

SEMI P49-0823 (Preliminary) SPECIFICATION FOR EXPERIMENTAL CURVILINEAR MULTIGON EXTENSION TO SEMI P39

This Standard was technically approved by the Microlithography Global Technical Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on May 10, 2023. Available at www.semiviews.org and www.semi.org in August 2023.

NOTICE: This is a Preliminary Standard. Preliminary Standards are approved for publication by a single chapter of a technical global committee for information and comment prior to letter balloting for its adoption as a Full-consensus Standard. Preliminary Standards are published for a period of no more than two years, unless a publication extension is granted by the International Standards Committee. This Preliminary Standard will be removed from publication, and will no longer be available for purchase, during the August 2025 publication cycle. Suggestions for improvements should be submitted to the Microlithography Global Technical Committee or Standards staff.

NOTICE: The section symbol (§) is used to reference a numbered section, paragraph, or header and all subordinate paragraphs, headers, and embedded material (i.e., EXCEPTIONs, lists) therein. The paragraph symbol (§§) is used to reference a particular numbered paragraph and embedded material therein. When duplicated (i.e., §§§ and §§§§) the symbols are used to reference more than one section or paragraph, respectively.

1 Purpose

1.1 The purpose of this Specification is to define an experimental record for SEMI P39 OASIS which is capable of describing curvilinear figures using splines. This Specification was drafted by the SEMI Curvilinear Format Task Force.

2 Scope

2.1 This Specification intends to reduce the size of OASIS files containing inverse lithography technology (ILT) figures, which can only be stored as piecewise linear polygons in an OASIS file. This is predicated on the assumption that storing spline control points instead of polygon vertex points will require fewer points to be stored for each figure, for a given standard of pattern fidelity.

2.2 This Specification enables interoperability between tools processing curvilinear figures, by providing a mechanism to pass spline-based figures from one tool to another without interpolation or information loss.

2.3 This Specification provides a basis for the task force participants to make a quantitative assessment of the degree of file size reduction that can be achieved through the use of this extension.

2.4 This extension defines a **superset** of SEMI P39 OASIS. Files created using this extension are, of necessity, incompatible with existing SEMI P39 OASIS readers.

NOTICE: SEMI Standards and Safety Guidelines do not purport to address all safety issues associated with their use. It is the responsibility of the users of the Documents to establish appropriate safety and health practices, and determine the applicability of regulatory or other limitations prior to use.

3 Referenced Standards and Documents

3.1 SEMI Standards and Documents

SEMI P39 — Specification for OASIS¹ – Open Artwork System Interchange Standard

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

4 Terminology

4.1 None.

¹ OASIS® is a registered trademark of Thomas Grebinski, and is licensed for use by SEMI.

5 Overview of Modifications

5.1 *New Magic Bytes* — SEMI P39 (§ 6.4) requires a <magic-bytes> sequence at the beginning of each file. This extension requires a different <magic-bytes> sequence: “%SEMI-OASIS-CURVILINEAR<CR><NL>” to allow readers to distinguish files containing curvilinear extensions from conventional SEMI P39 OASIS files.

5.2 *New Modal Variables* — The following additional modal variables are defined by this supplement, all of which are initialized to a state of *undefined* at the beginning of each cell. Referencing any of these new modal variables when they are *undefined* should be treated as a fatal error.

5.2.1 *multigon-perimeter-x-offset* and *multigon-perimeter-y-offset* are used to offset embedded holes within an exterior perimeter.

5.2.2 *multigon-implicit-ratio* is used for projecting the implicit control points of Type 2 implicit cubic piecewise Bézier sections.

5.3 New **MULTIGON** record as detailed below:

5.3.1 **MULTIGON** Record

5.3.1.1 A **MULTIGON** record represents a composite geometric figure, expressed as an exterior *perimeter*, plus zero or more associated interior hole *perimeters*. Each *perimeter* may comprise a mixture of piecewise linear and curvilinear *sections*. It has the following format:

‘35’ multigon-info-byte [layer-number] [datatype-number] perimeter-list [x] [y] [repetition]

5.3.1.2 The **multigon-info-byte** contains the bit pattern ‘000XYRDL’.

5.3.1.3 *x* and *y* are **signed-integer** coordinates representing either the absolute or the relative (*x,y*) location of the *anchor-location* of the multigon. **X** is 1 if *x* is present, and **Y** is 1 if *y* is present. When either *x* or *y* is unspecified, the value of modal variable *geometry-x* or *geometry-y*, respectively, is used instead. Refer to SEMI P39 § 21 for a discussion of how *absolute* and *relative* modes affect the interpretation of *x* and *y*.

