# Smooth Blended Subdivision Shading

J. Bakker[1], P.J. Barendrecht[1], J. Kosinka[1]

[1]Johann Bernoulli Institute, University of Groningen, Nijenborgh 9, 9747 AG Groningen, The Netherlands

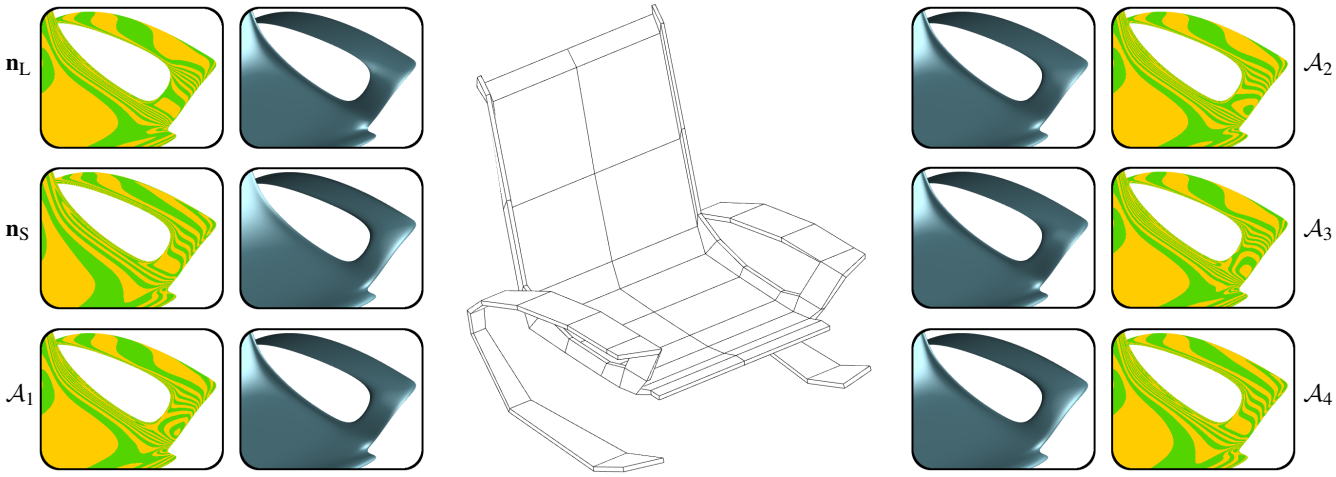


**Figure 1:** *Comparing 6 different normal fields for a region of a shaded Catmull-Clark subdivision surface. Left:* $\mathbf{n}_L$*: with surface normals artifacts are clearly visible.* $\mathbf{n}_S$*: subdivided normals yield a result that is too smooth.* $\mathcal{A}_1$*: a linear blend of* $\mathbf{n}_L$ *and* $\mathbf{n}_S$ *shows artifacts. Right: three different subdivision blends of* $\mathbf{n}_L$ *and* $\mathbf{n}_S$*.* $\mathcal{A}_2$*: blend weights at EVs set to* 1 *shows improvement over linear blending, but does not remove all artifacts.* $\mathcal{A}_3$*: weights at EVs after one subdivision step set via the limit stencil still shows artifacts.* $\mathcal{A}_4$*: blend weights at EVs and their one-ring neighbourhood set to* 1 *gives the best overall result. Reflection lines are shown next to each shaded rendering.*

**Abstract**

*The concept known as subdivision shading aims at improving the shading of subdivision surfaces. It is based on the subdivision of normal vectors associated with the control net of the surface. By either using the resulting subdivided normal field directly, or blending it with the normal field of the limit surface, renderings of higher visual smoothness can be obtained. In this work we propose a different and more versatile approach to blend the two normal fields, yielding not only better results, but also a proof that our blended normal field is* $C^1$*.*

**CCS Concepts**

•***Computing methodologies*** → ***Rendering; Shape modeling;***

## 1. Introduction

The concept of subdivision shading [AB08] proposes the application of a selected subdivision scheme to both the vertices and the normal vectors of a control net defining a subdivision surface. As such, a $C^1$ subdivided normal field $\mathbf{n}_S$ is obtained that is smoother than the $C^0$ normal field $\mathbf{n}_L$ of the limit surface. The advantage of using $\mathbf{n}_S$ is that renderings of increased visual smoothness can be obtained. In contrast, using only $\mathbf{n}_L$ might highlight undesired artefacts that often occur in subdivision surfaces around extraordinary vertices (EVs).

