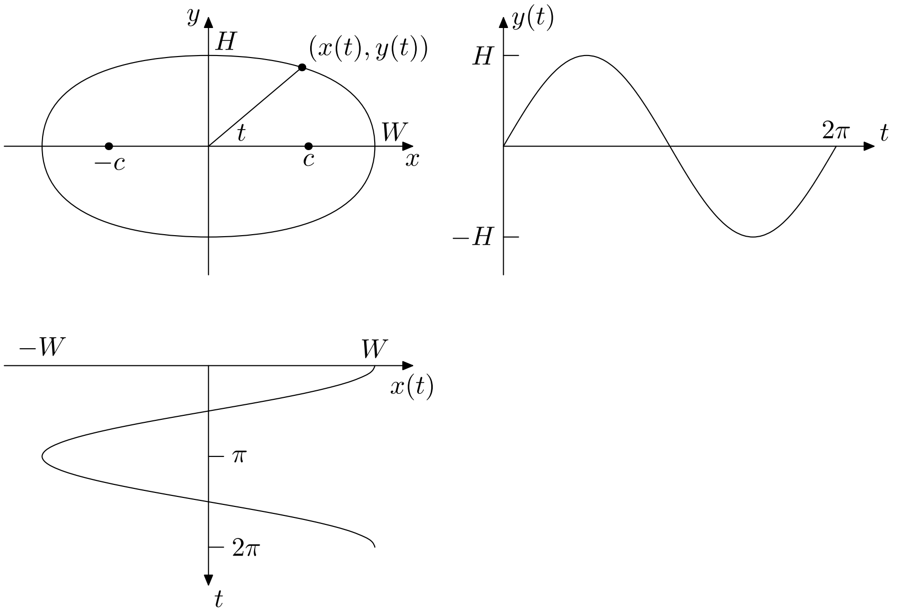
RW778 Graphics

Homework Assignment H1

Due: Wed 2006.08.02

1. Find the values of A, B, C, and D for the case of a world window $W = (l, r, t, b) = (-10, 10, -6, 6)$ and a viewport $V = (0, 600, 0, 400)$. [4]

2. An ellipse is the set of all points such that the sum of the distances to two foci is constant. The point $(c, 0)$ shown in the following Figure forms one focus, and $(-c, 0)$ forms the other. Show that H, W, and c are related by: $W^2 = H^2 + c^2$. [4]



The *eccentricity* $e = c/W$ of an ellipse is a measure of the ellipse's deviation from circularity, with an eccentricity of 0 for a true circle. As interesting examples, the planets in our solar system have very nearly circular orbits, with e ranging from $1/143$ (Venus) to $1/4$ (Pluto). Earth's orbit exhibits $e = 1/60$. As the eccentricity of an ellipse approaches unity, the ellipse flattens into a straight line. But e has to get very close to unity before this happens. What is the ratio H/W of height to width for an ellipse with $e = 0.99$? [2]

3. Hill suggests the following code for the Cohen-Sutherland clipping algorithm:

```
 1   double delx, dely;
 2   unsigned char code1, code2;
 3   RealRect w;
 4
 5   int clip(Point2 &p1, Point2 &p2) {
 6     do {
 7       if (p1.x < w.l) code1 |= 8;
 8       if (p2.x < w.l) code2 |= 8;
 9       if (p1.y < w.t) code1 |= 4;
10       if (p1.x < w.r) code1 |= 2;
11       if (p1.y < w.b) code1 |= 1;
12       if (p2.y < w.t) code2 |= 4;
13       if (p2.x < w.r) code2 |= 2;
14       if (p2.y < w.b) code2 |= 1;
15
16       if (code1|code2 == 0) return 1;  // trivial accept
17       if (code1&code2) return 0;  // trivial reject
18       delx = p2.x - p1.x, dely = p2.y - p1.y;
19       if (code1) chop(p1, code1); else chop(p2, code2);
20     } while(1);
21   }
22
23   void chop(Point2 &p, unsigned char code) {
24     if (code & 8)
25       p.y += (w.l - p.x) * dely / delx, p.x = w.l;
26     else if (code & 2)
27       p.y += (w.r - p.x) * dely / delx, p.x = w.r;
28     else if (code & 1)
29       p.x += (w.b - p.y) * delx / dely, p.y = w.b;
30     else if (code & 4)
31       p.x += (w.t - p.y) * delx / dely, p.y = w.t;
32   }
```

Consider a *vertical* line segment such that `delx` is zero. Why is the code in line 27 that would cause a divide by zero never reached? [2]

The performance of the Cohen-Sutherland algorithm would appear to be improved if we replaced lines such as "`else if (code & 2)`" with "`if (code & 2)`" and tried to do two line "chops" in succession. Explain why *as the code appears above* this approach can lead to erroneous endpoints being computed. [2]

Total marks: 14