to $\check{\mathbf{d}}$:

$$
\begin{aligned}
\check{\mathbf{d}}_{i2} &:= \hat{\mathbf{d}}_{i2},\ \check{\mathbf{d}}_{i1} := \frac{3\hat{\mathbf{d}}_{i1} + \hat{\mathbf{d}}_{i2}}{4}, \check{\mathbf{d}}_{i0} := \frac{2\hat{\mathbf{d}}_{i0} + 3\hat{\mathbf{d}}_{i1}}{5},\ i = 2,3; \\
\check{\mathbf{d}}_{11} &:= \frac{9\hat{\mathbf{d}}_{11} + 3(\hat{\mathbf{d}}_{21} + \hat{\mathbf{d}}_{12}) + \hat{\mathbf{d}}_{22}}{16}, \\
\check{\mathbf{d}}_{00} &:= \frac{4\hat{\mathbf{d}}_{00} + 6(\hat{\mathbf{d}}_{10} + \hat{\mathbf{d}}_{01}) + 9\hat{\mathbf{d}}_{11}}{25}, \\
\check{\mathbf{d}}_{10} &:= \frac{6\hat{\mathbf{d}}_{10} + 2\hat{\mathbf{d}}_{20} + 9\hat{\mathbf{d}}_{11} + 3\hat{\mathbf{d}}_{21}}{20}.
\end{aligned}
\tag{3}
$$

The remaining $\check{\mathbf{d}}_{ij}$ are defined by ij to ji 'diagonal' symmetry.

Since **t** is retained, the just-constructed surface ring joins the next, still to be constructed bi-3 ring C^2 along their common boundary (marked as thick green lines in Fig. 10 c). We summarize:

FASt Algorithm

If the input is a **c**-net, generate **d** and **q** according to Section 3.2. Iterate $(\mathbf{q},\mathbf{d}) \to (\tilde{\mathbf{q}},\check{\mathbf{d}})$:

a. $\mathbf{d} \to \mathbf{h} := \mathbf{h}\!\!\!- \cup \bullet$: Determine
the net $\mathbf{h}\!\!\!-$ by the formulas of Appendix 9.3 and
the nodes • by the formulas in Appendix 9.1.
b. $\mathbf{h} \to \mathbf{b}$: Convert
the non-uniform B-spline net **h** into $5n$ bi-cubic patches **b**.
c. $\hat{\mathbf{d}} \to \check{\mathbf{d}}$: by formulas (3). Note: $\hat{\mathbf{d}} \subset \mathbf{h}$.
$\mathbf{q} \to \tilde{\mathbf{q}}$: by formulas (1).
d. Update $(\mathbf{q},\mathbf{d}) := (\tilde{\mathbf{q}},\check{\mathbf{d}})$.

We re-iterate that the exposition in this section serves exclusively to *derive and justify* the FASt algorithm. The individual steps need not be implemented since the resulting linear transformation $(\mathbf{q},\mathbf{d}) \to (\tilde{\mathbf{q}},\check{\mathbf{d}})$ is compactly available as a family of matrices $\mathbf{M}_n$ of size $\bar{n} \times \bar{n}$, where n denotes the valence and $\bar{n} := 1 + (12+5)n$. The sequence of control nets **q** suffices if, as is common for Catmull-Clark subdivision, we only want to display the refined, faceted mesh.

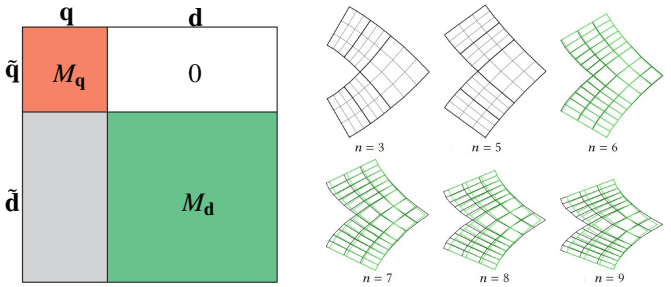
To generate the sequence of surface rings in BB-form, $(\mathbf{q},\mathbf{d}) \to \mathbf{b}$ has been encoded as a family of matrices $\mathbf{B}_n$, of size $(5n)16 \times \bar{n}$.

Two matrix multiplications therefore summarize the

FASt Algorithm: Iterate

i Output a surface ring: $\mathbf{b} = \mathbf{B}_n[\mathbf{q},\mathbf{d}]^t$.
ii Refine the control net: $[\mathbf{q},\mathbf{d}]^t \leftarrow \mathbf{M}_n[\mathbf{q},\mathbf{d}]^t$.

The non-zero entries of the matrices are listed in the Appendix.



(a) matrix M (b) characteristic maps

Figure 11: (a) Subdivision matrix M for limit analysis. (b) Characteristic maps: FASt in black; CCS, with patches split to have the same partition, in green. (For $n = 3$ and $n = 5$ the FASt characteristic map is visually identical to split CCS.)

4. Limit analysis

The split of the control net into **d** and **q** endows the subdivision matrix $\mathbf{M}_n$ with the structure displayed in Fig. 11 a. For a given n, let $M := \mathbf{M}_n$. Since the upper right submatrix of M is zero, the eigen-analysis of M splits into the eigen-analysis of the submatrices $M_{\mathbf{q}}$ and $M_{\mathbf{d}}$. The absolute maximal eigenvalues m_n of $M_{\mathbf{d}}$ (computed numerically) are

$$\begin{pmatrix} n & 3 & 5 & 6 & 7 & 8 & 9 & 10 \\ m_n & 0.151 & 0.179 & 0.196 & 0.214 & 0.225 & 0.232 & 0.238 \end{pmatrix}.$$

For $M_{\mathbf{q}}$, we can read off the refinement relation (1) that the leading terms of the eigenspectrum are (for each of three options of **q** with C^0 denoting the **q** that is C^1 only at the central point):

(a) **q** is C^2 : eigenvalues $1, \lambda, \lambda, \lambda^2$ (3-fold);
(b) **q** is C^1 : eigenvalues $1, \lambda, \lambda, \lambda^2$ (n-fold);
(c) **q** is C^0 : eigenvalues $1, \lambda, \lambda, \lambda^2$ ($2n$-fold).

Recall that λ is free to choose. When $\lambda \geq 1/2$ then m_n is less than λ^2 and the eigenvalues of $M_{\mathbf{q}}$ strictly dominate those of $M_{\mathbf{d}}$. That is, for $\lambda \geq 1/2$ the subdivision surface is curvature-bounded [7]. Note that $\lambda := \frac{1}{2}$ corresponds to uniform bi-cubic B-spline subdivision. However, for empirically *best shape*. we set the FASt default as

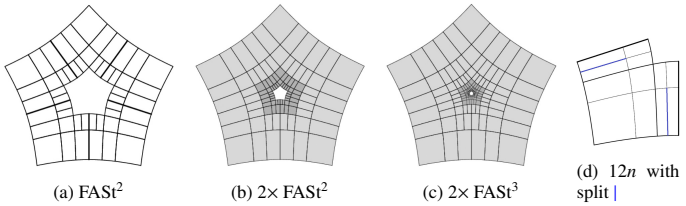
$$\lambda := \begin{cases} \lambda_{CC}, & \text{for } n = 3, 5, \\ \lambda_{FASt} := 0.57, & \text{for } n = 6, \ldots, 10. \end{cases} \quad (4)$$

(For comparison: the λ in PAS and QAS increases and is close to that of CCS. For $n = 3, 5$, $\lambda_{CC} < 0.57$.)

The limit analysis is completed by numerically establishing injectivity of the characteristic maps of FASt, black in Fig. 11 b. The outcome is not surprising given the injectivity of the characteristic map χ_{CC} of CCS and the fact that both maps look very similar, see Fig. 11 b, after two of the bi-3 pieces in each sector of χ_{CC} are uniformly split to yield the same layout of $5n$ bi-cubics, as for the FASt characteristic map. As intended, the lower value $\lambda = 0.57$ for $n > 5$ results in higher contraction speed that stretches the maps compared to CCS.