5.3.1.4 **R** is 1 if **repetition** is present. **L** is 1 if **layer-number** is present. **D** is 1 if **datatype-number** is present. Both **layer-number** and **datatype-number** are *unsigned-integers*. When **layer-number** and/or **datatype-number** are unspecified, they assume the value of modal variables *layer* and *datatype*, respectively.

5.3.1.5 The **perimeter-list** comprises one or more closed *perimeters*. Each *perimeter* has an implied tone— positive tone for the exterior *perimeter*, and negative tone for interior *perimeters* representing holes. The tone of the initial *perimeter*, representing the figure’s exterior, is always positive. No additional positive-tone *perimeters* are allowed. Zero or more subsequent negative-tone *perimeters* are permitted, representing interior holes. A *perimeter* has the following format: **perimeter-info-byte section-list [x-offset] [y-offset] [perimeter-repetition]**

5.3.1.6 The **perimeter-info-byte** contains the bit pattern ‘CXYR0000’.

5.3.1.7 The **C** bit denotes continuation. When **C** = 1, another *perimeter* follows. **C** = 0 denotes the final *perimeter*.

5.3.1.8 **x-offset** and **y-offset** are *signed-integers* representing an offset between the multigon’s *anchor-location* and the initial vertex of every interior *perimeter*. **X** is 1 if **x-offset** is present, **Y** is 1 if **y-offset** is present. For the exterior *perimeter* (the first *perimeter*), the multigon’s *anchor-location* is used as the initial vertex, so both **X** and **Y** bits must be 0. For each subsequent *perimeter*, when either **x-offset** or **y-offset** is unspecified, the value of modal variable *multigon-perimeter-x-offset* or *multigon-perimeter-y-offset*, respectively, is used instead. Modal variables *multigon-perimeter-x-offset* and *multigon-perimeter-y-offset* are in an *undefined* state at the start of each CELL and become *defined* whenever **x-offset** or **y-offset** is first encountered, respectively. They can subsequently be implicitly used and updated by successive *perimeters* across multigons.

5.3.1.9 **R** is 1 if **perimeter-repetition** is present, providing a way to more compactly describe groups of multiple identical hole perimeters. *Perimeter-repetition* has identical format as *repetition*, but it cannot reference the *repetition* modal and hence cannot have Type 0. Its occurrence also does not update the *repetition* modal variable. Note that **R** must be 0 for the first *perimeter*, since each MULTIGON record has exactly one exterior *perimeter*.

5.3.1.10 The **section-list** comprises one or more *sections* which, when concatenated, form a closed geometric figure. A *section* has the following format:

section-info-byte [ratio] point-list

5.3.1.11 The **section-info-byte** contains the bit pattern ‘CAAAR000’.

5.3.1.12 The **C** bit denotes continuation. When **C** = 1, an adjacent *section* follows, to be concatenated with the immediate *section*. **C** = 0 denotes the final *section* of a *perimeter*.

5.3.1.13 **R** = 1 denotes the presence of **ratio**, a positive *real* number used for deriving intermediate cubic Bézier control points for Type 2 implicit cubic piecewise Bézier *sections*. **R** must be 0 for all other *section* types. For an implicit cubic piecewise Bézier *section*, if **R** = 0, the value of modal variable *multigon-implicit-ratio* is used. Modal variable *multigon-implicit-ratio* is *undefined* at the start of each CELL and becomes *defined* when **ratio** is first encountered, after which it applies to all subsequent sections and MULTIGON records of that CELL, until it is changed by a subsequent section.

5.3.1.14 The **AAA** bits denote the section type as indicated in Table 1.

Table 1 Section Types

Type	Description
0	Piecewise linear section
1	Explicit cubic piecewise Bézier section
2	Implicit cubic piecewise Bézier section
3	Uniform cubic B-spline section, clamped
4	Uniform cubic B-spline section, periodic
5-7	Prohibited

5.3.1.14.1 A Type 0 piecewise linear *section* comprises one or more connected straight line segments.

5.3.1.14.1.1 When an exterior or interior *perimeter* consists entirely of one single Type 0 piecewise linear *section*, the implicit closure rules of SEMI P39 § 7 apply, and the final one or two points are omitted, depending on the type of *point-list*.