The original work mentions two different approaches (which we refer to as $\mathcal{A}_1$ and $\mathcal{A}_2$) with regard to blending $\mathbf{n}_S$ and $\mathbf{n}_L$. The motivation for blending them is that surface renderings benefit from using $\mathbf{n}_S$ only around EVs. Using $\mathbf{n}_S$ everywhere might result in surfaces that look too smooth in the sense that desired detail is lost. In both approaches, the vertices of the initial control net are initialised



DOI: 10.2312/egs.20181039

with a blending weight of 1 in case of EVs and 0 otherwise. Upon subdivision, the weights of the new vertices are either interpolated (bi)linearly ($\mathcal{A}_1$), or found by applying subdivision ($\mathcal{A}_2$). It follows that in the former case, $\mathbf{n}_S$ is only used in the one-ring neighbourhoods of EVs, resulting in sharp transitions in the blended normal field. In the latter case, the weights of updated EVs generally do not remain 1 for approximating subdivision schemes. As such, the blended normal field does not equal $\mathbf{n}_S$ at the limit positions of EVs, but also relies on $\mathbf{n}_L$. Therefore, the resulting blended normal field is not $C^1$ in either case.

In this work we improve upon subdivision shading by introducing a more versatile blending approach,

$$\mathbf{n}_B = (1 - b^p)\mathbf{n}_L + b^p \mathbf{n}_S, \tag{1}$$

where $\mathbf{n}_B$ is a $C^1$ blended normal field, $b$ a suitably chosen blending function, and $p \in \mathbb{R}$ a parameter that can be used to tune the blending per vertex or globally. Applying this approach typically results in better shading compared to [AB08]; see Fig. 1.

After discussing necessary preliminaries in Section 2, we focus on the blended normal field $\mathbf{n}_B$ and the subdivision blending function $b$ in Section 3 and prove that $\mathbf{n}_B$ is $C^1$ for suitable functions $b$. Details regarding implementation are considered in Section 4. Finally, Section 5 shows, compares, and discusses our results.

## 2. Preliminaries

This section briefly describes the building blocks required for studying the blended normal field $\mathbf{n}_B$.

**Subdivision surfaces** are a modelling technique commonly used to create geometries for animated movies, and are being adopted increasingly by both the video game and computer-aided design (CAD) industries. Starting from a coarse mesh, referred to as the *control net*, repeated refinement and smoothing results in a *limit surface* that is typically $C^2$ smooth everywhere except at so-called *extraordinary points* (EPs) where the surface is merely $G^1$, which becomes $C^1$ when a specific (re)parameterisation is used [Rei95]. These EPs are the limit positions of *extraordinary vertices* (EVs), which are vertices in the original control net with a valency $n$ different from the regular valency, which depends on the subdivision scheme and is 4 for Catmull-Clark [CC78] and 6 for Loop [Loo87] subdivision, the two schemes we focus on.

The refine and smooth operations, together referred to as a *subdivision step*, are captured by *stencils* which represent affine combinations of vertices (or attributes of vertices such as texture coordinates, colour, or weights). Although a vertex in the initial control net only reaches its eventual position on the limit surface after a theoretical infinite number of subdivision steps, it can also be projected to that position from the mesh using a *limit stencil*. As such, the position of an EP can be computed directly from the corresponding EV and its neighbourhood. Similarly, other limit stencils are available for the computation of tangent vectors (and therefore also the normal vector) at an EP [AS10]. Away from EPs the partial derivatives of the limit surface are readily computed using the relevant spline representation for the regular parts of the surface (which are uniform bicubic B-splines for Catmull-Clark and quartic three-directional box-splines for Loop subdivision) and can subsequently be used to compute the normal vector.

**Shading:** When rendering a subdivision surface, either as a dense mesh or as a parametric surface, the normals of the surface are required to compute diffuse and optionally specular reflections. Smooth normal fields are indispensable for aesthetically pleasing results, and therefore require surfaces of high geometrical continuity. In the case of subdivision surfaces, this demand is met away from the EPs. At the EPs the surface normal field $\mathbf{n}_L$ is merely continuous, which can cause artefacts when lighting the surface.

