Subdivision Exterior Calculus for Geometry Processing
Supplemental Material: Subdivision Rules

Fernando de Goes
Pixar Animation Studios

Mathieu Desbrun
Caltech

Mark Meyer
Pixar Animation Studios

Tony DeRose
Pixar Animation Studios

This supplemental material presents the rules needed to construct the subdivision matrices for discrete 0-, 1-, and 2-forms using the Loop scheme for triangle meshes and the Catmull-Clark scheme for quadrilateral meshes. We provide both interior and boundary rules for completeness. We denote by n the number of faces adjacent to a vertex. Boundary vertices are marked in black, while interior vertices are marked in white. Vertices without markers may be either interior or on the boundary.

A Loop Subdivision Rules

This section describes the rules for triangle mesh subdivision based on the Loop scheme. Following [Wang et al. 2006], we use the standard Loop subdivision rules from [Biermann et al. 2000] for 0-forms (with $\gamma = {}^3\!/_8$) and generalized half-box splines [Prautzsch et al. 2002] for 2-forms.

The Loop subdivision rules for 0-forms are shown in Fig. 1. Since we use the standard Loop subdivision rules, there is a single even interior stencil and a single odd interior stencil. The even and odd stencils for the boundary are simply the B-spline subdivision rules. The values for α and β are given by:

$$\alpha = \begin{cases} {}^3\!/_{16}, & \text{if } n = 3 \\ {}^3\!/_{8n}, & \text{otherwise} \end{cases} \quad \beta = \begin{cases} {}^1\!/_{12}, & \text{if } n = 3 \\ {}^1\!/_8, & \text{if } n = 4 \\ {}^1\!/_4 - {}^1\!/_{16} \sin^2(2\pi/5), & \text{if } n = 5 \\ {}^1\!/_4, & \text{if } n \geq 6 \end{cases}$$

The Loop subdivision rules for 1-forms are given in Fig. 2. The interior rules, both even and odd, have simple expressions. Subdivided even edges on the boundary and subdivided odd edges adjacent to the boundary also have relatively simple expressions. The even stencils for subdivided edges adjacent to the boundary, however, produce a much larger number of cases. These cases are defined by n, the number of faces adjacent to the boundary vertex, and how far the subdivided edge is from the boundary when walking counter-clockwise around the outgoing edges of the vertex.

The Loop subdivision rules for 2-forms at interior and boundary faces are given in Fig. 3. Similar to the 1-form boundary rules, the 2-form rules are defined by n and how far the subdivided face is from the boundary when walking counter-clockwise around the outgoing faces of the boundary vertex.

With these rules, subdivision commutes with exterior derivatives for any valence and configuration.

B Catmull-Clark Subdivision Rules

This section describes the rules for quadrilateral mesh subdivision. We use the standard Catmull-Clark subdivision rules described in [DeRose et al. 1998] for 0-forms and Doo-Sabin subdivision rules [Wang et al. 2006] for 2-forms.

The subdivision rules for 0-forms are shown in Fig. 4. During subdivision, a new vertex must be inserted for each vertex, edge, and face of the unrefined mesh. These three cases are denoted, respectively, as *Vertex Vertex*, *Edge Vertex* and *Face Vertex*. The boundary rules simply reproduce the standard B-spline subdivision. Values for β and γ are given by:

$$\beta = {}^3\!/_{2n}, \quad \gamma = {}^1\!/_{4n}.$$

The subdivision rules for 1-forms are given in Fig. 6. The interior, even boundary, and odd boundary-adjacent rules all have relatively simple expressions. However, the even boundary-adjacent rules include several special cases. Expressions remain simple for $n = 2$ (two boundary faces), and for the case when the subdivided edge is one-away from the boundary. The general case, on the other hand, is parameterized by the edge index e of the subdivided edge that indicates the number of edges when walking counter-clockwise around the boundary vertex one-ring starting at the boundary. As shown in Fig. 6, these subdivision rules involve three set of coefficients σ, ξ, and η. The vector of values for σ, ξ, and η for an edge index e is computed recursively w.r.t. e, based on the coefficients associated to the neighboring edge of index $e + 1$. Base case is set with $e = n - 2$. We give pseudocode in Alg. 1 implementing this recursive computation.

