수치해석
(2019학년도 1학기)

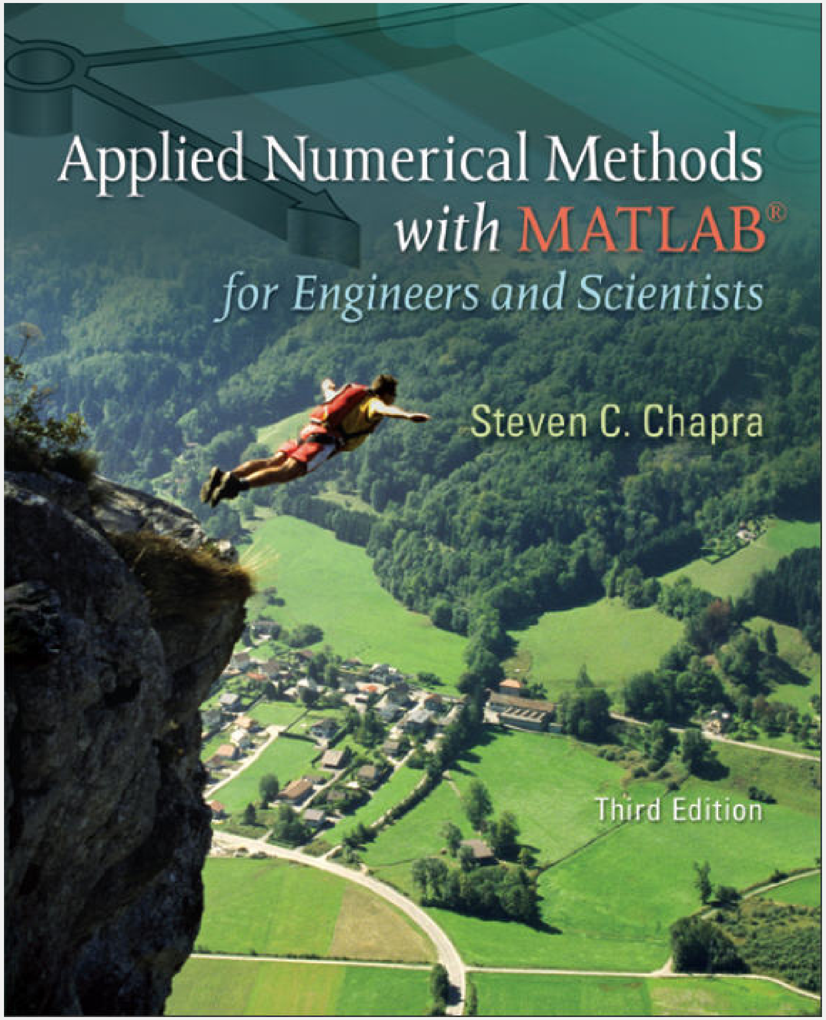
[1주/2차시 학습내용]: Basic Concept of Numerical Analysis, 수치 해석 기본 개념, 가속도 (Part One Modeling, Computers, and Error Analysis)

과학적인 물리 현상과 수치해석을 연계하여, 수치해석의 기본을 알아본다

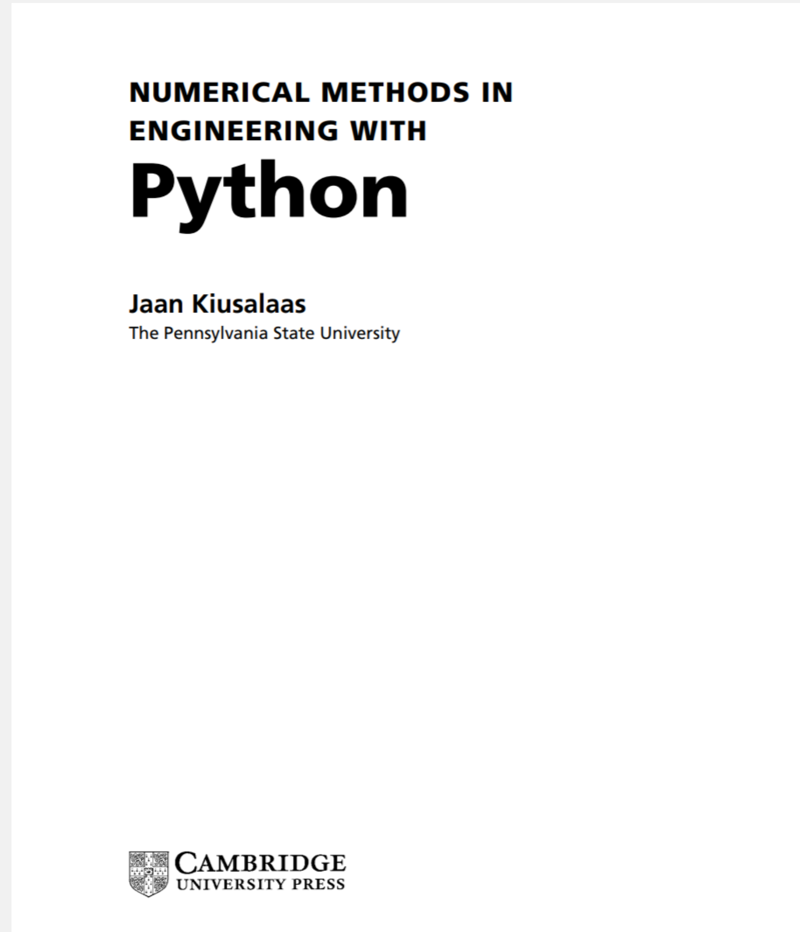
국민대학교 소프트웨어학부

교재

- Applied Numerical Methods with MATLAB for Engineers and Scientists
  - 3rd Ed.
  - Steven C. Chapra, Mc. Graw Hill, 2012



- Numerical Methods in Engineering with Python 3rd Edition, Kiusalaas, Jaan, Cambridge


NUMERICAL METHODS IN ENGINEERING WITH

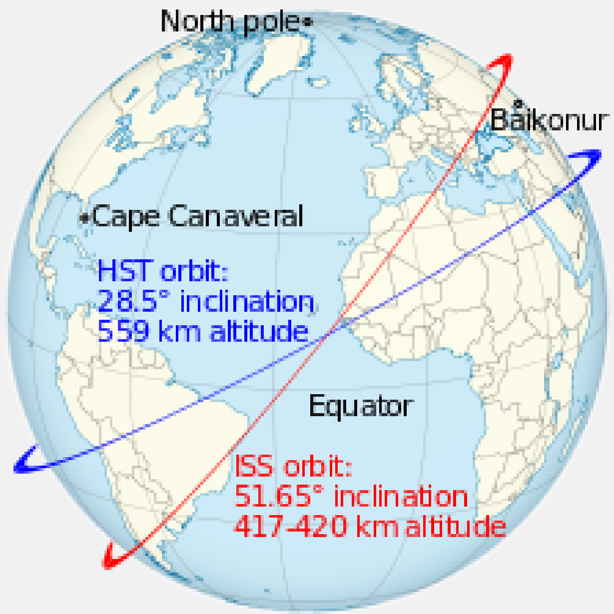
Python

Jaan Kiusalaas

The Pennsylvania State University

CAMBRIDGE UNIVERSITY PRESS

Movie (Interstellar, Gravity)

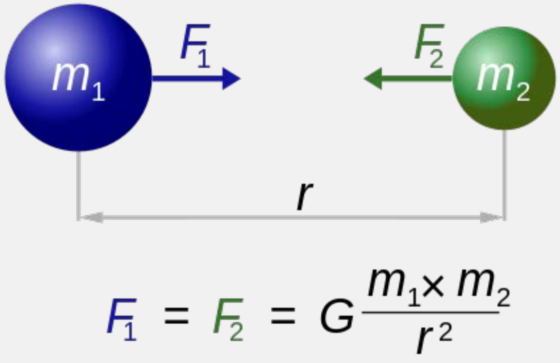


http://www.youtube.com/watch?v=OiTiKOy59o4

http://www.imdb.com/video/imdb/vi4172195865/
http://www.imdb.com/video/imdb/vi2340006169/

중력(重力, Gravity)

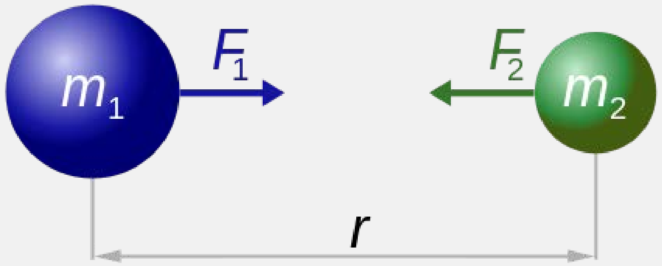
- **질량을 가진 두 물체 사이에 작용하는 힘**
- 아이작 뉴턴의 중력 이론
  - 질량을 가지는 두 물체 간의 거리가 r일 때, 두 물체 사이에 작용하는 중력의 세기
  - $F = G\frac{m_1 m_2}{r^2}$
    - F : 두 점질량 간의 **중력의 크기**
    - G : 중력 상수,
    - m1 : 첫 번째 점질량의 질량
    - m2 : 두 번째 점질량의 질량
    - r : 두 점질량간의 거리



https://ko.wikipedia.org/wiki/%EB%A7%8C%EC%9C%A0%EC%9D%B8%EB%A0%A5%EC%9D%98_%EB%B2%95%EC%B9%99

만유인력의 법칙(萬有引力-法則, law of universal gravity)

- **질량을 가진 물체 사이의 중력 끌림**
  - 점질량 m1 은 점질량 m2 를 두 질량의 곱과 두 질량 사이의 거리의 제곱에 반비례하는 힘 F2로 끌어 당긴다.
  - 두 힘 |F1|과 |F2|의 크기는 질량과 거리에 관계없이 항상 같다.
  - G는 중력상수이다.



$$F_1 = F_2 = G\frac{m_1 \times m_2}{r^2}$$

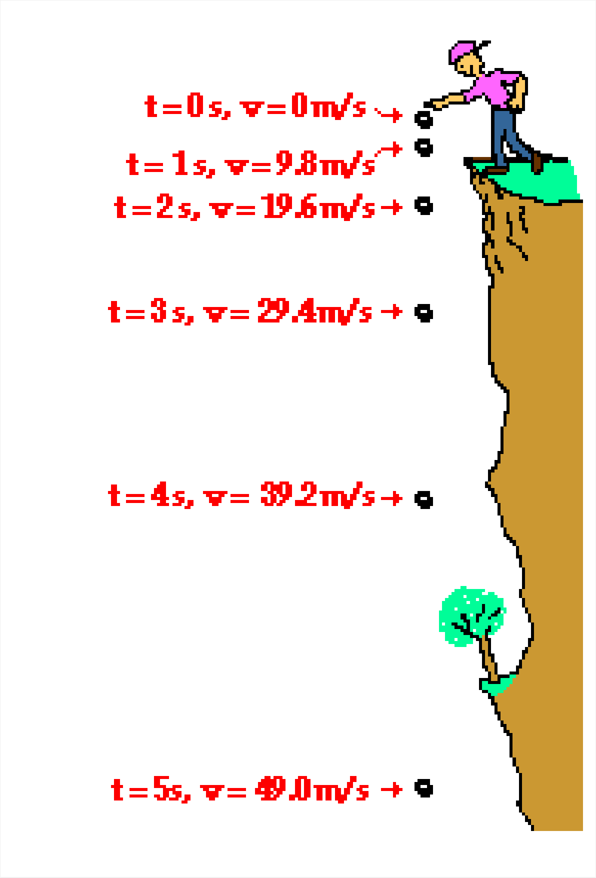
https://ko.wikipedia.org/wiki/%EB%A7%8C%EC%9C%A0%EC%9D%B8%EB%A0%A5%EC%9D%98_%EB%B2%95%EC%B9%99

중력(Gravity)과 중력 가속도(Acceleration due to gravity)

- 지구에서 중력은 물리적인 물체에 무게를 준다
  - 몸무게를 느끼는 것은 지구의 중력때문이다
  - 뉴턴이 사과가 떨어지는 것을 보고 지구와 사과는 서로 당긴다는 것을 알게 됨
- 달의 중력은 바다의 조수를 일으킨다.
- On Earth, gravity gives weight to physical objects, and the Moon's gravity causes the ocean tides.

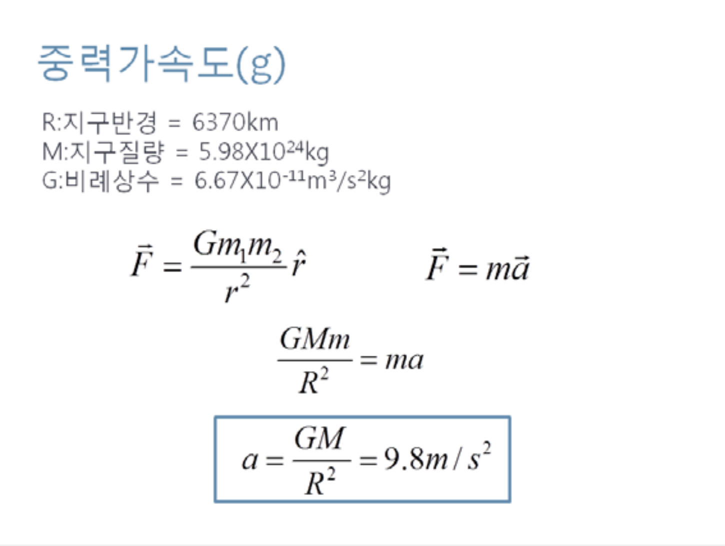
중력 가속도: Acceleration due to gravity

- **지구의 중력에 의해 운동하는 물체가 지니는 가속도**
  - Slow Velocity → Rapid Velocity
  - When the object is moving down toward Earth, **Earth's gravity affects (changes, draws) to the velocity of the moving object**
  - With velocity change rate of g=9.8m/s/s
- F=mg



중력가속도 (g)

- **속도의 변화가 있는 곳에 힘이 발생**
  - F=ma
- **질량을 가진 물체 사이에는 중력으로 인해 끌림이 있음 (만유인력)**
  - $F = G\frac{m_1 m_2}{r^2}$
  - 지구 중력에 의해 낙하하는 물체는 물체의 질량(m)에 관계없이 동일한 운동을 한다



중력가속도(g)

R:지구반경 = 6370km
M:지구질량 = 5.98×10^{24}kg
G:비례상수 = $6.67 \times 10^{-11} m^3/s^2kg$

$$\vec{F} = \frac{Gm_1m_2}{r^2}\hat{r} \qquad \vec{F} = m\vec{a}$$

$$\frac{GMm}{R^2} = ma$$

$$\boxed{a = \frac{GM}{R^2} = 9.8m/s^2}$$

가속도 (Acceleration)

- 가속도
  - 물체의 속도가 변화하는 비율이다.
  - The rate at which the velocity of an object changes
- 물체가 운동하는 경우
  - 정해진 시간 동안에 얼마나 속도가 변했나를 나타내는 값이다.
  - How much speed has been changed for the specified amount of time
- 가속도가 10m/s/s라는 것은?
  - 1초 동안에 10 m/s 씩 속도가 증가한다는 것이다.
  - During 1sec, the speed increase by 10 m/s

가속도 (Acceleration)

- 자동차 급 발진할 때
- Sudden start when driving a car
  - **속도의 변화가 있는 구간 (Velocity Change)**
  - Increase in velocity
    - 느린 속도→ 빠른 속도
    - Slow Velocity → Rapid Velocity
  - **가속도** 구간 (**acceleration**)
- **운전자는 뒤로 쏠리는 힘 받음**
  - Driver receives the **force** leaning **back**
- **Force (F) ∝ Acceleration (a) = Velocity Change**
- **F=ma (Newton's Second Law of Motion)**

감속도 (Deceleration)

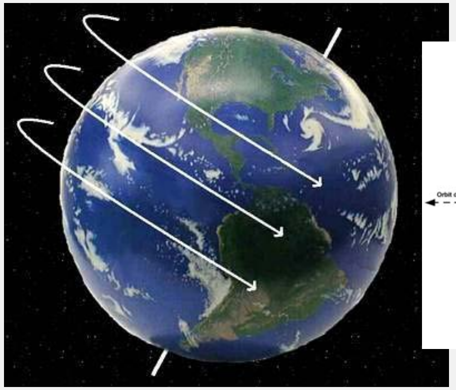
- 자동차 급 정거할 때
- Sudden braking when driving a car
  - **속도의 변화가 있는 구간 (Velocity Change)**
  - Decrease in velocity
    - 빠른 속도 → 느린 속도
    - Rapid Velocity → Slow Velocity
  - **감속도** 구간 (**Deceleration**)
- **운전자는 앞으로 쏠리는 힘 받음**
  - Driver receives the **force** leaning **forward**
- **Force (F) ∝ Deceleration (-a) = Velocity Change**
- **F=m(-a) (Newton's Second Law of Motion)**

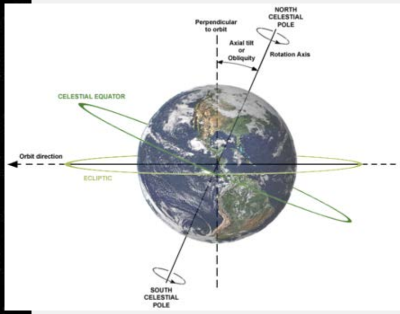
등속도 (Constant Velocity)

- 자동차가 순항할 때
- When car is cruising, there is no velocity change
  - **속도의 변화가 없는 구간 (No Velocity Change)**
    - rapid velocity (200km/h) → rapid velocity (200km/h)
    - slow velocity (50km/h) → slow velocity (50km/h)
  - **No acceleration, No deceleration**
- **뒤로 또는 앞으로 쏠리는 힘 발생 없음**
  - Driver receives **no force** leaning **back or no force** leaning **forward.**
  - 속도 변화가 없음 → 가속도, 감속도 구간이 아님
  - **Force (F) ∝ Acceleration (a) = Deceleration (-a)** = zero = **No Velocity Change**
  - $a = \frac{dv}{dt} = 0$ **F=ma =0 (Newton's Second Law of Motion)**

등속도 운동의 예(F=ma)

- The mass of the earth (m)
  - 약 59조 8천억톤
  - 59,800,000,000,000 t
  - Around 60,000 billion (10^9), 60 trillion (10^{12}) ton
- The earth's rotation velocity
  - 초속 30 km
  - 30km/sec
  - **Constant Velocity**
  - **Person on the Earth receives no force leaning back or no force leaning forward.**
  - F=ma=0





Earth's axial tilt is about 23.4°.

