The area between the graph of a function $x \mapsto (x, C(x))$ and the $x$-axis (hatched region in the figure below):

$$\int_{x_0}^{x_1} C(x) \, dx \quad (1)$$

If the curve is given parametrically, i.e., $t \mapsto (C_x(t), C_y(t))$, the integral (1) can be rewritten (by substituting $x = C_x(t)$, $x_0 = C_x(t_0)$, $x_1 = C_x(t_1)$, $C_x(t) = C_x(t^*) = C_y(t)$, and $dx = \frac{dC_x(t)}{dt} dt$) as

$$\int_{t_0}^{t_1} C_y(t) \frac{dC_x(t)}{dt} \, dt \quad (2)$$

If, furthermore, $t_0 \neq t_1$ and $(C_x(t_0), C_y(t_0)) = (C_x(t_1), C_y(t_1))$, i.e., the curve is cyclic, the integral (2) yields the area surrounded by the curve.

Assume that the cyclic curve is a spline composed of Bézier arcs $B_1$, $B_2$, ..., $B_n$ (each defined for $0 \leq t \leq 1$).

The area of the region surrounded by the spline

$$\sum_{i=1}^{n} \int_0^1 B^y_i(t) \frac{dB^x_i(t)}{dt} \, dt$$

is the sum of integrals:

In the sequel, I’ll skip the index $i$—calculations are exactly the same for each $i$; the functions $B(t) = (B^x(t), B^y(t))$ are third-degree polynomials:

$$B(t) = b_0(1 - t)^3 + 3b_1(1 - t)^2t + 3b_2(1 - t)t^2 + b_3t^3$$

where $b_0 = (b_{0x}, b_{0y})$, $b_1 = (b_{1x}, b_{1y})$, $b_2 = (b_{2x}, b_{2y})$, $b_3 = (b_{3x}, b_{3y})$ are points in the plane; $b_0, b_1$ are the nodes and $b_1, b_2$ are the control points of the Bézier arc $B$.

The computation of the antiderivative of the function $B^y(t) \frac{dB^x(t)}{dt}$ (a fifth-degree polynomial) is an elementary task (actually, it suffices to know that a derivative of $t^n$ is $nt^{n-1}$ and, thus, the integral of $t^n$ is $\frac{1}{n+1}t^{n+1}$). Skipping tedious calculations, I’ll present the final formula:

$$20 \int_0^1 B^y(t) \frac{dB^x(t)}{dt} \, dt = (b_{1x} - b_{0x})(10b_{0y}^3 + 6b_{1y}^3 + 3b_{2y}^3 + b_{3y}^3) + (b_{2x} - b_{1x})(4b_{0y}^3 + 6b_{1y}^3 + 6b_{2y}^3 + 4b_{3y}^3) + (b_{3x} - b_{2x})(b_{0y}^3 + 3b_{1y}^3 + 6b_{2y}^3 + 10b_{3y}^3) \quad (3)$$
The formula (3) stemmed from the discussion between Daniel H. Luecking and Laurent C. Siebenmann on MetaFont/MetaPost Discussion List (metafont@ens.fr, 2000; presently the MetaPost Discussion List is hosted by TUG—metapost@tug.org). Crucial was Luecking’s observation that three real multiplications per Bézier arc suffice to compute the area surrounded by a Bézier spline; division of the whole sum by 20 is a constant cost and thus can be neglected. Integer multiplication can be replaced by operations usually faster than real multiplication (e.g., $10a = 8a + 2a$, $8a = a$ shifted left by 3 bits, $2a = a$ shifted left by 1 bit).

Of course, such an optimization of the arithmetic operations makes sense only in a “production” implementation of the algorithm. The implementation at the level of MetaFont/MetaPost macros can be neither efficient nor precise. Nevertheless, the following code may sometimes prove useful:

```plaintext
vardef area(expr p) = % p is a Bézier segment; result = \int y dx
save xa, xb, xc, xd, ya, yb, yc, yd;
(xa,20ya)=point 0 of p;
(xb,20yb)=postcontrol 0 of p;
(xc,20yc)=precontrol 1 of p;
(xd,20yd)=point 1 of p;
(xb-xa)*(10ya + 6yb + 3yc + yd)
+(xc-xb)*( 4ya + 6yb + 6yc + 4yd)
+(xd-xc)* ( ya + 3yb + 6yc + 10yd)
enddef;

vardef Area(expr P) = % P is a cyclic path; result = area of the interior
area(subpath (0,1) of P)
for t=1 upto length(P)-1: + area(subpath (t,t+1) of P) endfor
enddef;
```

Observe that the macro `Area` computes a signed area (for the negative counterclockwise-oriented curves, and positive—for the clockwise-oriented ones). As a consequence, a non-trivial curve with selfintersection(s) (e.g., eight-shaped) may surround a region with the area equal to zero.

Observe also that the calculations can be carried out with respect to the $y$-axis, thus the following code

```plaintext
vardef area(expr p) = % p is a Bézier segment; result = \int y dx
save xa, xb, xc, xd, ya, yb, yc, yd;
(-20xa,ya)=point 0 of p;
(-20xb,yb)=postcontrol 0 of p;
(-20xc,yc)=precontrol 1 of p;
(-20xd,yd)=point 1 of p;
(yb-ya)*(10xa + 6xb + 3xc + xd)
+(yc-yb)* ( 4xa + 6xb + 6xc + 4xd)
+(yd-yc)* ( xa + 3xb + 6xc + 10xd)
enddef;

vardef Area(expr P) = % P is a cyclic path; result = area of the interior
area(subpath (0,1) of P)
for t=1 upto length(P)-1: + area(subpath (t,t+1) of P) endfor
enddef;
```

will yield the same results as the former one (within the accuracy of rounding errors).

*Gdańsk, April–May, 2011*