**Subdivision of normals:** As unit normals are points on the unit sphere $\mathbb{S}^2$, they should be subdivided as such, i.e., as points on $\mathbb{S}^2$. This procedure, called *spherical averaging*, is iterative within each subdivision step and relies on the *exponential map* and its inverse. Although it was observed in [AB08] that it results into a smooth normal field, a formal proof remains elusive. We implemented both linear and spherical averaging of normals and observed that the differences when applying subdivision a few times are negligible. In our experiments the maximum angle between these normals was close to zero degrees. All our results rely on linear averaging, which leads to a normal field $\mathbf{n}_S$ that is $C^1$ everywhere.

## 3. Smooth blended subdivision shading

We now focus on (1), showing that the use of certain blending functions $b$ defined on the subdivision surface ultimately results in a blended normal field $\mathbf{n}_B$ that is $C^1$ at EPs and reduces to the original limit surface normals in regular regions. We start by stating the following theorem (its proof can be found in the Appendix).

**Theorem 1** Let $b$ be a subdivision limit function of Catmull-Clark or Loop subdivision such that $b$ attains the value of 1 and a local maximum ($\nabla b = \mathbf{0}$) at each EP, and let $p$ be a positive real number. Then $\mathbf{n}_B$ defined in (1) is globally $C^1$ for $p \geqslant 1$. For $p < 1$ the claim is valid only up to a certain valency which depends on the value of $p$ and the underlying subdivision scheme; see Figure 2.

We consider two approaches for constructing a suitable blending function $b$. Our first attempt, $\mathcal{A}_3$, forces the blending weight at an EP to be 1 by employing a generalised limit stencil [LSNC09]. More precisely, the contribution $w_n$ of the EV itself in the limit stencil is computed, after which a blending weight of $\frac{1}{w_n}$ is assigned to

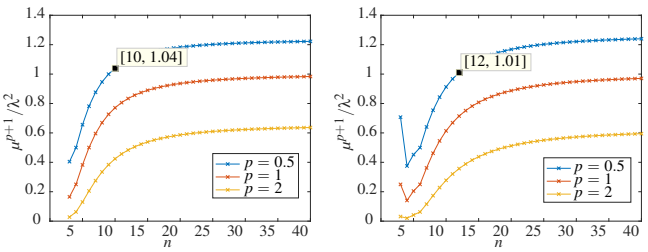


**Figure 2:** *The ratio $\mu^{p+1}/\lambda^2$ plotted against the valency n for different values of p (0.5, 1.0 and 2.0) for Catmull-Clark (left) and Loop (right). For both schemes, the ratio converges to 1 when $p = 1$, and stays below 1 for $p > 1$. For $p < 1$ the ratio becomes greater than 1 after a certain value of n, depending on the scheme.*

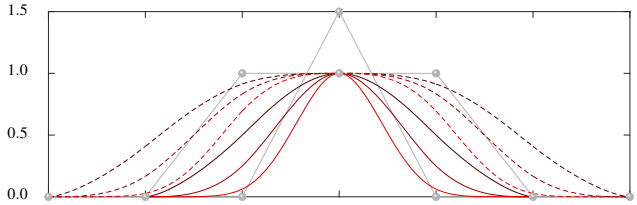


**Figure 3:** *The univariate equivalents of the two types of blending functions ($\mathcal{A}_3$ solid, $\mathcal{A}_4$ dashed) raised to different powers p (0.5, 1.0 and 2.0) shaded from dark red to bright red. The control nets are shown in grey.*

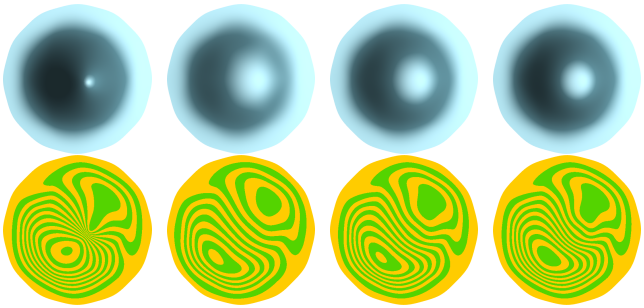

**Figure 4:** *A comparison of different values for p (from left to right: $\mathbf{n}_L$, $p = 0.5$, 1.0 and 2.0) using $\mathcal{A}_3$ for a Catmull-Clark subdivision surface with a central EP. The first row shows the shaded surface and the second row shows isophotes.*