5.3.1.14.1.2 When a Type 0 piecewise linear *section* is part of a *perimeter* containing multiple *sections*, the implicit closure rules of SEMI P39 § 7 do not apply; all vertex points of the section must be explicitly included in its *point-list*.

5.3.1.14.2 A Type 1 explicit cubic piecewise Bézier *section* comprises one or more cubic Bézier curves that are chained together such that the final control point of each curve is shared as the initial control point of the subsequent curve. Refer to Appendix 3 for more information on cubic Bézier curves. For this type, the *point-list* stores all four control points of member cubic Bézier curves.

5.3.1.14.2.1 An explicit cubic piecewise Bézier *section* comprising 3 cubic Bézier curves is depicted in Figure 1.

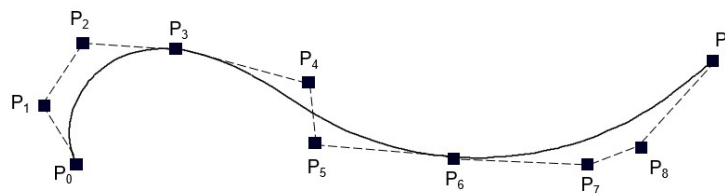
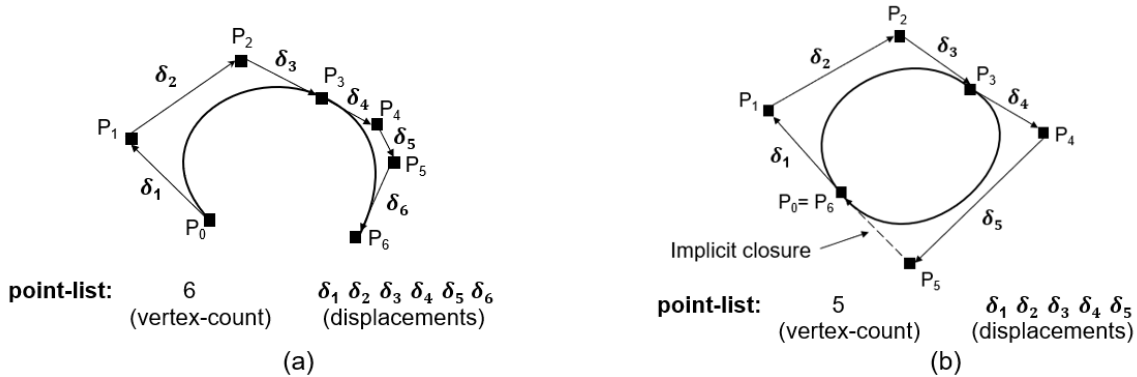


Figure 1
Type 1 Explicit Cubic Piecewise Bézier Section

5.3.1.14.2.2 For a *section* comprising *n* Type 1 explicit cubic Bézier curves, its *point-list's* vertex-count must be $(3 \times n)$, if the *section* is part of a *perimeter* containing multiple *sections*. This case is illustrated in Figure 2 (a). In this

example, the *section* comprises two explicit cubic Bézier curves and *point-list* contains six displacements, from δ_1 to δ_6 with P_0 as its initial vertex. If it is the only section in a *perimeter*, its *point-list*'s *vertex-count* must be $(3 \times n - 1)$, omitting the final point, which must be coincident with the *perimeter*'s initial point in that case. This case is illustrated in Figure 2 (b). In this example, the *section* comprises two explicit cubic Bézier curves and *point-list* contains five displacements, from δ_1 to δ_5 with P_0 as its initial vertex.



#1 (a) *section* is part of a *perimeter* containing multiple *sections*, (b) *section* is a single *perimeter section*.

Figure 2
Type 1 Explicit Cubic Piecewise Bézier Section Point-List

5.3.1.14.3 A Type 2 implicit cubic piecewise Bézier *section* comprises one or more cubic Bézier curves that are chained together such that the final control point of each curve is shared as the initial control point of the subsequent curve. For this type of *section*, the *point-list* stores only the two on-curve (initial and final) control points of each curve. The two intermediate control points are derived from the on-curve control points and their tangents according to a formula involving modal variable *multigon-implicit-ratio*. An implicit cubic piecewise Bézier *section* comprising three cubic Bézier curves is depicted in Figure 3.