5. Speed Comparison: Merging Consecutive Rings

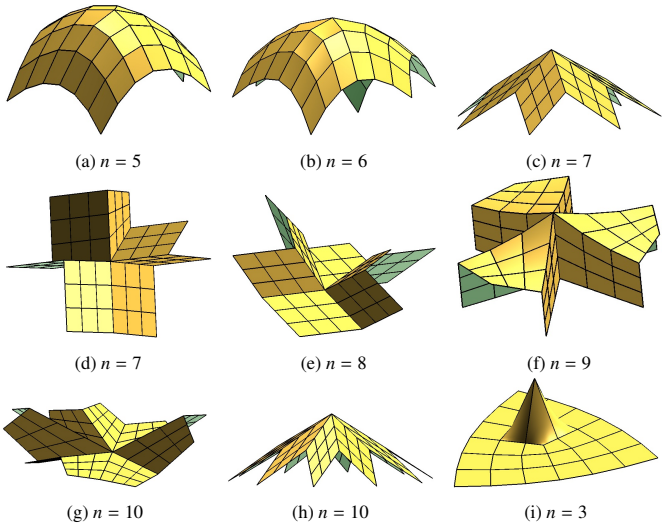
One way to visualize the savings of $5n$ pieces per ring for FASt, compared to alternative tuned subdivision [9] (MA18), [28] (MA19) and [29] (WM23) or QAS that each require $12n$ pieces, is to join two consecutive FASt rings to obtain $10n$ (FASt^2, Fig. 12 a,b), respectively three rings yielding $15n$ pieces for the triple ring (FASt^3, Fig. 12 c). Fig. 12 illustrates the very fast multi-step convergence using roughly the same number of pieces as the alternatives.



(a) FASt^2 (b) 2× FASt^2 (c) 2× FASt^3 (d) $12n$ with split |

Figure 12: Alternative point of view: merging FASt rings. (a) Merging consecutive FASt surface rings to yield $10n$ patches of one FASt^2 ring, i.e. fewer patches than the $12n$ patches for alternative algorithms. (b) Two FASt^2 rings. (c) Two FASt^3 rings, i.e. $15n$ pieces. (d) The blue uniform split (in one direction) gives FASt^2 the same $12n$ patch layout as tuned or QAS alternatives, to mix and match.

The eigen-structure of FASt^2 corresponds to the subdivision matrix M^2, i.e. the eigenvalues are squared and two nested rings form the characteristic map of FASt^2. FASt^2 produces the same curvature-bounded subdivision surfaces as FASt but with faster contraction due to $\lambda_{\text{FASt}^2} = \lambda^2_{\text{EAs}} < \frac{1}{3}$ for all $n \leq 10$. For $n = 3$, $\lambda_{\text{FASt}^2} \approx 0.1682$; for $n = 5$, $\lambda_{\text{FASt}^2} \approx 0.3025$; for $n = 6, \ldots, 10$ $\lambda_{\text{FASt}^2} \approx 0.3249$. That is FASt^2 achieves high contraction speed, useful for (isogeometric) engineering analysis, see Fig. 26 , while maintaining the good shape and even compatibility with the $12n$ framework of other curvature-bounded bi-cubic subdivision algorithms: Fig. 12 d shows how $10n$ patches can be split into $12n$ patches with the required structure.



(a) $n = 5$ (b) $n = 6$ (c) $n = 7$
(d) $n = 7$ (e) $n = 8$ (f) $n = 9$
(g) $n = 10$ (h) $n = 10$ (i) $n = 3$

Figure 13: Challenging **c**-nets: convexity preservation and higher-order-saddles.

6. Comparison to QAS and tuned subdivision

Since artistic perception is not universal, there are no agreed-upon worst case input scenarios. However, systematic testing points to the cases interrogated in the following shape gallery: convex input that should result in uniformly varying convex surfaces and higher-order, high-valence saddles, that should avoid pinching of highlight lines. Specifically, we compare FASt surfaces on the 9 challenging $\check{\mathbf{c}}$-nets, displayed in Fig. 13 , to surfaces generated by CCS, QAS, MA18 [9], MA19 [28] and

WM23 [29]. The goal of comparing FASt to QAS is to show that outcomes are of very similar quality despite using only 5/12 as many pieces. As is common for design, surface quality is measured as uniformity of the distribution of the surface highlight lines [30]. The set of 9 **č**-nets was selected as representative of a larger test set: Fig. 13 a,b,c,h outline convex shapes, Fig. 13 d,e,f,g outline higher-order-saddles and Fig. 13 i is a spike on an otherwise planar configuration. The n planar 4×4 subgrids in (c,d,e,g,h,i) and bilinear in (f) yield a predictable surface ring of regular bi-cubic patches, namely n planar (for (f) bilinear) pieces and transitions between them. This ring is shown in green surrounding the red subdivision region in images like Fig. 14 a that delineate the adjacent highlight line displays. The galleries demonstrate the following properties.

- Despite using only $5n$ pieces in place of $12n$, the shape of FASt is essentially on par with QAS.
- The fewer piece advantage of FASt becomes evident when merging two consecutive rings to $10n$ patches in FASt^2: very fast convergence compared to $12n$ patch algorithms.
- The shape of FASt surfaces is superior to tuned, bounded-curvature subdivision MA18, MA19 and WM23. These either suffer from first-step artifacts or have unwanted oscillations.

We display surfaces with 7 rings generated by CCS, MA18, MA19, WM23 and QAS subdivision and 6 FASt^2 rings. Zoom shows the innermost 3 rings, with the star-shaped hole indicating the finite contraction and faster contraction speed of FASt^2.

In Fig. 14 and Fig. 15, we focus the neighborhood of the extraordinary point, marked for example in (c) as a □-framed area. While the shape of MA18 is good for $n = 5$, it deteriorates for $n = 6$, Fig. 15. Already for $n = 5$, in Fig. 14, highlight lines on CCS (and WM23) slightly pinch together, MA19 develops oscillations, and WM23 is visibly not suitable for high-end design, see Fig. 14 (e). By contrast, QAS and FASt^2 show no artifacts. In (j-q), we therefore focus on the rings 5,6,7. (j) confirms CCS pinching, (k) clarifies that MA18 is not satisfactory after all. (Indeed, with increasing rings index, MA18 becomes worse and MA19 and WM23 are visually no better, or worse than CCS.) On the other hand, QAS and FASt^2 retain uniform highlight lines.

Increasing the valence to $n = 6$ in Fig. 15 makes the flaws observed for $n = 5$ more pronounced: CCS shows increased pinching of highlight lines, MA18 oscillates stronger and MA19, WM23 are clearly unsuitable for high-end design. We therefore only zoom in for QAS and FASt^2: Fig. 15 (j-l) show similar, high quality.

For the convex $n = 7$ configuration in Fig. 16, the flaws increase for CCS, MA18, MA19, WM23. That is, already for $n = 5, 6, 7$ tuned subdivision fails to deliver high-quality, uniform highlight line distributions, while QAS and FASt^2 succeed.

Starting with Fig. 17, the higher-order saddle configuration for $n \geq 7$ no longer require zoom for CCS, MA18, MA19, and WM23: all have regions of oscillating highlight lines at a frequency much higher than the input net indicates. Notably, the oscillation in the framed area in Fig. 17, away from the center, cannot be removed by the limit-focused tuned approaches. Again, FASt^2 is on par with QAS and has calm highlight lines.

