CS 348C — Fall 1995-6
Assignment 2 Solutions — Apostolos Lerios

Note

World Wide Web (WWW) documents referenced in these solutions can all be found under the course hierarchy

`http://graphics.stanford.edu/courses/cs348c-95-fall/`

Problem 1: Polynomial Specification.

It is not always possible to specify $f(t)$ unambiguously by f_1, f_2, and m where

$$\begin{aligned} f(0) &= f_1 \\ f(1) &= f_2 \\ f'\left(\frac{1}{2}\right) &= m. \end{aligned}$$

Assume that the $f(t)$ has the form

$$c_1 + c_2 t + c_3 t^2.$$

It follows that f_1, f_2, and m impose the following constraints on the quadratic coefficients c_1, c_2, and c_3:

$$\begin{aligned} c_1 &= f_1 \\ c_1 + c_2 + c_3 &= f_2 \\ c_2 + c_3 &= m. \end{aligned}$$

But these are not independent constraints, as subtracting the first equation from the second yields the third one. We draw two conclusions from this observation:

- m has to be equal to $f_2 - f_1$ for a quadratic $f(t)$ to exist with the desired properties. Thus f_1, f_2, and m do not *always* define $f(t)$ since I can assign them values in a way such that no quadratic $f(t)$ exists with the desired properties.
- If indeed $m = f_2 - f_1$, then there are infinite ways in which we may choose c_1, c_2, and c_3, since we have three variables constrained by only two equations. So, f_1, f_2, and m do not *unambiguously* define $f(t)$.

Problem 2: Bernstein Basis.

One way to solve this problem is by applying the de Casteljau algorithm to the given Bézier points of $\mathbf{q}(t)$. Using the interpolation weight

$$w = \frac{t-r}{s-r},$$

we obtain

$$
\begin{aligned}
\hat{\mathbf{q}}(r,r,t) &= (1-w)\hat{\mathbf{q}}(r,r,r) + w\hat{\mathbf{q}}(r,r,s) \\
\hat{\mathbf{q}}(r,s,t) &= (1-w)\hat{\mathbf{q}}(r,r,s) + w\hat{\mathbf{q}}(r,s,s) \\
\hat{\mathbf{q}}(s,s,t) &= (1-w)\hat{\mathbf{q}}(r,s,s) + w\hat{\mathbf{q}}(s,s,s)
\end{aligned}
$$

in the first level of interpolations,

$$
\begin{aligned}
\hat{\mathbf{q}}(r,t,t) &= (1-w)\hat{\mathbf{q}}(r,r,t) + w\hat{\mathbf{q}}(r,s,t) \\
\hat{\mathbf{q}}(s,t,t) &= (1-w)\hat{\mathbf{q}}(r,s,t) + w\hat{\mathbf{q}}(s,s,t)
\end{aligned}
$$

in the second level, and

$$
\hat{\mathbf{q}}(t,t,t) = (1-w)\hat{\mathbf{q}}(r,t,t) + w\hat{\mathbf{q}}(s,t,t)
$$

in the third and final level. But these are the same interpolations we do to find $\mathbf{p}(t)$ given its Bézier points, differing only in that we use t in place of w. Hence, if we perform repeated substitutions in the above equations so as to express $\mathbf{q}(t)$ in terms of its Bézier points, the coefficients of these points will be $B_1(w), \ldots, B_4(w)$. It follows that

$$
\begin{aligned}
\tilde{B}_1(t) &= B_1(w) = B_1\left(\frac{t-r}{s-r}\right) = \left(\frac{s-t}{s-r}\right)^3 \\
\tilde{B}_2(t) &= B_2(w) = B_2\left(\frac{t-r}{s-r}\right) = 3\left(\frac{s-t}{s-r}\right)^2 \frac{t-r}{s-r} \\
\tilde{B}_3(t) &= B_3(w) = B_3\left(\frac{t-r}{s-r}\right) = 3\left(\frac{s-t}{s-r}\right)\left(\frac{t-r}{s-r}\right)^2 \\
\tilde{B}_4(t) &= B_4(w) = B_4\left(\frac{t-r}{s-r}\right) = \left(\frac{t-r}{s-r}\right)^3 .
\end{aligned}
$$

A simpler way to solve this problem is based on the realization that the shape of $\mathbf{q}(t)$ is independent of the values we choose for r and s; the de Casteljau algorithm will always produce the same shape. In other words, different values for r and s effect alternate parameterizations, but do not modify the shape. It follows that $\mathbf{q}(t)$ and $\mathbf{p}(t)$ have the same shape, since the latter is just a special case of the former (with r and s equal to 0 and 1, respectively).

As we saw in class, alternate parameterizations are related to one another by affine parameter transformations. Since

$$
\begin{aligned}
\mathbf{q}(r) &= \mathbf{p}(0) \\
\mathbf{q}(s) &= \mathbf{p}(1),
\end{aligned}
$$

it follows that

$$
\mathbf{q}(t) = \mathbf{p}\left(\frac{t-r}{s-r}\right).
$$

Hence, from

$$\mathbf{q}(t) = \mathbf{b}_1\tilde{B}_1(t) + \mathbf{b}_2\tilde{B}_2(t) + \mathbf{b}_3\tilde{B}_3(t) + \mathbf{b}_4\tilde{B}_4(t)$$

and

$$\mathbf{p}\left(\frac{t-r}{s-r}\right) = \mathbf{b}_1 B_1\left(\frac{t-r}{s-r}\right) + \mathbf{b}_2 B_2\left(\frac{t-r}{s-r}\right) + \mathbf{b}_3 B_3\left(\frac{t-r}{s-r}\right) + \mathbf{b}_4 B_4\left(\frac{t-r}{s-r}\right),$$

we conclude that

$$\begin{aligned}&\mathbf{b}_1\tilde{B}_1(t) + \mathbf{b}_2\tilde{B}_2(t) + \mathbf{b}_3\tilde{B}_3(t) + \mathbf{b}_4\tilde{B}_4(t)\\ &\quad = \mathbf{b}_1 B_1\left(\frac{t-r}{s-r}\right) + \mathbf{b}_2 B_2\left(\frac{t-r}{s-r}\right) + \mathbf{b}_3 B_3\left(\frac{t-r}{s-r}\right) + \mathbf{b}_4 B_4\left(\frac{t-r}{s-r}\right).\end{aligned}$$

Equating the coefficients of $\mathbf{b}_1, \ldots, \mathbf{b}_4$, we obtain the same result as above.

Problem 3: Polar Forms.

