Tutorial 0: Introduction

Contents

A Preview of Calculus 2

The Area Problem 2

The Tangent Problem 4

Why do we learn calculus as computer scientists? 5

Octave 6

Basic commands 7

Plotting 8

A valentine gift 10

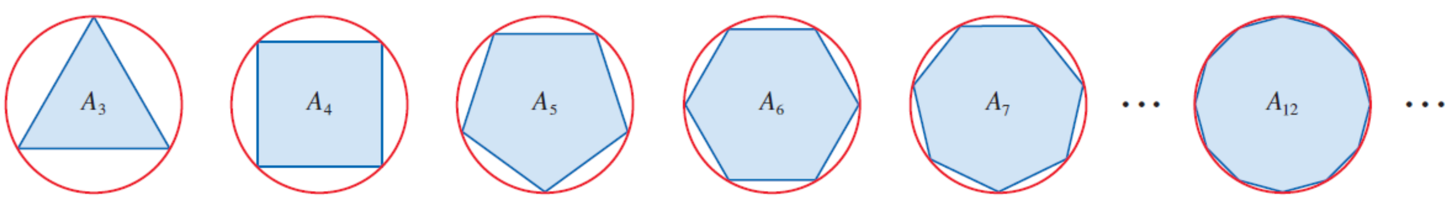
Exercises 10

A Preview of Calculus

- CALCULUS IS FUNDAMENTALLY DIFFERENT from the mathematics that you have studied previously:
    - calculus is less static and more dynamic
    - It is concerned with change and motion
    - it deals with quantities that approach other quantities
- We would like to be able to analyze quantities or processes that are undergoing continuous change.
- For example, if a stone falls 10 feet each second we could easily tell how fast it is falling at any time, the stone falls faster and faster, its speed changing at each instant.
- In studying calculus, we will learn how to model (or describe) such instantaneously changing processes and how to find the cumulative effect of these changes.
- Calculus revolves around two key problems involving the graphs of functions
    - The area problem
    - The tangent problem
    - And an unexpected relationship between them.

The Area Problem

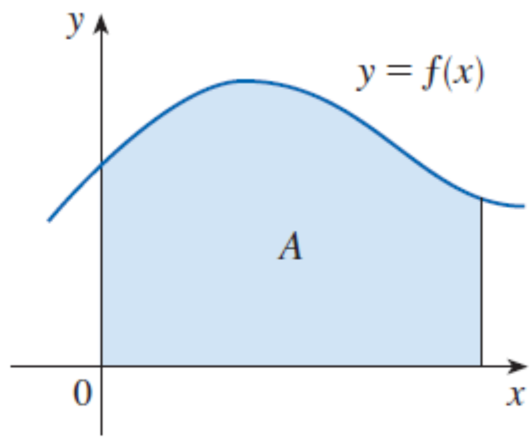
- Finding the area of a circle with inscribed regular polygons.



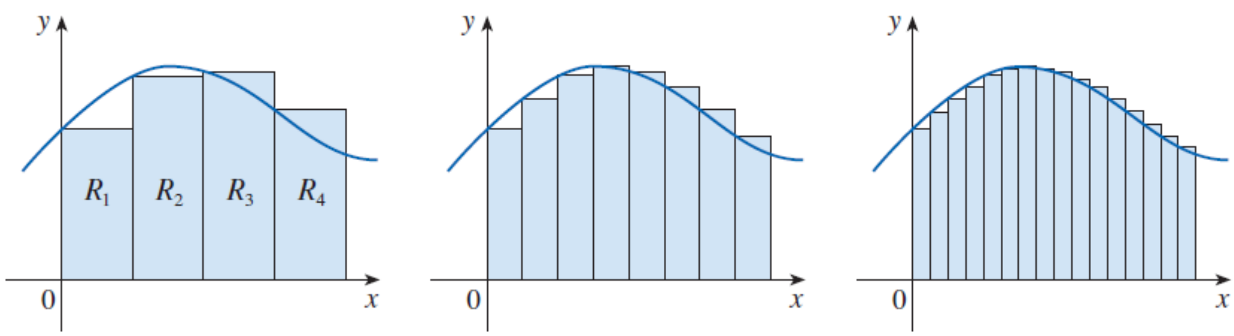
- Let A_n be the area of the inscribed regular polygon of n sides.
- As n increases, A_n gets closer and closer to the area of the circle.
- We say that the area A of the circle is the limit of the areas of the inscribed polygons, and we write

$$A = \lim_{n \to \infty} A_n$$

- Later, by indirect reasoning, it was proofed that the area of the circle: $A = \pi\, r^2$.
- Finding the area under the curve



