Lengths of Line Segments and Curves

Computer Assignment 1

Due Thursday, September 17

This assignment develops insights that will be useful when we compute

$$\lim_{x \to 0} \frac{\sin x}{x}$$

on Monday, September 14.

Least Upper Bounds

A number b is said to be an *upper bound* for a set $\mathcal{S}$ if $b \geq s$ for every $s \in \mathcal{S}$. For example, 0 is an upper bound for the interval $(-3, 0]$. So is 1. This example demonstrates that a set can have more than one upper bound. It also demonstrates that an upper bound for a set may or may not be a member of the set.

1. *True or False. If a number b is an upper bound for a set $\mathcal{S}$, and c is a number bigger than b, then c is an upper bound for $\mathcal{S}$, too.*

2. *Give an example of a set that has no upper bound.*

A number c is the *least* member of a set $\mathcal{S}$ if

- $c \in \mathcal{S}$, and
- there is no number $x \in \mathcal{S}$ that is smaller than c.

3. *For each of the following sets, determine whether it has a least member. If so, determine what that least member is.*

(a) $(-3, 0]$

(b) $[-3, 0]$

(c) $\{-3, 0\}$

(d) $(-\infty, 0)$

Now, we're going to combine the two previous ideas. Suppose we have a set $\mathcal{S}$ of numbers, and this set has an upper bound. Let $\mathcal{B}$ be the set of all upper bounds of $\mathcal{S}$. (By assumption, $\mathcal{B}$ is not empty.) It is a fundamental property of the real numbers that this set $\mathcal{B}$ has a least member. Unsurprisingly, we call the least member of $\mathcal{B}$ the *least upper bound* of $\mathcal{S}$.

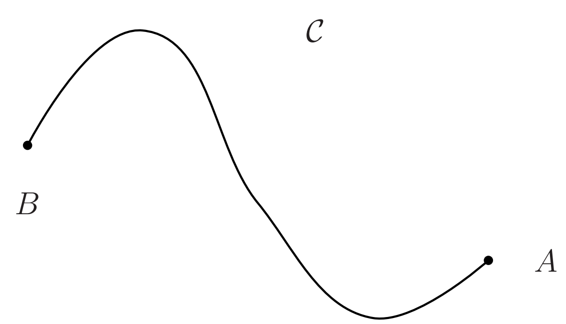
4. *For each of the following sets, determine whether or not it has an upper bound. If it does, determine its least upper bound.*

- **(a)** $(-3, 0]$
- **(b)** $(-3, 0)$
- **(c)** *The set of all integers.*
- **(d)** *The set of numbers whose squares are less than* 2.

Lengths of Curves

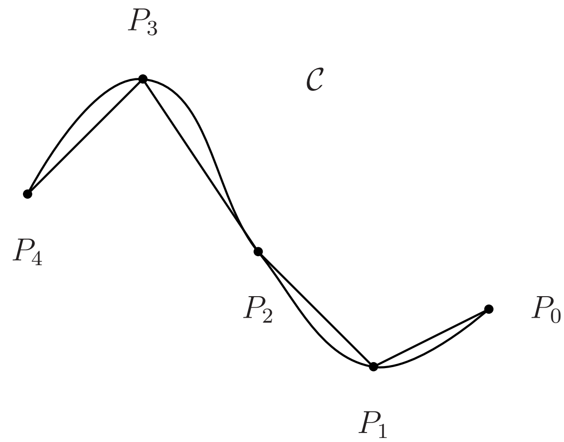
It is common to take the notion of the length of a curve for granted. How can we come up with a mathematically precise definition of this that will agree with our intuition?

Suppose we have a curve $\mathcal{C}$ that begins at a point A and ends at a point B.



If we take a finite sequence of points $P_0, P_1, \ldots, P_n$ in order on $\mathcal{C}$ (with $P_0 = A$ and $P_n = B$) and "connect the dots" by drawing a line segment from

P_0 to P_1, a line segment from P_1 to P_2, $\ldots$, and a line segment from P_{n-1} to P_n, we call each of the line segments a *chord* of $\mathcal{C}$ and we call this collection of chords a *chordal chain* for $\mathcal{C}$. The sum of the lengths of a chordal chain of $\mathcal{C}$ is called a *chord sum* of $\mathcal{C}$.



5. *Why should we expect that any reasonable definition of the length of a curve $\mathcal{C}$ will ensure that it is greater than every chord sum of $\mathcal{C}$?*

If the "dots" we connected were fairly close together, your intuition probably tells you that the chord sum is close to the length of $\mathcal{C}$. In fact, we will *define* the length of $\mathcal{C}$ to be the least upper bound of the set of all of its chord sums. (If the set of all of its chord sums has no upper bound, we either say that the length of $\mathcal{C}$ is undefined or we say that it has infinite length.)

The `Maple` worksheet on the last two pages of this handout (which is also available on this class's website) enables you to view chordal chains and calculate the corresponding chord sums for

- a semicircle of radius 1, and
- the graph of the function

$$f(x) = \begin{cases} \sqrt{|x|}\cos(1/x) & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

on the interval $[0, 1]$.

6. *Using this worksheet:*

(a) *Construct at least 5 random chordal chains (with different numbers of chords) for the semicircle and find the corresponding chord sums. Record your data in a table like:*

# of chords	chord sum
17	...
32	...
⋮	

Note 1: You will want to repeatedly execute steps (1), (2), and (3) (in the `Maple` *worksheet) in succession.*
Note 2: Not all chordal chains with the same number of chords yield the same chord sum.
Note 3: No two students should turn in the same table.