1. As we saw in class, the polar form of a curve $\mathbf{p}(t) = (p_x, p_y)$ is found by polarizing each coordinate individually. Considering $p_x(t) = 4t^2 + 6t + 7$ first, we use the properties of polar forms to polarize it.

   - Because polar forms are multiaffine, we can raise neither r nor s to a power. This restriction forces us to conclude that

     $$\hat{p}_x(r, s) = c_1 + c_2 r + c_3 s + c_4 rs$$

     for some constants $c_1, \ldots, c_4$.
   - Due to the symmetry of the polar forms,

     $$\hat{p}_x(r, s) = \hat{p}_x(s, r),$$

     which is equivalent to

     $$c_1 + c_2 r + c_3 s + c_4 rs = c_1 + c_2 s + c_3 r + c_4 sr.$$

     This equality reduces to

     $$(c_2 - c_3)(r - s) = 0,$$

     which must hold for all r and s; this is only possible if $c_3 = c_2$. Hence,

     $$\hat{p}_x(r, s) = c_1 + c_2(r + s) + c_4 rs.$$

   - Finally, the polar form must satisfy $\hat{p}_x(t, t) = p_x(t)$. From the above expression for $\hat{p}_x(r, s)$, we must have

     $$c_1 + 2tc_2 + c_4 t^2 = 4t^2 + 6t + 7$$

     for all t. This is only possible if $c_1 = 7$, $c_2 = 3$, and $c_4 = 4$. It follows that the polar form of $p_x(t)$ is

     $$\hat{p}_x(r, s) = 7 + 3(r + s) + 4rs.$$

When deriving the polar form of $p_y(t) = 2t + 1$, the first two steps are the same as the ones for $p_x(t)$. The last step requires that $c_1 = 1$, $c_2 = 1$, and $c_4 = 0$. Hence the polar form of $p_y(t)$ is

$$\hat{p}_y(r, s) = r + s + 1.$$

Combining the polar forms of the two coordinates yields the polar form of $\mathbf{p}(t)$:

$$\hat{\mathbf{p}}(r, s) = \begin{pmatrix} 7 + 3[r + s] + 4rs \\ r + s + 1 \end{pmatrix}.$$

2. The above detailed derivation of the polar form consists adequate proof that $\hat{\mathbf{p}}(r, s)$ is a multiaffine, symmetric function, satisfying $\hat{\mathbf{p}}(t, t) = \mathbf{p}(t)$.

3. In order to homogenize the domain, we replace the scalar argument r by the point $(r_w\,; r_x)$ on the projective line; similarly, $(s_w\,; s_x)$ replaces s. Homogenization must ensure that the Euclidean points r_x/r_w and s_x/s_w generate the same values for $\mathbf{p}(t)$ as r and s, respectively; hence, homogenization of the domain consists of mere substitution of r and s by the corresponding ratios:

$$\hat{\mathbf{p}}[(r_w\,; r_x), (s_w\,; s_x)] = \begin{pmatrix} 7 + 3\left[\frac{r_x}{r_w} + \frac{s_x}{s_w}\right] + \frac{r_x}{r_w}\frac{s_x}{s_w} \\ \frac{r_x}{r_w} + \frac{s_x}{s_w} + 1 \end{pmatrix}.$$

Homogenizing the domain alone has produced an expression which is ill-defined when either argument is a point-at-infinity. To remedy this, we homogenize the range as well. First, we add a weight coordinate to obtain

$$\hat{\mathbf{p}}[(r_w\,; r_x), (s_w\,; s_x)] = \begin{pmatrix} 1 \\ 7 + 3\left[\frac{r_x}{r_w} + \frac{s_x}{s_w}\right] + \frac{r_x}{r_w}\frac{s_x}{s_w} \\ \frac{r_x}{r_w} + \frac{s_x}{s_w} + 1 \end{pmatrix}.$$

Then, we take advantage of the fact that all triples $(\lambda\hat{p}_w\,; \lambda\hat{p}_x, \lambda\hat{p}_y)$, where $\lambda \neq 0$, represent the same projective point. Using $\lambda = r_w s_w$, we obtain the final result,

$$\hat{\mathbf{p}}[(r_w\,; r_x), (s_w\,; s_x)] = \begin{pmatrix} r_w s_w \\ 7r_w s_w + 3[r_x s_w + r_w s_x] + 4r_x s_x \\ r_x s_w + r_w s_x + r_w s_w \end{pmatrix}.$$

In the fully homogenized form, when one of the two arguments is a point-at-infinity, $\hat{\mathbf{p}}[(r_w\,; r_x), (s_w\,; s_x)]$ is well-defined; in particular, it is a point-at-infinity. However, we must still disallow the case where both arguments are points-at-infinity, as, in this case, $\lambda = 0$, and $\hat{\mathbf{p}}[(r_w\,; r_x), (s_w\,; s_x)] = (0; 0, 0)$, the prohibited triple.

4. Evaluating

$$\hat{\mathbf{p}}[(r_w;r_x),(s_w;s_x)] = \begin{pmatrix} r_w s_w \\ 7r_w s_w + 3[r_x s_w + r_w s_x] + 4r_x s_x \\ r_x s_w + r_w s_x + r_w s_w \end{pmatrix}$$

for the shown values of $(r_w;r_x)$ produces

$$\begin{aligned}
\hat{\mathbf{p}}[(1;1+h),(s_w;s_x)] &= \begin{pmatrix} s_w \\ 7s_w + 3[[1+h]s_w + s_x] + 4[1+h]s_x \\ [1+h]s_w + s_x + s_w \end{pmatrix} \\
\hat{\mathbf{p}}[(1;1),(s_w;s_x)] &= \begin{pmatrix} s_w \\ 7s_w + 3[s_w + s_x] + 4s_x \\ s_w + s_x + s_w \end{pmatrix} \\
\hat{\mathbf{p}}[(0;h),(s_w;s_x)] &= \begin{pmatrix} 0 \\ 3hs_w + 4hs_x \\ hs_w \end{pmatrix}.
\end{aligned}$$

By inspection, it follows that

$$\hat{\mathbf{p}}(1+h,\cdot) = \hat{\mathbf{p}}(1,\cdot) + h\hat{\mathbf{p}}(\delta,\cdot)$$

Important note: when we added $\hat{\mathbf{p}}(1,\cdot)$ and $h\hat{\mathbf{p}}(\delta,\cdot)$, we abolished our understanding of δ as the point-at-infinity on the projective line. The reason is that, while $h\hat{\mathbf{p}}(\delta,\cdot)$ yields the same point on the projective line no matter what $h \neq 0$ is, the sum $\hat{\mathbf{p}}(1,\cdot)+h\hat{\mathbf{p}}(\delta,\cdot)$, defined by adding pairwise the components $\hat{p}_w$, $\hat{p}_x$, and $\hat{p}_y$ of $(\hat{p}_w;\hat{p}_x,\hat{p}_y)$, produces different projective points for different values of h. The proper interpretation henceforth, for both arguments and results, is that of *sites*. This interpretation forces us to allow only 0 and 1 as permissible weight values; however, it gives us the flexibility to allow $h = 0$, interpreting $h\hat{\mathbf{p}}(\delta,\cdot) = (0;0,0)$ as the zero vector.