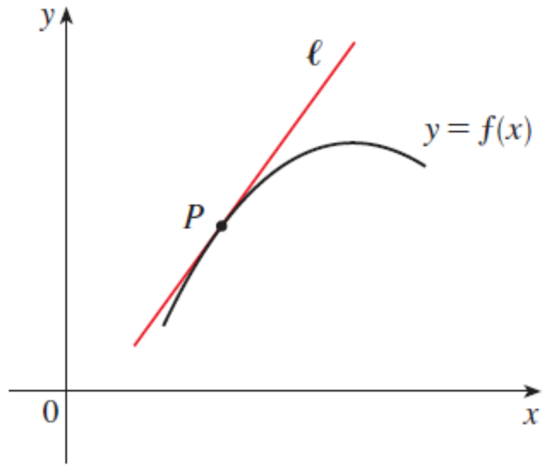
  - We approximate such an area by areas of rectangles.



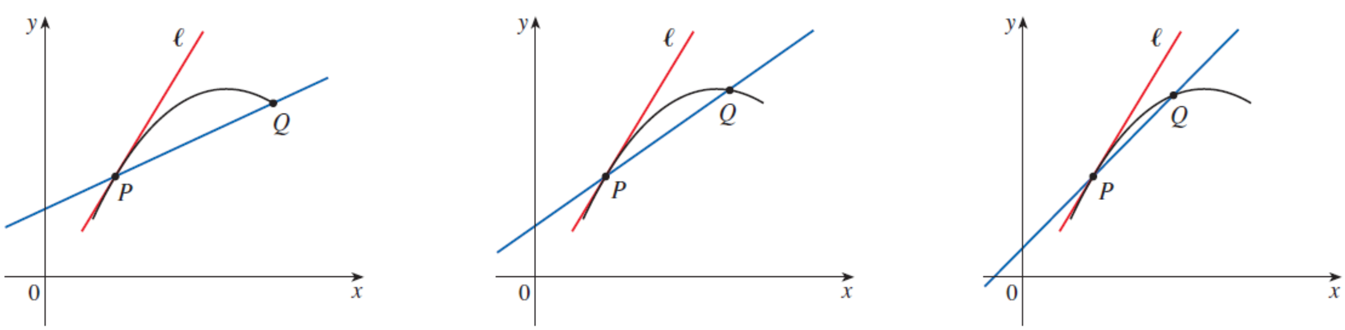
  - If we approximate the area A of the region under the graph of f by using n rectangles $R_1, R_2, \dots, R_n$, then the approximate area is
$$A_n = R_1 + R_2 + \cdots + R_n$$
  - Now imagine that we increase the number of rectangles (as the width of each one decreases) and calculate A as the limit of these sums of areas of rectangles: $A = \lim_{n \to \infty} A_n$
- The area problem is the central problem in the branch of calculus called **integral calculus**.

The Tangent Problem

- How to find an equation of the tangent line L to a curve with equation $y = f(x)$ at a given point P.

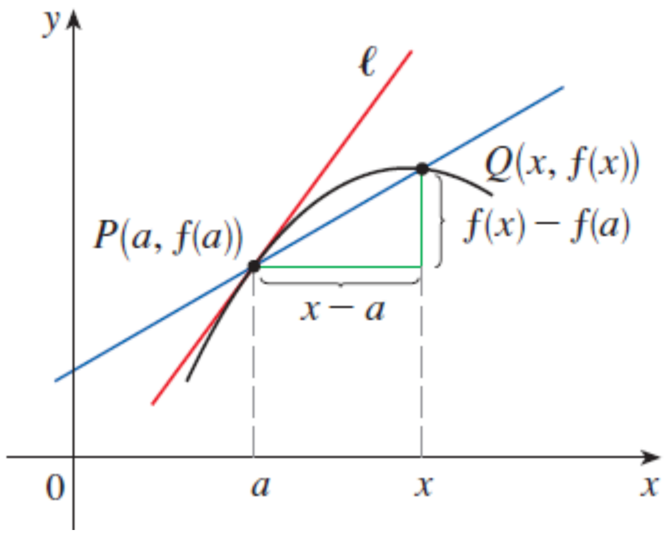


    - To find the function of the line L, we need its slope m.
    - But, to find the slope m, we need two points.
- To get around the problem we need an approximation to m by taking a nearby point Q on the curve and computing the slope m_{PQ} of the secant line PQ.
- As Q approaches P, the secant line PQ rotates and approaches the tangent line L as its limiting position.



