Finding limits in MATLAB [1]

If this MATLAB excercise is being counted in your grade, then please record your answers to all the questions on a separate sheet for submission.

In this exercise, you will learn how to use MATLAB to find limits of functions. All the limits we will calculate in this exercise will be of the form $\lim_{x\to c} f(x)$ (where c may be a number, a "one-sided number," or an infinity symbol). In order to work with these expressions, you want to declare x a *symbolic variable:*

```
>> syms x
```

Using a symbolic variable allows us to construct symbolic expressions for functions. Here is one that defines $f(x) = \cos x$.

```
>> f = cos(x)
```

Now we can evaluate $\lim_{x\to\pi} \cos x$ by entering:

```
>> limit(f, pi)
```

The result is as expected. Curiously enough, MATLAB's way of finding limits eliminates some of the surprises that its way of calculationg functions springs on us. Compare:

```
>> limit(f, pi/2)
>> cos(pi/2)
```

While the former gives the exact result for $\lim_{x\to\frac{\pi}{2}} \cos x$, the latter gives $\cos(\frac{\pi}{2})$ with a slight error.

Instead of defining functions symbolically, you also can enter directly their formulas into the `limit` command. For example, $\lim_{x\to 1} e^x$ can be calculated like this:

```
>> limit(exp(x), 1)
```

Curiously enough, MATLAB tells you that the answer to this one is the function value of the exponential function e^x at 1, but it does not give you a numerical answer. You can force the program to give you a numerical value by entering one of the following two commands:

```
>> double(ans)
```

or

```
>> exp(1)
```

The first of these commands tells MATLAB to convert the answer to the previous question into a numerical (*double* precision) format; the second one simply instructs MATLAB to compute e^1.

Problem 1 *How do you force* MATLAB *to show the number* e *with a precision of 15 places after the decimal point? Please record the command you enter and the actual value of* e *with this precision.* Hint: *If you have forgotten how to do this, you may want to consult the handout "Entering formulas in* MATLAB.

Now let us see what happens if we ask MATLAB to calculate some nonexisting limits. Enter:

```
>> limit(1/x^2, 0)
>> limit(1/x^3, 0)
```

While in the first case MATLAB tells you that $\lim_{x\to 0} \frac{1}{x^2} = \infty$, in the second case the answer is `NaN`, which stands for "Not a Number." As you know, in the second case we have $\lim_{x\to 0^+} \frac{1}{x^3} = \infty$ and $\lim_{x\to 0^-} \frac{1}{x^3} = -\infty$. Here is how we can get MATLAB to calculate these one-sided limits.

```
>> limit(1/x^3, x, 0, 'right')
>> limit(1/x^3, x, 0, 'left')
```

The `x` in the second argument of the `limit` command tells MATLAB that the limit is taken as the variable x approaches 0. In theory, this argument is optional and it should be possible to leave it out (we have been doing so up to this point), but for some reason my version of MATLAB bitterly complains if I do leave it out. You may want to give it a try and see how your machine reacts to leaving out the argument.

In a similar way, we can investigate what happens to the function $f(x) = \frac{|x|}{x}$ as x approaches zero.

```
>> limit(abs(x)/x, 0)
>> limit(abs(x)/x, x, 0, 'right')
>> limit(abs(x)/x, x, 0, 'left')
```

Problem 2 *Use* MATLAB *to investigate the existence of* $\lim_{x\to 1} \frac{x^2-1}{|x-1|}$, $\lim_{x\to 1^+} \frac{x^2-1}{|x-1|}$, *and* $\lim_{x\to 1^-} \frac{x^2-1}{|x-1|}$. *Indicate the commands you enter and* MATLAB*'s answers, as well as the interpretations of these answers in your own words.*

MATLAB can also be instructed to calculate limits at infinity. For example, the following commands give $\lim_{x\to\infty} \frac{4x^3-2x+6}{-x^3+x^2-1}$ and $\lim_{x\to -\infty} e^x$.

```
>> limit((4*x^3 - 2*x + 6)/(-x^3 + x^2 - 1), inf)
>> limit(exp(x), -inf)
```

Problem 3 *Try to find* $\lim_{x\to\infty} \sin x$ *with* MATLAB. *What command do you type; which answer does* MATLAB *give you, and how do you interpret this answer?*

Sometimes MATLAB gives you misleading answers. For example, the expression $\lim_{x\to 0} \sqrt{x}$ is meaningless from our point of view, since $\sqrt{x}$ is not a real number for $x < 0$; only $\lim_{x\to 0^+} \sqrt{x}$ makes sense. However, if you enter

```
>> limit(sqrt(x), 0)
```

MATLAB will give you the answer 0. What is going on here? While for negative arguments x the square root of x is not a real number, it is defined as a so-called *complex number.* Let us calculate a few square roots of negative numbers in MATLAB:

```
>> sqrt(-1)
>> sqrt(-0.1)
>> sqrt(-0.01)
>> sqrt(-0.0001)
```

MATLAB gives you answers of the form $a + bi$, where a, b are real numbers and i is the symbol for the *imaginary number* $\sqrt{-1}$. In the above examples, a is always zero. The number a is called the *real part* of the complex number $0 + bi$, and b is the *imaginary part*. If you want to know a little more about complex numbers, you may wish to read Section 1.1.6 of your textbook. However, even if you know only as much about complex numbers as I have just told you, it seems clear that as x approaches 0 from the left, the complex numbers $\sqrt{x}$ get closer and closer to zero. Thus from MATLAB's point of view, $\lim_{x\to 0^-} \sqrt{x} = 0$, and thus MATLAB tells you (mistakenly from our point of view) that $\lim_{x\to 0} \sqrt{x} = 0$.

Let us look at one more example of this kind.

```
>> limit(log(x), x, 0, 'left')
```

Now MATLAB is telling you that $\lim_{x\to 0^-} \ln x = -\infty$; which does not make sense from our point of view, since $\ln x$ is not a real number for $x < 0$. As you may have guessed, MATLAB calculates $\ln x$ as a complex number for $x < 0$, and looks at the behavior of these numbers as x approaches 0 from the left. Let us examine what pattern MATLAB "sees" by looking at these numbers:

```
>> log(-0.1)
>> log(-0.0001)
>> log(-0.00000001)
```

Problem 4 *What appears to happen to the real parts of* $\ln x$ *as* x *approaches zero from the left? What appears to happen to the imaginary parts of* $\ln x$ *as* x *approaches zero from the left? How would you interpret, in your own words, the meaning of* MATLAB*'s answer* `-inf` *to your question about* $\lim_{x\to 0^-} \ln x$*?*