5. We simply apply the result of the previous part repetitively to obtain

$$\begin{aligned}
&\hat{\mathbf{p}}(1+h,1+h) - \hat{\mathbf{p}}(1,1) \\
&\quad= \hat{\mathbf{p}}(1+h,1) + h\hat{\mathbf{p}}(1+h,\delta) - \hat{\mathbf{p}}(1,1) \\
&\quad= \hat{\mathbf{p}}(1,1) + h\hat{\mathbf{p}}(\delta,1) + h\hat{\mathbf{p}}(1,\delta) + h^2\hat{\mathbf{p}}(\delta,\delta) - \hat{\mathbf{p}}(1,1) \\
&\quad= 2h\hat{\mathbf{p}}(1,\delta) + h^2\hat{\mathbf{p}}(\delta,\delta).
\end{aligned}$$

6. Following the outlined steps, we obtain in order

- $$\mathbf{p}'(1) = \lim_{h\to 0} \frac{\mathbf{p}(1+h) - \mathbf{p}(1)}{h},$$

- $$\mathbf{p}'(1) = \lim_{h\to 0} \frac{\hat{\mathbf{p}}(1+h, 1+h) - \hat{\mathbf{p}}(1,1)}{h},$$

- $$\begin{aligned} \mathbf{p}'(1) &= \lim_{h\to 0} \frac{2h\hat{\mathbf{p}}(1,\delta) + h^2\hat{\mathbf{p}}(\delta,\delta)}{h} \\ &= \lim_{h\to 0}[2\hat{\mathbf{p}}(1,\delta) + h\hat{\mathbf{p}}(\delta,\delta)], \end{aligned}$$

- $$\mathbf{p}'(1) = 2\hat{\mathbf{p}}(1,\delta),$$

- $$\mathbf{p}'(1) = 2\hat{\mathbf{p}}[(1;1),(0;1)] = 2\begin{pmatrix} 0 \\ 3+4 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 14 \\ 2 \end{pmatrix}.$$

As curve derivatives are vectors, our resulting site has zero weight.

7. Differentiation yields
$$\mathbf{p}'(t) = (8t+6, 2),$$
and evaluation at $t = 1$ produces $(14, 2)$, which is indeed the same vector as our answer to the previous part.

Problem 4: Parametric Continuity; Theory.

1. First we will prove the forward direction of the theorem, namely that if two cubic curves $\mathbf{p}(t)$ and $\mathbf{q}(t)$ join with C^2 continuity at $t = t_0$, then
$$\hat{\mathbf{p}}(t_0, r, s) = \hat{\mathbf{q}}(t_0, r, s)$$
for all real numbers r and s.

C^2 continuity implies that
$$\begin{aligned} \mathbf{p}(t_0) &= \mathbf{q}(t_0) \\ \mathbf{p}'(t_0) &= \mathbf{q}'(t_0) \\ \mathbf{p}''(t_0) &= \mathbf{q}''(t_0). \end{aligned}$$

Expressing all terms in polar form, and simplifying, we obtain
$$\begin{aligned} \hat{\mathbf{p}}(t_0, t_0, t_0) &= \hat{\mathbf{q}}(t_0, t_0, t_0) \\ \hat{\mathbf{p}}(t_0, t_0, \delta) &= \hat{\mathbf{q}}(t_0, t_0, \delta) \\ \hat{\mathbf{p}}(t_0, \delta, \delta) &= \hat{\mathbf{q}}(t_0, \delta, \delta). \end{aligned}$$

Consider now the expression

$$\hat{\mathbf{p}}(t_0, t_0 + h_r, t_0 + h_s),$$

where h_r and h_s are arbitrary real numbers. Applying repetitively the given identity

$$\hat{\mathbf{p}}(t + h, r, s) = \hat{\mathbf{p}}(t, r, s) + h\hat{\mathbf{p}}(\delta, r, s),$$

we obtain

$$\begin{aligned}
\hat{\mathbf{p}}(t_0, t_0 + h_r, t_0 + h_s) &= \hat{\mathbf{p}}(t_0, t_0 + h_r, t_0) + h_s\hat{\mathbf{p}}(t_0, t_0 + h_r, \delta) \\
&= \hat{\mathbf{p}}(t_0, t_0, t_0) + h_r\hat{\mathbf{p}}(t_0, \delta, t_0) \\
&\quad h_s\hat{\mathbf{p}}(t_0, t_0, \delta) + h_s h_r\hat{\mathbf{p}}(t_0, \delta, \delta).
\end{aligned}$$

Using the derivative equalities above, and collecting terms, we deduce that

$$\begin{aligned}
\hat{\mathbf{p}}(t_0, t_0 + h_r, t_0 + h_s) &= \hat{\mathbf{q}}(t_0, t_0, t_0) + h_r\hat{\mathbf{q}}(t_0, \delta, t_0) \\
&\quad h_s\hat{\mathbf{q}}(t_0, t_0, \delta) + h_s h_r\hat{\mathbf{q}}(t_0, \delta, \delta) \\
&= \hat{\mathbf{q}}(t_0, t_0 + h_r, t_0) + h_s\hat{\mathbf{q}}(t_0, t_0 + h_r, \delta) \\
&= \hat{\mathbf{q}}(t_0, t_0 + h_r, t_0 + h_s).
\end{aligned}$$

Writing the sums $t_0 + h_r$ and $t_0 + h_s$ as r and s respectively, we conclude that

$$\hat{\mathbf{p}}(t_0, r, s) = \hat{\mathbf{q}}(t_0, r, s);$$

this equality holds for all real numbers r and s, since h_r and h_s were arbitrary real numbers to begin with.