    - This means that the slope m_{PQ} of the secant line becomes closer and closer to the slope m of the tangent line. We write
    $$m = \lim_{Q \to P} m_{PQ}$$
    - We say that m is the limit of m_{PQ} as Q approaches P along the curve.

- If P is the point $(a, f(a))$ and Q is the point $(x, f(x))$, then



$$m_{PQ} = \frac{f(x) - f(a)}{x - a}$$

Because x approaches a as Q approaches P, an equivalent expression for the slope of the tangent line is

$$\therefore m = \lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

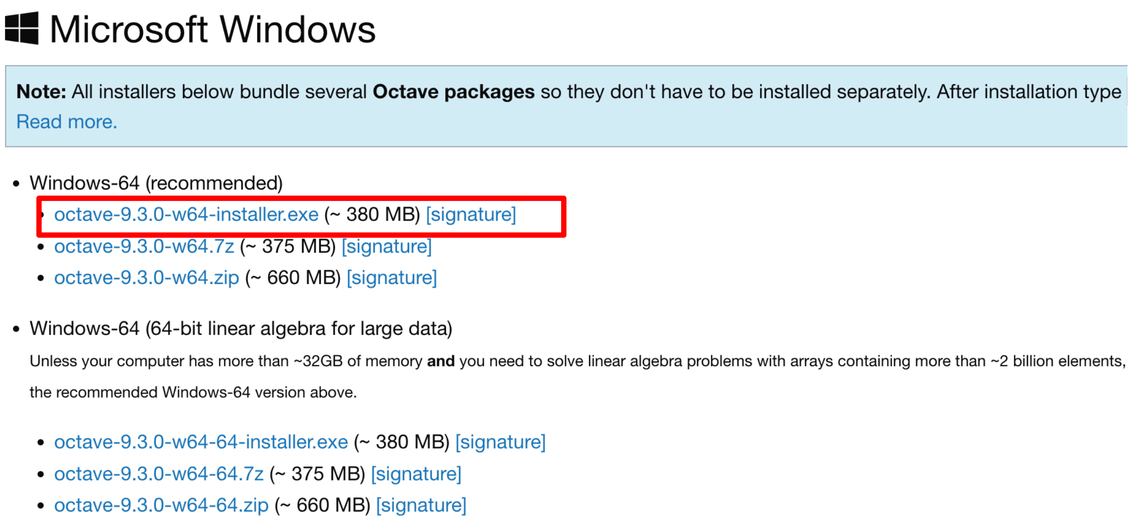
- The tangent problem has given rise to the branch of calculus called **differential calculus.**

Why do we learn calculus as computer scientists?