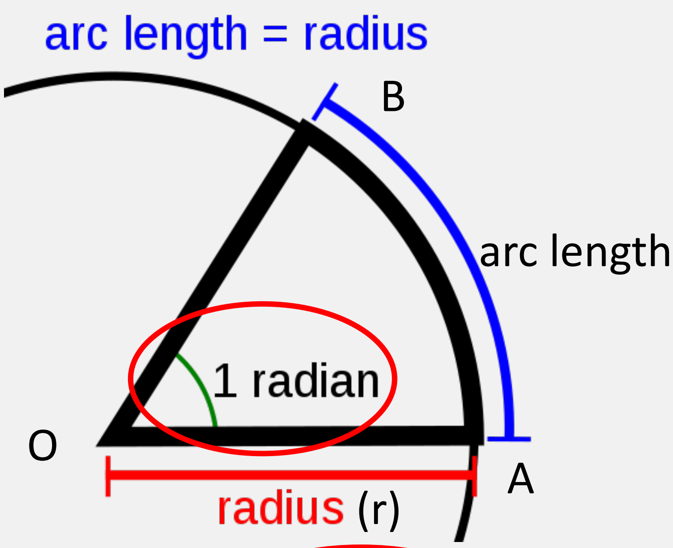
등속도 운동의 예 (F=ma)

- F=ma
  - m: Mass (질량)
  - a : Acceleration (가속도)
  - 지구의 자전은 등속운동으로, 가속도가 없기 때문에 $\frac{dv}{dt} = 0$, 즉 $a = 0$ 이다.
  - 질량(m)이 59조 8천억톤이지만 질량이 m인 지구 위에 있는 우리가 자전에 의해 받는 힘 (F)은 zero이다.
  - 지구상에 살고 있는 우리는 지구의 자전 속도를 전혀 느끼지 못하고 있다.

학습내용 (2주 / 1)
Part One Modeling, Computers, and Error Analysis,
교안: 라디안, 삼각 함수

Radian

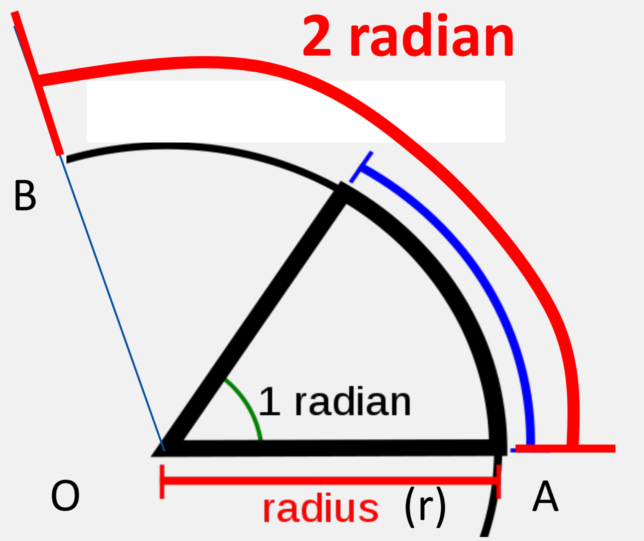
http://en.wikipedia.org/wiki/Radian

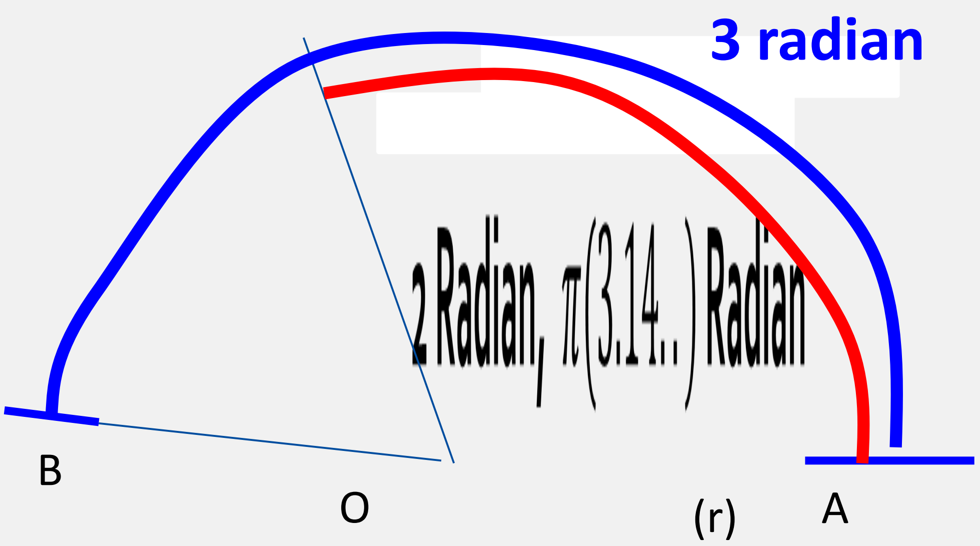


- 1 radian = 57.29 degree
  - $1 \times \frac{180^{\circ}}{\pi} = 57.29^{\circ}$

- **Radian**
  - Ratio between the length of an arc (AB) and its radius (r).
  - Radian is the standard unit of angular measure
- 1 radian = 57 degree = angle AOB
  - 반지름의 길이 (radius)와 호 AB의 길이가 같을 때, 각도 AOB의 크기를 1호도 (radian)이라 한다.
  - When the length of the radius (r) and the length of the arc AB is same, The size of the angle AOB is called as 1 radian.

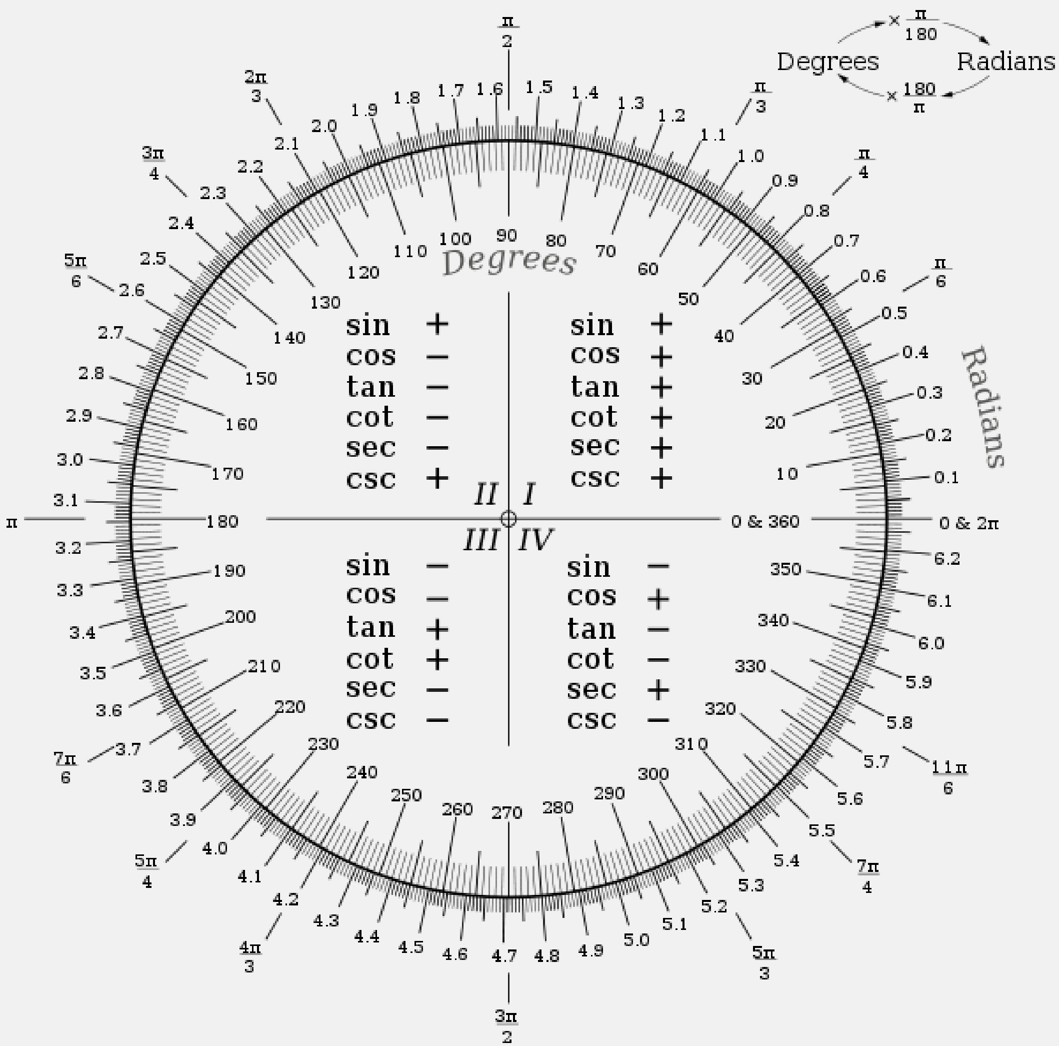
2 Radian, $\pi(3.14..)$ Radian





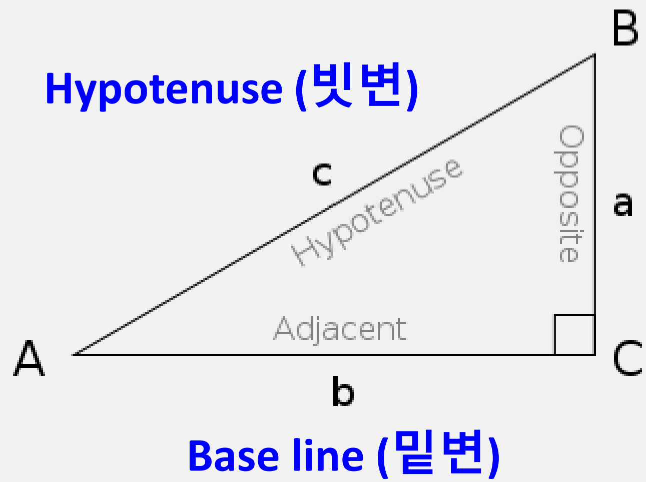
- 1 radian = 57.29 degree
  - $1 \times \frac{180^\circ}{\pi} = 57.29^\circ$
- 2 radian = 114.59 degree = angle AOB
  - $2 \times \frac{180^\circ}{\pi} = 114.59^\circ$
- 3 radian = 171.89 degree
  - $3 \times \frac{180^\circ}{\pi} = 171.89^\circ$
  - which is not 180°
- 3.14159 radian = 180 degree = π radian

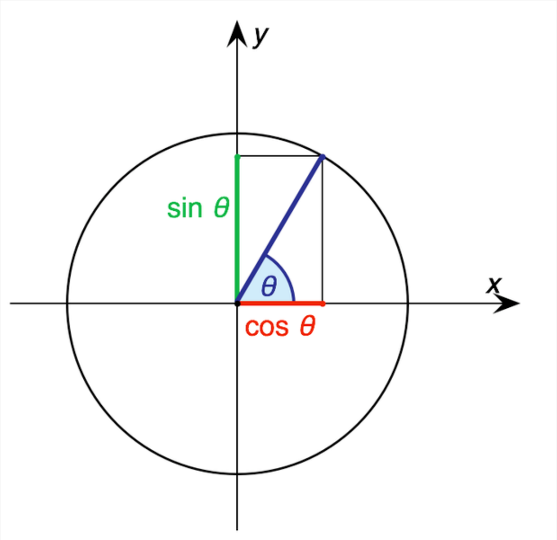
Degree and Radian



Trigonometry (삼각법)

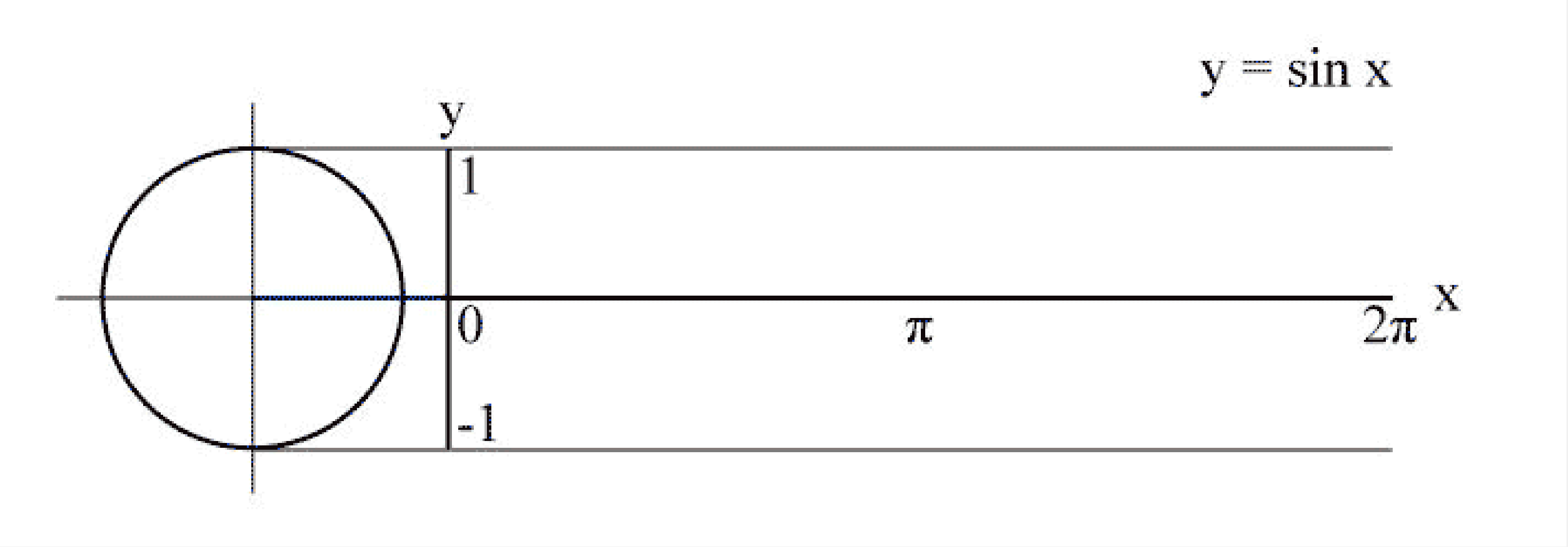
- A branch of mathematics that studies triangles and the relationships between their sides and the angles between these sides