It is just as easy to prove the reverse direction of the theorem. If

$$\hat{\mathbf{p}}(t_0, r, s) = \hat{\mathbf{q}}(t_0, r, s)$$

for all real numbers r and s, then, using $r = s = t_0$ yields

$$\hat{\mathbf{p}}(t_0, t_0, t_0) = \hat{\mathbf{q}}(t_0, t_0, t_0),$$

which is equivalent to C^0 continuity. Similarly, using $r = t_0$ and $s = t_0+1$, we obtain

$$\hat{\mathbf{p}}(t_0, t_0, t_0 + 1) = \hat{\mathbf{q}}(t_0, t_0, t_0 + 1),$$

which is equivalent to

$$\hat{\mathbf{p}}(t_0, t_0, t_0) + \hat{\mathbf{p}}(t_0, t_0, \delta) = \hat{\mathbf{q}}(t_0, t_0, t_0) + \hat{\mathbf{q}}(t_0, t_0, \delta);$$

we used

$$\hat{\mathbf{p}}(t + h, r, s) = \hat{\mathbf{p}}(t, r, s) + h\hat{\mathbf{p}}(\delta, r, s)$$

with $h = 1$ in our last derivation. By C^0 continuity, we conclude that

$$\hat{\mathbf{p}}(t_0, t_0, \delta) = \hat{\mathbf{q}}(t_0, t_0, \delta),$$

an equation which, along with C^0 continuity, proves that $\mathbf{p}(t)$ and $\mathbf{q}(t)$ join with C^1 continuity at $t = t_0$. Finally, using $r = s = t_0 + 1$, we obtain

$$\hat{\mathbf{p}}(t_0, t_0 + 1, t_0 + 1) = \hat{\mathbf{q}}(t_0, t_0 + 1, t_0 + 1),$$

which, after fully expanding both sides, yields

$$\begin{aligned} & \hat{\mathbf{p}}(t_0, t_0, t_0) + 2\hat{\mathbf{p}}(t_0, t_0, \delta) + \hat{\mathbf{p}}(t_0, \delta, \delta) \\ = \quad & \hat{\mathbf{q}}(t_0, t_0, t_0) + 2\hat{\mathbf{q}}(t_0, t_0, \delta) + \hat{\mathbf{q}}(t_0, \delta, \delta). \end{aligned}$$

And by C^1 continuity, this equation reduces to

$$\hat{\mathbf{p}}(t_0, \delta, \delta) = \hat{\mathbf{q}}(t_0, \delta, \delta),$$

an equation which, along with C^1 continuity, proves that $\mathbf{p}(t)$ and $\mathbf{q}(t)$ join with C^2 continuity at $t = t_0$.

2. Starting from the polar values

$$\hat{\mathbf{p}}(t_0, t_0, t_0) \quad \hat{\mathbf{p}}(t_0, t_0, s_0) \quad \hat{\mathbf{p}}(t_0, r_0, s_0),$$

we can perform two levels of polar interpolations to calculate polar values for any polar arguments comprising one (or more) occurences of t_0. We will show this by expressing $\hat{\mathbf{p}}(t_0, r, s)$ in terms of the above three polar values, where r and s are arbitrary real numbers. In the first level of interpolations, we obtain

$$\begin{aligned} \hat{\mathbf{p}}(t_0, t_0, s) &= (1 - w_1)\hat{\mathbf{p}}(t_0, t_0, t_0) + w_1\hat{\mathbf{p}}(t_0, t_0, s_0) \\ \hat{\mathbf{p}}(t_0, s_0, s) &= (1 - w_2)\hat{\mathbf{p}}(t_0, t_0, s_0) + w_2\hat{\mathbf{p}}(t_0, r_0, s_0), \end{aligned}$$

where

$$w_1 = \frac{s - t_0}{s_0 - t_0} \qquad w_2 = \frac{s - t_0}{r_0 - t_0}.$$

In the second level, we obtain

$$\hat{\mathbf{p}}(t_0, r, s) = (1 - w_3)\hat{\mathbf{p}}(t_0, t_0, s) + w_3\hat{\mathbf{p}}(t_0, s_0, s),$$

where

$$w_3 = \frac{r - t_0}{s_0 - t_0}.$$

All the interpolation weights w_1, w_2, and w_3 are well-defined since neither r_0 nor s_0 is equal to t_0.

A similar derivation can express the polar value $\hat{\mathbf{q}}(t_0, r, s)$, in terms of

$$\hat{\mathbf{q}}(t_0, t_0, t_0) \quad \hat{\mathbf{q}}(t_0, t_0, s_0) \quad \hat{\mathbf{q}}(t_0, r_0, s_0).$$

Observe that the given polar values for $\mathbf{q}(t)$ have the same arguments as the given values of $\mathbf{p}(t)$ in our previous derivation. Since the interpolation weights depend on the polar arguments alone, the weights that would appear in calculating $\hat{\mathbf{q}}(t_0, r, s)$ would be identical to the ones we used in the derivation for $\hat{\mathbf{p}}(t_0, r, s)$. But since

$$\begin{aligned}\hat{\mathbf{p}}(t_0, t_0, t_0) &= \hat{\mathbf{q}}(t_0, t_0, t_0)\\ \hat{\mathbf{p}}(t_0, t_0, s_0) &= \hat{\mathbf{q}}(t_0, t_0, s_0)\\ \hat{\mathbf{p}}(t_0, r_0, s_0) &= \hat{\mathbf{q}}(t_0, r_0, s_0),\end{aligned}$$

it follows that

$$\hat{\mathbf{p}}(t_0, r, s) = \hat{\mathbf{q}}(t_0, r, s).$$

Because, in this equality, r and s are arbitrary real numbers, the continuity theorem implies that $\mathbf{p}(t)$ and $\mathbf{q}(t)$ meet with C^2 continuity at t_0.

So far, we have shown that the given equalities are *sufficient* to guarantee C^2 continuity; they are also *necessary*. That is, if $\mathbf{p}(t)$ and $\mathbf{q}(t)$ meet with C^2 continuity at t_0, then, by the continuity theorem,

$$\hat{\mathbf{p}}(t_0, r, s) = \hat{\mathbf{q}}(t_0, r, s)$$

for all real numbers r and s. And hence, by evaluating this equality for

$$\begin{array}{ll} r = t_0 & s = t_0 \\ r = t_0 & s = s_0 \\ r = r_0 & s = s_0, \end{array}$$

we conclude that

$$\begin{aligned}\hat{\mathbf{p}}(t_0, t_0, t_0) &= \hat{\mathbf{q}}(t_0, t_0, t_0)\\ \hat{\mathbf{p}}(t_0, t_0, s_0) &= \hat{\mathbf{q}}(t_0, t_0, s_0)\\ \hat{\mathbf{p}}(t_0, r_0, s_0) &= \hat{\mathbf{q}}(t_0, r_0, s_0).\end{aligned}$$

Problem 5: Parametric Continuity; Geometry.