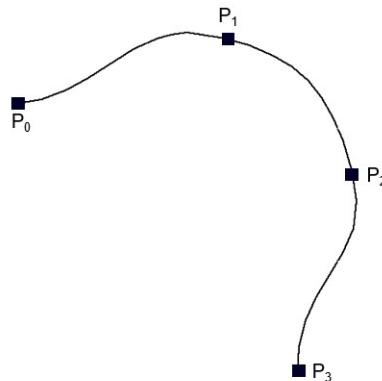
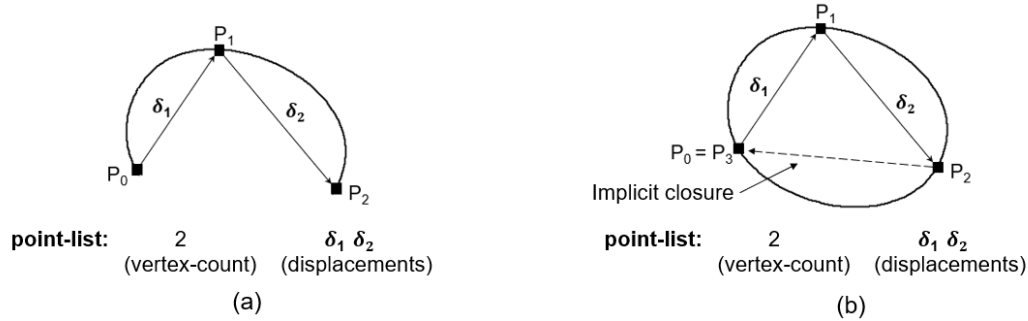


Figure 3
Type 2 Implicit Cubic Piecewise Bézier Section

5.3.1.14.3.1 For a *section* comprising n Type 2 implicit cubic Bézier curves, its *point-list*'s *vertex-count* must be n , if the *section* is part of a *perimeter* containing multiple *sections*. This case is illustrated in Figure 4 (a). In this example, the *section* comprises two implicit cubic Bézier curves and *point-list* contains two displacements, δ_1 and δ_2 with P_0 as its initial vertex. If it is the only section in a *perimeter*, its *point-list*'s *vertex-count* must be $(n - 1)$, omitting the final point, which must be coincident with the *perimeter*'s initial point in that case. This case is illustrated in Figure 4 (b). In this example, the *section* comprises three implicit cubic Bézier curves and *point-list* contains two displacements δ_1 and δ_2 with P_0 as its initial vertex.



#1 (a) section is part of a *perimeter* containing multiple *sections*, (b) section is a single *perimeter* section.

Figure 4

Type 1 Implicit Cubic Piecewise Bézier Section Point-List

5.3.1.14.3.2 Deriving Type 2 implicit cubic Bézier intermediate control points using the on-curve initial and final control points and *multigon-implicit-ratio* is a 3-step process. Figure 5 depicts the steps for the cubic Bézier curve defined between end control points P_1 and P_2 .

- Step 1: Compute tangents at points P_1 and P_2 . The tangent at each of the two on-curve control points depends on the previous and next on-curve points. For example, the tangent at P_1 is perpendicular to the angle bisector of the angle formed by P_1 with P_0 and P_2 , as Shown in Figure 5 (a).
- Step 2: Upon finding the tangents at P_1 and P_2 , an intermediate control point is projected along each tangent, in the direction toward the opposite control point, according to the formula $dist = b \times ratio$, where b is the distance between points P_1 and P_2 , as shown in Figure 5 (b).
- Step 3: The projected intermediate control points are rounded to the nearest integer coordinates and used. If the fractional part of an intermediate control point coordinate is exactly 0.5, it is rounded up, toward positive infinity, regardless of quadrant. Note that rounding should be performed in the coordinate system of the cell that contains the multigon, to ensure figure stability regardless of the orientation(s) at which the cell is placed within the layout hierarchy.