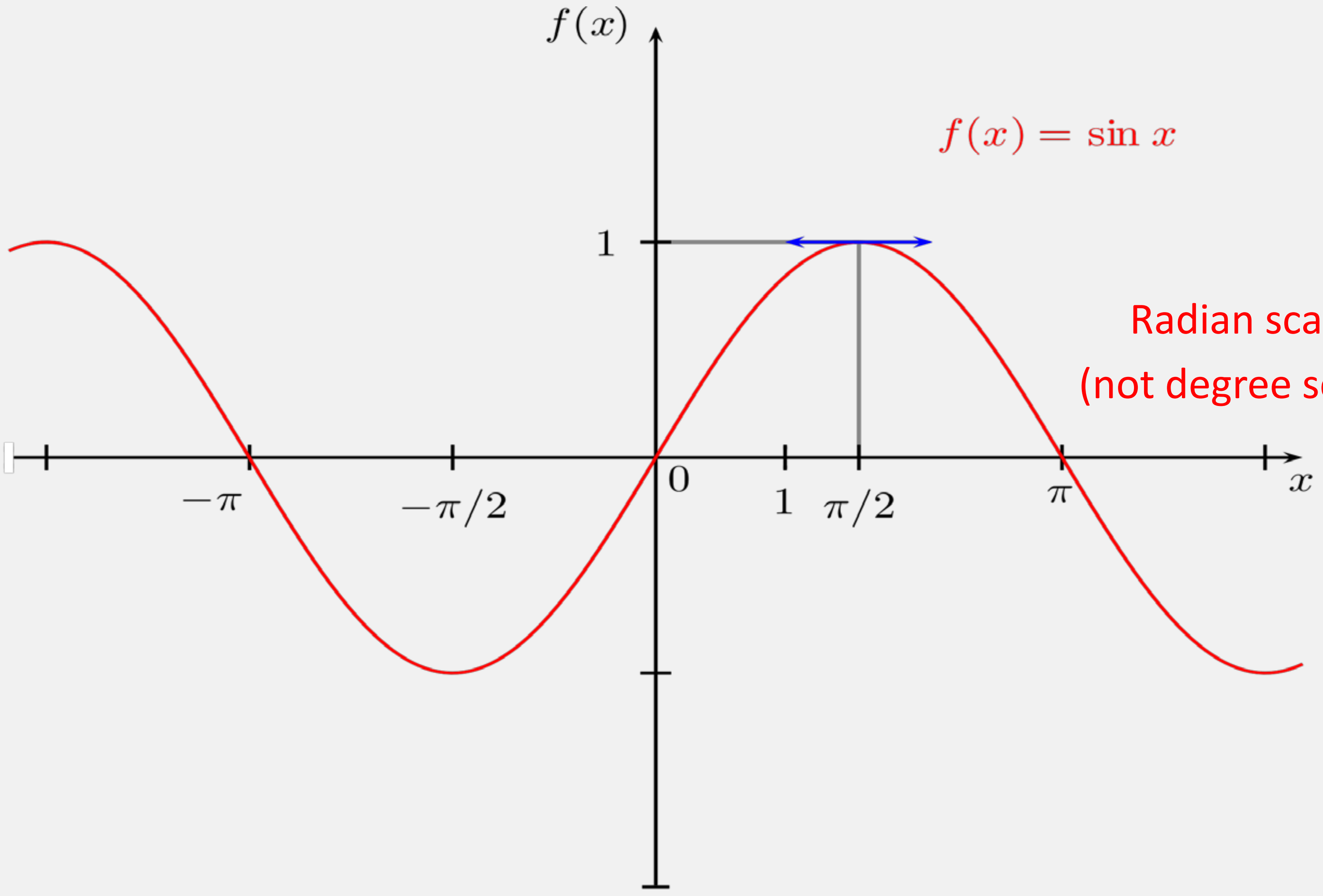


$$\sin A = \frac{\text{opposite}}{\text{hypotenuse}} = \frac{a}{c}. \qquad \cos A = \frac{\text{adjacent}}{\text{hypotenuse}} = \frac{b}{c}. \qquad \tan A = \frac{\text{opposite}}{\text{adjacent}} = \frac{a}{b} = \frac{\sin A}{\cos A}.$$

Trigonometric Function (삼각 함수)



Sine function

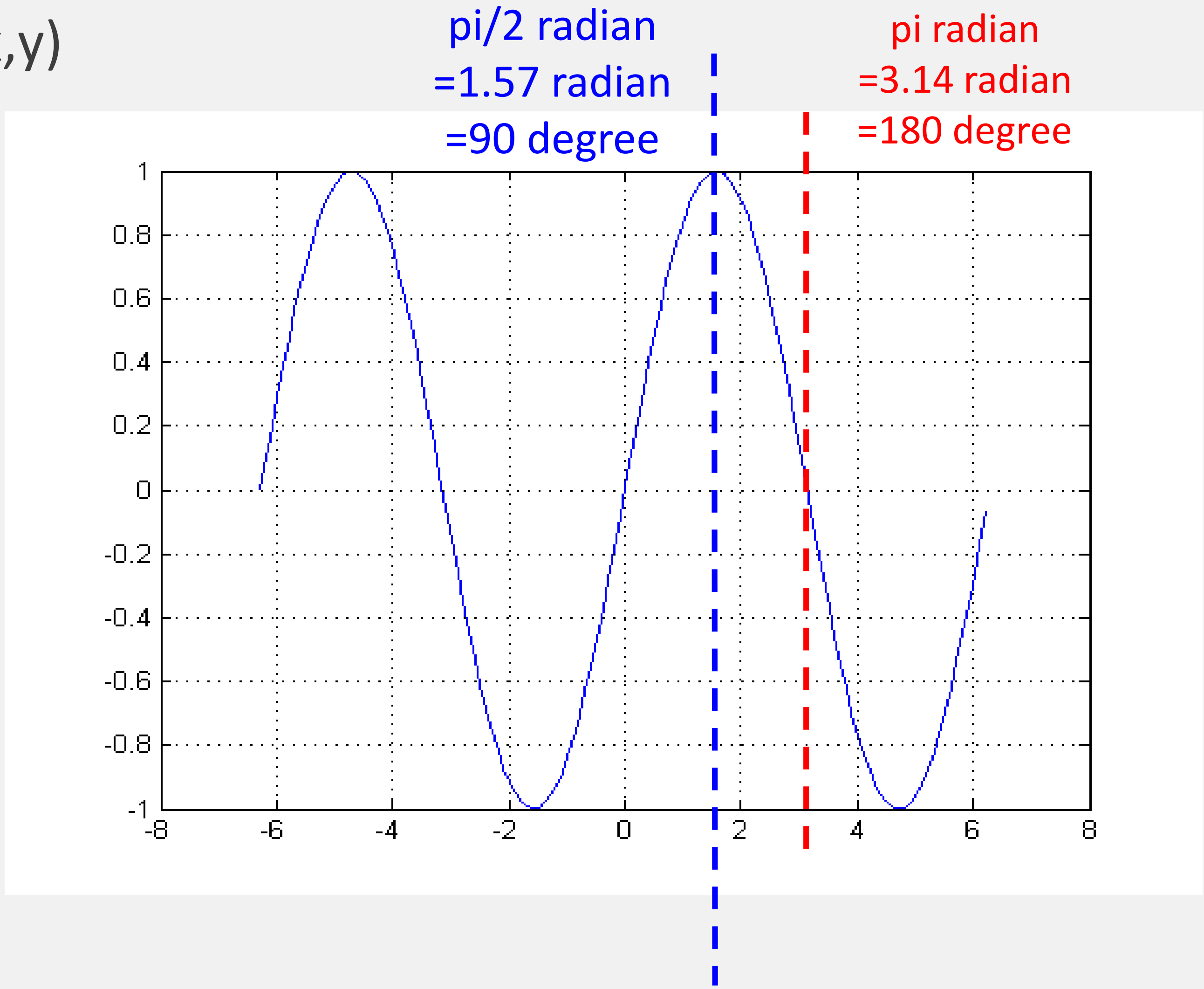


Radian scale

(not degree scale)

Draw Sin function with Matlab, How about Python?

- x=[-2*pi: 0.01:2*pi]
- y=sin(x) , plot(x,y)

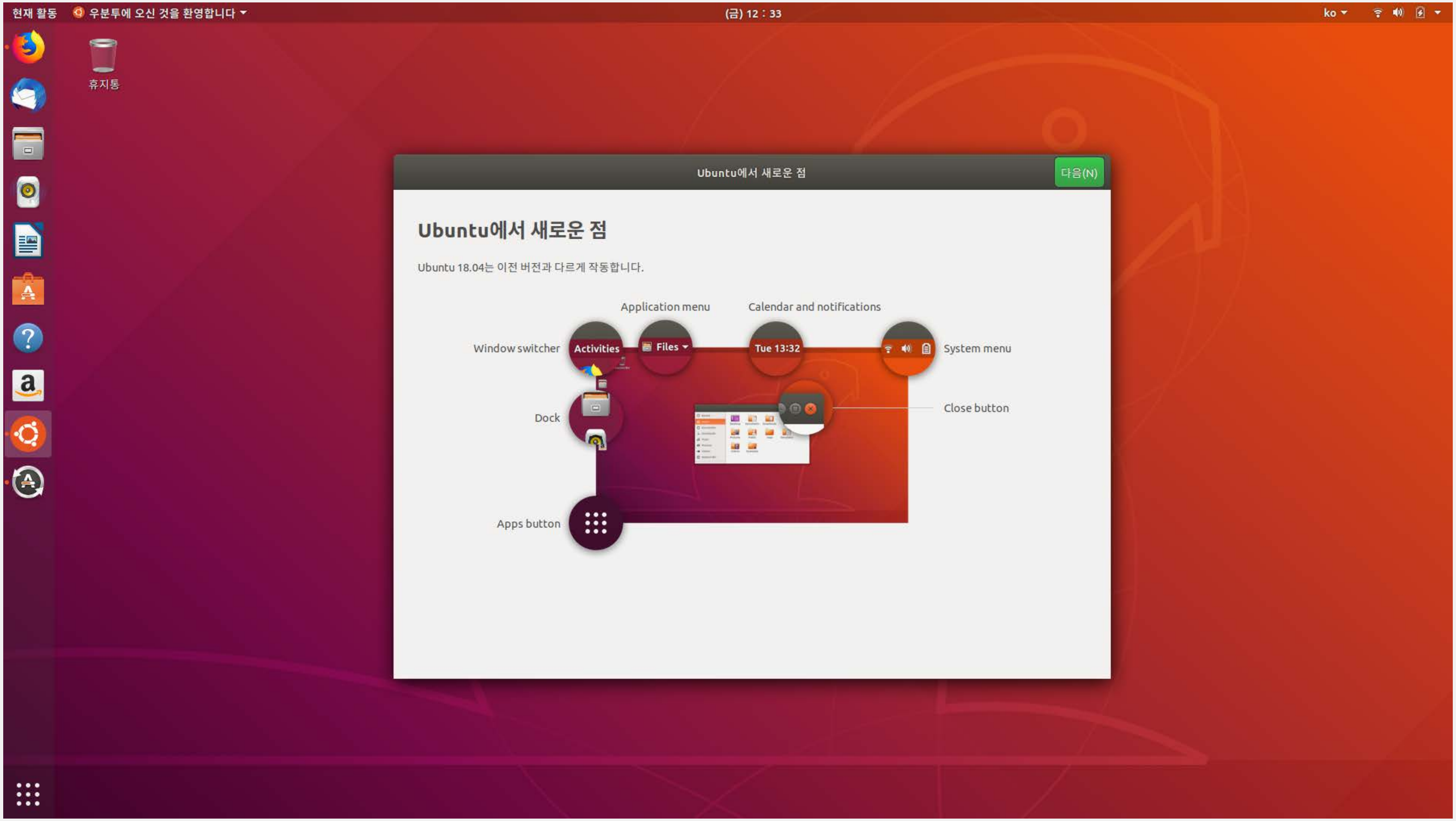


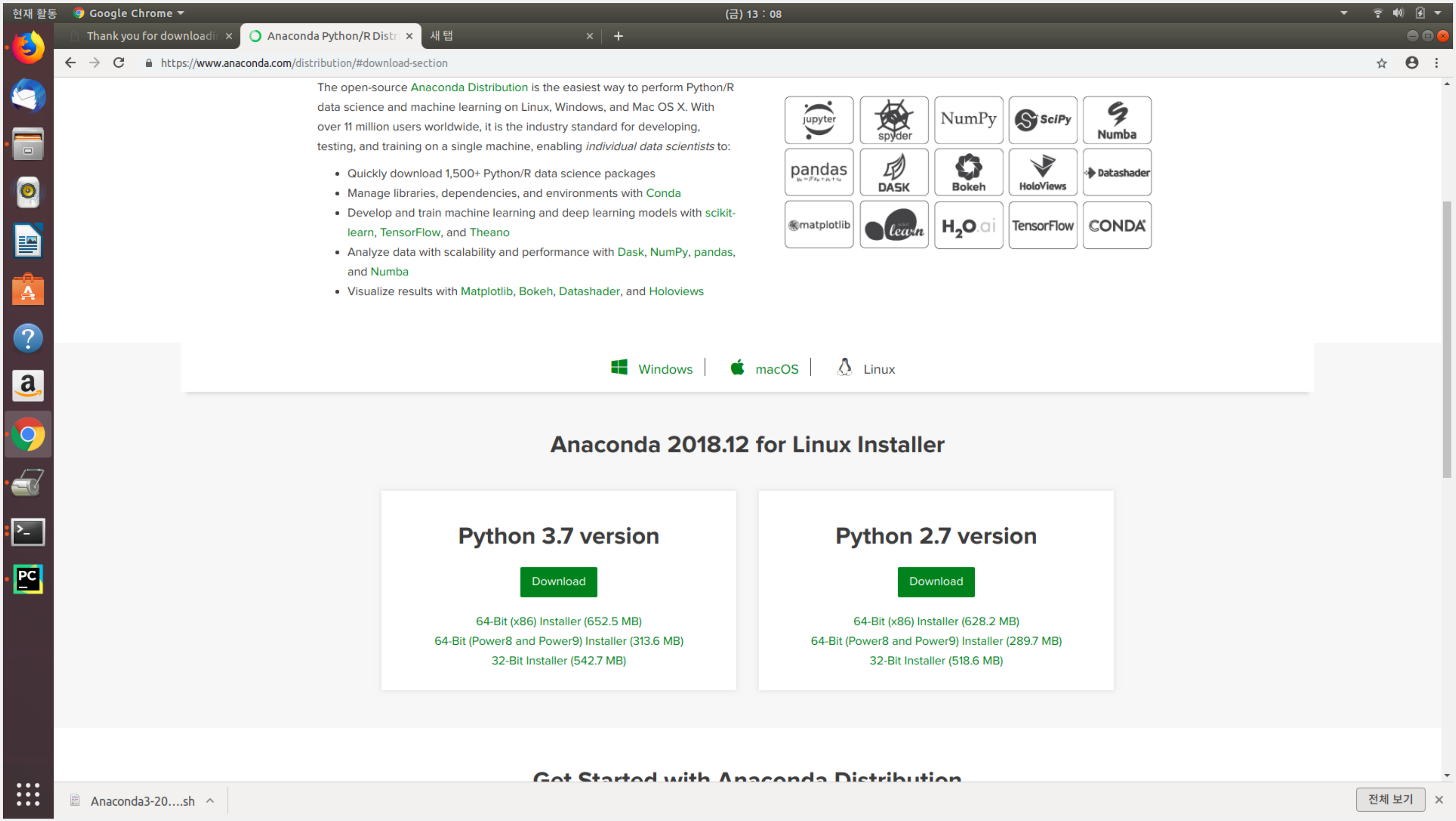
학습내용(2주/1차시)
Part One Modeling, Computers, and Error Analysis,
교안: Python 설치