Fig. 18, for $n = 8$, indicates that the findings for Fig. 17, for $n = 7$, are more pronounced as the valence increases. This is confirmed by Fig. 19 - surprisingly, for these high valences, MA19 and WM23 fare worse than CCS. The highlight lines of QAS and FASt^2 look optimal and similar and this 'calmness of

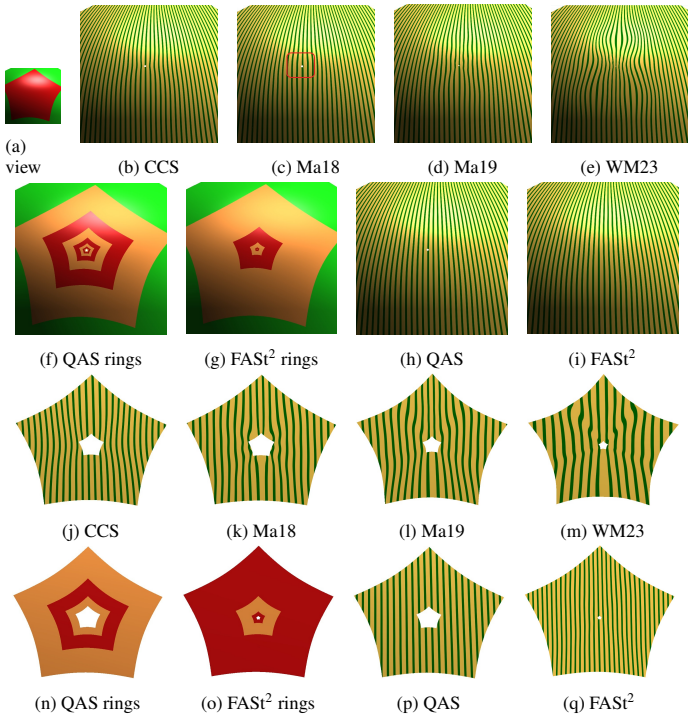


Figure 14: Convex $n = 5$ configuration input net Fig. 13 a. Row 1 and 2 show the area indicated by (a), row 3,4 the last three rings.

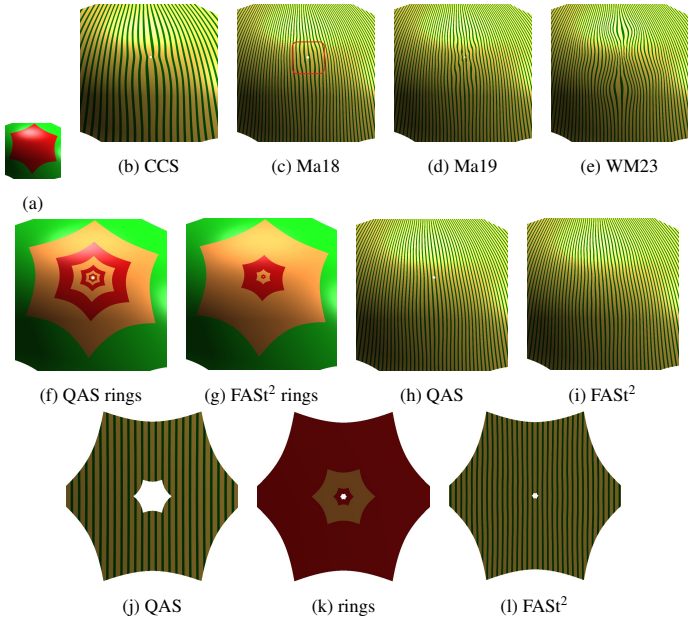
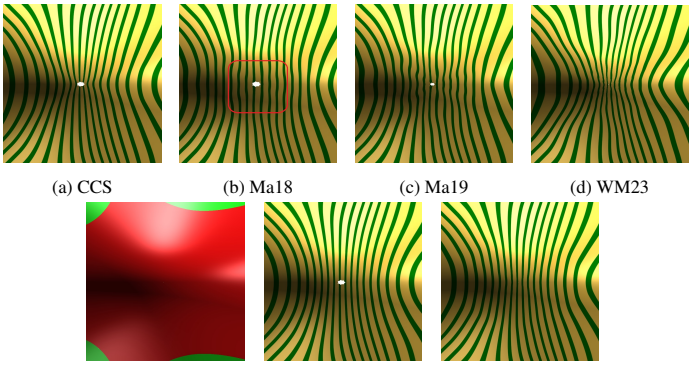


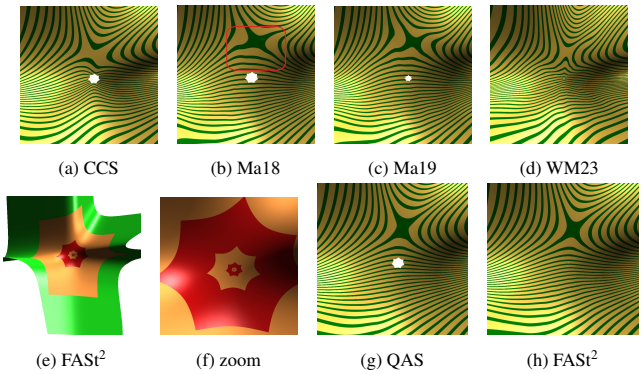
Figure 15: Convex $n = 6$ configuration input net Fig. 13 a. Row 1 and 2 show the area indicated by (a), row 3 the last three rings.



(a) CCS (b) Ma18 (c) Ma19 (d) WM23

(e) view (f) QAS (g) FASt^2

Figure 16: Convex $n = 7$ configuration input net Fig. 13 c.



(a) CCS (b) Ma18 (c) Ma19 (d) WM23

(e) FASt^2 (f) zoom (g) QAS (h) FASt^2

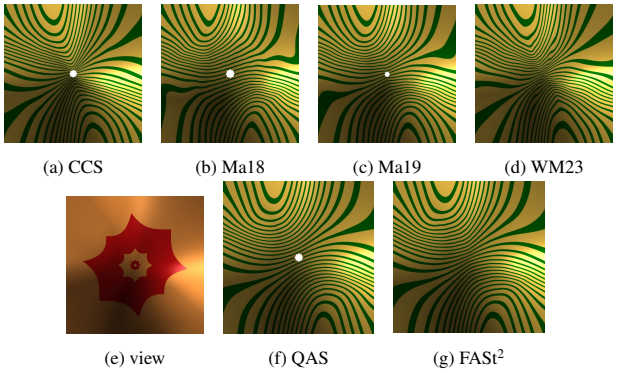
Figure 17: $n = 7$ configuration input net Fig. 13 d.

the highlight lines' is retained also for the $n = 10$ configuration of Fig. 20 .

Fig. 21 has minimally separated nodes of valence $n = 3, 5, 6$ (no quad contains more than one irregular node) so that for MA18, MA19 and WM23 one CCS refinement step must be applied. The arrow in (h) points to an unwanted ripple for CCS in the gold 5-sided area in (g). The red frame in (j) marks the vicinity of the extraordinary point where highlight lines increasingly oscillate for MA18, MA19 and WM23 in (j–l). MA19 improves near the extraordinary point, but has a poor transition in the first rings, while WM23 is poor in both locations. By contrast, FASt shows evenly distributed highlight lines.

Fig. 22 focuses on $n = 3$ configurations that usually require special care in high-end design. To not disturb the sourrounding surface, MA18 requires an initial CCS step that negatively affects surface quality; in convex configurations the impact is not as pronounced, but in Fig. 13 i, the first-step artifacts are well visible. As for other valences, FASt^2 yields a good highlight line distribution, in Fig. 22 c, that can be improved by starting with one QAS refinement step, Fig. 22 d. Applying one QAS step is no problem, since QAS and FASt^2 rings are compatible due to sharing the same **d**-net.

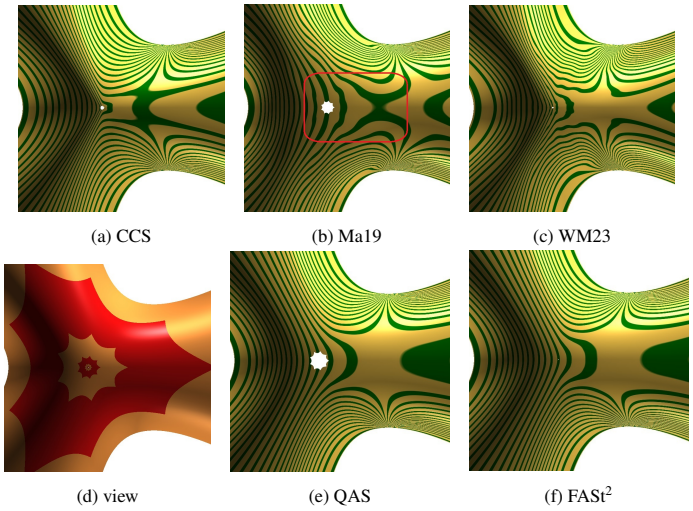
Fig. 23 compares two choices of **q**: a default option labeled (a) in Section 3.2 and the C^0 option labeled (c). The top row of Fig. 23 illustrates the default FASt with C^2 **q**: (a) the input **č**-net is a monkey saddle, $n = 6$; (b) shows the initialization of **q** according to Section 3.2 and Appendix 9.2 and the initialization of the **d**-net by uniform knot insertion in the **c**-net; (c) shows the calm highlight line distribution on the surface produced by FASt. The next two rows, Fig. 23 d–i, illustrate design based on modifying and placing a C^0 **q** without perturbing the input **d**-net. (Recall that C^0 is shorthand for **q** being C^1 at central point and C^0 across sector boundaries otherwise.) With $3 + 2n$ free coefficients, the C^0 **q** has n more free coefficients than the C^1 **q** to modify the surface shape. Fig. 23 f,i demonstrate that **q** *can* (but need not) be modified at more than one step of FASt.