All the plots referenced below adhere to the following conventions:

- The Bézier points of $\mathbf{p}(t)$ are drawn as black disks.
- Interpolated polar values of $\mathbf{p}(t)$ are drawn as gray disks.
- The Bézier points of $\mathbf{q}(t)$ are drawn as hollow circles with solid line style.
- Interpolated polar values of $\mathbf{q}(t)$ are drawn as hollow circles with dashed line style.
- The dotted line style is used to draw lines joining interpolated points, as well as the result of interpolation whenever it is drawn.

Note: the error tolerance in the figures is a quarter of an inch for most 300dpi laser printers.

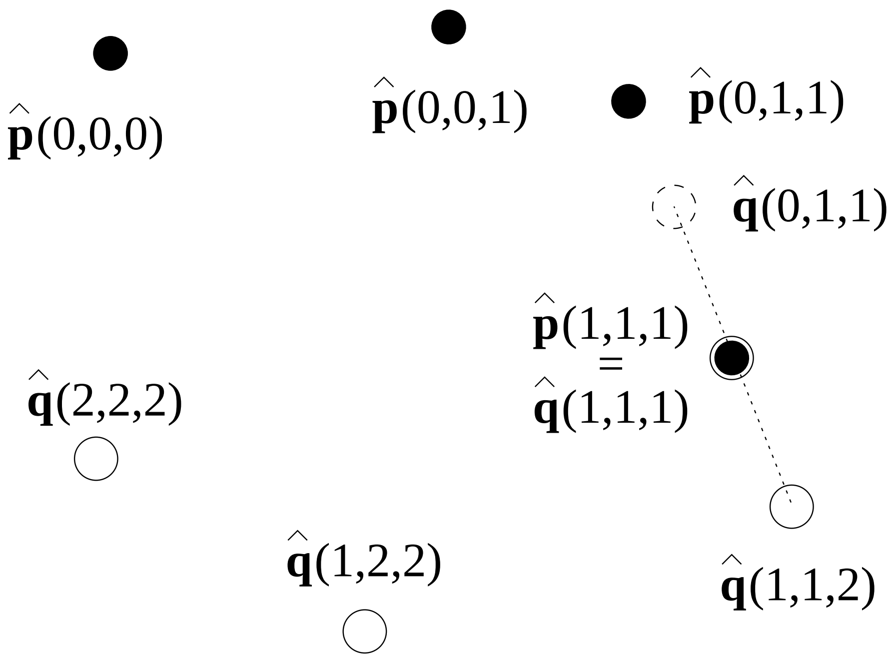

Figure 1: Solution to problem 5, part 1.

1. $\mathbf{p}(t)$ and $\mathbf{q}(t)$ meet with continuity C^0 at $t_0 = 1$; see figure 1.

2. $\mathbf{p}(t)$ and $\mathbf{q}(t)$ meet with continuity C^1 at $t_0 = 2$; see figure 2. $\hat{\mathbf{q}}(0,2,3)$ is not drawn due to space limitations, but it is obvious that it does not overlap $\hat{\mathbf{p}}(0,2,3)$.

3. $\mathbf{p}(t)$ and $\mathbf{q}(t)$ meet with continuity C^3 at $t_0 = 2$; see figure 3.

4. In the previous part, we saw that $\mathbf{p}(t)$ and $\mathbf{q}(t)$ meet with continuity C^3 at $t_0 = 2$. Hence, $\mathbf{p}(t)$ and $\mathbf{q}(t)$ are the same cubic curve; that is, $\mathbf{p}(t) = \mathbf{q}(t)$ for all t. It follows that $\mathbf{p}(t)$ and $\mathbf{q}(t)$ agree on all derivatives and thus meet with continuity C^3 at all t_0, including $t_0 = 3$.

5. $\mathbf{p}(t)$ and $\mathbf{q}(t)$ meet with continuity C^2 at $t_0 = 2$; see figure 4. $\hat{\mathbf{q}}(0,0,3)$ is not drawn due to space limitations, but it is clear that it does not overlap $\hat{\mathbf{p}}(0,0,3)$.

6. $\mathbf{p}(t)$ and $\mathbf{q}(t)$ meet with continuity C^{-1} at $t_0 = 2$ — that is, they do not meet at all; see figure 5. $\hat{\mathbf{p}}(2,2,2)$ is not drawn due to space limitations, but it is clear that it does not overlap $\hat{\mathbf{q}}(2,2,2)$.

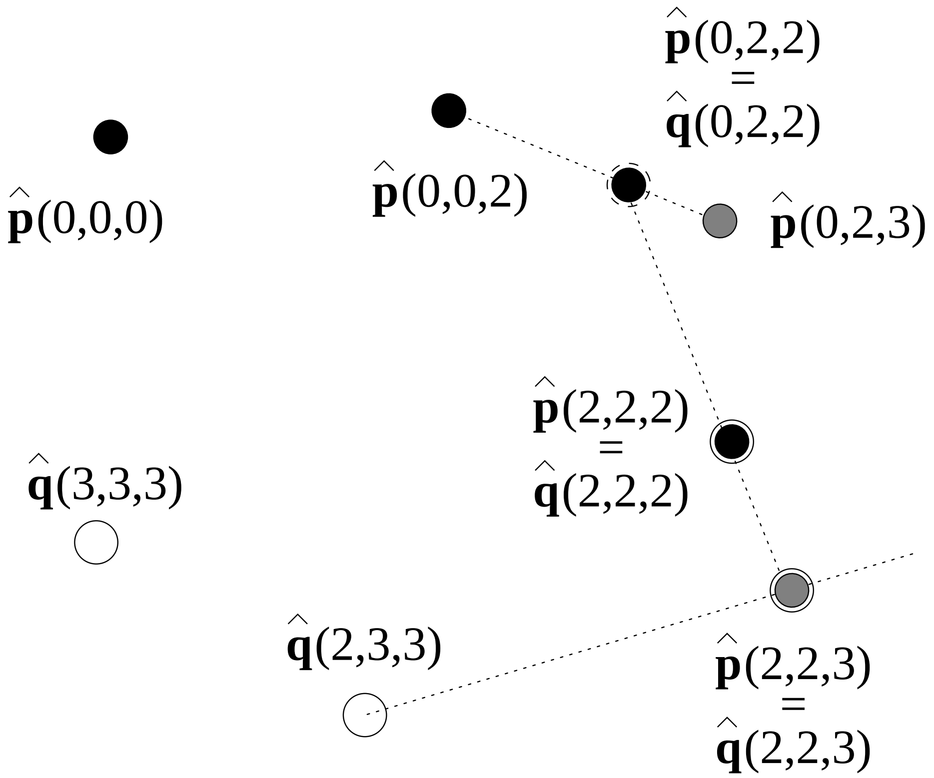

Figure 2: Solution to problem 5, part 2.

