Riemann Sums in MATLAB

Definition. Let $f(x)$ be a function on an interval $[a, b]$, and suppose this interval is partitioned by the values $a = x_0 < x_1 < ... < x_{n-1} < x_n = b$. Any sum of the form

$$R = \sum_{k=1}^{n} f(c_k) \triangle x_k,$$

where $\triangle x_k = x_k - x_{k-1}$ and $c_k \in [x_{k-1}, x_k]$ is referred to as a *Riemann sum* of f.

If we let P denote our choice of partition (the choice of values x_0, x_1, ..., x_n), and we let

$$\|P\| := \max(\triangle x_1, \triangle x_2, ..., \triangle x_n)$$

denote the norm of this partition, then we say f is *Riemann integrable* if

$$\lim_{\|P\| \to 0} \sum_{k=1}^{n} f(c_k) \triangle x_k$$

exists. Following Leibniz, our notation for this limit is

$$\int_a^b f(x)dx = \lim_{\|P\| \to 0} \sum_{k=1}^{n} f(c_k) \triangle x_k.$$

Example 1. As our first example, we will consider the case in which c_k is always chosen as the right endpoint of the interval $[x_{k-1}, x_k]$. If we take a regular partition with n intervals, then each interval has length $\triangle x = \frac{b-a}{n}$, and the k^{th} endpoint is

$$x_k = a + k \triangle x.$$

The Riemann sum becomes

$$R = \sum_{k=1}^{n} f(a + k \triangle x) \triangle x.$$

Suppose we would like to approximate the integral

$$\int_0^2 e^{-x^2} dx$$

with $n = 4$. We have $\triangle x = \frac{2-0}{4} = .5$ and the values

$$\begin{aligned} x_0 &= 0 \\ x_1 &= .5 \\ x_2 &= 1 \\ x_3 &= 1.5 \\ x_4 &= 2.0. \end{aligned}$$

The Riemann sum is

$$R = \sum_{k=1}^{4} f(0 + .5k).5 = .5(e^{-.5^2} + e^{-1^2} + e^{-1.5^2} + e^{-2^2}) = .6352.$$

More generally, we can write a MATLAB function M-file that carries out this calculation for any function f (defined as an inline function), endpoints a and b and regular partition with n points. See *rsum1.m*.

```
function value=rsum1(f,a,b,n)
%RSUM1: Computes a Riemann Sum for the function f on
%the interval [a,b] with a regular partition of n points.
%The points on the intervals are chosen as the right endpoints.
value = 0;
dx = (b-a)/n;
for k=1:n
c = a+k*dx;
value = value + f(c);
end
value = dx*value;
```

We run this in MATLAB with the following lines in the Command Window.

```
>>f=inline('exp(-x^2)')
f =
Inline function:
f(x) = exp(-x^2)
>>rsum1(f,0,2,4)
ans =
0.6352
>>rsum1(f,0,2,10)
ans =
0.7837
>>rsum1(f,0,2,100)
ans =
0.8723
>>rsum1(f,0,2,1000)
ans =
0.8811
```

To four decimal places, the correct value is .8821. △

One interesting aspect of the Riemann sum is that the points c_k need not be chosen in the same place on each interval. That is, suppose we partition the interval $[0, 1]$ with $0 = x_0 < x_1 = \frac{1}{2} < x_2 = 1$. In this case, a possible Riemann sum is

$$f(0)\frac{1}{2} + f(1)\frac{1}{2}.$$

Here $\triangle x_1 = \triangle x_2 = \frac{1}{2}$, and we have chosen c_1 as the left endpoint of the interval $[0, \frac{1}{2}]$ and c_2 as the right endpoint of the interval $[\frac{1}{2}, 1]$.

Example 2. As our second example, we will consider the case in which c_k is randomly selected on the interval $[x_{k-1}, x_k]$. In this case, we revise *rsum1.m* into *rsum2.m*.

```
function value=rsum2(f,a,b,n)
%RSUM2: Computes a Riemann Sum for the function f on
%the interval [a,b] with a regular partition of n points.
%The points on the intervals are chosen randomly.
value = 0;
dx = (b-a)/n;
for k=1:n
c = dx*rand + (a + (k-1)*dx);
value = value + f(c);
end
value = dx*value;
```

The only tricky line here is the one that defines c_k as a random number on the interval $[x_{k-1}, x_k]$. In order to avoid a digression on the simulation of random variables, let's simply accept that this line works.[1] The implementation is as follows (assuming $f(x) = e^{-x^2}$ has already been defined as an inline function).

```
>>rsum2(f,0,2,10)
ans =
0.8665
>>rsum2(f,0,2,100)
ans =
0.8818
>>rsum2(f,0,2,1000)
ans =
0.8821
```

Notice that your values may vary slightly from these due to the random nature of the program. Nonetheless, you should find that Riemann sums based on random values are actually converging more rapidly to the correct solution than did Riemann sums based on right endpoints.

[1] Okay, if you're really curious. Given any interval $[a, b]$, the points $c = (1 - r)a + rb$ move from a to b as r goes from 0 to 1. Our generic interval is $[a + (k - 1)\triangle x, a + k\triangle x]$ and *rand* denotes a value randomly chosen between 0 and 1. Therefore $c = (1 - \text{rand})(a + (k - 1)\triangle x) + \text{rand}(a + k\triangle x)$, which gives the formula we're using.

Assignments

1. Alter the M-file *rsum1.m* so that it computes Riemann sums of the given function by taking the values c_k as the left endpoints of each interval. Use your M-file to estimate

$$\int_0^2 e^{-x^2} dx$$

   for regular partitions with $n = 10, 100, 1000$.

2. Alter the M-file *rsum1.m* so that it computes Riemann sums of the given function by taking the values c_k as the midpoints of each interval. Use your M-file to estimate

$$\int_0^2 e^{-x^2} dx$$

   for regular partitions with $n = 10, 100, 1000$.