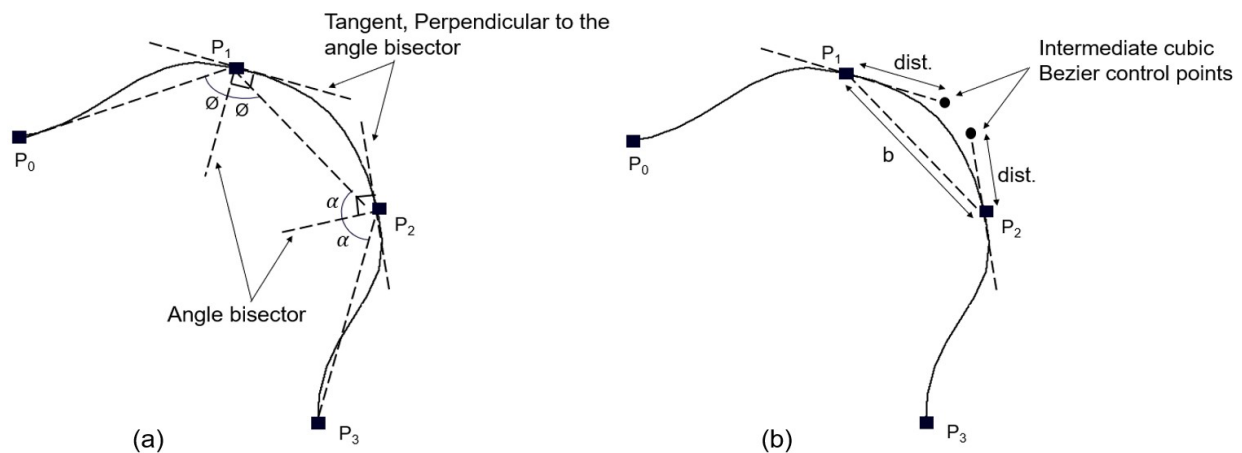


Figure 5

Type 2 Implicit Cubic Bézier Curve Intermediate Control Points Projection

5.3.1.14.3.3 Some extra restrictions apply to the points in a Type 2 (*implicit*) section. Successive duplicate points are not allowed in the *section's point-list*. A degenerate case where the angle at a control point (angles $\emptyset$ or α in Figure 5 [a]) becomes 0 degrees is also forbidden.

5.3.1.14.3.4 In a *perimeter* comprising multiple *sections*, a Type 2 (*implicit*) *section* must connect to either another Type 2 (*implicit*) *section* or to a Type 1 (*explicit*) *section*. Where two adjacent Type 2 (*implicit*) *sections* connect, tangent computation and intermediate control point projection along the tangent, at the *section* junction point, follows the same 3-step process described above. Where a Type 2 (*implicit*) *section* connects to a Type 1 (*explicit*) *section*, the tangent of the Type 1 (*explicit*) Bézier curve at their shared control point (P_3 in Figure 6), is used, and the intermediate control point is projected along that tangent in the direction away from the adjacent *explicit* Bézier curve's control point P_4 , using the process described in steps 2 and 3. Note that, if control points P_3 and P_4 are coincident, the tangent at point P_3 is undefined. In that case, connecting a Type 1 (*explicit*) *section* to a Type 2 (*implicit*) *section* at that point is forbidden.

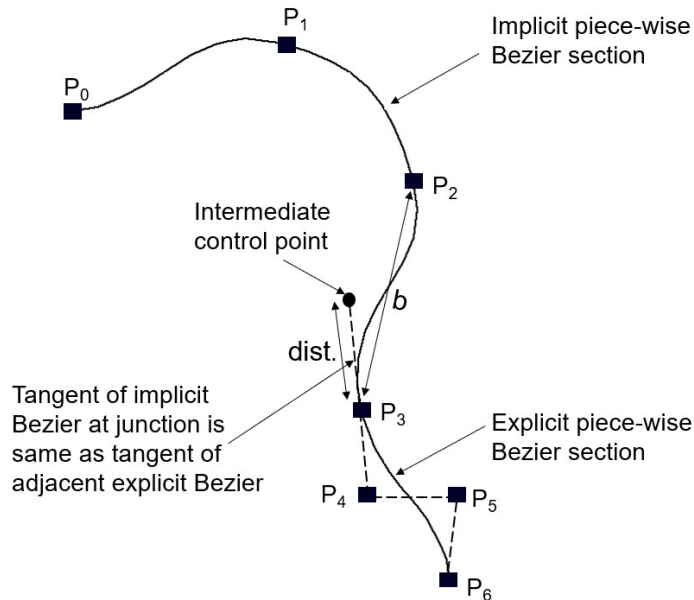
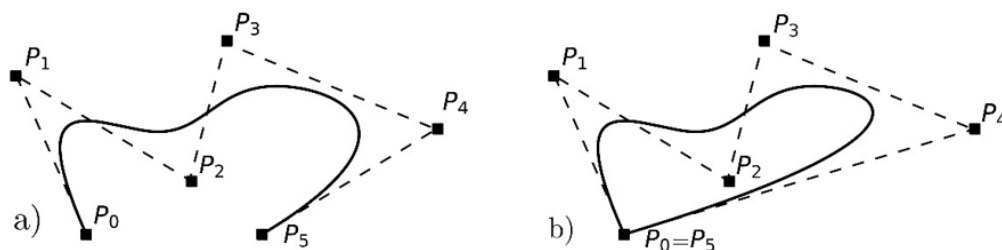


