Exercise: identify this curve?

- Fix a $t$ and express $P_0^1, P_1^1, P_2^1$ as linear combinations of $P_0, \ldots, P_3$
- Then express $P_0^2$ and $P_1^2$ as linear combinations of $P_0^1, P_1^1$, and $P_2^1$
- Finally, express $P_0^3$ as a linear combination of $P_0^2$ and $P_1^2$
- Answer: cubic Bezier!
How would you have rendered Bezier?

- Treat it as a generic polynomial curve and apply Horner’s rules or differences methods

- But Bezier curves are somewhat special and there is another alternative, from the previous exercise:
Bézier curve via de Casteljau

- Original four control points \( P_0, P_1, P_2, P_3 \) become seven new control points \( l_0, l_1, l_2, l_3 = r_0, r_1, r_2, r_3 \).
- Each set of new control points control half of the Bezier curve.
- In the limit, the control points obtained form the Bézier curve determined by \( P_0, P_1, P_2, P_3 \).
A proof (aside)

- Bézier curve $\mathbf{p}(t) = T \mathbf{M}_B \mathbf{P}$
- $\mathbf{p}(1/2) = P_0/8 + 3P_1/8 + 3P_2/8 + P_3/8 = l_3$
- Reparameterize first half of $\mathbf{p}(t)$: $t \in [0, 1/2]$ to $q(s)$: $s \in [0, 1]$

$$q(s) = p(s/2) = [1 \quad s/2 \quad s^2/4 \quad s^3/8] \mathbf{M}_B \mathbf{P} =$$

$$S = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1/2 & 0 & 0 \\
0 & 0 & 1/4 & 0 \\
0 & 0 & 0 & 1/8
\end{bmatrix}, \quad \mathbf{M}_B \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{S} \mathbf{M}_B$$

$$P = \mathbf{S} \mathbf{M}_B \begin{bmatrix} l_0 \\
l_1 \\
l_2 \\
l_3 \end{bmatrix}$$

- Second half of $\mathbf{p}(t)$ is similar
de Casteljau = subdivision

This is a **subdivision** scheme:

- Subdivide to obtain new points (**refinement** procedure)
- New points ($l$'s and $r$'s) are **weighted averages** of the old ($P$'s)
- Note: de Casteljau’s is not interpolatory except at the boundary

![Diagram showing the de Casteljau algorithm]

In general, $p^{(k+1)} = Sp^{(k)}$, $S$ is a **subdivision matrix**

\[
\begin{bmatrix}
    l_0 \\
    l_1 \\
    l_2 \\
    l_3 \\
    r_1 \\
    r_2 \\
    r_3
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & 0 & 0 \\
    1/2 & 1/2 & 0 & 0 \\
    1/4 & 1/2 & 1/4 & 0 \\
    1/8 & 3/8 & 3/8 & 1/8 \\
    0 & 1/4 & 1/2 & 1/4 \\
    0 & 0 & 1/2 & 1/2 \\
    0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    P_0 \\
    P_1 \\
    P_2 \\
    P_3
\end{bmatrix}
\]
Cubic Bézier via subdivision

- Keep subdividing until sufficiently fine, then connect adjacent control points obtained to form polygonal curve
- A recursive algorithm
- Involve only additions and divisions by 2 — shifts
- Very fast
- Multi-resolution!
How about B-Splines?

- B-splines can also be generated via subdivision, in the same form
  \[ \mathbf{c}^{(k+1)} = S \mathbf{c}^{(k)} \]

- Consider any curve represented in \( l \)-th degree B-spline basis (the \( B \)'s)
  \[ p(t) = \sum_i p_i B_i^l(t) \]

  where \( l \) is the B-spline degree, \( i \) the index, and \( p_i \)'s are control points.

- In matrix form, we have \( p(t) = \mathbf{B}(t) \mathbf{p} \), where
  \( \mathbf{p} \): column vector of control points
  \( \mathbf{B}(t) \): row vector of B-spline bases
B-Splines via subdivision

- Continue from matrix representation: \( p(t) = B(t) \ p = \)

- Eventually, we shall rewrite

\[
p(t) = B(t) \ p = B(2t) \ S \ p
\]

where

- \( S \) is the subdivision matrix

- \( p' = S \ p \) is the new, refined set of control points

- \( B(2t) \) represent refined B-spline basis functions

- Let us focus on uniform B-splines
What are splines?

- An $m$-th degree spline is a piecewise polynomial of degree $m$ that is $C^{m-1}$.

- A spline curve is defined by a knot sequence; the knots are at parametric $t$ values where the polynomial pieces join.

- Most common are uniform knot spacing, i.e., $t = 0, 1, 2, \ldots$ Nonuniform knot spacing or repeated knots are also possible.

- A spline basis often serves as a blending function with local control.