(a) CCS (b) Ma18 (c) Ma19 (d) WM23

(e) view (f) QAS (g) FASt^2

Figure 18: $n = 8$ configuration input net Fig. 13 e.



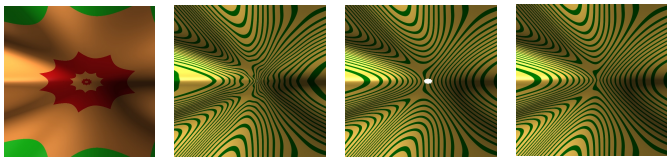
(a) CCS (b) Ma19 (c) WM23

(d) view (e) QAS (f) FASt^2

Figure 19: $n = 9$ configuration input net Fig. 13 f.

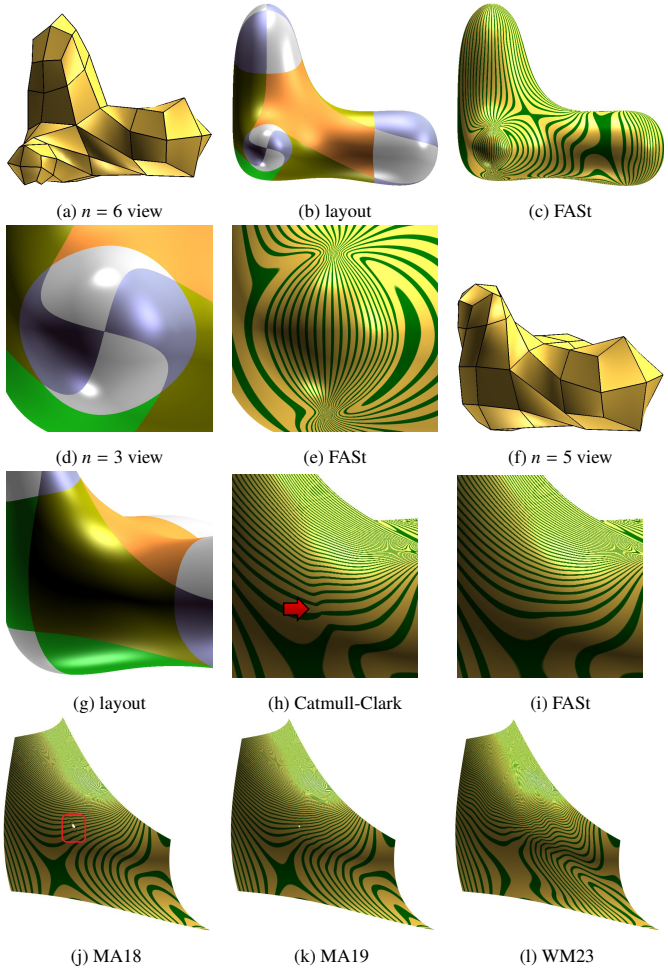
7. Speed, Export, Texture and Finite Elements

The output of FASt consists of standard bi-cubic patches. These can be grouped together as B-spline surfaces or individually output as pieces in Bézier form. That is, the rings of the FASt surface can be exported to CAD programs, leaving a minuscule hole that, if needed, can be capped by the approach in [31]. Fig. 24 shows the bi-cubic pieces of FASt, at the excessive subdivision level 10, exported to and displayed by freeCAD, a



(a) view (b) WM23 (c) QAS (d) FASt^2

Figure 20: $n = 10$ configuration for input net Fig. 13 g.



(a) $n = 6$ view (b) layout (c) FASt

(d) $n = 3$ view (e) FASt (f) $n = 5$ view

(g) layout (h) Catmull-Clark (i) FASt

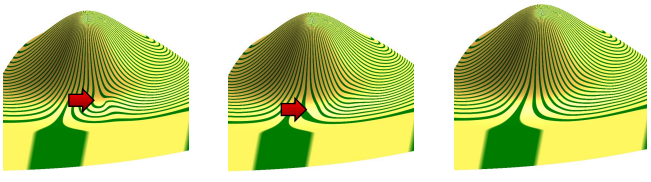
(j) MA18 (k) MA19 (l) WM23

Figure 21: FASt for $n = 3, 4, 5, 6$. Comparison on the 5-sided (gold) region. The reader should zoom in on MA18 and MA19 to discover the flaws in the limit. The flaws in WM23 and Catmull-Clark (CCS) are easily visible.

freely-available NURBS renderer. Fig. 25 illustrates that applying textures to the FASt surface is no different than for CCS: one uses FASt for sampling the texture.

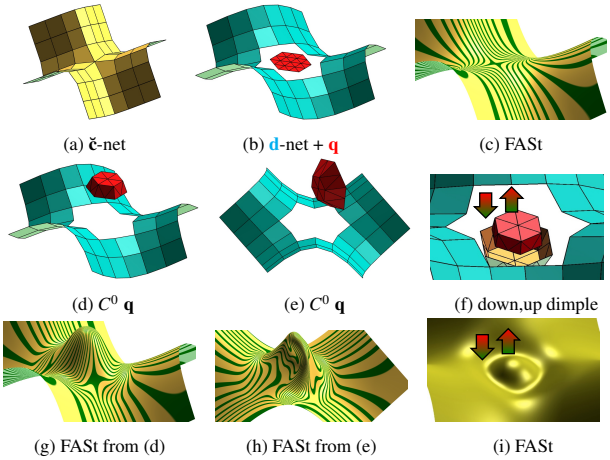
Fig. 26 demonstrates that piecewise bi-cubic subdivision lends itself naturally to solving differential equations directly in terms of subdivision functions on the subdivision surface. Here, merging steps, as discussed in Section 5 and illustrated in Fig. 12, allows for very fast convergence. Using the standard Galerkin formulation, weighted by the first fundamental form, Fig. 26, top, illustrates the progression of temperature isolines from two heat sources on the FASt surface of Fig. 25 a. Fig. 26, bottom, shows the same for sunlight on the frog extremities of Fig. 24 b.

Limitations. Like any high-quality surface construction, to avoid special rules, FASt assumes an input mesh with at most one extraordinary node per quad. For extreme valences $n > 9$,



(a) MA18 (b) FASt (c) QAS+FASt^2

Figure 22: Input net Fig. 13 i: $n = 3$ configuration.

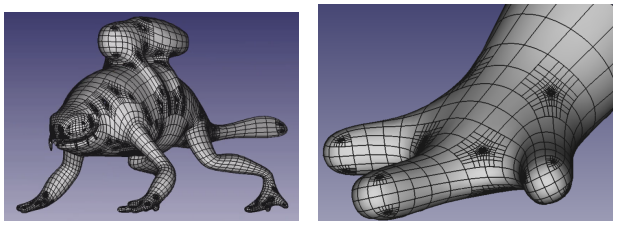


(a) $\check{\mathbf{c}}$-net (b) $\mathbf{d}$-net + $\mathbf{q}$ (c) FASt

(d) C^0 $\mathbf{q}$ (e) C^0 $\mathbf{q}$ (f) down,up dimple

(g) FASt from (d) (h) FASt from (e) (i) FASt

Figure 23: FASt using (b,c) standard C^2 $\mathbf{q}$, (d,e,g,h) C^0 $\mathbf{q}$ and (f,i) a series of (two, superimposed) C^0 $\mathbf{q}$ to create a down, then an up dimple.



(a) monsterfrog (b) zoom

Figure 24: FASt implemented using the formulas of the Appendix with export as NURBS, rendered with FreeCAD .

the fast convergence paired with the low number of pieces per ring induces very slight highlight line oscillations in FASt.