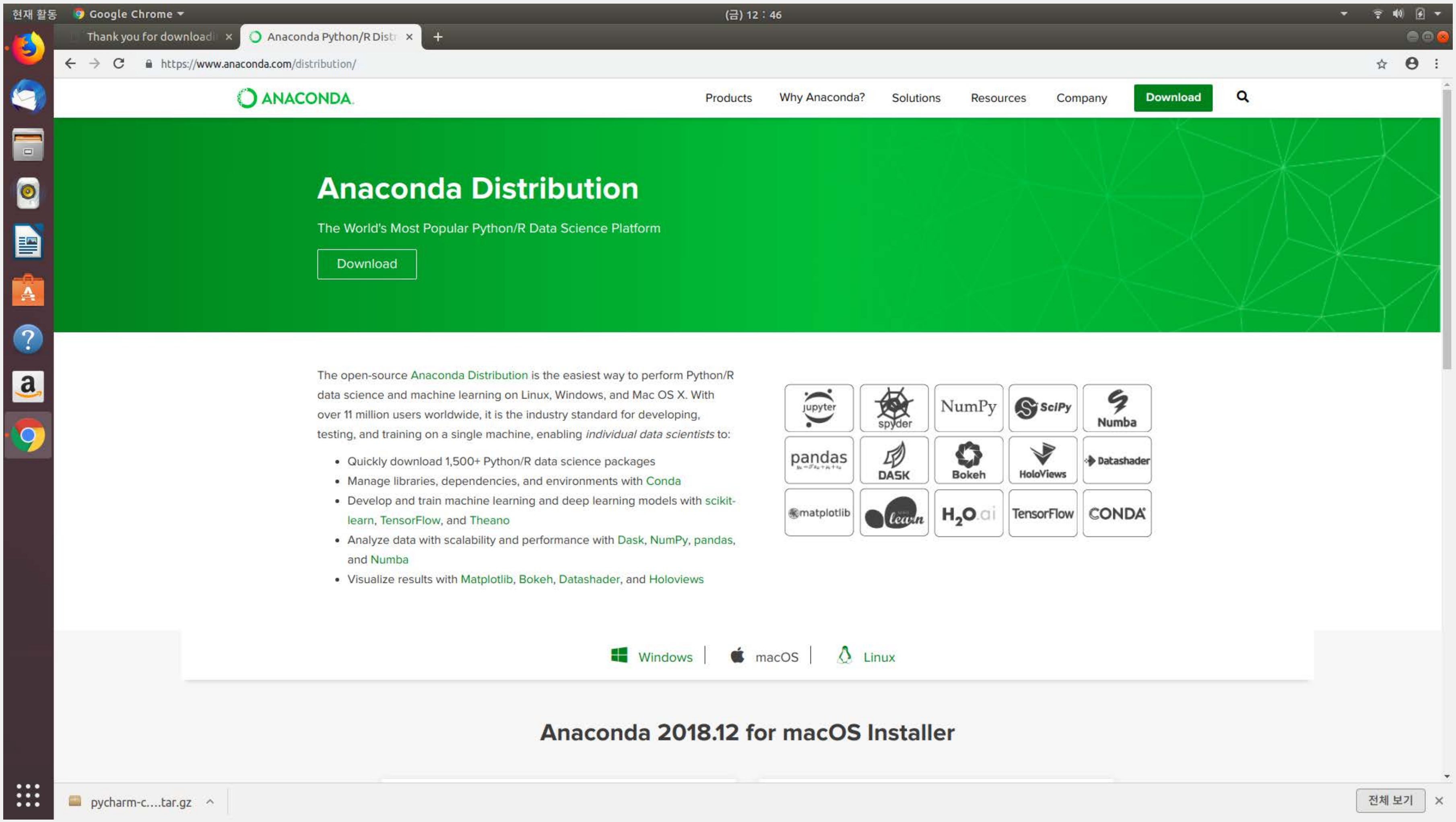
학습내용 2주/1차시 : 과학 현상을 수치적으로 분석하기위한 코딩 환경에 대해 알아본다

NumPy and SciPy

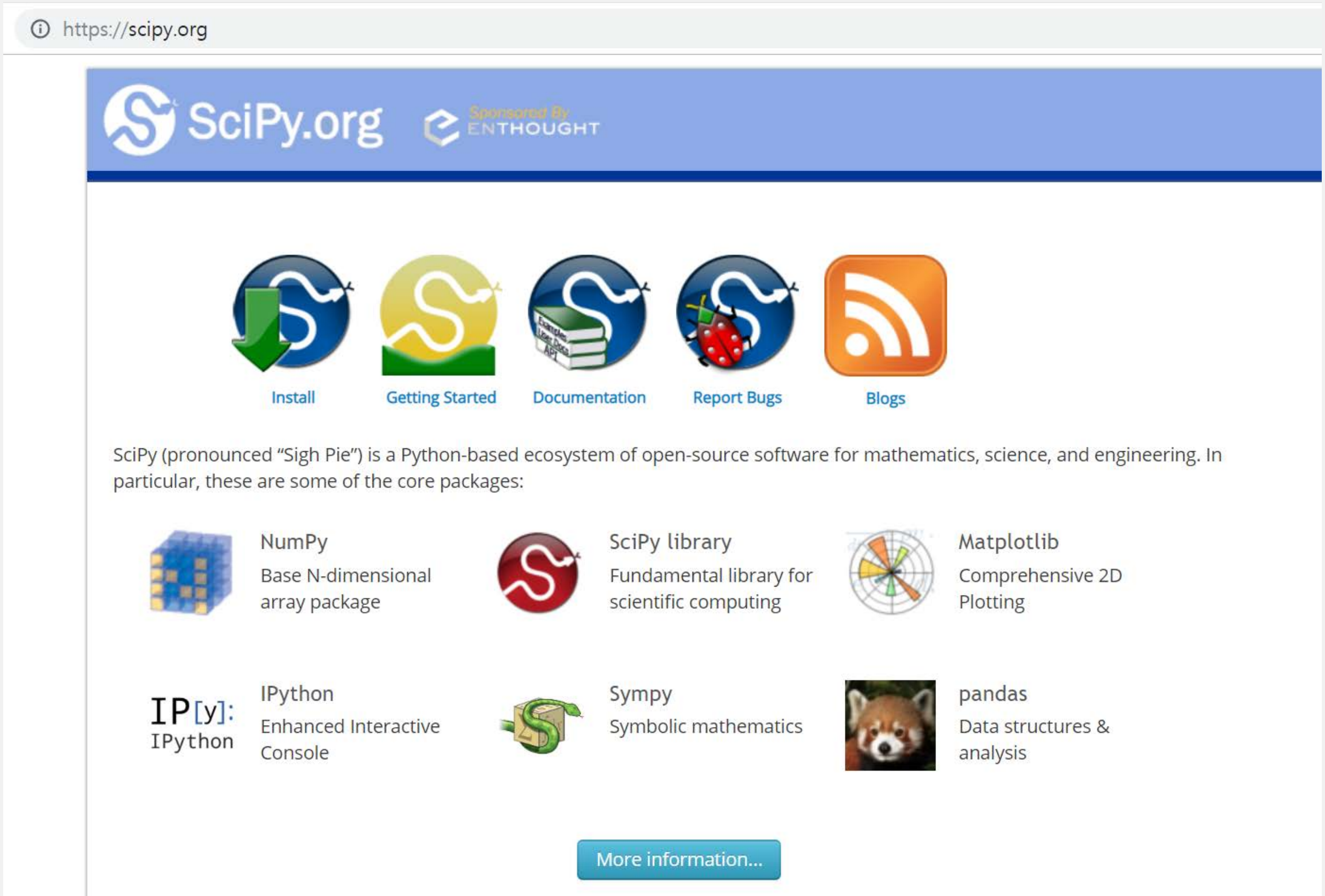
Prof. Sang-Chul Kim


현재 활동
우분투에 오신 것을 환영합니다
(금) 12 : 33
ko
휴지통
Ubuntu에서 새로운 점
다음(N)
Ubuntu에서 새로운 점
Ubuntu 18.04는 이전 버전과 다르게 작동합니다.
Application menu
Calendar and notifications
Window switcher
Activities
Files
Tue 13:32
System menu
Dock
Close button
Apps button


현재 활동
Google Chrome
(금) 13 : 08
Thank you for downloadi
Anaconda Python/R Distr
새 탭
https://www.anaconda.com/distribution/#download-section
The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 11 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling individual data scientists to:
Quickly download 1,500+ Python/R data science packages
Manage libraries, dependencies, and environments with Conda
Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews
jupyter
spyder
NumPy
SciPy
Numba
pandas
DASK
Bokeh
HoloViews
Datashader
matplotlib
learn
H2O.ai
TensorFlow
CONDA
Windows
macOS
Linux
Anaconda 2018.12 for Linux Installer
Python 3.7 version
Download
64-Bit (x86) Installer (652.5 MB)
64-Bit (Power8 and Power9) Installer (313.6 MB)
32-Bit Installer (542.7 MB)
Python 2.7 version
Download
64-Bit (x86) Installer (628.2 MB)
64-Bit (Power8 and Power9) Installer (289.7 MB)
32-Bit Installer (518.6 MB)
Anaconda3-20....sh
전체 보기


Google Chrome
(금) 12 : 46
Thank you for download
Anaconda Python/R Distr
https://www.anaconda.com/distribution/
ANACONDA.
Products
Why Anaconda?
Solutions
Resources
Company
Download
Anaconda Distribution
The World's Most Popular Python/R Data Science Platform
Download
The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 11 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling individual data scientists to:
Quickly download 1,500+ Python/R data science packages
Manage libraries, dependencies, and environments with Conda
Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews
jupyter
spyder
NumPy
SciPy
Numba
pandas
DASK
Bokeh
HoloViews
Datashader
matplotlib
learn
H2O.ai
TensorFlow
CONDA
Windows
macOS
Linux
Anaconda 2018.12 for macOS Installer
pycharm-c....tar.gz
전체 보기

NumPy and SciPy



NumPy

- NumPy는 행렬이나 일반적으로 대규모 다차원 배열을 쉽게 처리 할 수 있도록 지원하는 파이썬의 라이브러리이다.
- NumPy는 데이터 구조 외에도 수치 계산을 위해 효율적으로 구현된 기능을 제공한다.

배열 생성

```
>>> import numpy as np
>>> x = np.array([1, 2, 3])
>>> x
array([1, 2, 3])
>>> y = np.arange(10) # like Python's range, but returns an array
>>> y
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

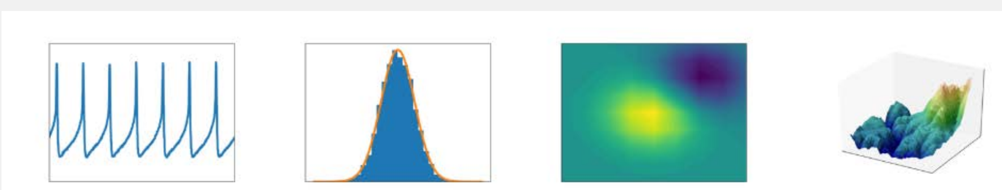
NumPy 배열 API 함수

최근접 이웃 탐색

```
>>> # # # Pure iterative Python # # #
>>> points = [[9,2,8],[4,7,2],[3,4,4],[5,6,9],[5,0,7],[8,2,7],[0,3,2],[7,3,0],[6,1,1],[2,9,6]]
>>> qPoint = [4,5,3]
>>> minIdx = -1
>>> minDist = -1
>>> for idx, point in enumerate(points): # iterate over all points
        dist = sum([(dp-dq)**2 for dp,dq in zip(point,qPoint)])**0.5  # compute the euclidean distance for each point to q
        if dist < minDist or minDist < 0:  # if necessary, update minimum distance and index of the corresponding point
            minDist = dist
            minIdx = idx

>>> print 'Nearest point to q: ', points[minIdx]
Nearest point to q: [3, 4, 4]

>>> # # # Equivalent NumPy vectorization # # #
>>> import numpy as np
>>> points = np.array([[9,2,8],[4,7,2],[3,4,4],[5,6,9],[5,0,7],[8,2,7],[0,3,2],[7,3,0],[6,1,1],[2,9,6]])
>>> qPoint = np.array([4,5,3])
>>> minIdx = np.argmin(np.linalg.norm(points-qPoint,axis=1)) # compute all euclidean distances at once and return the index of the smallest one
>>> print 'Nearest point to q: ', points[minIdx]
Nearest point to q: [3 4 4]
```

시각화 패키지 Matplotlib 소개

- Matplotlib는 파이썬에서 자료를 차트(chart)나 플롯(plot)으로 시각화(visulaization)하는 패키지이다.
- Matplotlib는 다음과 같은 정형화된 차트나 플롯 이외에도 저수준 api를 사용한 다양한 시각화 기능을 제공한다.
  - 라인 플롯(line plot), 스캐터 플롯(scatter plot), 컨투어 플롯(contour plot), 서피스 플롯(surface plot), 바 차트(bar chart), 히스토그램(histogram), 박스 플롯(box plot)



https://datascienceschool.net/

데이터시각화 예습 (데이터사이언스 스쿨)

- https://datascienceschool.net/view-notebook/d0b1637803754bb083b5722c9f2209d0/

라인 플롯

가장 간단한 플롯은 선을 그리는 라인 플롯(line plot)이다. 라인 플롯은 데이터가 시간, 순서 등에 따라 어떻게 변화하는지 보여주기 위해 사용한다.

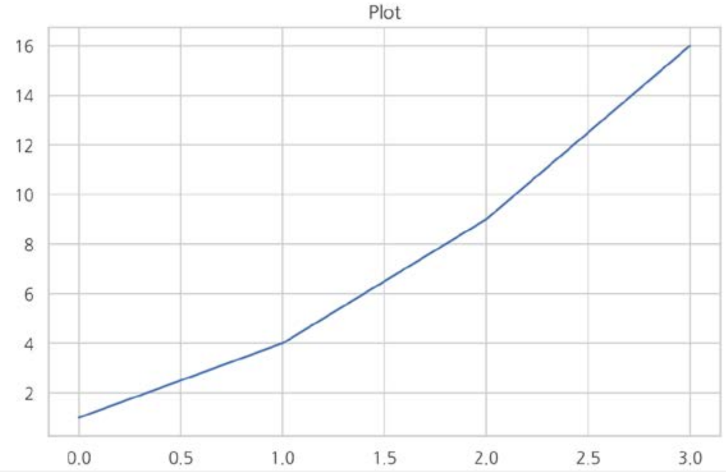
명령은 pylab 서브패키지의 `plot` 명령을 사용한다.

- http://Matplotlib.org/api/pyplot_api.html#Matplotlib.pyplot.plot

만약 데이터가 1, 4, 9, 16 으로 변화하였다면 다음과 같이 `plot` 명령에 데이터 리스트 혹은 `ndarray` 객체를 넘긴다.

In [2]:

