

Lab C: Tables, Sums and Integration

Tables and Sums

Mathematica has some nice features for making lists of numbers and for adding up the terms in the list. Suppose, for example, you wanted a list of the first 30 perfect squares. The following command will give it to you: `Table[n^2, {n, 1, 30}]`

The list `{n, 1, 30}` tells *Mathematica* that `n` is the variable to adjust when moving from term to term, and that the lowest value of `n` to use is 1; the highest, 30. By default, *Mathematica* uses a *step size* of 1, that is it tries `n = 1, 2, 3, 4, ...`

1. Get *Mathematica* to list for you the first 30 odd numbers (1, 3, ...).

See what `Table[n^2, {n, 1, 30, 3}]` does. Be sure you understand the the 3 is doing in this command.

2. Imitate this to get *Mathematica* to list the first 30 odd numbers in a different way.

In addition to making a list, *Mathematica* can add up the terms for you. Simply replace `Table` with `Sum`. For example, `Sum[n^2, {n, 1, 30}]` yields the sum of the first 30 squares (the result is 9455). Note: It is often handy to first use `Table[]` to make sure that you have set up the sum correctly, since you will be able to see the terms involved listed out. Once you think you have it correct, switch to `Sum[]` to get the answer.

3. a) Have *Mathematica* determine the sum of the first 30 odd numbers.
 b) Now combine your use of `Sum[]` and `Table[]` to make a list of sums of consecutive odd numbers. Your list should start 1, 4, 9 ..., since $1+3=4$ and $1+3+5=9$.
 c) Do you notice anything interesting about your list? Can you explain this connection?

Applying Tables and Sums to Approximate Integrals

Now suppose you want to approximate an integral using one of the methods we have discussed in class. For each of them, we need to compute $f(x_i^*)$ for a number of values of x_i^* .

Let's get started by writing a function `xistar[a_, b_, n_, i_]:=...` that produces the endpoint of subinterval i if the interval $[a, b]$ is divided into n equal pieces. If you do this correctly, you should be able to use `Table[]` to generate the list of the endpoints, for example:

```
In[20]= Table[xistar[1,3,20,i],{i,0,20}]
```

```
Out[20]= {1, --, -, --, -, -, -, --, -, --, 2, --, --, --, --, -, --, --, --, --, 3}
          11 6 13 7 3 8 17 9 19      21 11 23 12 5 13 27 14 29
          10 5 10 5 2 5 10 5 10      10 5 10 5 2 5 10 5 10
```

4. Use your function `xistar[]` together with `Table[]` to generate a list of the endpoints of the subintervals needed to apply Simpson's Rule to approximate $\int_1^4 f(x) dx$ with 15 subintervals ($n = 15$).

Now we have all the tools we need to use midpoints, trapezoids, left Riemann sums, right Riemann sums, or Simpson's rule to estimate an integral. Let's approximate

$$\ln(4) = \int_1^4 1/x \, dx .$$

For example, if we wanted to do left Riemann sums with $n = 15$, we could do the following sorts of things. Look over this *Mathematica* output and make sure you understand why it does what it does.

```
In[48]:= Clear[f]
         f[x_]:=1/x
```

```
In[49]:= Table[xistar[1,4,15,i],{i,0,14}]
```

```
Out[49] = {1, -, -, -, -, 2, --, --, --, --, 3, --, --, --, --}
           5 5 5 5      5 5 5 5      5 5 5 5
```

```
In[50]:= Table[f[xistar[1,4,15,i]],{i,0,14}]
```

```
Out[50] {1, -, -, -, -, -, --, --, --, --, -, --, --, --, --}
           6 7 8 9 2 11 12 13 14 3 16 17 18 19
```

```
In[51]:= ((4-1)/15) * Sum[f[xistar[1,4,15,i]],{i,0,14}]
```

```
Out[51]= 113634299
         -----
         77597520
```

```
In[52]:= ((4-1)/15) * Sum[f[xistar[1,4,15,i]],{i,0,14}] //N
```

```
Out[52] = 1.4644114
```

```
In[53]:= leftSum[f_,a_,b_,n_] := ((b-a)/n)* Sum[f[xistar[a,b,n,i]],{i,0,n-1}]
```

```
In[54]:= leftSum[f,1,4,15]
```

```
Out[54]= 113634299
         -----
         77597520
```

Notice how defining `leftSum[]` can save a lot of typing if you need to do this several times. Now it's your turn to give it a try.

5. Use *Mathematica* to approximate $\ln(4)$ using trapezoids, midpoints, and Simpson's rule with $n = 15$, 150, and 1500.

You will probably want to split up the sums into portions that have the same coefficient (1,2, or 4). You will also have to think a little bit about how to get the endpoints you need. Note that midpoints of n intervals are the odd numbered endpoints of $2n$ intervals.

6. How accurate are your estimates? Use the error bound formulas to give bounds on the errors of your estimates.
7. For each method, how large must n be to approximate $\ln(4)$ within 10^{-4} ? within 10^{-8} ?