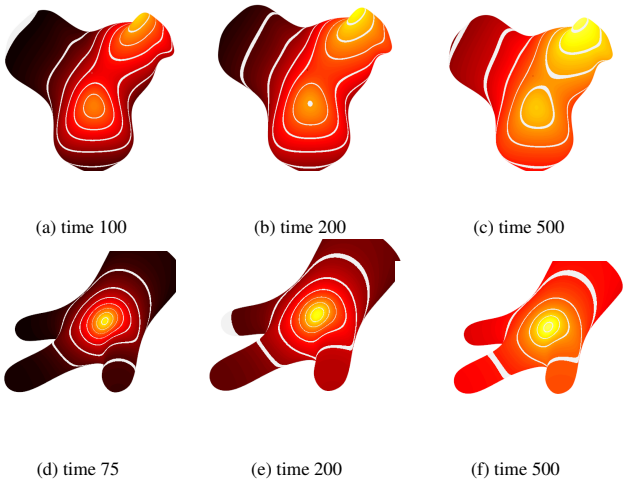
8. Conclusion

The surface quality of FASt, measured as uniform distribution of highlight lines, is comparable to the high-end, industrial-design quality QAS algorithm, but on a much lower budget: $5n$ pieces in place of $12n$. We conjecture that, for $n > 4$ and bi-cubic subdivision with maximal smoothness, $5n$ pieces is the minimal number. Here maximal smoothness means C^2 everywhere except at the extraordinary point, where the surface is C^1 and curvature-bounded.

Acknowledgements. Monsterfrog model attributed to Bay Raitt. Jörg Peters acknowledges the support of the Fields Institute, Toronto, CA. and U.S. Navy Grant number SBIR N251-060. Seth Barber rendered Figures 24, 25, 26.

(a) control net (b) $n = 6$ at center (c) $n = 3$ corners

Figure 25: Textures on an FASt surface with valence 3,4,5,6.



(a) time 100 (b) time 200 (c) time 500

(d) time 75 (e) time 200 (f) time 500

Figure 26: Geodesic flow with sources (a,b,c) at two $n = 3$ corners on the FASt surface whose **c**-net is displayed in Fig. 25 a, (d,e,f) on the surface in Fig. 24. The flow is computed using FASt elements to solve the heat equation on FASt geometry.

References

[1] OpenSubdiv, Opensubdiv, may 12 (2022). URL https://graphics.pixar.com/opensubdiv/docs/intro.html

[2] E. Catmull, J. Clark, Recursively generated B-spline surfaces on arbitrary topological meshes, Computer-Aided Design 10 (1978) 350–355.

[3] K. Karciauskas, J. Peters, U. Reif, Shape characterization of subdivision surfaces – case studies, Computer-Aided Geometric Design 21 (6) (2004) 601–614.

[4] J. Peters, U. Reif, Shape characterization of subdivision surfaces – basic principles, Computer-Aided Geometric Design 21 (6) (2004) 585–599.

[5] U. H. Augsdörfer, N. A. Dodgson, M. A. Sabin, Tuning subdivision by minimising gaussian curvature variation near extraordinary vertices, Computer Graphics Forum 25 (3) (2006) 263–272.

[6] M. A. Sabin, Cubic recursive division with bounded curvature, in: Curves and surfaces, Elsevier, 1991, pp. 411–414.

[7] J. Peters, U. Reif, Subdivision Surfaces, Vol. 3 of Geometry and Computing, Springer-Verlag, New York, 2008.

[8] K. Karčiauskas, J. Peters, Quadratic-attraction subdivision, Computer Graphics Forum 42 (5) (2023) e14 900.

[9] Y. Ma, W. Ma, Subdivision schemes with optimal bounded curvature near extraordinary vertices, Computer Graphics Forum 37 (7) (2018) 455–467.

[10] X. Wang, W. Ma, A class of new tuned primal subdivision schemes with high-quality limit surface in extraordinary regions, ACM Transactions on Graphics (TOG) 43 (6) (2024) 1–17.

[11] K. Karčiauskas, J. Peters, Point-augmented bi-cubic subdivision surfaces, in: Computer Graphics Forum, Vol. 41, Wiley Online Library, 2022, pp. 13–23.

[12] F. Kälberer, M. Nieser, K. Polthier, Quadcover - surface parameterization using branched coverings, Comput. Graph. Forum 26 (3) (2007) 375–384.

[13] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, D. Zorin, Quad-mesh generation and processing: A survey, in: Computer graphics forum, Vol. 32, Wiley Online Library, 2013, pp. 51–76.

[14] M. Lyon, M. Campen, L. Kobbelt, Quad layouts via constrained t-mesh quantization, in: Computer Graphics Forum, Vol. 40, Wiley Online Library, 2021, pp. 305–314.

[15] N. Pietroni, S. Nuvoli, T. Alderighi, P. Cignoni, M. Tarini, et al., Reliable feature-line driven quad-remeshing, ACM Transactions on Graphics 40 (4) (2021) 1–17.

[16] U. H. Augsdörfer, N. A. Dodgson, M. A. Sabin, Artifact analysis on B-splines, box-splines and other surfaces defined by quadrilateral polyhedra, Comp Aid Geom Design 28 (3) (2011) 177–197.

[17] X. Li, G. T. Finnigan, T. W. Sederberg, G 1 non-uniform catmull-clark surfaces, ACM Transactions on Graphics (TOG) 35 (4) (2016) 1–8.

[18] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, in: J. Hodgins, J. C. Hart (Eds.), Proc ACM SIGGRAPH, Vol. 22(3) of ACM Tr Graph, 2003, pp. 477–484.

[19] K. Karčiauskas, J. Peters, Concentric tesselation maps and curvature continuous guided surfaces, Computer Aided Geometric Design 24 (2) (2007) 99–111.

[20] H. Prautzsch, Freeform splines, Comput. Aided Geom. Des 14 (3) (1997).

[21] A. Levin, Modified subdivision surfaces with continuous curvature, in: ACM SIGGRAPH 2006 Papers, 2006, pp. 1035–1040.

[22] K. Karčiauskas, J. Peters, Quadratic-attraction subdivision with contraction-ratio λ= 1/2, Computers & Graphics 123 (2024) 104001.

[23] C. de Boor, A Practical Guide to Splines, Springer, 1978.

[24] C. de Boor, B-form basics, in: G. Farin (Ed.), Geometric Modeling: Algorithms and New Trends, SIAM, 1987, pp. 131–148.

[25] G. Farin, Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, Academic Press, 1988.

[26] M. Halstead, M. Kass, T. DeRose, Efficient, fair interpolation using catmull-clark surfaces, in: L. Valastyan, L. Walsh (Eds.), Proceedings of the 20th annual conference on Computer graphics and interactive techniques, Siggraph 1993, ACM Press, New York, NY, USA, 1993, pp. 35–44.

[27] C. De Boor, B-form basics, Mathematics Research Center, University of Wisconsin-Madison, 1986.

[28] Y. Ma, W. Ma, Subdivision schemes for quadrilateral meshes with the least polar artifact in extraordinary regions, Comput. Graph. Forum 38 (7) (2019) 127–139.

[29] X. Wang, W. Ma, An extended tuned subdivision scheme with optimal convergence for isogeometric analysis, Computer-Aided Design 162 (2023) 103544.

[30] K.-P. Beier, Y. Chen, Highlight-line algorithm for realtime surface-quality assessment, Comp-Aid Design 26 (4) (1994) 268–277.

[31] K. Karčiauskas, J. Peters, Improved caps for improved subdivision surfaces, Computer-Aided Design (2023) 103 543.

9. Appendix: Nonzero M_n entries

Recall the structure of M_n from Fig. 11. $M_{\mathbf{q}}$ provided by (1). This Appendix contains the remainig complete set of non-zero entries of M_n for implementing FASt for valence $n = 3, 5, \ldots, 10$. Note that while many stencils of the refined control-net are sparse, visual display as weights on the control net makes little sense since stencils can have up to $\bar{n} = 17n + 1$ five-digit entries.