Figure 6
Implicit Cubic Bézier Tangent at Implicit and Explicit Piecewise Bézier Section Junction

5.3.1.14.4 Uniform Clamped Cubic B-Spline

5.3.1.14.4.1 A Type 3 uniform clamped cubic B-spline *section* comprises at least four control points. Note that such B-splines can be either open or closed. Figure 7 shows both open and closed uniform clamped cubic B-splines. The B-spline curve is obtained via a mathematical formula from the control points (refer to Appendix 2).



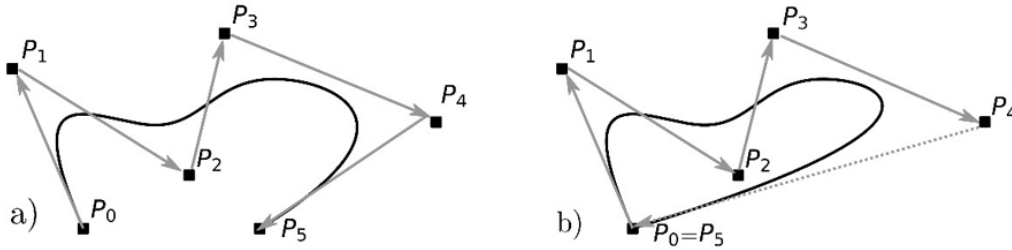
#1 a) open B-spline, and b) closed B-spline where the first control point (P_0) and the last (P_5) are coincident. Dashed lines correspond to the control polygon of the B-spline.

Figure 7
Uniform Clamped Cubic B-Spline

5.3.1.14.4.2 In the case of an open B-spline section (more than one *section* in the *perimeter*), the *point-list* stores all control points except the initial vertex, which is taken from the *anchor-location* of the *perimeter* for the first *section*, or else from the final vertex of the previous *section*.

5.3.1.14.4.3 In the case of a closed B-spline *section* (the only *section* in the *perimeter*), the final displacement is omitted since the *section* is closed (refer to ¶ 5.3.1.16 for details). The first vertex is taken from the *anchor-location* of the *perimeter*.

5.3.1.14.4.4 Both cases are illustrated in Figure 8 below.

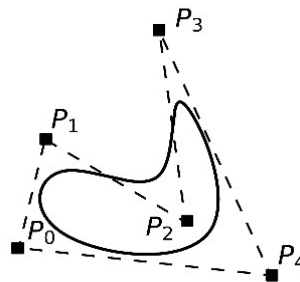


#1 a) open B-spline, and b) closed B-spline. The grey solid arrows show the point list deltas. The grey dashed arrow shows the implicit (not stored) point list delta.

Figure 8
Point List Describing Uniform Clamped Cubic B-Spline

5.3.1.14.5 Uniform Periodic Cubic B-Spline

5.3.1.14.5.1 A Type 4 periodic cubic B-spline is necessarily the sole *section* of a *perimeter* and cannot be mixed with any other *sections*. The section must have at least three control points. Figure 9 shows a periodic B-spline with five control points.



#1 Dashed lines correspond to the control polygon of the B-spline.

Figure 9
Periodic Cubic B-spline with Five Control Points

5.3.1.15 A *point-list* is used to specify the control points of each spline-based *section*. The format of *point-lists* is defined in SEMI P39 (§ 7.7). The *perimeter's* initial vertex is also used as initial vertex of first *section's point-list*. For every subsequent *section*, the final point of the previous *section* is used as initial vertex of the *point-list*. For spline-based sections, Type 4 all-angle *g-delta* based *point-lists* are appropriate, and the use of other *point-list types* is discouraged.