- Resulting spline curve is given by a set of control points blended by shifted or translated versions of the spline basis.
Example: uniform B-splines

- B-splines: one particular class of spline curves

Degree 0-3 uniform B-splines

A piecewise linear curve ($C^0$) obtained by blending five uniform degree-1 B-splines with control points

Note **local control** and **increased continuity**
Key property of uniform B-splines

- A uniform B-spline can be written as a **linear combination** of translated \((k)\) and **dilated** or **compressed** \((2t)\) copies of itself.

- This is the key to connect B-splines to subdivision.

Degree one, \(l = 1\):

![Diagram showing linear combination of B-splines with degree one]

1/2 1 1/2
Technical details (aside)

- B-spline of degree $l$, $B_l(t)$, is $C^{l-1}$ continuous, $l \geq 1$
- The $i$-th B-spline, $B_i^l$, is simply a translate of the B-spline $B_l(t)$ or $B_i^0(t)$: $B_i^l(t) = B_l(t - i)$ — right shift of $i$ units
- B-splines satisfy the refinement equation

$$B_l(t) = \frac{1}{2^l} \sum_{k=0}^{l+1} \binom{l+1}{k} B_l(2t - k)$$

- A uniform B-spline can be written as a linear combination of translated $(k)$ and dilated or compressed $(2t)$ copies of itself
- This is the key to connect B-splines to subdivision
B-spline via subdivision

- Using the refinement equation, we have

\[ B(t) = B(2t) \cdot S \]

where the entries of \( S \) are given by the refinement equation.

\[
S_{2i+k,i} = s_k = \frac{1}{2^l} \binom{l+1}{k}
\]

- Thus, \( p(t) = B(t) \cdot p = B(2t) \cdot Sp \)

We have changed B-spline bases \( B(t) \) to \( B(2t) \), where each element of \( B(2t) \) is half as wide as one in \( B(t) \) and the sequence in \( B(2t) \) are spaced twice as dense.
Refinement of B-splines

Linear B-spline case; this extends to B-splines of any degree.
What have we done?

- **Refined the B-spline basis** functions, and at the same time,
- **Refined the set of control points** $p$
- Twice as many new control points $p' = Sp$:
  - One new point (an **odd point**) is inserted between two consecutive control points in $p$
  - Each control point in $p$ (an **even point**) is either retained (**interpolatory**) or moved (**approximating**) in $p'$
- $S$ is the **subdivision matrix**
**Subdivision matrix $S$**

<table>
<thead>
<tr>
<th></th>
<th>even points</th>
<th>odd points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>1 0 0 0 0 0</td>
<td>0 0 1/2 1/2 0</td>
</tr>
<tr>
<td>Uniform</td>
<td>1/2 1/2 0 0</td>
<td>0 0 1/2 1/2</td>
</tr>
<tr>
<td>Cubic</td>
<td>0 1 0 0 0</td>
<td>0 0 0 1/2</td>
</tr>
<tr>
<td></td>
<td>0 1/2 1/2 0</td>
<td>0 0 0 1/2 1/2</td>
</tr>
<tr>
<td></td>
<td>0 0 1 0</td>
<td>0 0 0 0 1</td>
</tr>
</tbody>
</table>

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
1/2 & 1/2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 1/2 & 1/2 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1/2 & 1/2 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
4/8 & 0 & 0 & 0 & 0 \\
6/8 & 1/8 & 0 & 0 & 0 \\
4/8 & 4/8 & 0 & 0 & 0 \\
1/8 & 6/8 & 1/8 & 0 & 0 \\
0 & 4/8 & 4/8 & 0 & 0 \\
0 & 1/8 & 6/8 & 1/8 & 0 \\
0 & 0 & 4/8 & 4/8 & 0 \\
0 & 0 & 1/8 & 6/8 & 1/8 \\
0 & 0 & 0 & 4/8 & 4/8 \\
0 & 0 & 0 & 1/8 & 6/8 \\
0 & 0 & 0 & 0 & 4/8
\end{bmatrix}
\]

For linear uniform B-spline and for uniform cubic B-splines.
Cubic B-splines via subdivision

\[
\begin{align*}
\frac{1}{8}P_0 + \frac{3}{4}P_1 + \frac{1}{8}P_2 \\
\frac{1}{2}P_1 + \frac{1}{2}P_2 \\
\frac{1}{8}P_1 + \frac{3}{4}P_2 + \frac{1}{8}P_3 \\
\frac{1}{2}P_0 + \frac{1}{2}P_1 \\
\frac{1}{2}P_2 + \frac{1}{2}P_3
\end{align*}
\]
Convergence of subdivision (aside)

\[ p_j = S_j p^0 \]

- The recursively refined set of control points converge to the actual spline curve \( p(t) = \sum p_i B_i(t) \)