```
plt.title("Plot")
plt.plot([1, 4, 9, 16])
plt.show()
```

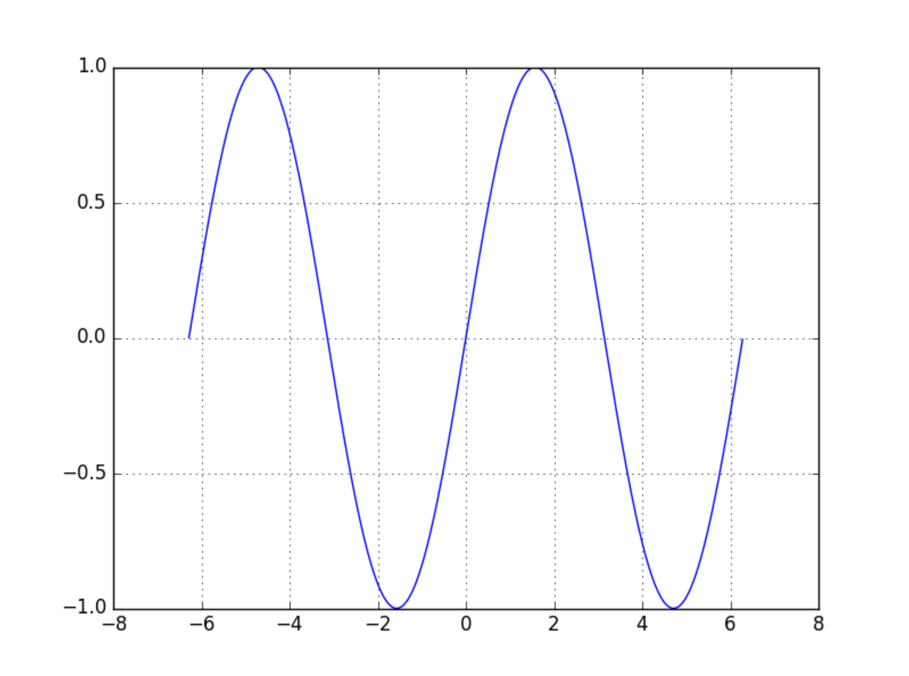


Plotting Sine wave using Python

2주 2차시: pycharm을 통한 파이썬 코딩 환경에 익숙해지기

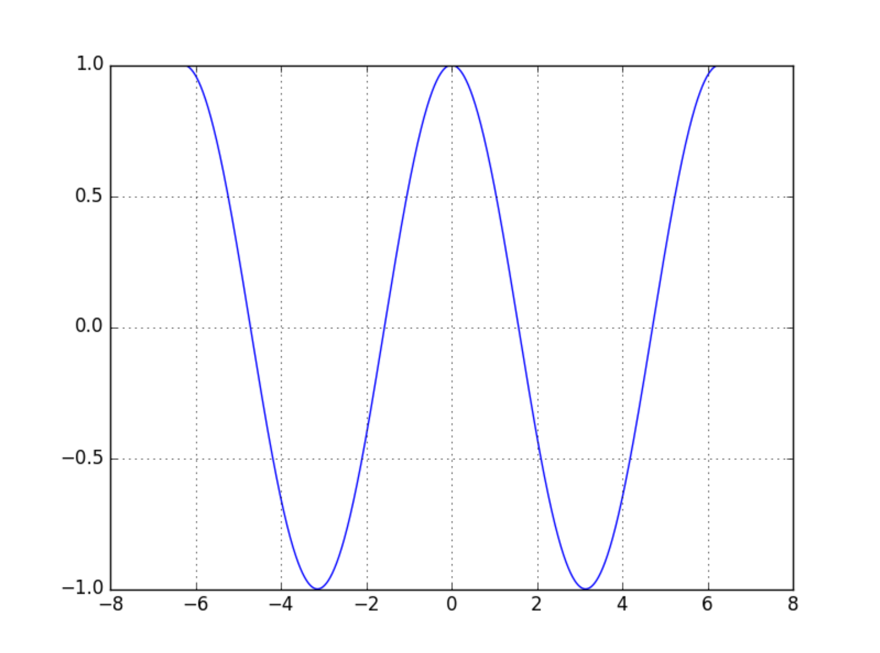
Plotting Sine wave using Python

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(1)
t=np.arange(-2*np.pi, 2*np.pi, 0.01)
y=np.sin(t)
plt.plot(t,y)
plt.grid(True)
plt.show()
```



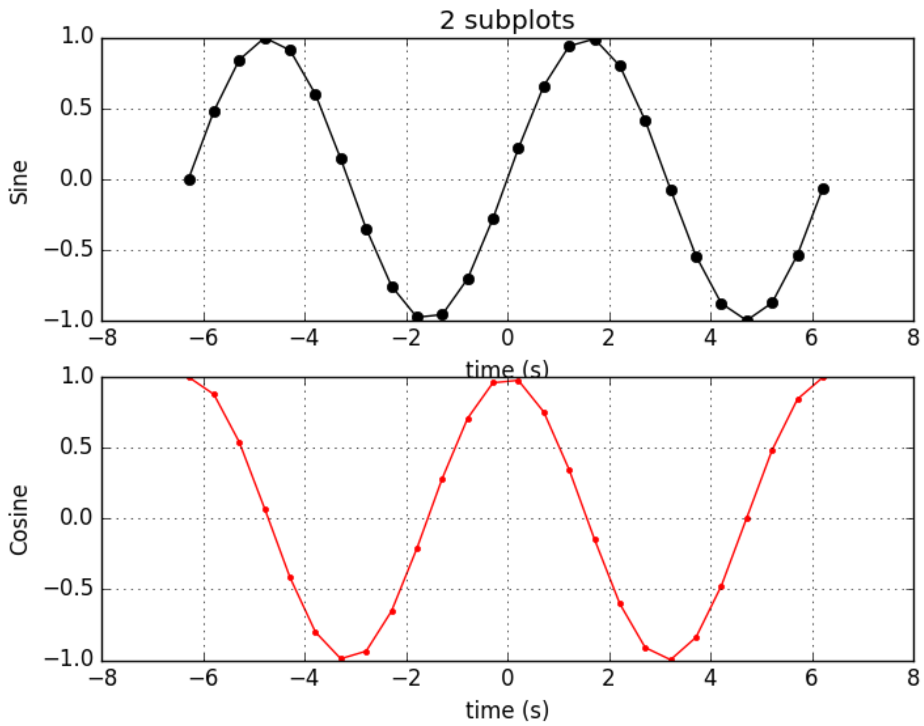
Plotting Cosine wave using Python

```
import numpy as np
import matplotlib.pyplot as plt

plt.figure(2)
t=np.arange(-2*np.pi, 2*np.pi, 0.01)
y1=np.cos(t)
plt.plot(t,y1)
plt.grid(True)
plt.show()
```



Sub-Plotting for Sine and Cosine

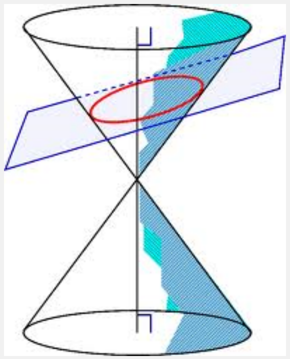
```
import numpy as np
import matplotlib.pyplot as plt
t=np.arange(-2*np.pi, 2*np.pi,
0.5)
y1=np.sin(t)
y2=np.cos(t)
plt.subplot(2, 1, 1)
plt.plot(t, y1, 'ko-')
plt.title('2 subplots')
plt.xlabel('time (s)')
plt.ylabel('Sine')
plt.grid(True)

plt.subplot(2, 1, 2)
plt.plot(t, y2, 'r.-')
plt.xlabel('time (s)')
plt.ylabel('Cosine')
plt.grid(True)
plt.show()
```

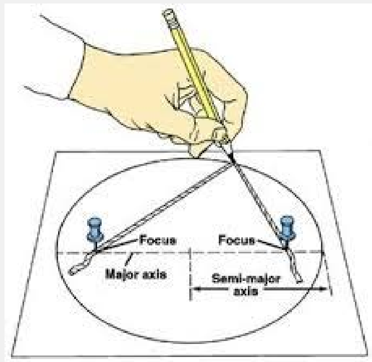


학습내용(3주/1 차시)
Part One Modeling, Computers, and Error Analysis,
교안: 타원, 쌍곡선, 쌍곡선 함수

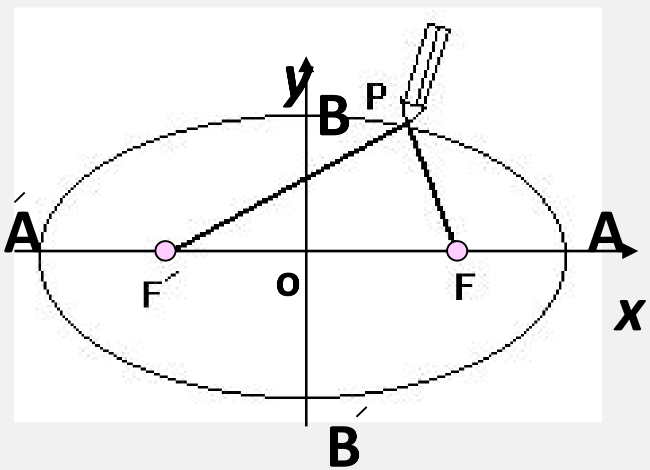
학습내용 3주/1차시: 타원, 쌍곡선, 쌍곡선 함수 이해하기

타원 (Ellipse)

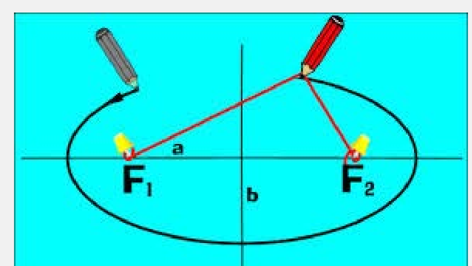




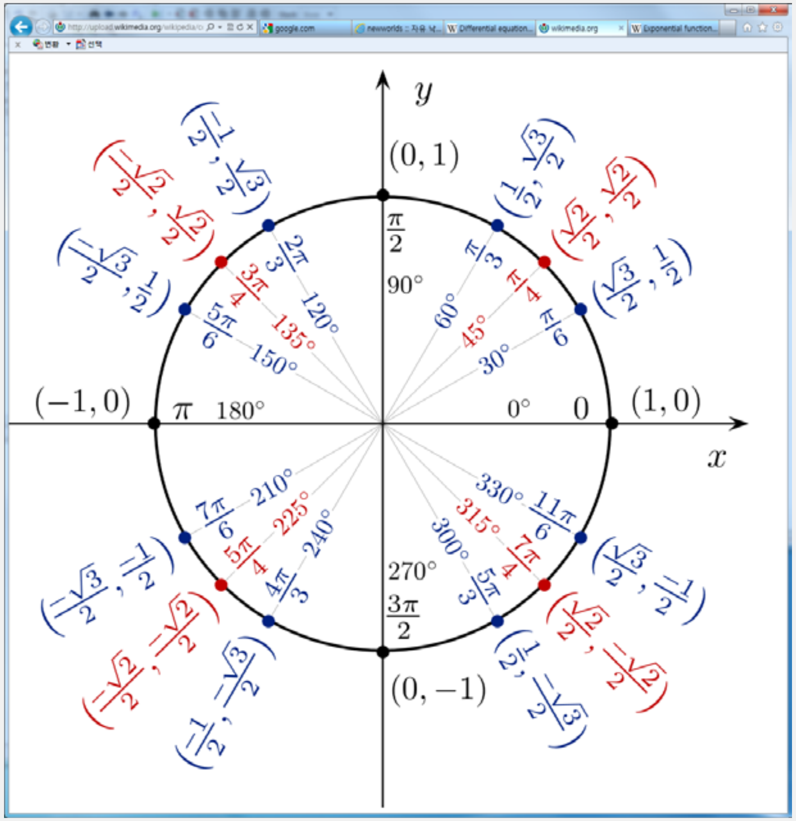
- A set of points providing constant length sum (major axis length) from two vertices F, F' above the plane
- 평면 위에서 두 정점 F, F' 으로 부터의 거리의 합(장축의 길이)이 일정한 점들의 집합
- F, F' : 타원의 초점 (focus)
- A, A', B, B' : 타원의 꼭지점 (vertex)
- $\overline{AA'}$: 장축 (major axis)
- $\overline{BB'}$: 단축 (minor axis)
- O :타원의 중심 (center)



$$\overline{PF} + \overline{PF'}$$
$$= \boldsymbol{Constant\ Length\ Sum}$$
$$= \overline{AA'}$$
$$= \boldsymbol{Major\ Axis\ Length}$$

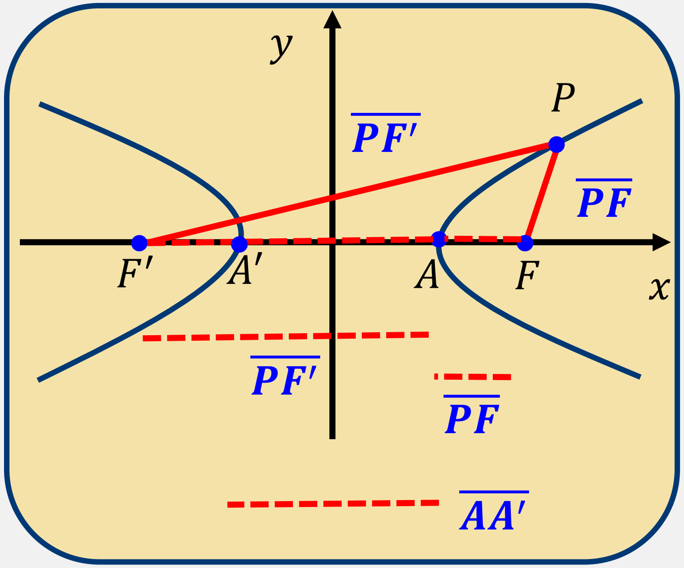


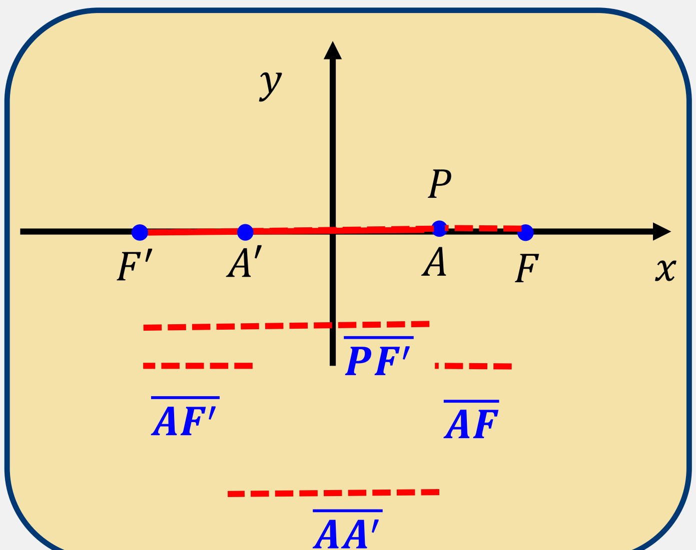
Unit Circle (단위원)의 특징에서 나온 sin, cos



- 단위 원 위의 점은 (x, y)로 표시가능
- 단위 원 위의 점의 특징은 x좌표, y 좌표의 값을 제곱해서 더한 값이 1이 된다.
- 이런 단위 원의 x 좌표의 집합을 cos으로 표시하고, 단위 원의 y 좌표의 집합을 sin으로 표시한다
- $x^2+y^2=1$
- $(x, y) = (cosa, sina)$
- $cosa^2+sina^2=1$

쌍곡선 (Hyperbolic)





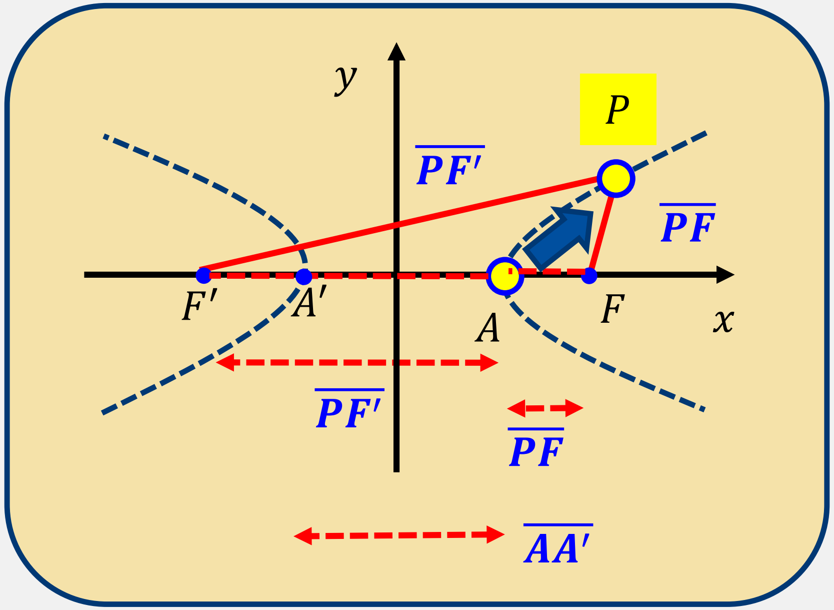
- 평면 위에서 두 정점 F, F' 으로 부터의 거리의 차(주축의 길이)가 일정한 점들의 집합.
- A set of points providing constant length difference (principal axis length) from two vertices F, F ' above the plane
- *F*, *F'* : 초점(focus)
- *A*, *A'*: 꼭지점 (vertex)
- $\overline{AA'}$: 주축 (principal axis)

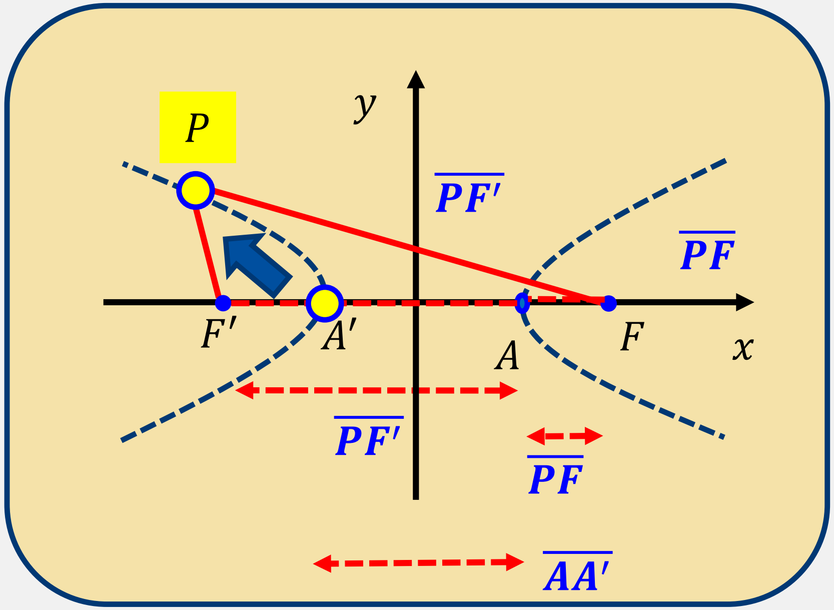
$$\overline{PF}-\overline{PF'}$$
$$= \textit{Constant Length Sum}$$
$$= \overline{AA'}$$
$$= \textit{Principal Axis Length}$$

$$\overline{PF'}-\overline{AF'}=\overline{PF'}-\overline{AF}=\overline{AA'}$$

P점이 A점에 있다고 가정하자.
그러면, PF'길이에서 PF길이를 빼면, AA'의 길이가 나타난다.
여기서 PF길이와 AF와 같다. AF는 A'F'길이와 같다.
그래서 PF'에는 A'F'의 길이가 포함되어 있기 때문에 PF'에서 A'F'의 길이를 빼면, AA'의 길이가 나온다.