Problem 6: B-Spline Curve Drawing.

Programming part.

A sufficiently efficient solution to this part is shown below:

```
static int Degree;
static int KnotCount;
static int *Multiplicities;
static Point *deBoorP;
static double *Knots;

/* The Indexes array, of size IndexCount=2*Degree+s, contains a
   representation of the knot sequence of the spline, as a list of
   indexes into the Knots array. In the Indexes array, the indexes of
   the knots are repeated as many times as their multiplicity
   dictates. */
```

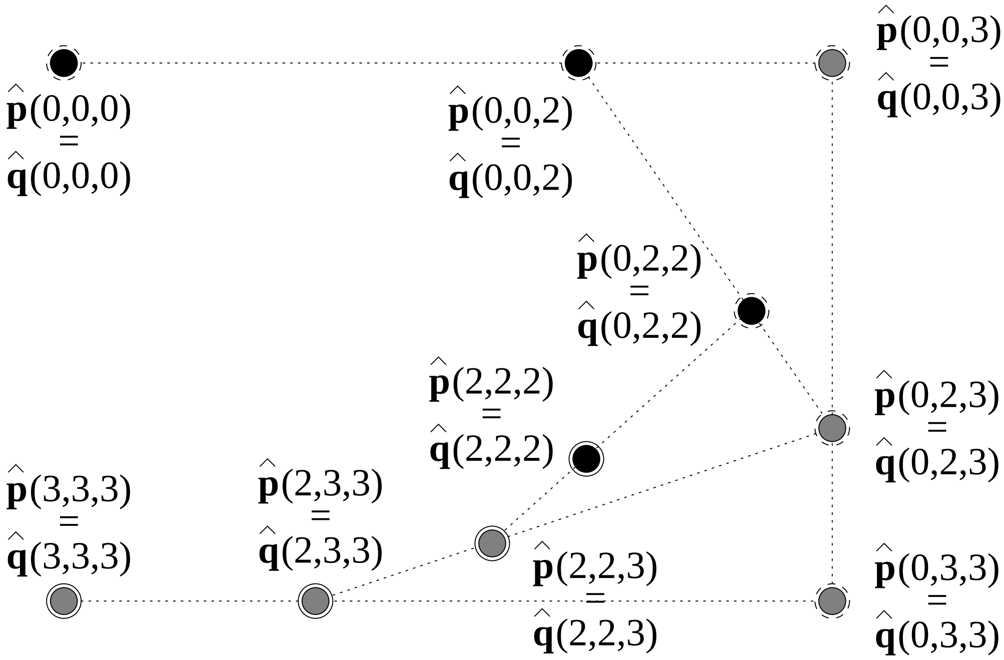

Figure 3: Solutions to problem 5, parts 3 and 4.

```
static int IndexCount;
static int *Indexes;

/* The values of t and Direction during the last invocation of
   GetSplinePoint(). */

static double Lastt;
static int LastDirection;


/* Macros. */

#define MIN(x,y) (((x)<(y))?(x):(y))
#define MAX(x,y) (((x)>(y))?(x):(y))


/* Initialization: retain basic system parameters in local variables,
   and allocate and initialize local storage. */

void InitSystem(int NewDegree,
                int NewKnotCount,
                double *NewKnots,
```

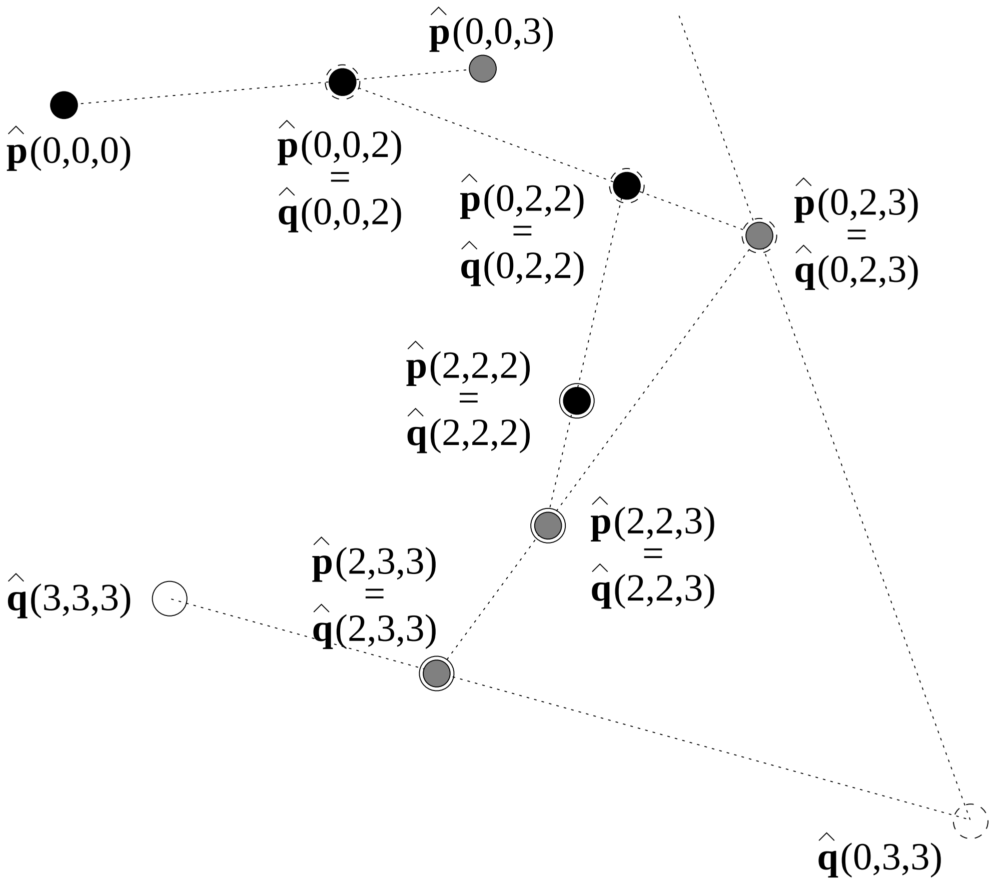

Figure 4: Solution to problem 5, part 5.

```
              int *NewMultiplicities,
              Point *NewdeBoorP) {

int i,j,k;

Degree=NewDegree;
KnotCount=NewKnotCount;
Knots=NewKnots;
Multiplicities=NewMultiplicities;
deBoorP=NewdeBoorP;

IndexCount=2*Degree;
for (i=0;i<NewKnotCount-2;i++)
  IndexCount+=Multiplicities[i];
Indexes=(int *)malloc(sizeof(double)*IndexCount);
```