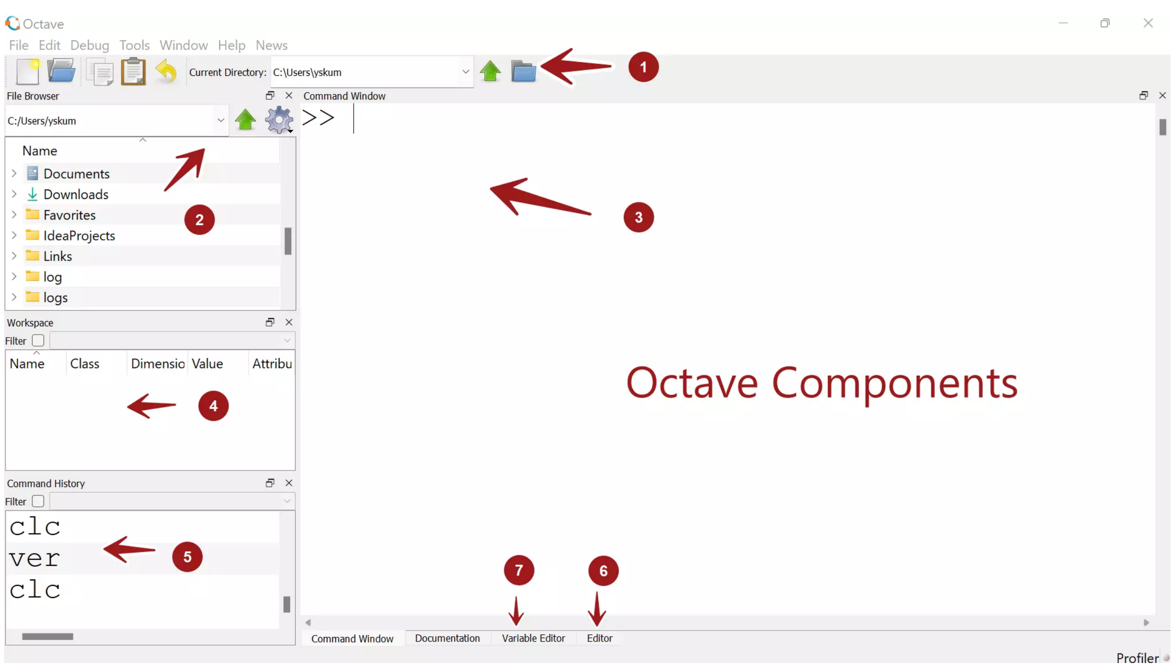
1. **Scientific computing**: writing software programs and libraries for solving problems/equations involving integrals and differentiations.
   a. Examples: Matlab, Scipy
2. **Computer graphics and simulations**
3. **Optimization**:
   a. Gradient Descent algorithm
4. **Automation**:
   like robotics, automation can require quantifying a lot of human behavior.

Octave

- Octave is a high-level language, primarily intended for numerical computations. It provides a convenient command line interface for solving linear and nonlinear problems numerically.
- To download Octave, go to https://octave.org/download, scroll down and select octave-9.3.0-w64-installer.exe



- Once you install and run it, you will see a window like this



1. Shows the current working directory.
2. The file browser displays the directories and files in the current directory.
3. The Command Window is the window where Octave commands are executed and the execution results are displayed to the user.
4. The Workspace window displays all the variables that are currently being used by the Octave. Details like the variable name, dimension, value, etc are displayed.
5. The Command History window displays the previous Octave commands entered in the Command Window.
6. The Editor window is where we can enter the code, save script files, execute, debug and run the Octave scripts.
7. A window that shows the variables and their values to manipulate them.

Basic commands

- 2+3

```
>> 2+3

ans = 5
```

- $\frac{\log_{10} 100}{\log_{10} 10}$

```
>> log10(100)/log10(10)

ans = 2
```

- $\left\lfloor \frac{1+\tan(1.2)}{1.2} \right\rfloor$

```
>> floor((1+tan(1.2))/1.2)

ans = 2
```

- $\sqrt{3^2 + 4^2}$

```
>> sqrt(3^2 + 4^2)

ans = 5
```

- $e^{i\,\pi}$

```
>> e^(i*pi)

ans = -1.0000e+00 + 1.2246e-16i
```

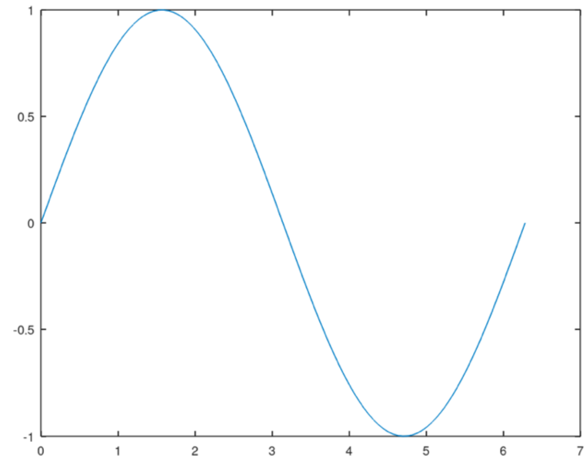
Plotting

- Plot $y = \sin(x)$ vs $x = [0: 2 * \pi]$ with the following commands.

```
>> x = linspace(0, 2*pi, 100);

>> y = sin(x);

>> plot(x,y);
```