Following the technique of [8] to eliminate unnecessary rounding, we present simplified rational-weight subdivision formulas, whose outcome is essentially indistinguishable from the derived FASt. That is, the simplified formulas preserve the shape of the subdivision surfaces but have only 5 digits. The simplified formulas are derived by multiplying by 10^5 the high-digit stencil weights of FASt and retaining the resulting integer parts to be placed into tables. The integer weights are corrected to sum to 10^5 and so guarantee a partition of 1. The truncation preserves the quality of the subdivision surfaces. All comparisons in the paper were made with the listed tables of simplified formulas. The tables are indexed by the extraordinary point-valence n and have the pattern:

$$DQ^n := 10^5 \begin{pmatrix} \hat{\nu}_{11} & \hat{\nu}_{12} & \hat{\nu}_{13} & \hat{\nu}_{14} & \hat{\nu}_{15} & \hat{\nu}_{16} \\ \hat{\nu}_{21} & \hat{\nu}_{22} & \hat{\nu}_{23} & \hat{\nu}_{24} & \hat{\nu}_{25} & \hat{\nu}_{26} \end{pmatrix}, \; D_1^n := 10^5 \begin{pmatrix} \dot{\nu}_{04} & \dot{\nu}_{14} & \dot{\nu}_{24} \\ \dot{\nu}_{03} & \dot{\nu}_{13} & \dot{\nu}_{23} \\ \dot{\nu}_{02} & \dot{\nu}_{12} & \dot{\nu}_{22} \\ \dot{\nu}_{01} & \dot{\nu}_{11} & \dot{\nu}_{21} \\ \dot{\nu}_{00} & \dot{\nu}_{10} & \dot{\nu}_{20} \end{pmatrix},$$

$$D_2^n := 10^5 \begin{pmatrix} \ddot{\nu}_{04} & \ddot{\nu}_{14} & \ddot{\nu}_{24} & \ddot{\nu}_{34} \\ \ddot{\nu}_{03} & \ddot{\nu}_{13} & \ddot{\nu}_{23} & \ddot{\nu}_{33} \\ \ddot{\nu}_{02} & \ddot{\nu}_{12} & \ddot{\nu}_{22} & \ddot{\nu}_{32} \\ \ddot{\nu}_{01} & \ddot{\nu}_{11} & \ddot{\nu}_{21} & \ddot{\nu}_{31} \\ \ddot{\nu}_{00} & \ddot{\nu}_{10} & \ddot{\nu}_{20} & \ddot{\nu}_{30} \end{pmatrix}; \; Q_k^n := 10^5 \begin{pmatrix} \mu_{10}^k & \mu_{11}^k & \ldots & \mu_{1,n-1}^k \\ \mu_{20}^k & \mu_{21}^k & \ldots & \mu_{2,n-1}^k \\ \mu_{30}^k & \mu_{31}^k & \ldots & \mu_{3,n-1}^k \\ \mu_{40}^k & \mu_{41}^k & \ldots & \mu_{4,n-1}^k \\ \mu_{50}^k & \mu_{51}^k & \ldots & \mu_{5,n-1}^k \\ \mu_{60}^k & \mu_{61}^k & \ldots & \mu_{6,n-1}^k \end{pmatrix},$$

where $k = 2, 3, 5$.

9.1. *Tables for new control points* •

Only the new control points $\bullet_1^s$ and $\bullet_2^s$ of Fig. 10 b need to be defined.

$$\begin{aligned}
\bullet_1^s &:= \sum_{i=0}^{2}\sum_{j=0}^{2} \dot{\nu}_{ij}\mathbf{d}_{ij}^s + \sum_{i=0}^{2} \dot{\nu}_{i3}(\mathbf{d}_{3i}^{s-1} + \mathbf{d}_{3i}^s) \\
&\quad + \sum_{i=0}^{2} \dot{\nu}_{i4}(\mathbf{d}_{2i}^{s-1} + \mathbf{d}_{i2}^{s+1}) + \sum_{j=1}^{6} \hat{\nu}_{1j}\mathbf{q}_j^s\,, \\
\bullet_2^s &:= \sum_{i=0}^{2}\sum_{j=0}^{2} \ddot{\nu}_{ij}(\mathbf{d}_{ij}^s + \mathbf{d}_{ji}^{s+1}) + \sum_{j=0}^{2} \ddot{\nu}_{3j}\mathbf{d}_{3j}^s \\
&\quad + \sum_{i=0}^{2} \ddot{\nu}_{i3}(\mathbf{d}_{3i}^{s-1} + \mathbf{d}_{3i}^{s+1}) + \sum_{i=0}^{2} \ddot{\nu}_{i4}(\mathbf{d}_{2i}^{s-1} + \mathbf{d}_{i2}^{s+2}) \\
&\quad + \hat{\nu}_{21}\mathbf{q}_1^s + \hat{\nu}_{22}(\mathbf{q}_2^s + \mathbf{q}_4^{s+1}) + \hat{\nu}_{23}(\mathbf{q}_3^s + \mathbf{q}_6^{s+1}) \\
&\quad + \hat{\nu}_{24}(\mathbf{q}_4^s + \mathbf{q}_2^{s+1}) + \hat{\nu}_{25}(\mathbf{q}_5^s + \mathbf{q}_5^{s+1}) + \hat{\nu}_{26}(\mathbf{q}_6^s + \mathbf{q}_3^{s+1})\,.
\end{aligned}$$

where $\dot{\nu}_{ij}$, $\ddot{\nu}_{ij}$ and $\hat{\nu}_{ij}$ are from D_1^n, D_2^n, DQ^n divided by 10^5 and

$$DQ^3 := \begin{pmatrix} 6049 & -15573 & 49756 & -15573 & 38712 & 49756 \\ -105577 & 54028 & -12439 & 74763 & -13819 & 457 \end{pmatrix},$$

$$D_1^3 := \begin{pmatrix} 149 & -1461 & -2924 \\ 172 & -2300 & -2259 \\ -53 & 304 & 3624 \\ 0 & -7 & 304 \\ 0 & 0 & -53 \end{pmatrix},\ D_2^3 := \begin{pmatrix} -37 & 365 & 731 & 0 \\ -43 & 575 & 564 & 0 \\ 13 & -76 & -883 & -1704 \\ 0 & 1 & -226 & -913 \\ 0 & 0 & 39 & 168 \end{pmatrix},$$

$$DQ^5 := \begin{pmatrix} -1168 & 5551 & 17986 & 5551 & 29927 & 17986 \\ 88 & 10014 & -4496 & 11657 & 7909 & 21841 \end{pmatrix},$$

$$D_1^5 := \begin{pmatrix} -143 & 669 & -2360 \\ -172 & 810 & 3498 \\ 393 & -1706 & 22182 \\ 16 & -21 & -1706 \\ -4 & 16 & 393 \end{pmatrix},\ D_2^5 := \begin{pmatrix} 35 & -167 & 589 & 0 \\ 42 & -202 & -874 & 0 \\ -97 & 426 & -2092 & 9701 \\ -4 & 5 & 437 & 308 \\ 0 & -4 & -93 & -37 \end{pmatrix},$$

$$DQ^6 := \begin{pmatrix} -499 & 5841 & 15223 & 5841 & 28835 & 15223 \\ 3688 & 6438 & -3805 & 9667 & 9313 & 19851 \end{pmatrix},$$

$$D_1^6 := \begin{pmatrix} -222 & 949 & -2328 \\ -468 & 1799 & 3445 \\ 944 & -3825 & 28767 \\ -8 & 204 & -3825 \\ -7 & -8 & 944 \end{pmatrix},\ D_2^6 := \begin{pmatrix} 55 & -236 & 581 & 0 \\ 116 & -449 & -860 & 0 \\ -236 & 955 & -1952 & 16903 \\ 2 & -50 & 760 & -730 \\ 1 & 2 & -187 & 207 \end{pmatrix},$$

$$DQ^7 := \begin{pmatrix} 154 & 5479 & 14409 & 5479 & 27892 & 14409 \\ 4545 & 4753 & -3602 & 9216 & 10224 & 17554 \end{pmatrix},$$

$$D_1^7 := \begin{pmatrix} -259 & 1095 & -2416 \\ -844 & 2938 & 2883 \\ 1507 & -5820 & 33554 \\ -78 & 613 & -5820 \\ -1 & -78 & 1507 \end{pmatrix},\ D_2^7 := \begin{pmatrix} 64 & -273 & 603 & 0 \\ 211 & -734 & -720 & 0 \\ -376 & 1454 & -1746 & 22709 \\ 18 & -152 & 977 & -2284 \\ 0 & 18 & -269 & 590 \end{pmatrix},$$