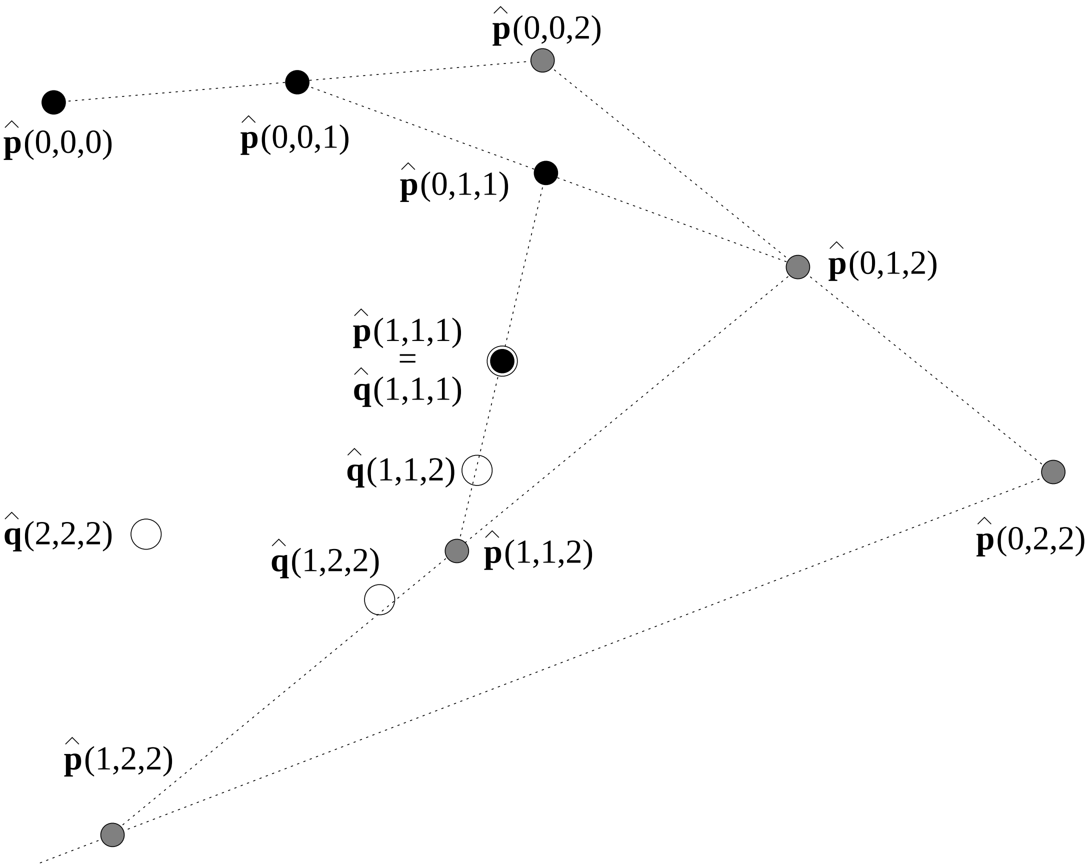

Figure 5: Solution to problem 5, part 6.

```
  for (i=0;i<Degree;i++)
    Indexes[i]=0;
  for (j=1;j<KnotCount-1;j++)
    for (k=0;k<Multiplicities[j-1];k++)
      Indexes[i++]=j;
  for (;i<IndexCount;i++)
    Indexes[i]=KnotCount-1;
}

/* Invalidate the GetSplinePoint() cache. */

void NewKnots() {

  LastDirection=0;
```

```
}

/* The influence of a point is found by the examining its label, as
   stored in the Indexes array. This is best understood by considering
   a spline of degree 2, with knot sequence (with multiplicities)
   00123344. The de Boor points of the spline correspond to the polar
   arguments 00, 01, 12, 23, 33, 34, and 44.

   The third segment of the spline, segment 2, is traversed when t
   takes values between 2 and 3. Hence its de Boor points are 12, 23,
   and 33. In general, the label of the first de Boor point of a
   segment is such that the element of the knot sequence right after
   the label corresponds to the second endpoint of the
   segment. Continuing our example, 12 is the first point of segment 2
   because right after 12, the knot sequence contains 3, the parameter
   value corresponding to the second endpoint of segment 2. Similarly,
   the label of the last de Boor point of a segment is such that the
   element of the knot sequence right before the label corresponds to
   the first endpoint of the segment. Continuing our example, 33 is
   the last point of segment 2 because right before 33, the knot
   sequence contains 2, the parameter value corresponding to the first
   endpoint of segment 2.

   In order to find the influence of a point, we reverse the above
   argument. We consider the elements of the knot sequence immediately
   before and after the point's label, and figure out which segments
   have them as endpoints. Continuing our example, point 23 influences
   segments 1 through 2. The first segment is 1 because 1 precedes 23
   in the knot sequence; this means that point 23 is a de Boor point
   of the segment starting at t=1, i.e. segment 1, while failing to
   influence segment 0: to do so, it would have to be (or precede) the
   last point of segment 0, and thus 0 (or less) would precede its
   label (23) in the knot sequence. Similarly, the last segment is 2
   because 3 appears in the knot sequence immediately after 23; this
   means that point 23 is a de Boor point of the segment ending at
   t=3, i.e. segment 2, while failing to influence segment 3: to do
   so, it would have to be (or follow) the first point of segment 3,
   and thus 4 (or higher) would follow its label (23) in the knot
   sequence. */

void GetPointInfluence(int Point,
                       int *FirstSegment,
                       int *LastSegment) {

  *FirstSegment=Indexes[MAX(0,Point-1)];
  *LastSegment=Indexes[MIN(IndexCount-1,Point+Degree)]-1;
```

```
}

/* Point retrieval. */

void GetSplinePoint(double t,
                    int Direction,
                    Point *InterpolatedP) {

  static int HigherIndex;

  int i,j,k;
  int FirstPoint;
  int NextPoint,PreviousLevel;
  double From,To,w;

  /* Parse the knot sequence (with multiplicities) until we find the
     spline segment which corresponds to the given parameter value t
     and given Direction. For example, if t=1.5 and the knot sequence
     (with multiplicities) of a degree 2 spline is 001122, then we
     choose the second segment.

     HigherIndex is set to the index of the first Indexes element
     which points to the knot value at the chosen segment's second
     endpoint. Continuing our example, HigherIndex is set to 4, since
     Indexes[4]=2 is the first Indexes element pointing to Knots[2]=2,
     which is the knot value at the second endpoint of the second
     segment.

     FirstPoint is set to the index of the spline's de Boor point
     which constitutes the first of the Degree+1 de Boor points of the
     chosen segment. Continuing our example, FirstPoint is set to 2
     since the spline's de Boor points have polar arguments 00, 01,
     11, 12, 22, and the first de Boor point of the second segment has