P점은 반대편에서도 생성 가능



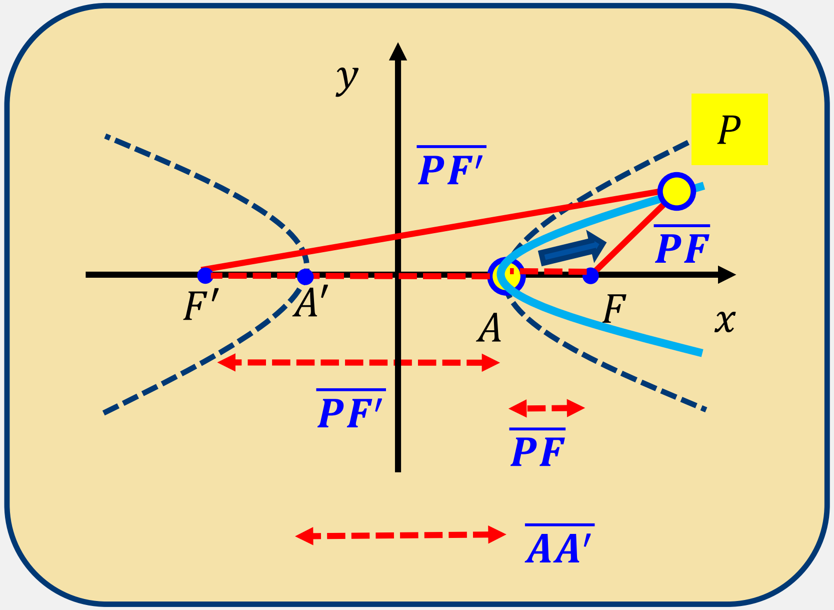


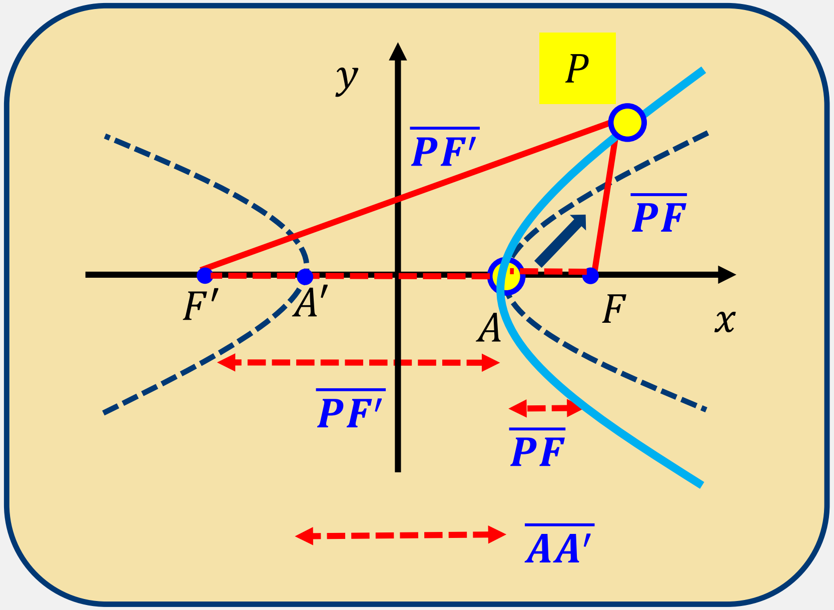
P점이 A점 위로 올라간다고 가정하자.
그러면, PF'길이에서 PF길이를 뺀다는 것은, P점이 A점 의 위치에 있을 때를 상기하여, PF'에서 A'F'의 길이를 빼면, AA'의 길이가 나온다.

P점이 A'점 위에서 올라간다고 가정하자.
그러면, PF 길이에서 PF' 길이를 뺀다는 것은, P점이 A'점 의 위치에 있을 때를 상기하여, PF에서 AF의 길이를 빼면, AA'의 길이가 나온다.

$$\overline{PF} - \overline{PF'}$$
$$= \textit{Constant Length Sum}$$
$$= \overline{AA'}$$
$$= \textit{Principal Axis Length}$$

폭이 좁은 쌍곡선과 폭이 넓은 쌍곡선

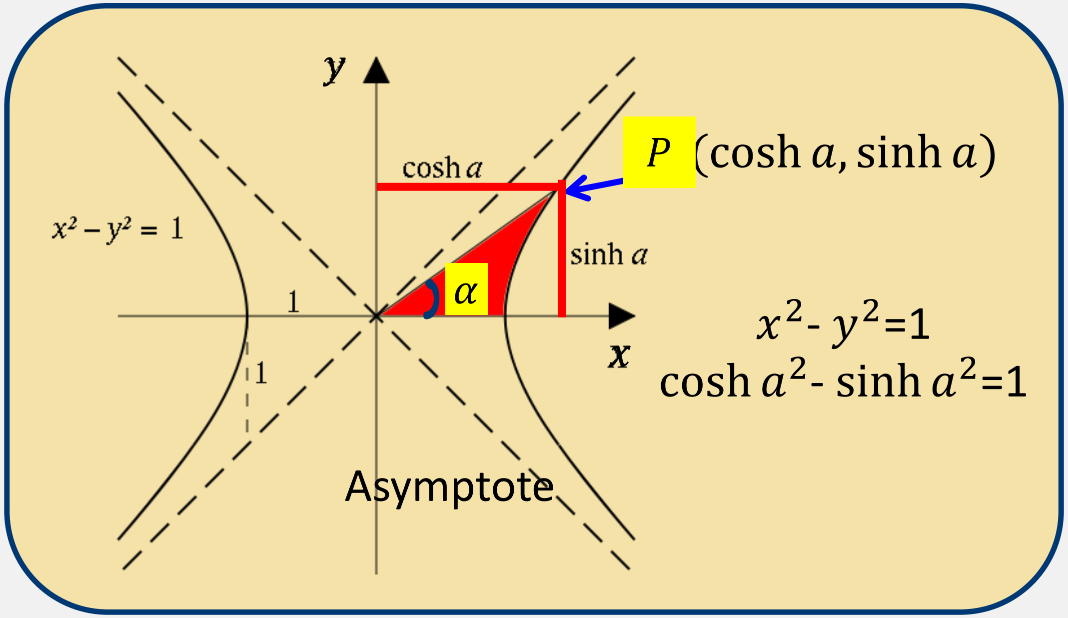


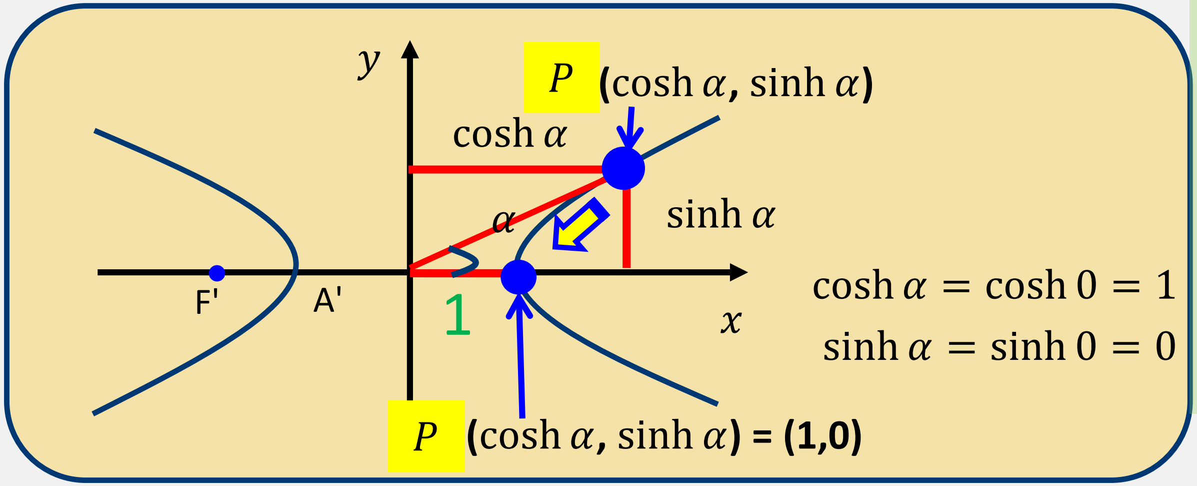


P점이 A점 위로 올라갈 때, x축에 붙어서 올라간다면 폭이 좁은 쌍곡선이 만들어질 것이다.

P점이 A점 위로 올라갈 때, x축에서 떨어져서 y축 쪽으로 붙어서 올라간다면 폭이 넓은 쌍곡선이 만들어질 것이다.

Unit Hyperbolic (단위 쌍곡선)의 특징에서 나온 sinh, cosh





- The difference of squared vales of x and y coordinate gives one
- A set of x coordinate point of hyperbolic is called as **cosine hyperbolic** (cosh).
- A set of y coordinate point of hyperbolic is called as **sine hyperbolic** (sinh).

sinh, cosh 특징 확인

- Matlab
  - 어떤 수(예: 1.2, 2)를 제곱해서 빼면, 1이 나옴
  - (cosh(1.2*pi))^2-(sinh(1.2*pi))^2=1
  - (cosh(2*pi))^2-(sinh(2*pi))^2=1

- Python
  - 어떤 수(예: 1.2, 2)를 제곱해서 빼면, 1이 나옴

```
(np.cosh(1.2*np.pi))**2-
(np.sinh(1.2*np.pi))**2
```

```
0.9999999999994316
```

```
(np.cosh(2*np.pi))**2-
(np.sinh(2*np.pi))**2
```

```
1.000000000145519
```

쌍곡선 함수와 지수 함수와의 관계

- 쌍곡선 함수 그래프 그려보기

$$\sinh x = \frac{e^x - e^{-x}}{2} = \frac{e^{2x} - 1}{2e^x}$$

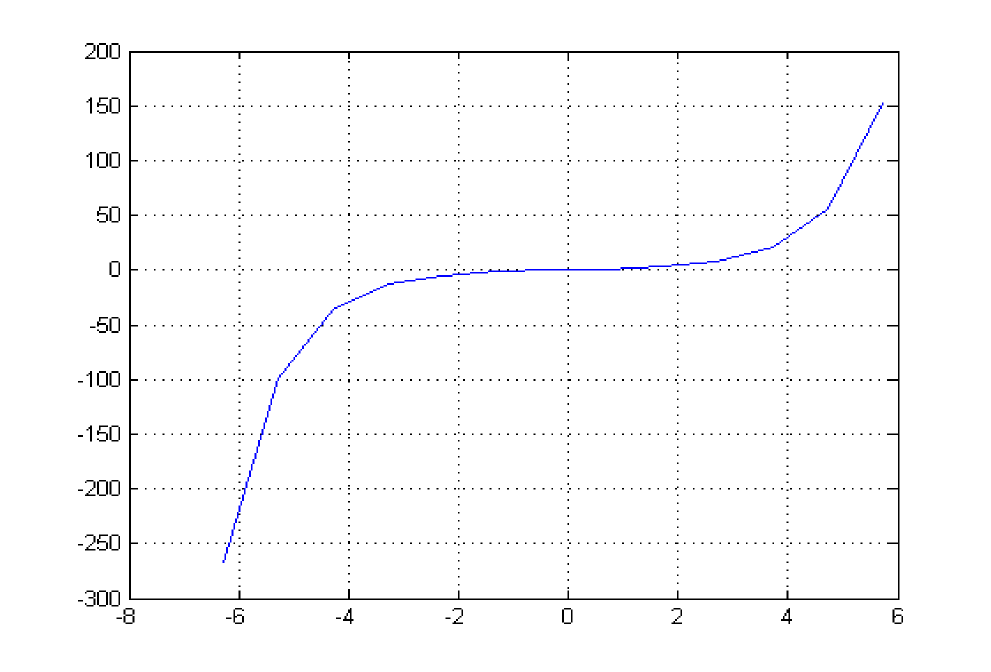
$$\cosh x = \frac{e^x + e^{-x}}{2} = \frac{e^{2x} + 1}{2e^x}$$

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

$$\int \frac{du}{a^2 - u^2} = a^{-1} \tanh^{-1}\left(\frac{u}{a}\right) + C; u^2 < a^2$$

sinh(x) 함수 그래프 그려보기

```
x=[-2*pi: 2*pi] (or x=[-2*pi: 0.01: 2*pi] for more detail)
[-6.2832   -5.2832   -4.2832   -3.2832   -2.2832   -1.2832   -0.2832
 0.7168 1.7168    2.7168    3.7168    4.7168    5.7168]
y1=sinh(x), plot(x, y1)
[-267.7449  -98.4956  -36.2286  -13.3115   -4.8530   -1.6655   -0.2870
 0.7798 2.6936    7.5330   20.5544   55.9013  151.9660]
```



- $e^0 = 1$
- $e^1 = 2.71828^{\ 1}$
- $e^2 = 2.71828^{\ 2} = 7.3891\ldots$
- $e^3 = 2.71828^{\ 3} = 20.0855\ldots$
- $e^4 = 2.71828^{\ 4} = 54.5982\ldots$

지수 (Exponential) 함수에 관해

- $e^0 = 1$
- $e^1 = 2.71828^{1}$
- $e^2 = 2.71828^{2} = 7.3891...$
- $e^3 = 2.71828^{3} = 20.0855...$
- $e^4 = 2.71828^{4} = 54.5982...$