$$DQ^8 := \begin{pmatrix} 678 & 4990 & 14315 & 4990 & 27137 & 14315 \\ 4711 & 3768 & -3578 & 9254 & 10984 & 15471 \end{pmatrix},$$

$$D_1^8 := \begin{pmatrix} -277 & 1194 & -2545 \\ -1230 & 4044 & 2214 \\ 2014 & -7505 & 37014 \\ -170 & 1073 & -7505 \\ 10 & -170 & 2014 \end{pmatrix},\ D_2^8 := \begin{pmatrix} 68 & -298 & 636 & 0 \\ 307 & -1010 & -553 & 0 \\ -503 & 1875 & -1540 & 27194 \\ 42 & -268 & 1119 & -3854 \\ -2 & 42 & -338 & 997 \end{pmatrix},$$

$$DQ^9 := \begin{pmatrix} 1084 & 4527 & 14476 & 4527 & 26550 & 14476 \\ 4673 & 3105 & -3619 & 9480 & 11635 & 13700 \end{pmatrix},$$

$$D_1^9 := \begin{pmatrix} -287 & 1271 & -2679 \\ -1586 & 5029 & 1582 \\ 2445 & -8871 & 39549 \\ -267 & 1512 & -8871 \\ 25 & -267 & 2445 \end{pmatrix},\ D_2^9 := \begin{pmatrix} 71 & -317 & 669 & 0 \\ 396 & -1256 & -395 & 0 \\ -610 & 2217 & -1359 & 30638 \\ 66 & -377 & 1213 & -5258 \\ -6 & 66 & -393 & 1375 \end{pmatrix},$$

$$DQ^{10} := \begin{pmatrix} 1391 & 4128 & 14713 & 4128 & 26095 & 14713 \\ 4561 & 2623 & -3678 & 9768 & 12189 & 12229 \end{pmatrix},$$

$$D_1^{10} := \begin{pmatrix} -293 & 1336 & -2801 \\ -1896 & 5872 & 1030 \\ 2802 & -9966 & 41441 \\ -360 & 1902 & -9966 \\ 41 & -360 & 2802 \end{pmatrix},\ D_2^{10} := \begin{pmatrix} 73 & -333 & 700 & 0 \\ 474 & -1467 & -257 & 0 \\ -700 & 2491 & -1204 & 33302 \\ 90 & -475 & 1277 & -6454 \\ -10 & 90 & -437 & 1705 \end{pmatrix}.$$

9.2. *Tables for initializing* $\mathbf{q}$ *from* $\mathbf{c}$

The μ_{ij} are defined by

$$Q_2^3 := \begin{pmatrix} 0 & 0 & 0 \\ 5 & -41 & 36 \\ -94 & -94 & 188 \\ 36 & -41 & 5 \\ 6851 & 1922 & 6851 \\ 13021 & 13021 & 23956 \end{pmatrix},\ Q_3^3 := \begin{pmatrix} 0 & 0 & 0 \\ 1 & -12 & 177 \\ -61 & -61 & 783 \\ 177 & -12 & 1 \\ 8620 & 418 & 8620 \\ 9045 & 9045 & 32711 \end{pmatrix},$$

$$Q_5^3 := \begin{pmatrix} 0 & 0 & 0 \\ 29 & -14 & -98 \\ -109 & -110 & -109 \\ 29 & -98 & -14 \\ 9257 & 2675 & 2675 \\ 19579 & 10441 & 19579 \end{pmatrix},\ Q_2^5 := \begin{pmatrix} -11 & 4 & 13 & 4 & -11 \\ 23 & -2 & -25 & -12 & 17 \\ -20 & 53 & 53 & -20 & -66 \\ 17 & -12 & -25 & -2 & 23 \\ 3459 & 1442 & 196 & 1442 & 3459 \\ 9769 & 3367 & 3367 & 9769 & 13726 \end{pmatrix},$$

$$Q_3^5 := \begin{pmatrix} -32 & 18 & 5 & 18 & -32 \\ 75 & -24 & 4 & -41 & 60 \\ -52 & 15 & 15 & -52 & -275 \\ 60 & -41 & 4 & -24 & 75 \\ 5729 & 497 & -56 & 497 & 5729 \\ 9979 & 858 & 858 & 9979 & 23852 \end{pmatrix},\ Q_5^5 := \begin{pmatrix} -41 & 5 & 11 & 11 & 5 \\ 85 & -6 & -13 & -18 & -24 \\ -131 & 70 & 14 & 70 & -131 \\ 85 & -24 & -18 & -13 & -6 \\ 6138 & 2183 & 118 & 118 & 2183 \\ 16855 & 2813 & 2370 & 2813 & 16855 \end{pmatrix},$$

$$Q_2^6 := \begin{pmatrix} 38 & 0 & -38 & -38 & 0 & 38 \\ -108 & 47 & 156 & 108 & -47 & -156 \\ 81 & -81 & -163 & -81 & 81 & 163 \\ -156 & -47 & 108 & 156 & 47 & -108 \\ 3206 & 1515 & -176 & -176 & 1515 & 3206 \\ 8286 & 3834 & 1609 & 3834 & 8286 & 10511 \end{pmatrix},\ Q_3^6 := \begin{pmatrix} 136 & 153 & -18 & -18 & 153 & 136 \\ -357 & -255 & 114 & 75 & -447 & -510 \\ 409 & 81 & -162 & 81 & 409 & 492 \\ -510 & -447 & 75 & 114 & -255 & -357 \\ 6270 & 1486 & -494 & -494 & 1486 & 6270 \\ 9250 & 347 & 1387 & 347 & 9250 & 19193 \end{pmatrix},$$

$$Q_5^6 := \begin{pmatrix} 59 & 115 & 38 & -95 & 38 & 115 \\ -258 & -207 & 0 & 270 & -86 & -408 \\ 328 & 162 & -162 & -162 & 162 & 328 \\ -258 & -408 & -86 & 270 & 0 & -207 \\ 6284 & 3191 & -190 & -480 & -190 & 3191 \\ 14852 & 2684 & 1498 & 1498 & 2684 & 14852 \end{pmatrix},\ Q_2^7 := \begin{pmatrix} 219 & 54 & -151 & -243 & -151 & 54 & 219 \\ -504 & -37 & 457 & 607 & 300 & -232 & -591 \\ 405 & -144 & -586 & -586 & -144 & 405 & 650 \\ -591 & -232 & 300 & 607 & 457 & -37 & -504 \\ 3479 & 1755 & -393 & -1350 & -393 & 1755 & 3479 \\ 6763 & 4047 & 1868 & 1868 & 4047 & 6763 & 7972 \end{pmatrix},$$

$$Q_3^7 := \begin{pmatrix} 775 & 296 & -49 & -113 & -49 & 296 & 775 \\ -1697 & -462 & 180 & 334 & 132 & -970 & -2083 \\ 1422 & 112 & -298 & -298 & 112 & 1422 & 2291 \\ -2083 & -970 & 132 & 334 & 180 & -462 & -1697 \\ 8259 & 2384 & -834 & -800 & -834 & 2384 & 8259 \\ 8143 & 599 & 986 & 986 & 599 & 8143 & 14365 \end{pmatrix},\ Q_5^7 := \begin{pmatrix} 708 & 423 & -13 & -162 & -162 & -13 & 423 \\ -1730 & -765 & 159 & 409 & 465 & -73 & -1312 \\ 1673 & 406 & -375 & -438 & -375 & 406 & 1673 \\ -1730 & -1312 & -73 & 465 & 409 & 159 & -765 \\ 8295 & 4624 & -329 & -1007 & -1007 & -329 & 4624 \\ 11958 & 3289 & 505 & 2131 & 505 & 3289 & 11958 \end{pmatrix},$$

$$Q_2^8 := \begin{pmatrix} 354 & 146 & -146 & -354 & -354 & -146 & 146 & 354 \\ -760 & -235 & 428 & 840 & 760 & 235 & -428 & -840 \\ 640 & 0 & -640 & -905 & -640 & 0 & 640 & 905 \\ -840 & -428 & 235 & 760 & 840 & 428 & -235 & -760 \\ 3343 & 1948 & -25 & -1420 & -1420 & -25 & 1948 & 3343 \\ 5736 & 3846 & 1955 & 1172 & 1955 & 3846 & 5736 & 6519 \end{pmatrix},$$