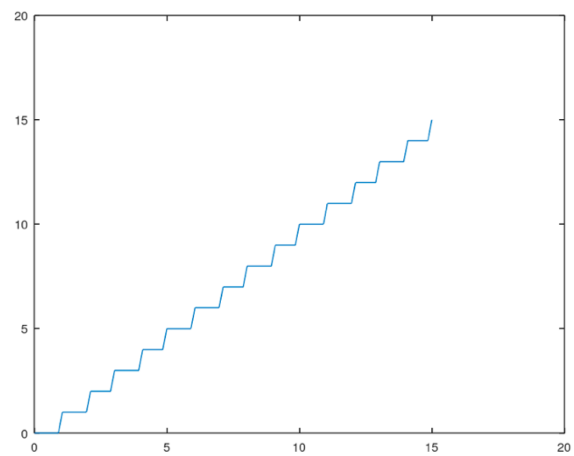
  - Using ";" after each statement prevents printing the output.
  - We can write the same program but in a script file in the editor.

- Plot the $y = \lfloor x \rfloor$ for $x = [0: 15]$

```
>> x = linspace(0, 15, 100);

>> y = floor(x);

>> plot(x,y);
```



- To plot more than one function in one graph

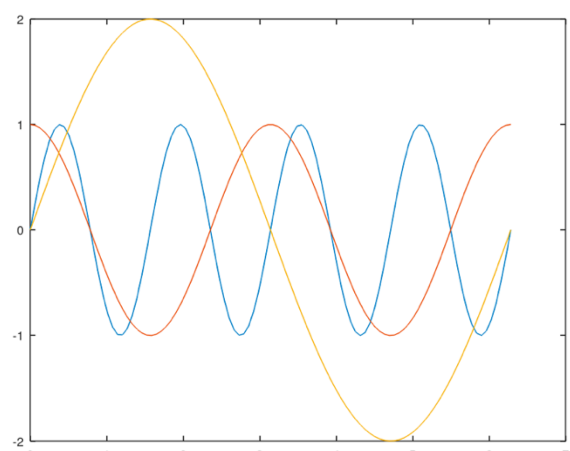
```
x = linspace(0, 2*pi);

a = sin(4 * x);

b = cos(2 * x);

c = 2 * sin(x);

plot(x, a, x, b, x, c);
```



- We can define arrays in Octave as follows

```
>> arr = [1,2,3,4,5,10];

>> sum(arr)

ans = 25

>> mean(arr)

ans = 4.1667

>> max(arr)

ans = 10

>> min(arr)

ans = 1
```

- Matrix operations

```
>> a = [1, 2; 3, 4];

>> b = [5, 6; 7, 8];

>> a + b

ans =

    6    8

   10   12

>> a * b

ans =

   19   22

   43   50

>> a .* b

ans =

   5   12

   21   32
```

A valentine gift

```
n = 800;
A = 1.995653;
B = 1.27689;
C = 8;
r=linspace(0,1,n);
theta=linspace(-2,20*pi,n);
[R,THETA]=ndgrid(r,theta);

% define the number of petals we want per cycle. Roses have 3 and a bit.
petalNum=3.6;

x = 1 - (1/2)*((5/4)*(1 - mod(petalNum*THETA, 2*pi)/pi).^2 - 1/4).^2;
phi = (pi/2)*exp(-THETA/(C*pi));
y = A*(R.^2).*(B*R - 1).^2.*sin(phi);

R2 = x.*(R.*sin(phi) + y.*cos(phi));

X=R2.*sin(THETA);

Y=R2.*cos(THETA);

Z=x.*(R.*cos(phi)-y.*sin(phi));

% % define a red map for our rose colouring
red_map=linspace(1,0.25,10)';
red_map(:,2)=0;
red_map(:,3)=0;
clf
surf(X,Y,Z,'LineStyle','none')
view([-40.50 42.00])
colormap(red_map)
```

Exercises

1. Install Python3, Octave https://octave.org/download, and install Symbolic package https://octave.sourceforge.io/symbolic/
2. Compute the addition, multiplication, subtraction, and of $a = 100$ and $b = 50$.
3. Plot the function $y = x^2$ for $x = [-10:10]$
4. Given a circle with radius 8.42. Compute its area and circumference.
5. Practice octave: https://www.youtube.com/playlist?list=PLCq1Hx5o5Rk0UH6Bnxx7OednyhVPOEgyj