Rectangular Approximations to Integrals

Introduction

The purpose of this lab is to acquaint you with some rectangular approximations to integrals.

Rectangular Approximations

In lecture you have learned that the area under a curve between two points a and b can be approximated as sum of areas of rectangles -The Riemann sum.

Suppose $f(x)$ is a non-negative, continuous function defined on some interval $[a, b]$. Then by the area under the curve $y = f(x)$ between $x = a$ and $x = b$ we mean the area of the region bounded above by the graph of $f(x)$, below by the x-axis, on the left by the vertical line $x = a$, and on the right by the vertical line $x = b$. All of the numerical methods in this lab depend on subdividing the interval $[a, b]$ into subintervals of uniform length.

In these simple rectangular approximation methods, the area above each subinterval is approximated by the area of a rectangle, with the **height** of the rectangle being chosen according to some rule. In particular, the left-end-point, right-end-point and midpoint rules.

The Maple `student` package has commands for visualizing these rectangular area approximations.

```
> with(student):
> rightbox(x^2,x=0..4,7);
> leftbox(x^2,x=0..4,10);
```

There are also Maple commands `leftsum` and `rightsum` to sum the areas of the rectangles, see the examples below. Note the use of `evalf` to obtain the desired numerical answers.

```
> rightsum(x^2,x=0..4,7);
> evalf(rightsum(x^2,x=0..4,7));
> evalf(leftsum(x^2,x=0..4,10));
```

Accuracy

It should be clear from the graphs that adding up the areas of the rectangles only approximates the area under the curve. However, by increasing the number of subintervals the accuracy of the approximation can be improved. One way to measure how good the approximation is by finding the difference between an upper bound of the area and the lower bound. The example in your book uses the following example:

```
>f:=x->1-x^2;
>leftbox(f(x),x=0..1,4);
>rightbox(f(x),x=0..1,4);
```

Since the left-end-point rule gives an area larger than the real area, it is called the **upper sum**. By contrast, the right-end-point rule gives an area smaller than the real area and is called the **lower sum**. Therefore the true area is in between and the error obtained by using four rectangles cannot be larger than

```
>aa:=evalf(leftsum(f(x),x=0..1,4));
>bb:=evalf(rightsum(f(x),x=0..1,4));
>aa-bb;
```

Exercises

1. A Plot $f(x) = 7x^3 + 14$ on the domain $-1 \leq x \leq 2$ using the right-end-point rule (`rightbox`) and the left-end-point rule (`leftbox`) with 5 rectangles. Which rule over estimates the real area and which under estimates the real area.

   B Plot $g(x) = 20x - \frac{x^4}{2} + 100$ on the domain $3 \leq x \leq 4$ using the right-end-point rule (`rightbox`) and the left-end-point rule (`leftbox`) with 5 rectangles. Which rule over estimates the real area and which under estimates the real area.

   C Plot $h(x) = 1 - 7x$ on the domain $-5 \leq x \leq 0$ using the right-end-point rule (`rightbox`) and the left-end-point rule (`leftbox`) with 5 rectangles. Which rule over estimates the real area and which under estimates the real area.

   D Looking at your graphs, what determines whether the rule will over or under estimate the area. Can you state your answer using the first or second derivative of a function?

2. Plot the following functions using the right and left end-point rules using 10 rectangles on the given domains.

   A $j(x) = \frac{3+x}{x^2}$ on the domain $1 \leq x \leq 5$

   B $k(x) = \sqrt{9 - x^2}$ on the domain $0 \leq x \leq 3$

   C $l(x) = e^{\frac{-(x^2)}{\pi}}$ on the domain $\frac{10\pi}{16} \leq x \leq \pi$

   D All the functions are decreasing on the given intervals. Looking at the graphs, which rule will give an area approximation slightly closer to the real area. Why? State your answer using the first or second derivative of a function.

3. For the functions `g(x)` and `j(x)` on their given domains, find the upper and lower sum of the area approximations. Then experiment with the number of rectangles to find the minimum number that keeps the error within 0.1.