

Lab 1—

Sequences and Series

Chapter 8 of Stewart is concerned with sequences and series. The ultimate goal of the chapter is to investigate the representation of functions by “infinite polynomials” (power series). These are a very powerful tool in that we can frequently deduce a power series representation for a function from a problem when we have not been given the function itself. The progression of the chapter is to first define sequences and their limits, then define series (infinite sums) in terms of sequences, and finally define power series in terms of ordinary series. In this lab we will look at the tools available in *Mathematica* for working with sequences and series.

■ Sequences

A sequence is an infinite string (or ordered list) of numbers, $\{a_1, a_2, \dots\}$. In principle, the terms of the sequence could come from anywhere with no rhyme or reason to a particular entry. In practice, however, if we are to work with a sequence, we need some method of generating the terms. Sometimes, although not always, we may have an explicit formula for a_n . For example,

$$a_n = \frac{2n^2 - 3}{n^2 + 2n}$$

In other cases, including many interesting examples, the terms of a sequence may be given *recursively*. That is, in terms of prior entries. For example, the well known Fibonacci sequence (introduced to describe the “Family Planning” habits of rabbits) can be described as:

$$\begin{aligned} a_1 &= a_2 = 1 \\ a_n &= a_{n-1} + a_{n-2} \quad (n \geq 3) \end{aligned}$$

Since in *Mathematica* a sequence is just a list, one tool available to us for analyzing a sequence is **ListPlot** for viewing a portion of the sequence graphically. If the terms of the sequence are given explicitly, $a_n = f(n)$ for some function f , as in the first example, then we can easily generate a portion of the sequence for plotting with **Table**:

```
Clear[f];
f[n_] :=  $\frac{2n^2 - 3\sqrt{n}\text{Cos}[n]}{n^2 + 2n - 1}$ ;
firstpart = Table[f[n], {n, 1, 40}];
lp = ListPlot[firstpart, PlotRange -> All, PlotStyle -> PointSize[.01], AxesOrigin -> {0, 0}];
```

ListPlot will plot a sequence $\{a_1, a_2, \dots\}$ as the list of points $\{(1, a_1), (2, a_2), \dots\}$. We can see from the graph that the terms of the sequence appear to be settling down. A limit of a sequence is a number L to which the terms of the sequence are tending. *Mathematica* has a built-in function **Limit** for symbolically evaluating the limit of functions. In this case,

```
Limit[f[n], n -> ∞]
```

[Note: ∞ can be entered from one of the palettes, as the word Infinity (remember to capitalize), or as `ESC`inf`ESC`.]

In this example the function that produces the terms is defined for all x . Consequently, it is reasonable to ask for the limit of $f(x)$ as $x \rightarrow \infty$ through all real values. This is just asking whether f has a horizontal asymptote, and we can use our previous techniques that were for working with infinite limits (in this case dividing top and bottom by n^2). Pursuing this example further graphically:

```
fp = Plot[f[x], {x, 1, 40}, DisplayFunction -> Identity, PlotStyle -> RGBColor[0, 1, 0]];
Show[lp, fp, DisplayFunction -> $DisplayFunction];
```

We can show analytically that f does have a horizontal asymptote at $y = 2$ and we can see this in the picture. If the graph as a whole is converging to a horizontal asymptote $y = L$, then certainly the same is true when considering just the integer values, and so the sequence will have a limit of L . Be careful, however, as the opposite is not true! If f does not have a limit, it may still be the case that the sequence defined by f does, simply because f may be well behaved on the integers but poorly behaved in between. This is illustrated in the following example.

```
Clear[f];
f[n_] := Sin[ ( 2 n^3 - 1 ) / ( n^2 + 1 ) Pi ];
firstpart = Table[f[n], {n, 1, 20}];
lp = ListPlot[firstpart, PlotRange -> All,
  DisplayFunction -> Identity, PlotStyle -> PointSize[.02], AxesOrigin -> {0, 0}];
fp = Plot[f[x], {x, 1, 20}, DisplayFunction -> Identity, PlotStyle -> RGBColor[0, 1, 0]];
Show[lp, fp, DisplayFunction -> $DisplayFunction];
```

It is also important to realize that traditionally we are very sloppy about distinguishing between these two types of limits. We usually rely solely on whether we are using n or x as the variable. This distinction makes no difference to *Mathematica*—a variable is a variable! For the above example the limit of the sequence should be what? (Use your “hand” techniques to determine to what the inside of sine is converging.) However, *Mathematica* tells us:

```
In[10]:= Limit[f[n], n -> ∞]
```

which is the answer appropriate for allowing n to pass through *all* reals, not just the integers.

Do this: Find the limit of the sequence $\{\sqrt{n+2} - \sqrt{n}\}_{n=1}^{\infty}$. [This is problem #21 from §8.1.]

■ Recursively defined sequences

For recursively defined sequences the only difference is getting *Mathematica* to understand how the sequence is defined. *Mathematica* has no difficulty working with recursively defined functions. For example, to compute the eighth Fibonacci number:

```

Clear[f]
f[1] = 1;
f[2] = 1;
f[n_] := f[n - 1] + f[n - 2]

f[8]

```

Note that we start by giving the initial cases and then the general rule. However, this is **not** the way we will define a recursive sequence as it is *very* slow. This is because every time it needs a new value, it must go back and recalculate all preceding values. A slight variation in the above definition instructs *Mathematica* to save any previously computed values in memory for later use. This significantly improves speed! The same Fibonacci definition in this alternate form is:

```

In[36]:= Clear[f];
         f[1] = 1;
         f[2] = 1;
         f[n_] := f[n] = f[n - 1] + f[n - 2]

```

Note only the last line has changed and now has the general form:

```
f[n_] := f[n] = <recursive formula>
```

Be absolutely sure you use the proper structure for the definition of a function of this type!