The subdivision rules for 2-forms are given in Fig. 5. Looking at the faces in the coarse one-ring of the vertex corresponding to the subdivided face (shown in black), the subdivision weights follow a simple pattern: the face to be subdivided receives a weight of f_0, the left and right neighbors (in the one-ring) receive a weight of f_1, and all other faces receive a weight of f_2. Note that, if there are only three faces ($n = 3$), the weights are simply $\{f_1, f_0, f_1\}$ (no faces have weight f_2). The constant f_0 is computed such that the sum of all face weights is equal to ${}^1\!/_4$. The boundary subdivision rules for 2-forms follow the same pattern, e.g., if the subdivided face was adjacent to the boundary, then the weights are $\{f_0, f_1, f_2, \ldots, f_2\}$.

With these rules, subdivision commutes with exterior derivatives for any valence and configuration.

References

BIERMANN, H., LEVIN, A., AND ZORIN, D. 2000. Piecewise smooth subdivision surfaces with normal control. In *ACM SIGGRAPH*, 113–120.

DEROSE, T., KASS, M., AND TRUONG, T. 1998. Subdivision surfaces in character animation. ACM SIGGRAPH, 85–94.

PRAUTZSCH, H., BOEHM, W., AND PALUSZNY, M. 2002. *Bezier and B-Spline Techniques*. Springer-Verlag New York, Inc.

WANG, K., WEIWEI, TONG, Y., DESBRUN, M., AND SCHRÖDER, P. 2006. Edge subdivision schemes and the construction of smooth vector fields. *ACM Trans. Graph. 25*, 3, 1041–1048.