5.3.1.16 For any *perimeter* containing one single spline *section*, implicit closure is used—the final point is omitted from its *point-list* and implied from its initial point. For any *perimeter* with multiple *sections*, the final point of the final *section* must always be explicitly included in its *point-list*.

5.3.1.17 Each successive **MULTIGON** record updates all multigon-related modal variables.

5.3.1.18 Any of the following instances should be treated as a fatal error:

5.3.1.18.1 Any *perimeter* which does not form a closed figure after interpolation.

5.3.1.18.2 The presence of *perimeter-repetition* for the exterior *perimeter* of a multigon.

5.3.1.18.3 Within a multiple-section *perimeter*:

5.3.1.18.3.1 Any Type 1 *section* containing n explicit cubic Bézier curves whose *point-list's vertex-count* is not $(3 \times n)$.

5.3.1.18.3.2 Any Type 2 *section* containing n implicit cubic Bézier curves whose *point-list's vertex-count* is not (n) .

5.3.1.18.4 Within a single-section *perimeter*:

5.3.1.18.4.1 Any Type 1 *section*, whose *point-list's vertex-count* is not $(3 \times n - 1)$.

5.3.1.18.4.2 Any Type 2 *section* containing n implicit cubic Bézier curves whose *point-list's vertex-count* is not $(n - 1)$.

5.3.1.18.5 Successive duplicate points within the *point-list* of a Type 2 implicit cubic Bézier *section*.

5.3.1.18.6 Any Type 2 implicit cubic Bézier on-curve control point for which the tangent is undefined.

5.3.1.18.7 Implicit use of modal variables *multigon-perimeter-x-offset*, *multigon-perimeter-y-offset*, *multigon-implicit-ratio*, *layer*, or *datatype* when they are in the undefined state.

5.3.1.18.8 Any *open uniform clamped B-spline point-list* containing fewer than three points.

5.3.1.18.9 Any *closed uniform clamped B-spline point-list* containing fewer than two points.

5.3.1.18.10 Any Type 4 uniform periodic B-spline *section* which is not the sole *section* in its *perimeter*.

5.3.1.18.11 Any *point-list* within a multigon with fewer than two points.

5.3.1.19 The interpretation of self-intersecting *perimeters*, re-entrant *perimeters*, and *perimeters* with zero-area regions is application-dependent.

APPENDIX 1

EQUATIONS AND DESCRIPTIONS

NOTICE: The material in this Appendix is an official part of SEMI P49 and was approved by full letter ballot procedures on May 10, 2023.

A1-1 Cubic Bézier Curve Equations and Description

A1-1.1 A cubic Bézier curve is a parametric curve defined by four control points, P_0 , P_1 , P_2 , and P_3 , that define the shape of the curve. The curve coincides with end control points P_0 and P_3 . The curve is given by:

$$\mathbf{B}(t) = (1-t)^3 \mathbf{P}_0 + 3(1-t)^2 t \mathbf{P}_1 + 3(1-t)t^2 \mathbf{P}_2 + t^3 \mathbf{P}_3, \quad 0 \leq t \leq 1 \quad (\text{A1-1})$$

A1-1.2 The figure below shows examples of cubic Bézier curves.

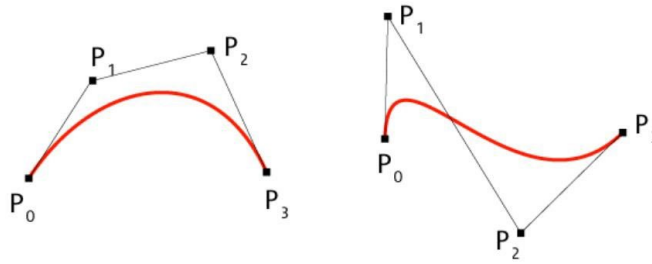


Figure A1-1
Cubic Bézier Curve Examples

A1-2 B-Spline Curve Equations and Description

A1-2.1 B-spline curves are a generalization of Bézier curves. The general equation of a B-spline curve is the following:

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i \quad (\text{A1-2})$$

where:

p = degree of the B-spline,

$\{P_i\} = n+1$ control points,

$\{N_{i,p}\}_i$ = basis functions, which are piecewise polynomials of degree p , and

u = real parameter of the parametric B-spline curve.