- Have geometric rate of convergence, i.e., difference decrease by constant factor (see notes) — \( ||\varepsilon_j|| < c \gamma^j \)

- Can thus obtain spline curves via subdivision, just like de Casteljau for Bezier curves!
Idea of subdivision

- A subdivision curve (or surface) is the limit of a sequence of successively refined control polygon (or control mesh)
Subdivision

- An effective and efficient way to model and render smooth curves and surfaces, e.g., Bezier and B-splines, via **local refinement**

- Two aspects:
  - **Topological rule**: where to insert new vertices? Are old vertices kept?
  - **Geometrical rule**: spatial location of the new vertices – typically given as an average of nearby new or old vertices

- First introduced to graphics by Ed Catmull and Chaikin in the 1970’s

- One of the most intensely studied subjects of geometric modeling (1990’s) and ubiquitous in modeling and animation software now
Subdivision surfaces in animation

- Subdivision surfaces in Geri’s game:
  
  http://mrl.nyu.edu/~dzorin/sig99/derose/sld001.htm
Surface example: Catmull-Clark

- Works on quadrilateral meshes
- Topological rules:
  - One new point per face and edge; retain the old vertices
  - Connect face point with all adjacent edge points
  - Connect old vertex with all adjacent edge points
Catmull-Clark subdivision

- Geometric rules (subdivision masks shown below)

- This is all nice if the quadrilateral mesh connectivity is regular, i.e., a rectangular grid, but not always the case
Extraordinary vertices

- In a quadrilateral mesh, a vertex whose valence is not 4 is called an **extraordinary vertex**

- In a triangle mesh, an extraordinary vertex has valence $\neq 6$

- Geometric rules for extraordinary vertices are different

**Exercise:** For a closed triangle mesh, can all vertices have degree 6?
Catmull-Clark and B-splines

- Even if original mesh has faces other than quadrilaterals, after one subdivision, all faces become quadrilaterals.

- **Number of extraordinary vertices never increase**

- Over rectangular (regular) region, the limit is *bicubic B-spline surface*, i.e., $C^2$.

- Continuity at extraordinary vertices: $C^1$.

- There are many other types of subdivision surfaces with different schemes giving different levels of continuity.
Advantages

- **Efficient to compute/render** with simple algorithms: weighted averages within a local neighborhood

- Flexible **local control** of surface features

- **Provable smoothness** if well designed

- **One-piece and seamless**; can model surfaces with arbitrary topology (same topology as control mesh) with relative ease

- **Compact** representation: base mesh + (fixed) rules

- Natural **level-of-detail** (hierarchical) representation
Subdivision surface vs. mesh

- Subdivision surfaces are smooth limit surfaces
- But in practice, e.g., rendering, only a few subdivisions are needed to produce a mesh that is dense enough
- **Polygonal meshes**: a much more general geometric representation
  - Does not have to result from subdivision – *irregular connectivity* vs. *subdivision connectivity*
  - Typically obtained from discretization of math representation or reconstruction out of a point cloud
Derivation of B-spline basis

- ... via convolution
- Recall: B-spline bases defined by a knot sequence
- In uniform case (uniform B-splines), i.e., uniform spacing of the knots, B-spline basis can be defined via repeated convolution

\[
B_l(t) = (B_{l-1} \otimes B_0)(t) = \int B_{l-1}(s)B_0(t - s)ds
\]

- \(B_0(t)\), degree-0 B-spline, is the box function at \(t = 0\)
Convolution

- An integral that computes a “running weighted average”

\[(f \otimes g)(t) = \int f(s)g(t - s)ds\]

- Kernel/weighting function \(g\) is often **symmetric** about 0
B-splines via convolution

\[ B_0(t) \]

\[ B_1(t) \]

\[ B_2(t) \]

\[ \ldots \ldots \]
A few words on convolution (aside)

- Function $g$ first **reversed**: differ from **cross correlation**
  - To ensure **commutativity**: $f \otimes g = g \otimes f$
  - Convolution is also **associative**: $f \otimes (g \otimes h) = (f \otimes g) \otimes h$
  - And **distributive over addition**: $f \otimes (g + h) = f \otimes g + f \otimes h$

- **Discrete convolution in 1D**: **serial products**
  - $\{f_0, f_1, \ldots, f_{m-1}\} \otimes \{g_0, g_1, \ldots, g_{n-1}\} = \{f_0g_0, f_0g_1 + f_1g_0, \ldots, f_{m-1}g_{n-1}\}$
  - Length of resulting sequence: $n + m - 1$
  - Matrix formulation: multiplication by a **Toeplitz matrix**

- **Circular convolution** defined by a **circulant matrices**, i.e., $C_{ij} = C_{kl}$ if and only if $i - j \equiv k - l \pmod{n}$