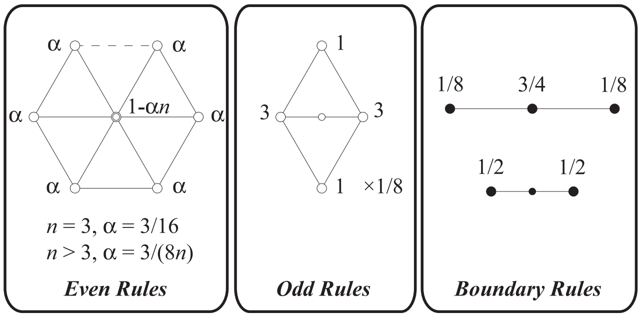

Figure 1: *Loop subdivision rules for* 0*-forms.*

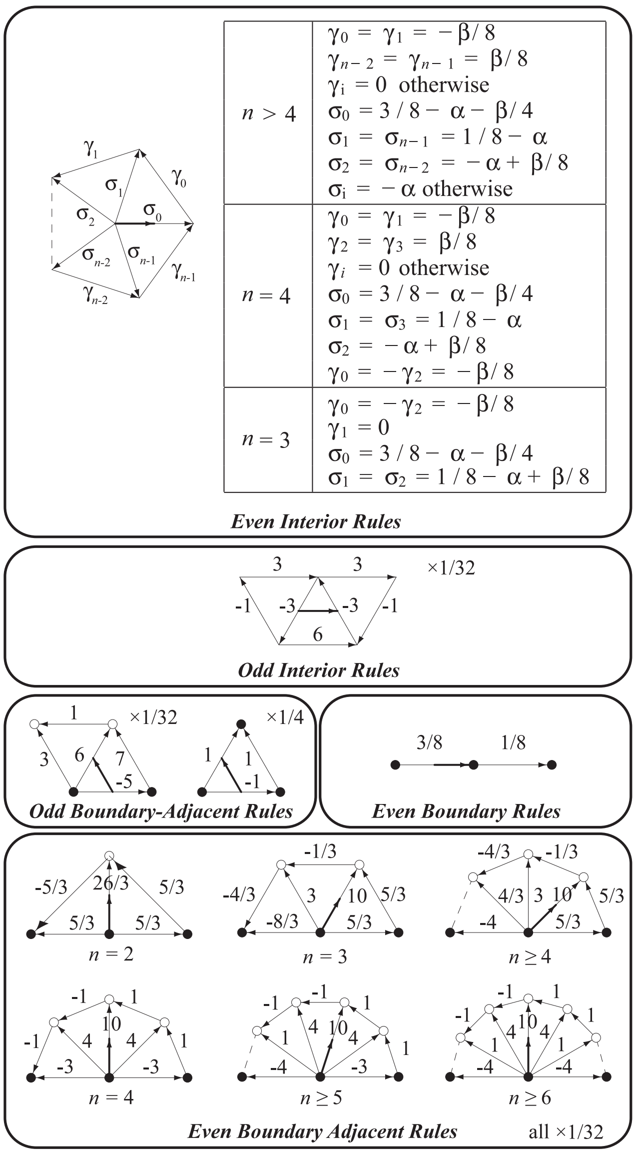

Figure 2: *Loop subdivision rules for* 1*-forms.*

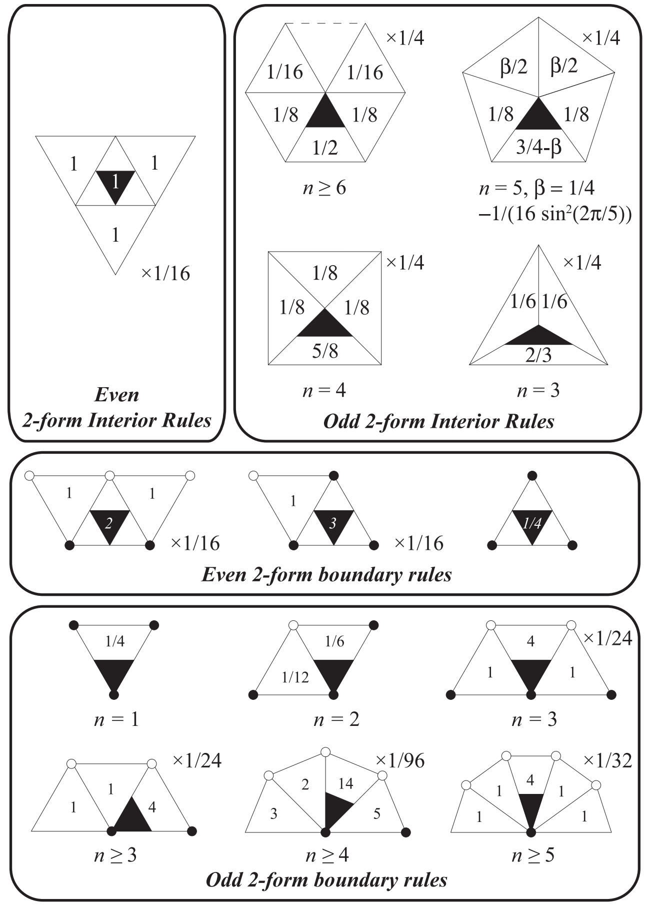

Figure 3: *Loop subdivision rules for* 2*-forms.*

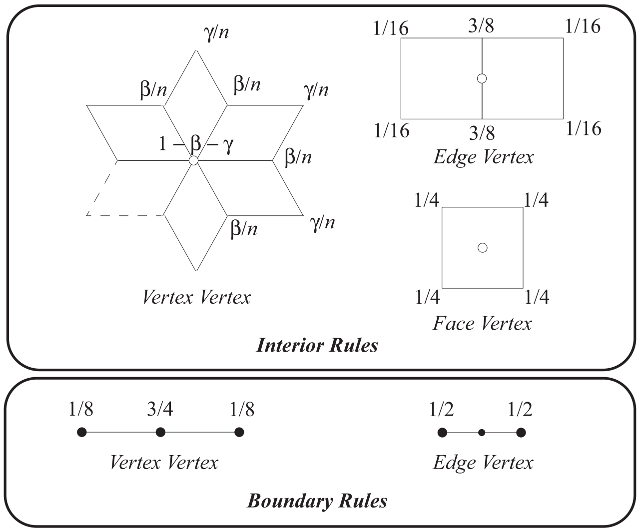

Figure 4: *Catmull-Clark subdivision rules for* 0*-forms.*

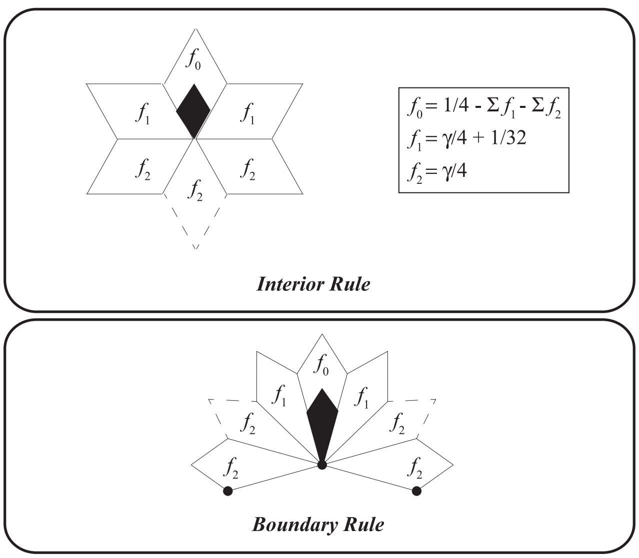

Figure 5: *Catmull-Clark subdivision rules for* 2*-forms.*

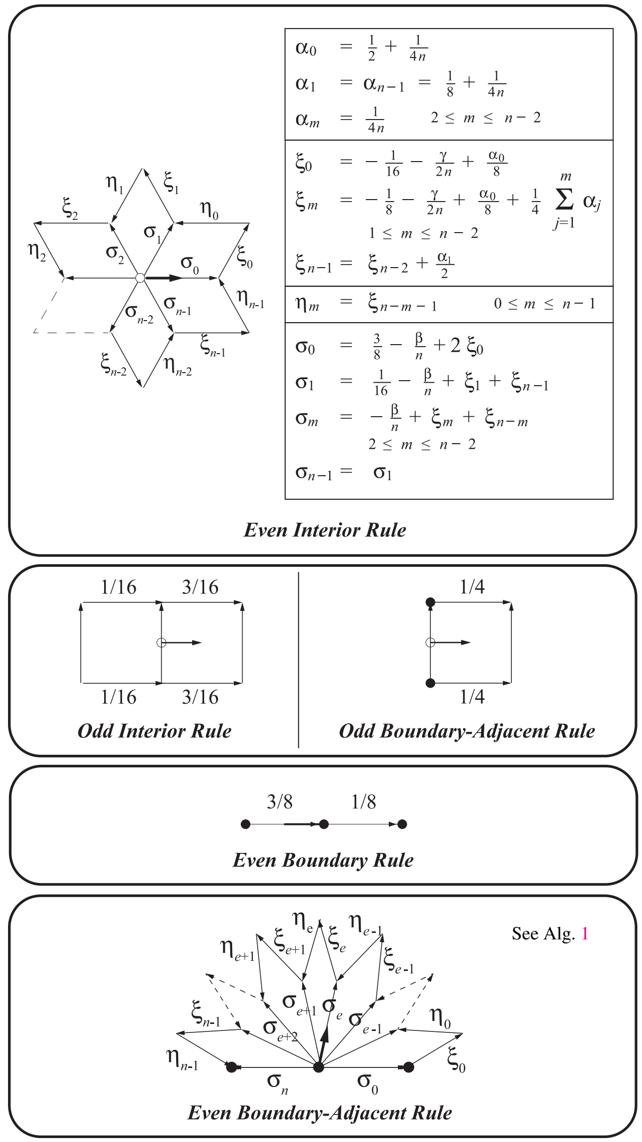

Figure 6: *Catmull-Clark subdivision rules for* 1*-forms.*

Algorithm 1 Computing coefficients of Catmull-Clark subdivision rules for 1-forms at *even boundary-adjacent edges*.

Routine GetWeights(e, n):
// e is the index of the outgoing even boundary-adjacent edge.
// n is the number of faces incident to the boundary vertex.
// for $1 \leq e \leq n$ - 1 and $n \geq 2$

Initialize $\sigma_i = 0$, for $0 \leq i \leq n$
Initialize $\eta_i = \xi_i = 0$, for $0 \leq i < n$
if $n == 2$ **then**
  $\sigma_1 = 3/8$; $\xi_0 = 1/16$; $\eta_1 = -1/16$;
  return $\{\sigma, \xi, \eta\}$
end if