the EV. Assigning zero weights to vertices in the one-ring neighbourhood $\mathcal{N}_1$ of the EV then guarantees a blending weight of 1 at the EP, but only if $\mathcal{N}_1$ does not contain other EVs. In the latter scenario, where a face contains multiple EVs, this procedure clearly does not work. Although EVs can be separated further by applying subdivision, a denser mesh is not always desired. As such, we consider a second approach, $\mathcal{A}_4$, which is more generally applicable. In this setting, we assign a blending weight of 1 to EVs and to the vertices in their $\mathcal{N}_1$. Clearly, this satisfies the required conditions for constructing a suitable $b$. Note that for $\mathcal{A}_3$, only the two-ring neighbourhood $\mathcal{N}_2$ of an EV is affected, whereas for $\mathcal{A}_4$, it is the three-ring neighbourhood $\mathcal{N}_3$ (see Figure 3).

The blending functions resulting from applying either approach can subsequently be raised to a power $p$. Naturally, this does not affect the support of $b$, but determines the rate of decay of $b$ around the EP; see Figure 3. The value of $p$ can differ per vertex (or per region if a face contains multiple EVs). A comparison of different values of $p$ can be found in Figure 4. One can see that a lower value for $p$ makes the result look smoother, while for a larger $p$-value the effect is more local. The (default) value for $p$ in all other figures was set to 1.

Ultimately, we obtain a normal field with subdivided normals at EPs, limit surface normals in the regular regions, and a $C^1$ blend of the two in the regions around EPs that correspond to the two- or three-ring of faces around their associated EVs.

## 4. Implementation

In this section we briefly describe the steps involved in implementing the proposed blending method.

**Initialization:** The *geometry* is represented by the provided control net (imported from e.g. an OBJ file) and is thus given. We do not assume the file to contain normals. Instead, mesh vertex *normals* are calculated after a custom number of subdivision steps by computing an area-weighted average of the normals of incident faces. Likewise, the *blending weights* can be initialised after a custom number of subdivision steps for approaches $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_4$. In case $\mathcal{A}_3$ is used, EVs have to be sufficiently separated first.

**Subdivision:** For all three types of subdivision (geometry, normals, blending weights) the same subdivision stencils are used. The subdivision depth of each type can be controlled independently, such that the subdivision of normals and/or blending weights can start later than the subdivision of geometry.

**Blending:** After completing the last subdivision step, the two normal fields are blended using Equation 1 based on the computed blending weights.

**Limit positions and normals:** Given a mesh and vertex normals, we obtain the limit positions and normals by applying the limit stencils of the used subdivision scheme. For our results in Section 5 we always applied these limit stencils for the geometry. Likewise, for the normals we applied these limit stencils when calculating subdivided normals.

## 5. Results and discussion

We first compare the blending weights for all approaches $\mathcal{A}_k$, $k \in \{1,2,3,4\}$; see Figure 5. In scenarios with neighbouring EVs and/or extraordinary faces, or EVs with high valencies, $\mathcal{A}_4$ performs best.

Figure 6 and Figure 7 show isophotes for various (blended) normal fields in the case of Catmull-Clark and Loop subdivision, respectively. The images clearly show that the subdivided normals and the smooth blending of these normals with the surface normals are smoother than the surface normals.

In the case of $\mathcal{A}_3$, it may seem possible to construct a suitable $b$ also in the case when there are edge- or face-connected EVs in the control net. By collecting the limit stencils into one (possibly) global linear system and inverting its matrix, one can achieve $b = 1$

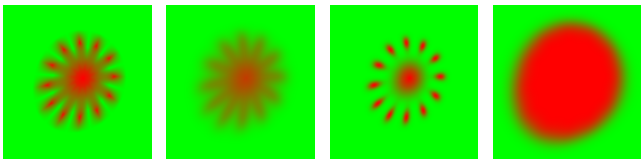

**Figure 5:** *Comparing the blending weights of $\mathcal{A}_1 \ldots \mathcal{A}_4$ (left to right) for a once-subdivided polar configuration (EV with a high valency, $n = 12$, surrounded by triangles). $\mathcal{A}_3$ has to be subdivided once more before the weights can be initialised.*