A1-2.2 Basis functions are defined from a knot vector $\{u_0, \dots, u_m\}$, which is a non-decreasing sequence of numbers. The number of knots ($m+1$) is defined from the number of control points ($n+1$), and the degree p of the B-spline via the relation $m = n + p + 1$.

A1-2.3 One can compute B-spline basis functions with the following recursive formula²:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (\text{A1-4})$$

² The recursive formula to compute B-spline basis functions can be simplified and optimized in the case of uniform clamped and periodic B-splines.

A1-2.4 Denominators from the left and right terms can be zero. Each time a zero denominator is encountered, it multiplies a basis function which is zero by recursive definition, and the whole term must be set to zero by definition.

A1-2.5 In the following, a description of two B-spline flavors used in the format is provided (uniform clamped and periodic).

A1-3 Properties and Convention

A1-3.1 The B-spline curve is defined if the parameter value verifies $u \in [u_p, u_{m-p}]$, which is by convention $[0.0, 1.0]$.

A1-4 Cubic B-Splines

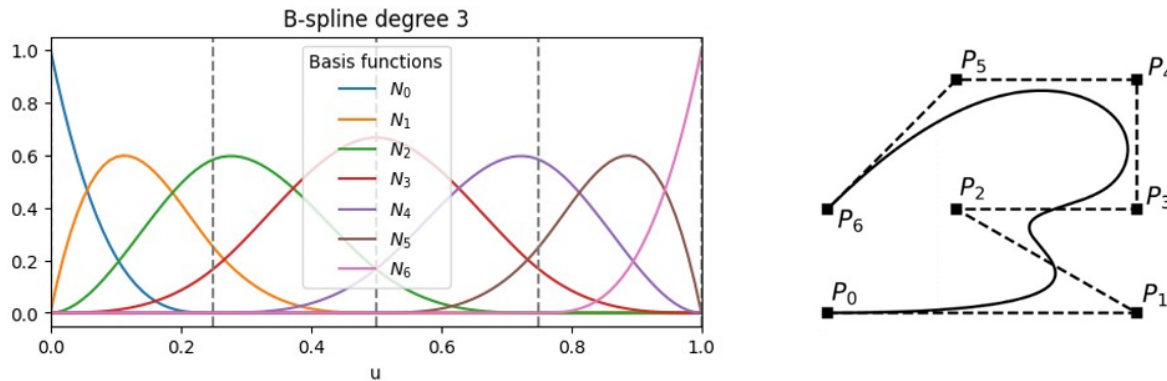
A1-4.1 B-splines used in this format/formula are cubic ($p = 3$). In the following, we keep the p notation for general purposes.

A1-5 Uniform Clamped B-Spline

A1-5.1 A uniform clamped B-spline is a B-spline described by a uniform clamped knot vector. Firstly, a knot vector is clamped when the first $p + 1$ knots are equal, and the last $p + 1$ knots are equal. Secondly, a clamped knot vector is uniform when its internal knots $\{u_p, \dots, u_{m-p}\}$ are equally spaced.

A1-5.2 Note that, for a given number of control points, a uniform clamped knot vector is uniquely defined, and for this reason, it does not need to be stored.

A1-5.3 Example: In the case of seven control points, the cubic uniform clamped knot vector is $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$. Figure A1-2 shows a uniform clamped B-spline curve, and its basis functions defined from seven control points.



- #1 a) Basis functions with respect to the parameter value. Dashed lines correspond to the knots, and the degree from the basis functions is omitted in the legend for clarity. b) Cubic uniform clamped B-spline where $\{P_0, P_1, P_2, P_3, P_4, P_5, P_6\}$ are the control points.

Figure A1-2
Cubic Uniform Clamped Basis Functions (Left) and B-Spline Curve (Right) for Seven Control Points

A1-6 Uniform Periodic B-Spline

A1-6.1 A uniform periodic B-spline is described by a uniform knot vector with the p last control points respectively equal to the p first control points. In this case, the number of free control point in $n + 1 - p$. A uniform knot vector is a knot vector in which all knots are uniform, meaning equally spaced. In the format, these free control points are stored in the *point-list*.

A1-6.2 Example — In the case of five free control points, the knot vector of a periodic B-spline is $\{-0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6\}$. Figure A1-3 shows a periodic B-spline with five free control points $\{P_0, P_1, P_2, P_3, P_4\}$.