(b) *How close is your largest chord sum to* π*?*

7. *It turns out that the graph of* f *(which is called "wiggle" in the* `Maple` *worksheet) is infinite. Assuming this to be true, answer the following questions:*

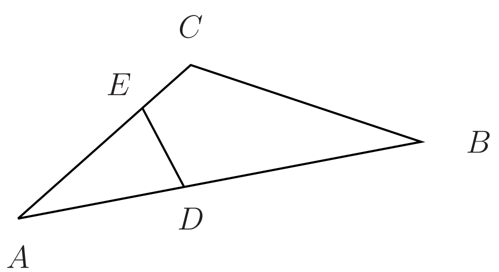
(a) *The chordal chains generated by the* `Maple` *worksheet are set up to use (in addition to the "mandatory" beginning and ending dots at* $(0, 0)$ *and* $(1, \cos 1)$*) dots at points of the form* $(1/(k\pi), f(1/(k\pi)))$ *which will be close to the peaks and valleys of the graph of* f*, thus giving a large chord sum for a given number of chords. What if, instead, we had placed dots at points of the form* $(1/((k + 1/2)\pi), f(1/((k + 1/2)\pi)))$*? Would the chord sums have been big or small?*

(b) *Can you get chord sums larger than 10? Than 100? Than 1000? Than 10000? (If you can, state how many chords there were in the chordal chains corresponding to these chord sums.)*
Note: You will want to repeatedly execute steps (4), (5), and (6) (in the `Maple` *worksheet) in succession.*

Sides of Triangles

(In the following, if P and Q are any two points in the plane, $\overline{PQ}$ will stand for the line segment from P to Q, and PQ will stand for the length of that line segment.)

Suppose we have a triangle $\triangle ABC$ in the plane (*i.e.*, a triangle whose vertices we label A, B, and C). Suppose, furthermore, that D is a point on $\overline{AB}$ and E is a point on $\overline{AC}$, and that AD is the same as AE. For example, the triangle (plus $\overline{DE}$) might look like:



In this picture, it seems obvious that $BC \geq DE$, but what if $\triangle ABC$ looked a little different, or the points D and E, while still equidistant from A and located on, respectively, $\overline{AB}$ and $\overline{AC}$, were in different places on those sides?

8. *Show that, in any case, $BC \geq DE$ by doing the following:*

- **(a)** *If $AC = AB$:*
  - **(i)** *Explain why $\triangle ABC$ and $\triangle ADE$ are similar triangles.*
  - **(ii)** *Explain why the fact that they are similar triangles implies that $BC \geq DE$.*
- **(b)** *If $AC < AB$:*
  - **(i)** *Draw a line parallel to $\overline{DE}$ and passing through C. Explain why this line intersects $\overline{AB}$.*
  - **(ii)** *Call the point of intersection F. Explain why we know $CF \geq DE$. (Hint: See (a).)*

(iii) *Note that $\angle CFB$ is obtuse. Draw a line perpendicular to $\overline{CF}$ and passing through B. Let G be the point where this line intersects the line passing through C and F. (G won't be on the line* segment *$\overline{CF}$.) Using the Pythagorean Theorem, explain why $BC > CG$ and, therefore, $BC > CF$.*

(iv) *Combine the results of (ii) and (iii) to get $BC \geq DE$.*

(c) *Explain why, by symmetry, the case $AB < AC$ is just like the case $AC < AB$.*

Maple Worksheet

```
(The actual commands are in this typeface.  Other text is
"commentary" and doesn't need to be typed in.)
```

The procedure "randseq" takes a positive integer n as input and creates a random sequence of $n+1$ points on the upper unit semicircle.

```
randseq := proc(n:posint)
map(x -> [cos(evalf(Pi)*x),sin(evalf(Pi)*x)],
sort([0,1,stats[random,uniform](n-1)]));
end:
```

The procedure "chordsum" takes a sequence of points as input and calculates the corresponding chord sum.

```
chordsum := proc(rs) local i;
add(sqrt((rs[i+1][1]-rs[i][1])^2+(rs[i+1][2]-rs[i][2])^2),
i=1..nops(rs)-1);
end:
```

The procedure "plotchords" takes a sequence of points as input and plots the chords obtained by connecting adjacent points.

```
plotchords := proc(rs) local i,ls,ps;
ls := seq(plottools[line](rs[i],rs[i+1],color=red, linestyle=1),
i=1..nops(rs)-1);
ps := seq(plottools[point](rs[i],color=green),i=1..nops(rs));
plots[display]([ls,ps],scaling='CONSTRAINED');
end:
```

(1) Replace the "a" in the following command by some positive integer. Then x will be a random sequence of points on the upper unit semicircle.

```
x := randseq(a):
```

(2) Executing the following command plots the chords corresponding to the sequence of points you generated in step (1).

```
plotchords(x);
```

(3) Executing the following command calculates the chord sum for the sequence of points you generated in step (2).

```
chordsum(x);
```

"wiggle" is the function that maps 0 to 0 and maps x to $\cos(1/x)\sqrt{|x|}$ if x is not 0.

```
wiggle := proc(x)
if x=0 then
0;
else
cos(1/x)*sqrt(abs(x));
fi:
end:
```

Executing the following command will then plot the graph of wiggle(x) on the interval $[0, 1]$. I claim that this curve has infinite length.

```
plot(wiggle(y),y=0..1,numpoints=200);
```

(4) Replace the "c" in the following command by a positive integer. Then z will be a sequence of $c + 2$ points on the curve that should correspond to a fairly large chord sum.

```
z := [[1.,evalf(cos(1))],seq([1/(evalf(Pi)*k),
wiggle(1/(evalf(Pi)*k))],k=1..c),[0.,0.]]:
```

(5) Execute the following command to find the corresponding chord sum.

```
chordsum(z);
```

(6) Execute the following command to see the chords.

```
plotchords(z);
```