$$Q_3^8 := \begin{pmatrix} 1393 & 302 & -284 & -25 & -25 & -284 & 302 & 1393 \\ -2800 & -378 & 610 & 65 & 242 & 561 & -1103 & -3298 \\ 2071 & -167 & -488 & -59 & -488 & -167 & 2071 & 3561 \\ -3298 & -1103 & 561 & 242 & 65 & 610 & -378 & -2800 \\ 9093 & 2422 & -1524 & -434 & -434 & -1524 & 2422 & 9093 \\ 7814 & 1982 & 252 & 736 & 252 & 1982 & 7814 & 11431 \end{pmatrix},$$

$$Q_5^8 := \begin{pmatrix} 1431 & 746 & -232 & -256 & 13 & -256 & -232 & 746 \\ -3110 & -1296 & 623 & 491 & 93 & 695 & 236 & -2047 \\ 2783 & 521 & -758 & -306 & -306 & -758 & 521 & 2783 \\ -3110 & -2047 & 236 & 695 & 93 & 491 & 623 & -1296 \\ 9479 & 5339 & -773 & -1397 & -48 & -1397 & -773 & 5339 \\ 9993 & 4418 & 637 & 864 & 864 & 637 & 4418 & 9993 \end{pmatrix},$$

$$Q_2^9 := \begin{pmatrix} 410 & 218 & -75 & -334 & -436 & -334 & -75 & 218 & 410 \\ -838 & -385 & 247 & 765 & 924 & 651 & 73 & -539 & -899 \\ 731 & 165 & -477 & -897 & -897 & -477 & 165 & 731 & 954 \\ -899 & -539 & 73 & 651 & 924 & 765 & 247 & -385 & -838 \\ 2958 & 1945 & 393 & -971 & -1510 & -971 & 393 & 1945 & 2958 \\ 5073 & 3605 & 1935 & 845 & 845 & 1935 & 3605 & 5073 & 5653 \end{pmatrix},$$

$$Q_3^9 := \begin{pmatrix} 1735 & 430 & -477 & -184 & 212 & -184 & -477 & 430 & 1735 \\ -3338 & -594 & 990 & 296 & -302 & 560 & 811 & -1333 & -3780 \\ 2461 & -160 & -858 & -3 & -3 & -858 & -160 & 2461 & 4014 \\ -3780 & -1333 & 811 & 560 & -302 & 296 & 990 & -594 & -3338 \\ 8756 & 2727 & -1766 & -976 & 557 & -976 & -1766 & 2727 & 8756 \\ 7638 & 2874 & 255 & 152 & 152 & 255 & 2874 & 7638 & 10154 \end{pmatrix},$$

$$Q_5^9 := \begin{pmatrix} 1822 & 1028 & -267 & -499 & 60 & 60 & -499 & -267 & 1028 \\ -3715 & -1792 & 701 & 931 & -118 & 87 & 1075 & 202 & -2498 \\ 3298 & 814 & -982 & -555 & 130 & -555 & -982 & 814 & 3298 \\ -3715 & -2498 & 202 & 1075 & 87 & -118 & 931 & 701 & -1792 \\ 9212 & 5587 & -497 & -2024 & 73 & 73 & -2024 & -497 & 5587 \\ 9101 & 4975 & 1115 & 182 & 447 & 182 & 1115 & 4975 & 9101 \end{pmatrix},$$

$$Q_2^{10} := \begin{pmatrix} 423 & 262 & 0 & -262 & -423 & -423 & -262 & 0 & 262 & 423 \\ -834 & -471 & 71 & 587 & 878 & 834 & 471 & -71 & -587 & -878 \\ 750 & 286 & -286 & -750 & -927 & -750 & -286 & 286 & 750 & 927 \\ -878 & -587 & -71 & 471 & 834 & 878 & 587 & 71 & -471 & -834 \\ 2564 & 1839 & 666 & -505 & -1230 & -1230 & -505 & 666 & 1839 & 2564 \\ 4582 & 3398 & 1934 & 751 & 298 & 751 & 1934 & 3398 & 4582 & 5034 \end{pmatrix},$$

$$Q_3^{10} := \begin{pmatrix} 1844 & 627 & -449 & -420 & 149 & 149 & -420 & -449 & 627 & 1844 \\ -3450 & -971 & 946 & 738 & -275 & -112 & 914 & 659 & -1610 & -3791 \\ 2652 & 140 & -1005 & -347 & 278 & -347 & -1005 & 140 & 2652 & 3986 \\ -3791 & -1610 & 659 & 914 & -112 & -275 & 738 & 946 & -971 & -3450 \\ 7895 & 3107 & -1300 & -1582 & 306 & 306 & -1582 & -1300 & 3107 & 7895 \\ 7551 & 3269 & 343 & -111 & 166 & -111 & 343 & 3269 & 7551 & 9638 \end{pmatrix},$$

$$Q_5^{10} := \begin{pmatrix} 1930 & 1216 & -101 & -625 & -155 & 234 & -155 & -625 & -101 & 1216 \\ -3774 & -2128 & 375 & 1182 & 237 & -347 & 484 & 1175 & -135 & -2695 \\ 3402 & 1176 & -839 & -896 & 47 & 47 & -896 & -839 & 1176 & 3402 \\ -3774 & -2695 & -135 & 1175 & 484 & -347 & 237 & 1182 & 375 & -2128 \\ 8271 & 5483 & 247 & -2097 & -656 & 681 & -656 & -2097 & 247 & 5483 \\ 8758 & 5209 & 1379 & -85 & 191 & 191 & -85 & 1379 & 5209 & 8758 \end{pmatrix}.$$

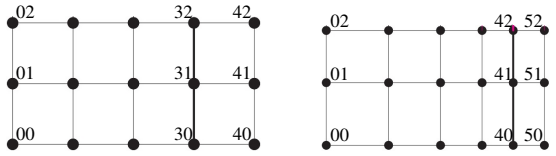
For the C^0 option of $\mathbf{q}$, we need only set

$$\bar{\mathbf{q}}_2^s := \mathbf{p} + \frac{2}{n}\sum_{j=0}^{n-1} \cos(\frac{2\pi}{n}j)\mathbf{q}_2^{s+j},$$

and redefine $\mathbf{q}_4^{s-1} = \mathbf{q}_2^s := \bar{\mathbf{q}}_2^s$, $s = 0, \ldots, n-1$.

9.3. *Knot insertion:* $\mathbf{d}$ *to* $\mathbf{h}$

Compute $\check{D} := RD$, where



(a) **d**-net (b) **ℏi**-net

Figure 27: Labels of the parts of the **d**- and **ℏi**-nets used in the explicit formulas $\check{D} := RD$ below of non-uniform **d** to **ℏi** refinement.

$$R := \frac{1}{144}\begin{pmatrix}
144 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 144 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 24 & 120 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 72 & 72 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 18 & 108 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 72 & 72 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 144 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 144 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 24 & 120 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 72 & 72 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 108 & 18 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 72 & 72 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 24 & 0 & 0 & 0 & 0 & 120 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 24 & 0 & 0 & 0 & 0 & 120 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & 20 & 0 & 0 & 0 & 20 & 100 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & 12 & 0 & 0 & 0 & 60 & 60 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 18 & 3 & 0 & 0 & 15 & 90 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & 12 & 0 & 0 & 0 & 60
\end{pmatrix}, \quad 15 \\ 60$$

$$D := (\mathbf{d}_{00}, \ldots, \mathbf{d}_{40}, \mathbf{d}_{01}, \ldots, \mathbf{d}_{41}, \mathbf{d}_{02}, \ldots, \mathbf{d}_{42})^T,$$

$$\check{D} := (\mathbf{ℏi}_{00}, \ldots, \mathbf{ℏi}_{50}, \mathbf{ℏi}_{01}, \ldots, \mathbf{ℏi}_{41}, \mathbf{ℏi}_{02}, \ldots, \mathbf{ℏi}_{42})^T.$$

and the remaining nodes of **ℏi** by structural symmetry across the diagonal of each sector of the surface ring construction.