$f_2 = \gamma/4$
$f_1 = 1/32 + \gamma/4$
$f_0 = 1/4 - 2f_1 - (n-3)f_2$

if $e == 1$ **then**
  $\sigma_0 = 3/16 - f_0$; $\sigma_1 = 1/4 + f_0 - f_1$; $\sigma_2 = 1/16 + f_1 - f_2$; $\sigma_n = f_2 - 1/8$;

$$\xi_i = \begin{cases} 1/4 - f_0, & \text{for } i = 0 \\ 1/16 - f_1, & \text{for } i = 1 \\ -f_2, & \text{otherwise} \end{cases} \qquad \eta_i = \begin{cases} 3/16 - f_0, & \text{for } i = 0 \\ -f_1, & \text{for } i = 1 \\ -f_2, & \text{otherwise} \end{cases}$$

else if $e == n - 1$ **then**
  $\sigma_n = 3/16 - f_0$; $\sigma_{n-1} = 1/4 + f_0 - f_1$; $\sigma_{n-2} = 1/16 + f_1 - f_2$; $\sigma_0 = f_2 - 1/8$;

$$\xi_i = \begin{cases} f_0 - 3/16, & \text{for } i = n\text{-}1 \\ f_1, & \text{for } i = n\text{-}2 \\ f_2, & \text{otherwise} \end{cases} \qquad \eta_i = \begin{cases} f_0 - 1/4, & \text{for } i = n\text{-}1 \\ f_1 - 1/16, & \text{for } i = n\text{-}2 \\ f_2, & \text{otherwise} \end{cases}$$

else
  $\{A, B, C\} =$ GetWeightVector(e,n)
  $\sigma_0 = A[0]$; $\sigma_{e-1} = A[1]$; $\sigma_e = A[2]$; $\sigma_{e+1} = A[3]$; $\sigma_n = A[5]$;
  if $n > 4$ and $e < n$-2 **then**
    $\sigma_{e+2} = A[4]$
  end if

$$\xi_i = \begin{cases} B[0], & \text{for } 0 \leq i \leq e\text{-}2 \\ B[1], & \text{for } i = e\text{-}1 \\ B[2], & \text{for } i = e \\ B[3], & \text{for } i = e\text{+}1 \\ B[4], & \text{otherwise} \end{cases} \qquad \eta_i = \begin{cases} C[0], & \text{for } 0 \leq i \leq e\text{-}2 \\ C[1], & \text{for } i = e\text{-}1 \\ C[2], & \text{for } i = e \\ C[3], & \text{for } i = e\text{+}1 \\ C[4], & \text{otherwise} \end{cases}$$

end if
return $\{\sigma, \xi, \eta\}$

SubRoutine GetWeightVector(e, n):
// for $2 \leq e \leq n$ - 2 and $n \geq 3$

$f_2 = \gamma/4$; $f_1 = 1/32 + \gamma/4$; $f_0 = 1/4 - 2f_1 - (n-3)f_2$;

if $e == n - 2$ **then**
  // Base case, edge two-away from boundary
  $A[0] = \gamma/2 - 1/8$; $A[1] = f_1 - f_2 + 1/16$;
  $A[2] = f_0 - f_2 + 1/4$; $A[3] = f_1 - f_2 + 1/16$;
  $A[4] = 0$; $A[5] = 1/8 - f_0 - 2f_1 + f_2$;
  $B[0] = \gamma/2$; $B[1] = f_1 + f_2$;
  $B[2] = f_0 + f_1 - 3/16$; $B[3] = f_0 + 2f_1 - f_2 - 1/4$;
  $B[4] = 0$
else
  // Recursive case
  $\{\widetilde{A}, \widetilde{B}, \widetilde{C}\}$ = GetWeightVector($e+1$, n)
  $A[0] = \widetilde{A}[0] + f_2$; $A[1] = f_1 - f_2 + 1/16$;
  $A[2] = \widetilde{A}[1] + f_0 - f_1 + 3/16$ $A[3] = \widetilde{A}[2] + f_1 - f_0 - 3/16$;
  $A[4] = \widetilde{A}[3] + f_2 - f_1 - 1/16$; $A[5] = \widetilde{A}[5] - f_2$
  $B[0] = \widetilde{B}[0] + f_2$; $B[1] = \widetilde{B}[0] + f_1$;
  $B[2] = \widetilde{B}[1] + f_0 - 3/16$; $B[3] = \widetilde{B}[2] + f_1 - 1/16$;
  $B[4] = \widetilde{B}[3] + f_2$
end if
$C[0] = B[0]$ $C[1] = B[1] - 1/16$ $C[2] = B[2] - 1/16$
$C[3] = B[3]$ $C[4] = B[4]$
return $\{A, B, C\}$