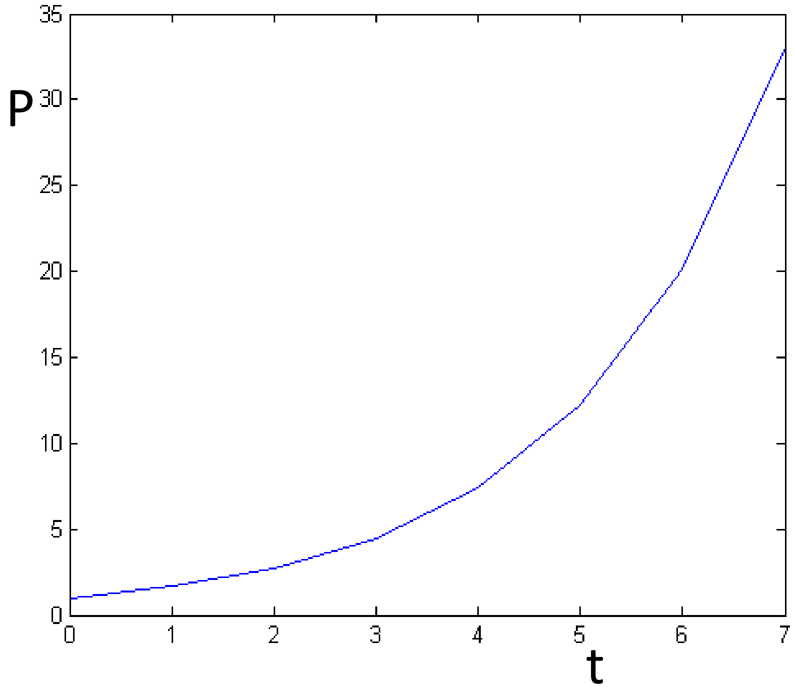
- exp(1)
- exp(1)^2
- exp(1)^3

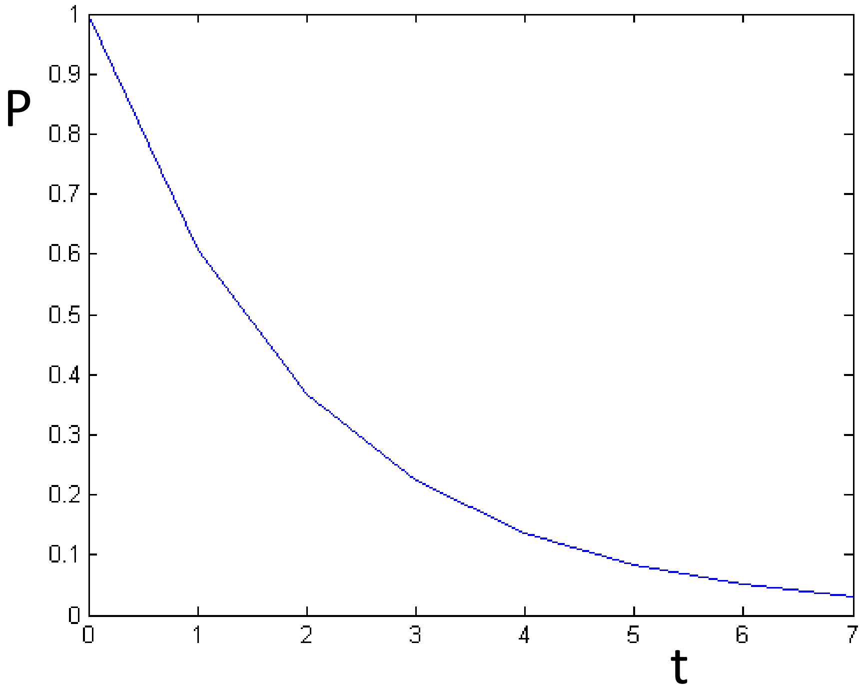
- $e^0 = 1$
- $e^1 = 2.71828^{1}$
- $e^2 = 2.71828^{2} = 7.3891...$
- $e^3 = 2.71828^{3} = 20.0855...$
- $e^4 = 2.71828^{4} = 54.5982...$


```
>>> np.exp(0)
1.0
>>> np.exp(1)
 2.7182818284590451
>>>  np.exp(2)
 7.3890560989306504
```

지수 함수 (Exponential function)

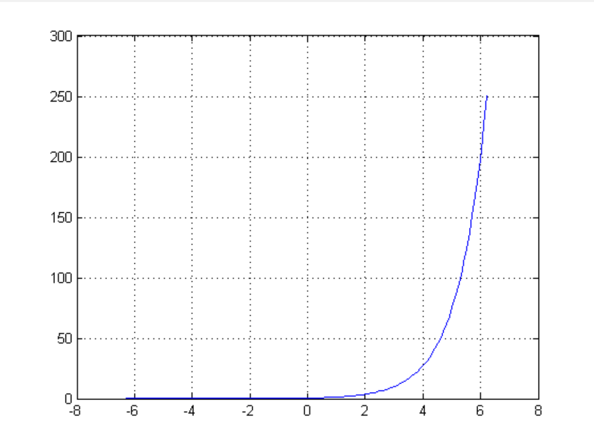
- $P=e^{0.5t}$
- $P=e^{-0.5t}$



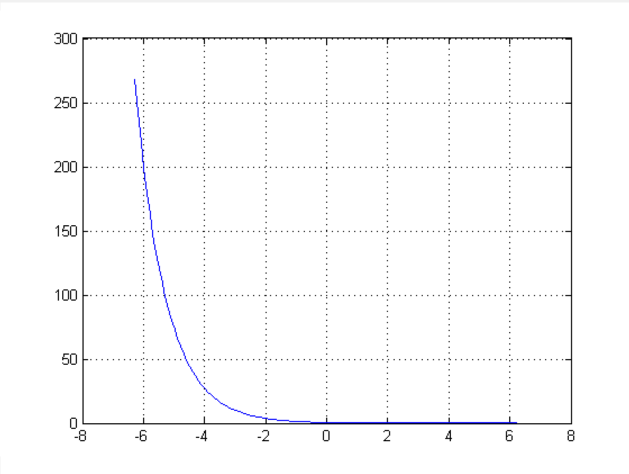


쌍곡선 함수와 지수 함수의 관계

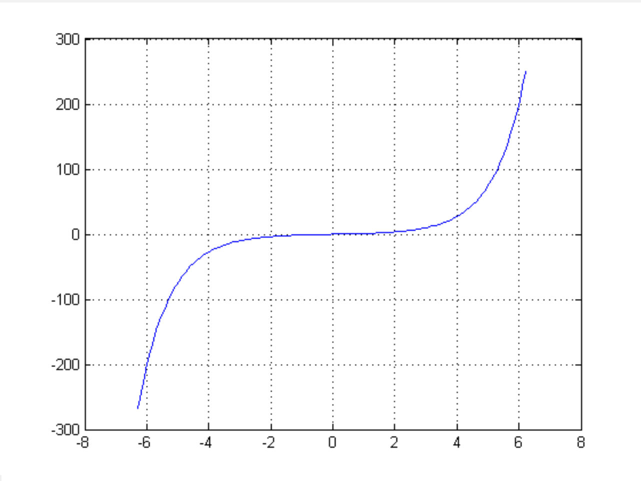
$$\sinh x = \frac{e^{x} - e^{-x}}{2}$$



$$\frac{e^{x}}{2}$$



$$\frac{e^{-x}}{2}$$



$$\frac{e^{x} - e^{-x}}{2}$$

x=[-8: 8]

y1=exp(x)/2

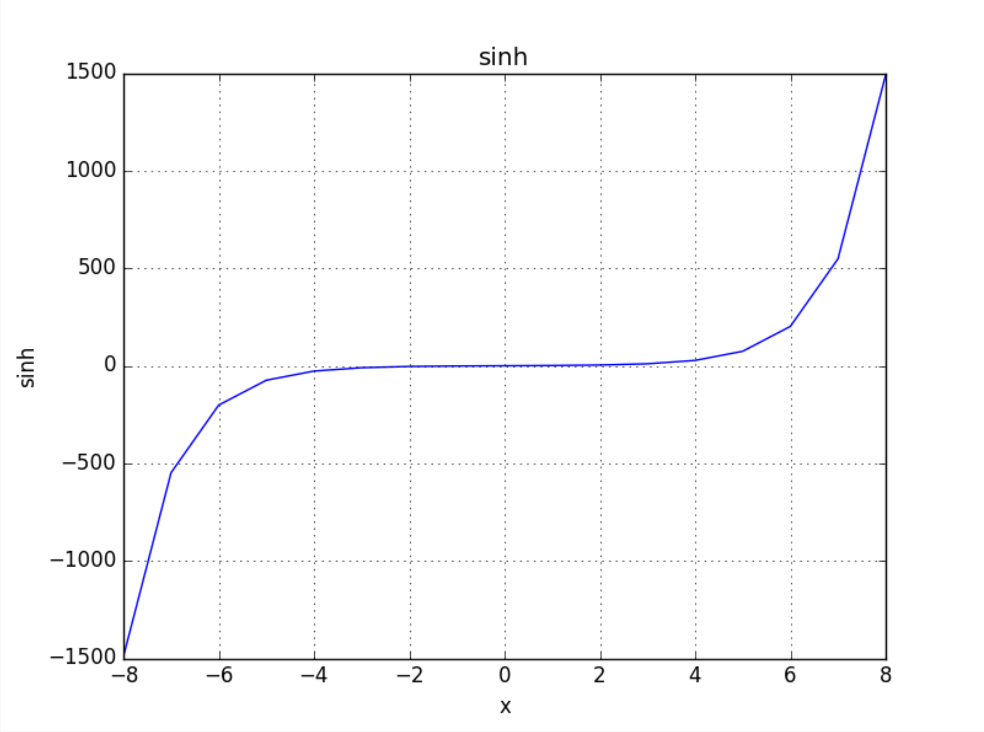
y2=exp(-x)/2

y3=y1-y2

plot(x,y3)

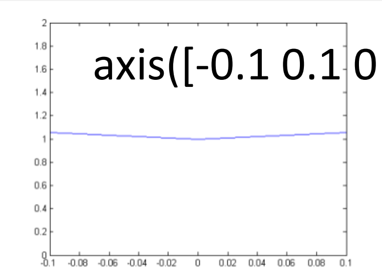
Plot Sinh

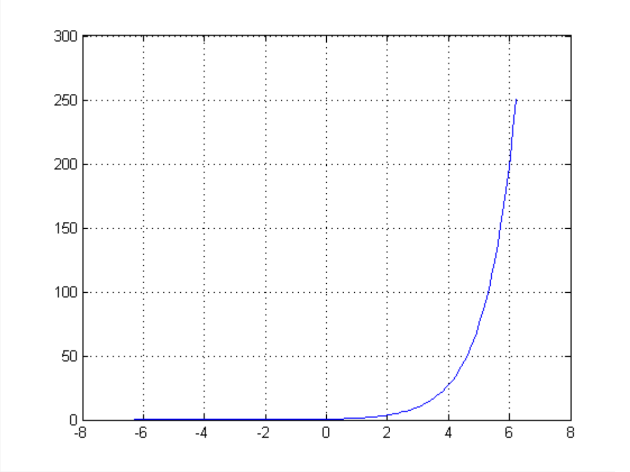
```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-8,8+1,1)
y1=np.exp(x)/2
y2=np.exp(-x)/2
y3=y1-y2

plt.figure(1)
plt.plot(x, y3)
plt.xlabel('x')
plt.ylabel('sinh')
plt.title('sinh')
plt.grid(True)
plt.show()
```

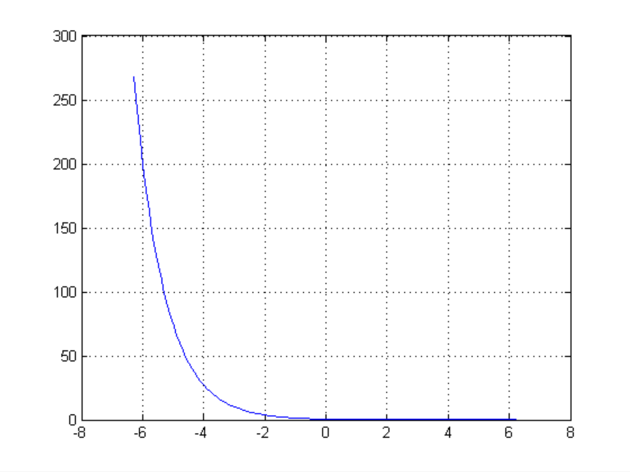


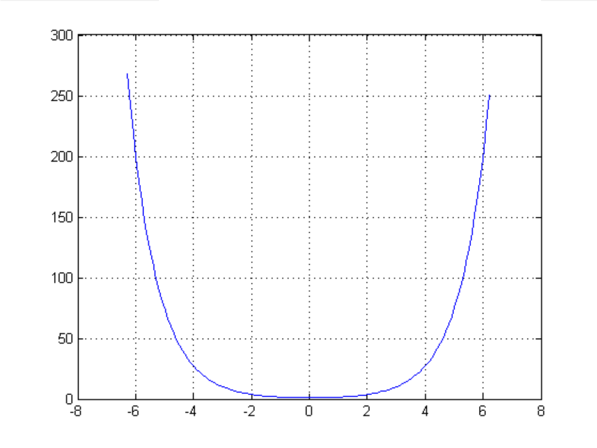
쌍곡선 함수와 지수 함수의 관계

$$\cosh x = \frac{e^x + e^{-x}}{2}$$









$$\frac{e^x}{2}$$

$$\frac{e^{-x}}{2}$$

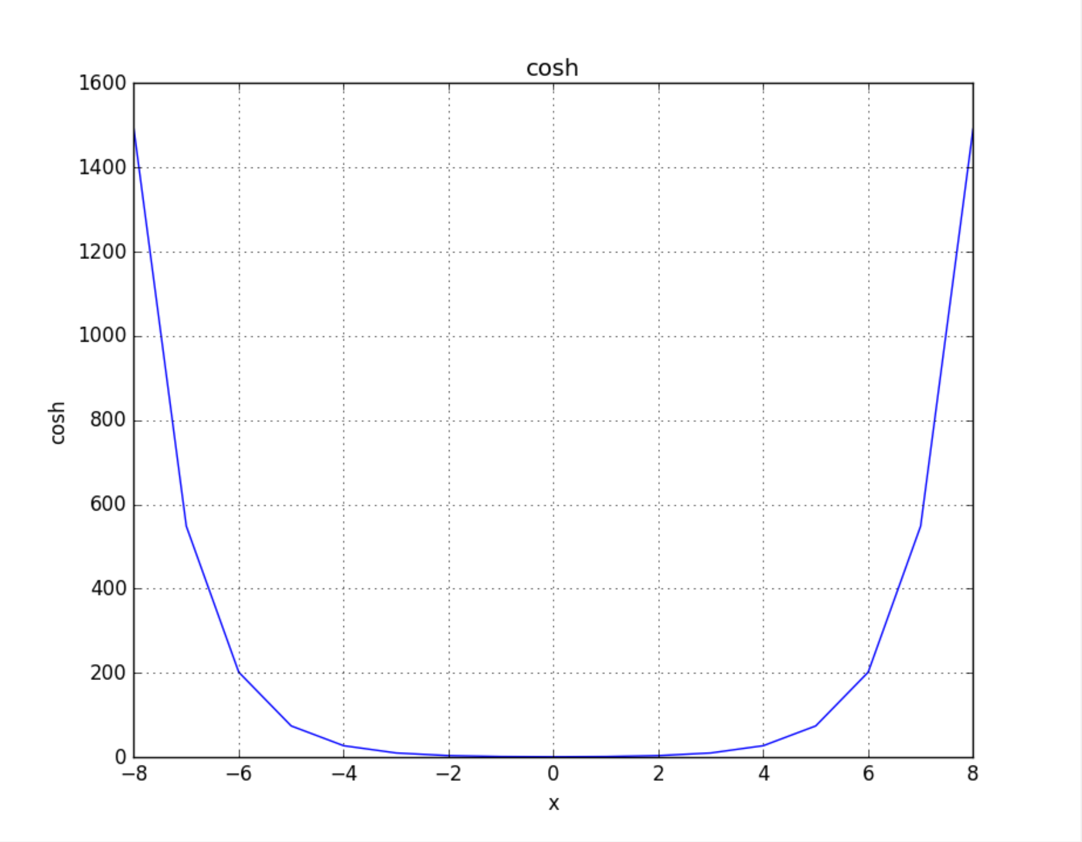
$$\frac{e^x + e^{-x}}{2}$$

```
x=[-8: 8]
y1=exp(x)/2
y2=exp(-x)/2
```

```
y4=y1+y2
plot(x,y4)
```