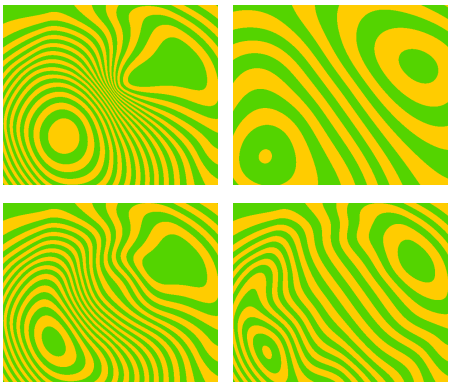

**Figure 6:** *A comparison of* $\mathbf{n}_\mathrm{L}$ *(top left),* $\mathbf{n}_\mathrm{S}$ *(top right),* $\mathcal{A}_3$ *(bottom left) and* $\mathcal{A}_4$ *(bottom right) for a region of a Catmull-Clark subdivision surface.*

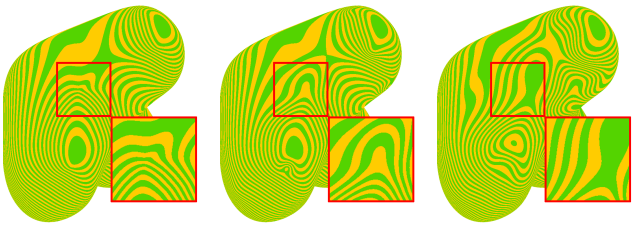

**Figure 7:** *A comparison of* $\mathbf{n}_\mathrm{L}$ *(left),* $\mathcal{A}_3$ *(mid) and* $\mathcal{A}_4$ *(right) for a Loop subdivision surface.*

at all EPs (such matrix is singular only in very special cases). However, we also need $\nabla b = \mathbf{0}$ at all EPs, see Theorem 1. All combined, this results into three linear equations per EP, which will, in general, result into an over-constrained linear system with no solution. In summary, $\mathcal{A}_4$ is the preferred approach in most situations.

**Acknowledgements** This paper is based on the first author's MSc research internship at the University of Groningen.

## References

[AB08] ALEXA M., BOUBEKEUR T.: Subdivision shading. *ACM Trans. Graph. 27*, 5 (Dec. 2008), 142:1–142:4. 1, 2

[AS10] ANDERSSON L.-E., STEWART N. F.: *Introduction to the mathematics of subdivision surfaces*. SIAM, 2010. 2

[CC78] CATMULL E., CLARK J.: Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10*, 6 (1978), 350–355. 2

[KPR04] KARČIAUSKAS K., PETERS J., REIF U.: Shape characterization of subdivision surfaces — case studies. *Computer Aided Geometric Design 21*, 6 (2004), 601–614. 4

[Loo87] LOOP C.: *Smooth subdivision for surfaces based on triangles*. Master's thesis, University of Utah, 1987. 2

[LSNC09] LOOP C., SCHAEFER S., NI T., CASTAÑO I.: Approximating subdivision surfaces with Gregory patches for hardware tessellation. In *ACM Transactions on Graphics (TOG)* (2009), vol. 28, ACM, p. 151. 2

[PR04] PETERS J., REIF U.: Shape characterization of subdivision surfaces — basic principles. *Computer Aided Geometric Design 21*, 6 (2004), 585–599. 4

[Rei95] REIF U.: A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design 12*, 2 (1995), 153–174. 2, 4

[RS01] REIF U., SCHRÖDER P.: Curvature integrability of subdivision surfaces. *Adv. in Comp. Mathematics 14*, 2 (Feb 2001), 157–174. 4

## Appendix Proof of Theorem 1

*Proof* We assume that the geometry $\mathbf{s}$, the normal field $\mathbf{n}_\mathrm{S}$, and also $b$ are (locally) parametrised using the characteristic map [Rei95], and that the extraordinary point is at $\mathbf{0}$.
Observe that $b$, $\mathbf{n}_\mathrm{L}$, and $\mathbf{n}_\mathrm{S}$ are all, by construction, at least $C^1$ away from $\mathbf{0}$ and so we only need to establish that $\mathbf{n}_\mathrm{B}$ is also $C^1$ at $\mathbf{0}$. This does not follow directly from (1), because $\mathbf{n}_\mathrm{L}$ is only $C^0$ at $\mathbf{0}$, which in turn means that the gradient of $\mathbf{n}_\mathrm{L}$ at $\mathbf{0}$ may not exist.
To resolve this, we apply the machinery devised in [RS01], and further refined [PR04] and applied [KPR04] later, and study the behaviour of $1-b$ and $\mathbf{n}_\mathrm{L}$ (and their derivatives) as we approach $\mathbf{0}$. To simplify notation, we denote the oriented directional derivative operator by $D$ and write for instance $D(b)$ instead of $D_\mathbf{d}(b)$ for a particular (unit) direction $\mathbf{d}$ in the characteristic map at $\mathbf{0}$.