     polar arguments 11. */

  if ((Direction!=LastDirection) || (t<Lastt))
    HigherIndex=Degree;
  while (((Direction<0) && (t>Knots[Indexes[HigherIndex]])) ||
         ((Direction>0) && (t>=Knots[Indexes[HigherIndex]])))
    HigherIndex++;
  FirstPoint=HigherIndex-Degree;

  /* Copy the chosen segment's de Boor points to the InterpolatedP
     array. */

  for (NextPoint=0;NextPoint<=Degree;NextPoint++) {
```

```
    for (k=0;k<3;k++)
      InterpolatedP[NextPoint][k]=deBoorP[FirstPoint][k];
    FirstPoint++;
  }

  /* Perform the repeated linear interpolations leading to the
     computation of the spline point. NextPoint always contains the
     index of the first empty element of the InterpolatedP
     array. PreviousLevel always contains the index of the first of
     the two entries in InterpolatedP which are used to obtain new
     ones by interpolation. */

  PreviousLevel=0;
  for (i=Degree;i>0;i--) {
    for (j=0;j<i;j++) {

      From=Knots[Indexes[HigherIndex-i+j]];
      To=Knots[Indexes[HigherIndex+j]];
      w=(t-From)/(To-From);
      for (k=0;k<3;k++)
        InterpolatedP[NextPoint][k]=
          ((1.0-w)*InterpolatedP[PreviousLevel][k]+
           w*InterpolatedP[PreviousLevel+1][k]);

      NextPoint++;
      PreviousLevel++;
    }
    PreviousLevel++;
  }

  /* Cache most recent arguments. */

  Lastt=t;
  LastDirection=Direction;
}

/* Cleanup: deallocate local storage. */

void Cleanup(void) {

  free(Indexes);
}
```

If you want to try it out, the source code of `scurvy` is now available on-line as a tar (archive) file; see the WWW page

`software/scurvy/index.html`

Written questions.

1. The knot sequence of the spline defined in `question.sc` is

$$0, 0, 0, 1, 2, 2, 3, 3, 3.$$

   It follows that the de Boor points are assigned to the segments as shown in the table below:

Segment	000	1	22	333
0...1	000 00 0	 1 1 1	 2 22	
1...2	00 0	1 1 1	 2 22 22	 3
2...3		1	22 22 2	 3 33 333

2. Segments $1\ldots 2$ and $2\ldots 3$ share only two de Boor points. Equivalently, the third knot (whose value is 2) has multiplicity 2. It follows that the *guaranteed* continuity between these two segments is C^1. However, the *actual* continuity, for the specified de Boor points, is C^2. The reason is that the polar values of the two segments agree not only for polar arguments 122 and 223 (as the table indicates), but also for 123:

   - The polar value of the segment $1\ldots 2$ for polar arguments 123 can be found from 012 and 122 using the de Boor algorithm. Denoting by $\mathbf{p}(t)$ the equation of the curve associated with the segment $1\ldots 2$, we obtain

$$\begin{aligned} \hat{\mathbf{p}}(1,2,3) &= \frac{3}{2}\hat{\mathbf{p}}(1,2,2) - \frac{1}{2}\hat{\mathbf{p}}(0,1,2) \\ &= \frac{3}{2}(12,-6,0) - \frac{1}{2}(8,6,0) \\ &= (14,-12,0). \end{aligned}$$

   - Similarly, we can find the polar value of the segment $2\ldots 3$ for polar arguments 123 from 223 and 233. Denoting by $\mathbf{q}(t)$ the equation of the curve associated with the segment $2\ldots 3$, we obtain

$$\begin{aligned} \hat{\mathbf{q}}(1,2,3) &= 2\hat{\mathbf{q}}(2,2,3) - \hat{\mathbf{q}}(2,3,3) \\ &= 2(17,-6,0) - (20,0,0) \\ &= (14,-12,0). \end{aligned}$$

By problem 4, part 2, it follows that $1\ldots2$ and $2\ldots3$ meet with C^2 continuity at $t=2$. These two segments do not meet with C^3 continuity at $t=2$ because they may disagree when the polar arguments do not include any 2's, e.g.

$$\begin{aligned}
\hat{\mathbf{p}}(3,3,3) &= 2\hat{\mathbf{p}}(2,3,3)-\hat{\mathbf{p}}(1,3,3)\\
&= 2\left[2\hat{\mathbf{p}}(2,2,3)-\hat{\mathbf{p}}(1,2,3)\right]-\\
&\quad \left[\frac{3}{2}\hat{\mathbf{p}}(1,2,3)-\frac{1}{2}\hat{\mathbf{p}}(0,1,3)\right]\\
&= 2\left[2(17,-6,0)-(14,-12,0)\right]-\\
&\quad \left[\frac{3}{2}(14,-12,0)-\frac{1}{2}\left[\frac{3}{2}\hat{\mathbf{p}}(0,1,2)-\frac{1}{2}\hat{\mathbf{p}}(0,0,1)\right]\right]\\
&= (40,0,0)-\left[(21,-18,0)-\frac{1}{2}\left[\frac{3}{2}(8,6,0)-\frac{1}{2}(3,6,0)\right]\right]\\
&= (40,0,0)-\left[(21,-18,0)-(5.25,3,0)\right]\\
&= (24.25,21,0)\\
&\neq \hat{\mathbf{q}}(3,3,3)=(23,6,0).
\end{aligned}$$