- Use _ where it is supposed to be used!
- Use := where it is supposed to be used!
- Use = where it is supposed to be used!

A recursive definition works by applying the rule over and over until it reduces it down to the known cases.

Example: Consider the sequence consisting of ratios of successive Fibonacci numbers, $b_n = a_n / a_{n-1}$. [See §8.1, #37] We already have the function f to give us the Fibonacci numbers, so we can define $b_n = g(n) = f(n) / f(n - 1)$:

```

In[31]:= Clear[g];
         g[n_] := f[n] / f[n - 1];
         ListPlot[Table[g[n], {n, 2, 10}],
                 PlotRange -> All, PlotStyle -> PointSize[.03], AxesOrigin -> {0, 0}];

```

On the (rather shaky) basis of this small sample, it certainly appears that this sequence is converging. To get an approximation of the limit we can look at a few of the higher terms:

```

g[20] // N
g[30] // N

```

Be careful not to rely too heavily on this sort of empirical analysis. Anything that involves an infinite process can have very strange things happen. It is best to use these techniques as an aide to the methods in the book, (i.e. should you be trying to show that a given sequence converges or diverges?) and for double checking an answer obtained by those methods.

Important: Notice that we have been careful to **Clear** our function before making a new definition. This is always a good habit, but is **vital** when using the recursive definition structure shown above! With f defined as above **it is impossible** to redefine f to be used as anything else if it is not cleared first!!!! This is not an overstatement. f is now useless as anything else until it has been cleared! If you forget and try, *Mathematica* will simply give you bogus output with no comment. **You have been warned!** Consider trying to redefine f as $f(x) = x^2$ without first clearing:

```
In[34]:= f[x_] := x^2
         f[4]
```

Example: Problem #49 of §8.1 defines the *arithmetic-geometric mean* of two numbers a and b as the common limit of two codependent sequences. The definitions of the sequences $\{a_n\}$ and $\{b_n\}$ are given by:

$$a_1 = \frac{a+b}{2}, \quad b_1 = \sqrt{ab}$$

$$a_n = \frac{a_{n-1} + b_{n-1}}{2}, \quad b_n = \sqrt{a_{n-1} b_{n-1}}$$

We can use *Mathematica*'s capabilities to define these sequences and then plot them to observe their behavior. For this example we will start with $a = 10$ and $b = 1$.

```
In[86]:= Clear[f, g, a, b]
         a = 10; b = 1;
         f[1] = (a + b) / 2;
         g[1] = Sqrt[a b];
         f[n_] := f[n] = (f[n - 1] + g[n - 1]) / 2
         g[n_] := g[n] = Sqrt[f[n - 1] g[n - 1]]

In[92]:= am = ListPlot[Table[f[n], {n, 1, 5}], PlotStyle -> {PointSize[.02], RGBColor[1, 0, 0]},
         PlotRange -> All, DisplayFunction -> Identity];
         gm = ListPlot[Table[g[n], {n, 1, 5}], PlotStyle -> {PointSize[.015], RGBColor[0, 0, 1]},
         PlotRange -> All, DisplayFunction -> Identity];
         Show[am, gm, DisplayFunction -> $DisplayFunction];
```

Do this: Let $\{a_n\}$ be the sequence defined by $a_1 = 1$ and $a_n = \sqrt{2 + a_{n-1}}$. Plot the first part of this sequence and hypothesize whether it converges or diverges. Estimate the limit if it appears to be convergent. [See §8.1, #44]

■ Series

A series is an infinite sum. Given a sequence $\{a_1, a_2, \dots\}$, the sum of all these terms, $a_1 + a_2 + \dots$, would be a series. This is a formal sum, as *a priori* we do not know that there is any actual number we can reasonably assign as the value of this sum. To determine a value we look at the limit of the *sequence of partial sums*, $s_n = a_1 + a_2 + \dots + a_n$. Therefore, if $a_n = f(n)$, then by definition the value of the series would be

$$\text{Limit}[\text{Sum}[f[i], \{i, 1, n\}], n \rightarrow \infty]$$

if this limit exists. However, it is not necessary to express it this way as **Sum** allows us to use an upper limit of ∞ (even though clearly *Mathematica* cannot literally add up an infinite number of terms). For example, to compute the value of the series $1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots$ we can use:

```
Sum[1/n^2, {n, 1, \infty}]
```

We can also explicitly ask *Mathematica* to compute the sum numerically:

```
NSum[1/n^2, {n, 1, \infty}]
```

Again be careful about relying blindly on such results. It is possible for *Mathematica* to be fooled. Consider the following example:

```
NSum[1/n^2 + 10^(-8)/n, {n, 1, ∞}]
```

Mathematica reports that this series converges, when it in fact includes a multiple of the harmonic series $\sum n^{-1}$, which will cause it to diverge.

Do this: Compute the sum of the series $\sum_{k=1}^{\infty} \frac{k^3}{4^k}$.