Let $\lambda$ be the sub-dominant eigenvalue of multiplicity 2 and $\mu$ the subsub-dominant eigenvalue of the subdivision scheme in question (Catmull-Clark or Loop). Further, let $\mathbf{s}^l$ denote the spline ring of subdivision level $l$ and $I^l$ and $II^l$ its first and second fundamental form, respectively. As shown in [PR04, Theorem 3.1], the dominant terms of these forms are $\lambda^{2l}$ and $\mu^l$, respectively. Thus, the shape operator (Weingarten map) $S^l = II^l (I^l)^{-1}$ of $\mathbf{s}^l$ behaves (typically diverges) as $\left(\frac{\mu}{\lambda^2}\right)^l$ when $l \to \infty$ at $\mathbf{0}$, and so does $D(\mathbf{n}_\mathrm{L}^l)$, the directional derivative of the normal vector $\mathbf{n}_\mathrm{L}^l = \frac{\partial_1 \mathbf{s}^l \times \partial_2 \mathbf{s}^l}{\|\partial_1 \mathbf{s}^l \times \partial_2 \mathbf{s}^l\|}$ of $\mathbf{s}^l$.
By assumption, $b(\mathbf{0}) = 1$ and thus $\mathbf{n}_\mathrm{B}(\mathbf{0}) = \mathbf{n}_\mathrm{S}(\mathbf{0})$ by (1) for any value of $p$. As $b$ is continuously differentiable and attains a maximum at $\mathbf{0}$, it follows that $D(b^p) = pb^{p-1}D(b)$ vanishes at $\mathbf{0}$. For the spline ring $\mathbf{s}^l$ of level $l$, (1) reads

$$\mathbf{n}_\mathrm{B}^l = (1-b_l^p)\mathbf{n}_\mathrm{L}^l + b_l^p \mathbf{n}_\mathrm{S}^l,$$

where $b_l$ is the spline ring of $b$ of level $l$. Differentiating this yields

$$D(\mathbf{n}_\mathrm{B}^l) = -D(b_l^p)\mathbf{n}_\mathrm{L}^l + (1-b_l^p)D(\mathbf{n}_\mathrm{L}^l) + D(b_l^p)\mathbf{n}_\mathrm{S}^l + b_l^p D(\mathbf{n}_\mathrm{S}^l). \quad (2)$$

We now push $l$ to the limit, $l \to \infty$, i.e., we approach $\mathbf{0}$. Since $D(b_l^p)$ vanishes in the limit as observed above, the first and third summand on the right-hand side of (2) vanish. As expected, the only problematic term is the product $(1-b_l^p)D(\mathbf{n}_\mathrm{L}^l)$. But this behaves as

$$\mu^{lp}\left(\frac{\mu}{\lambda^2}\right)^l = \left(\frac{\mu^{p+1}}{\lambda^2}\right)^l$$

in the limit. This shows that the problematic term vanishes at $\mathbf{0}$ provided that $\mu^{p+1} < \lambda^2$. This is indeed the case for the Catmull-Clark and Loop subdivision schemes at any valency for $p \geqslant 1$ as depicted in Figure 2. For $p < 1$, this is only valid up to a certain valency depending on the value of $p$. For example, with $p = \frac{1}{2}$, the condition holds for Catmull-Clark up to $n = 9$ and for Loop up to $n = 11$. However, we note that the default value is $p = 1$ and that in practice $p \ll 1$ is rarely used, especially around vertices of high valency. Consequently, $D(\mathbf{n}_\mathrm{B})$ is well defined for $p \geqslant 1$ and behaves as $D(\mathbf{n}_\mathrm{S})$ at $\mathbf{0}$, which completes the proof. □