RC5 Suggested activities (Python version):

1. Write a Python function that computes the arc length of the path P (given on p. 278) from $t = 0$ to $t = s$ for a given $0 \le s \le 1$. The choice of numerical integration method is up to you.

2. Write a program that, for any input $0 \le s \le 1$, finds the parameter $t^*(s)$ that is s of the way along the path. In other words, the arc length from $t = 0$ to $t = t^*(s)$ divided by the arc length from $t = 0$ to $t = 1$ should be equal to s. Use the Bisection Method to locate the point $t^*(s)$ to three correct decimal places. What function is being set to zero? What bracketing interval should be used to start the Bisection Method?

3. Equipartition the path of Figure 5.6 into n subpaths of equal length, for $n = 4$ and $n = 20$. Plot analogues of Figure 5.6, showing the equipartitions.

4. Replace the Bisection Method in Step 2 with Newton's Method, and repeat Steps 2 and 3. What is the derivative needed? What is a good choice for the initial guess? Is computation time decreased by this replacement?

5. Use Python animation commands to demonstrate traveling along the path, first at the original parameter $0 \le t \le 1$ speed and then at the (constant) speed given by $t^*(s)$ for $0 \le s \le 1$.

6. Experiment with equipartitioning a path of your choice. Choose a path defined by parametric equations (see Parametric Equation in Wikipedia for ideas), partition it into equal arc length segments, and animate as in Step 5.