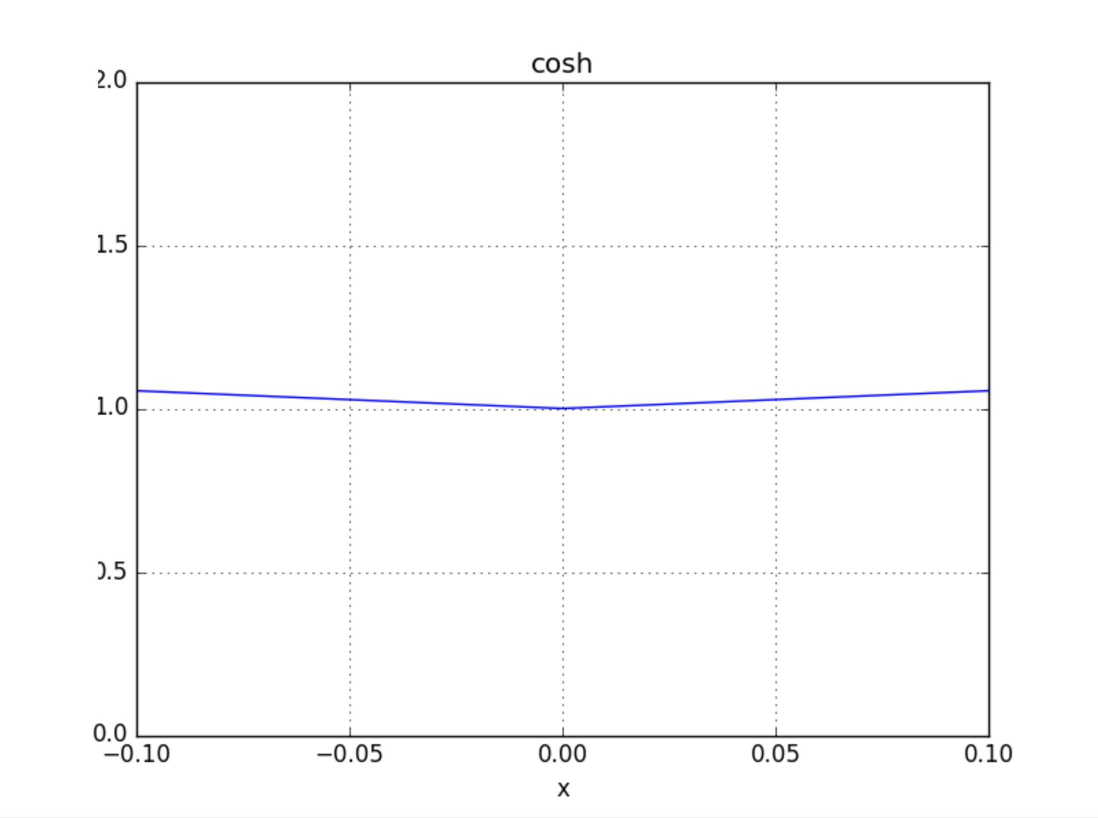
Plot cosh()

```
import numpy as np
import matplotlib.pyplot as
plt
x=np.arange(-8,8+1,1)
y1=np.exp(x)/2
y2=np.exp(-x)/2
y4=y1+y2
plt.figure(1)
plt.plot(x, y4)
plt.xlabel('x')
plt.ylabel('cosh')
plt.title('cosh')
plt.grid(True)
plt.show()
```



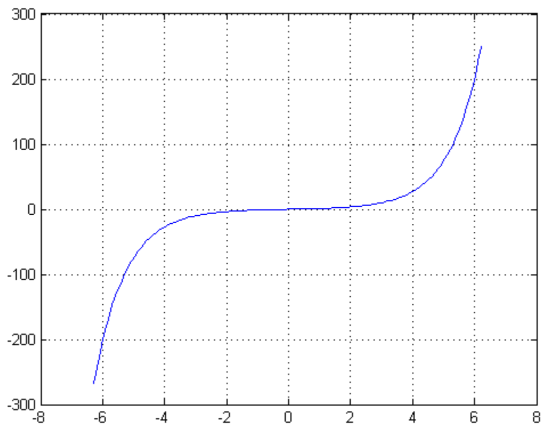
xlim & ylim

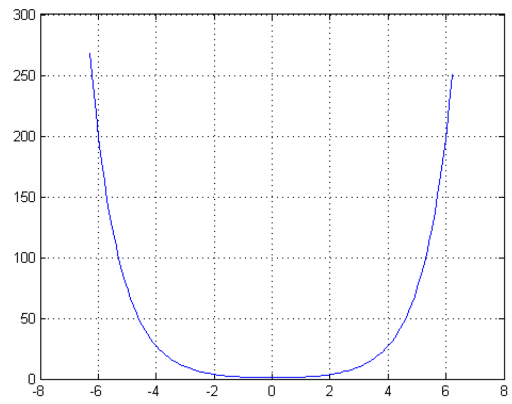
```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-8,8+1,1)
y1=np.exp(x)/2
y2=np.exp(-x)/2
y4=y1+y2
plt.figure(1)
plt.plot(x, y4)
plt.xlabel('x')
plt.ylabel('cosh')
plt.title('cosh')
plt.grid(True)
plt.xlim(-0.1, 0.1)
plt.ylim(0, 2)
plt.show()
```

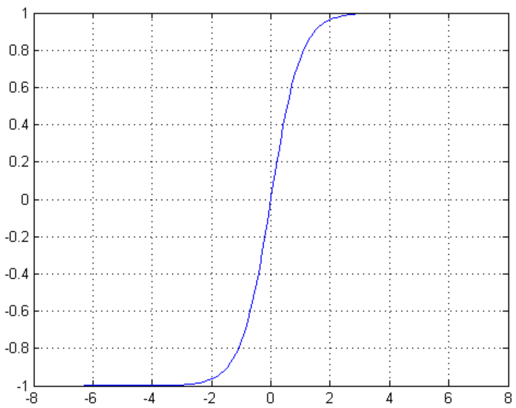


쌍곡선 함수와 지수 함수의 관계

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{\frac{e^x - e^{-x}}{2}}{\frac{e^x + e^{-x}}{2}} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$





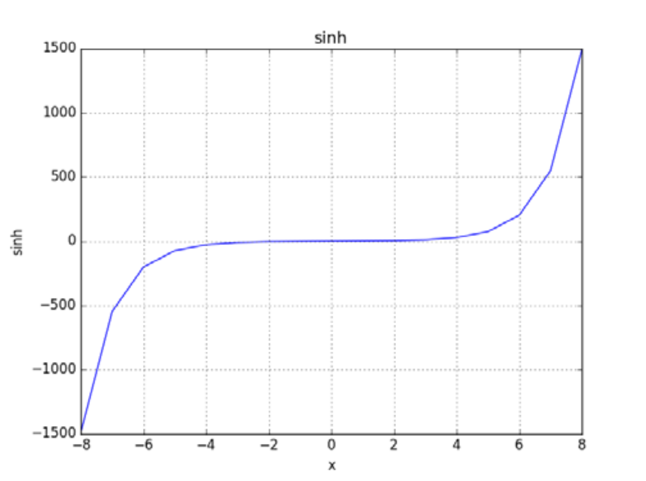


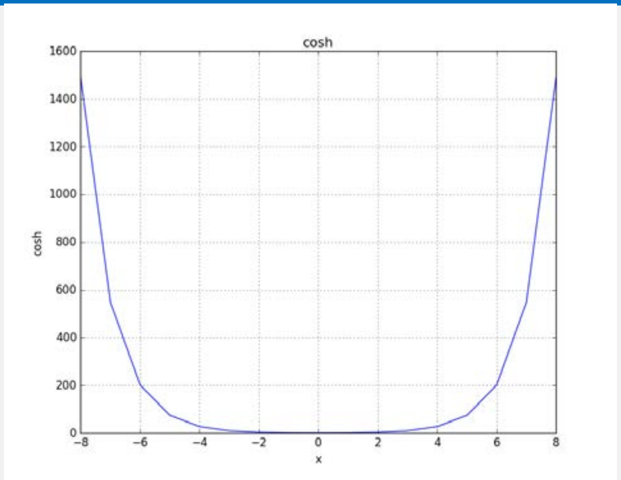
$$\frac{e^x - e^{-x}}{2}$$

$$\frac{e^x + e^{-x}}{2}$$

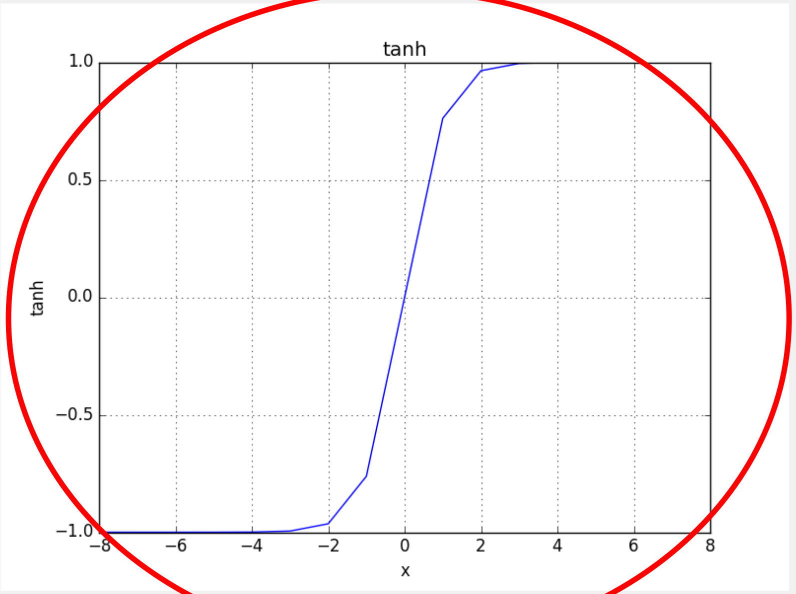
$$\frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Draw tanh



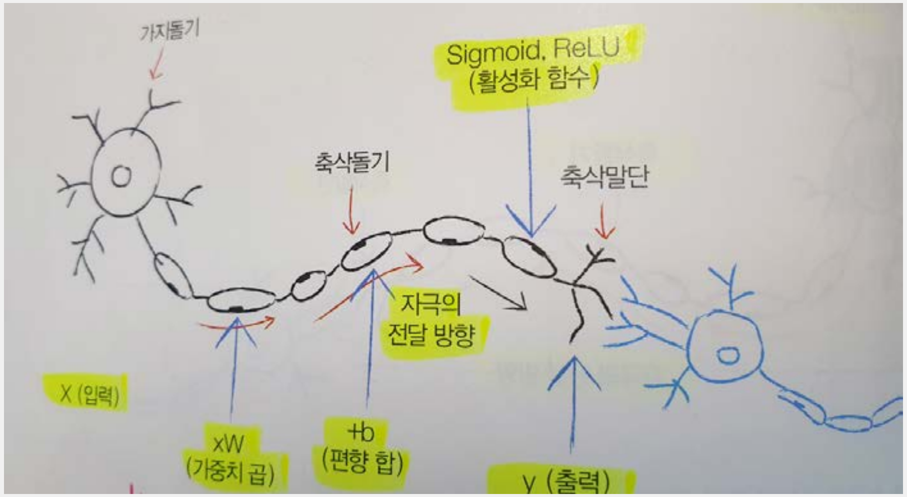



```python
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-8,8+1,1)
y1=np.exp(x)/2
y2=np.exp(-x)/2
y3=y1-y2
y4=y1+y2
y5=y3/y4
plt.figure(1)
plt.plot(x, y5)
plt.xlabel('x')
plt.ylabel('tanh')
plt.title('tanh')
plt.grid(True)
plt.show()
```



실제 뉴런과 인공 뉴런의 동작원리

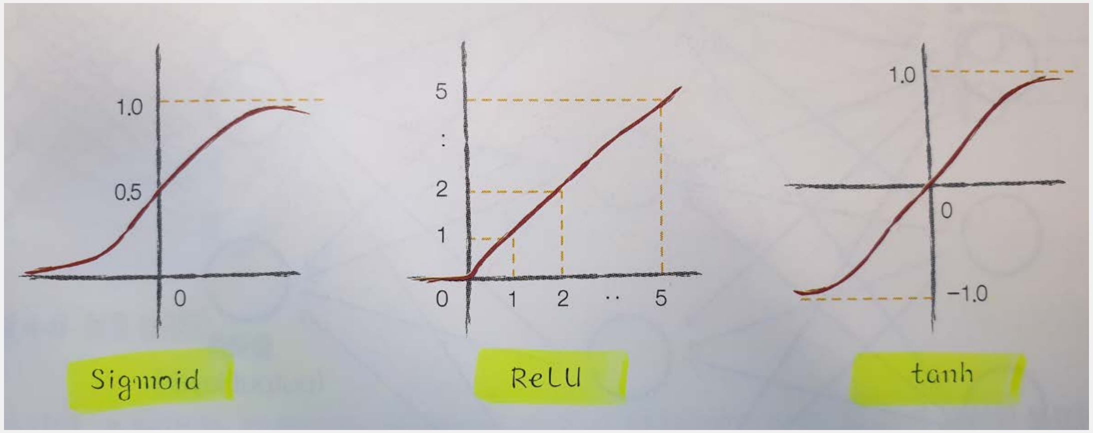
- 가지돌기에서 신호를 받아들이고, 이 신호가 축삭돌기를 지나, 축삭단말로 전달된다. 축삭돌기를 지나는 동안 신호가 약해지거나, 강하게 전달되기도 한다. 축삭돌기까지 전달된 신호는 다음 뉴런의 가지돌기로 전달된다.
- 인공 뉴런은 가지돌기에 해당하는 입력값 x부분, 축삭돌기에 해당하는 가중치(w)를 곱하고, 편향(b)를 더하고, 활성화 함수를 거치는 과정, 축삭단말에 해당하는 결과값 y 부분으로 구현된다.
- y=Sigmoid(x*w+b)



한빛미디어 3분 딥러닝

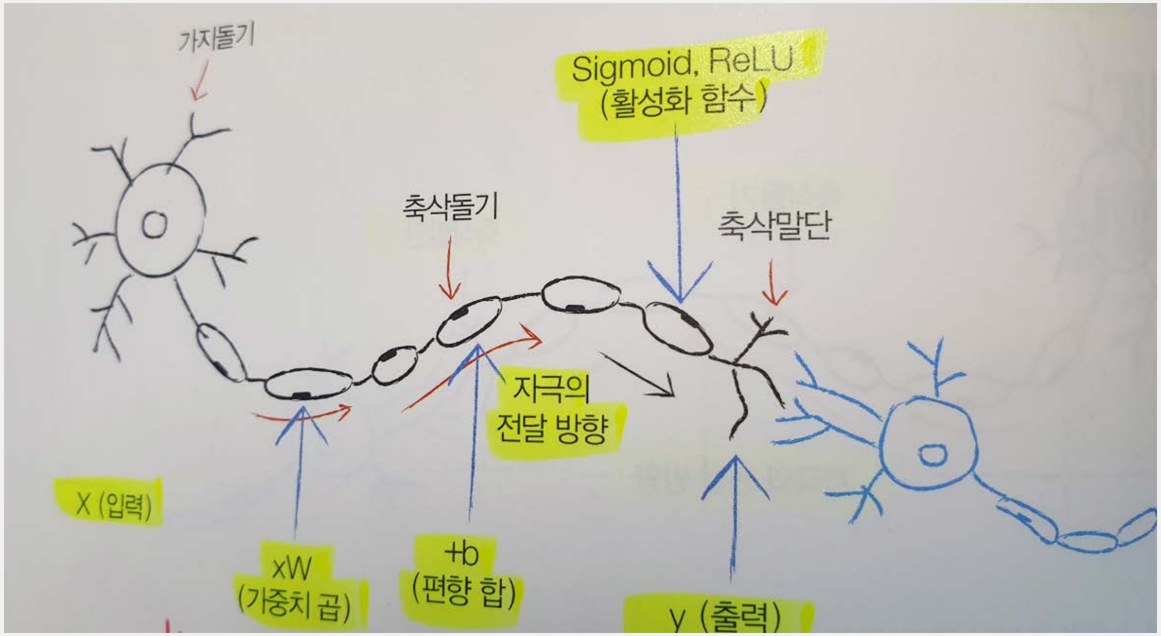
활성화 함수(activation function)와 tanh()

- y=Sigmoid(x*w+b)
  - y: 출력, Sigmoid: 활성화함수, x:입력, w:가중치, b: 편향
  - w:가중치, b: 편향을 찾아내는 것이 학습(Learning)
- 대표적인 활성화 함수는 Signoid (시그노이드), ReLU(렐루), tanh (쌍곡탄젠트) 함수가 있다
- 최근에는 ReLU가 많이 사용되면, 입력값이 0보다 작으면 항상 0을 0보다 크면, 입력값을 그대로 출력한다
- 학습목적에 따라 다른 활성화 함수를 사용



뉴런의 기본동작

- 입력값 x에 가중치(w)를 곱하고, 편향(b)를 더한 뒤 활성화 함수(Signoid (시그노이드), ReLU(렐루), tanh (쌍곡탄젠트))를 거쳐 결과값 y를 만들어 내는 것, 이것이 인공 뉴런의 기본이다.
- y=Sigmoid(x*w+b)



학습 또는 훈련

- 그리고 원하는 y값을 만들어내기 위해 w와 b의 값을 변경해가면서 적절한 값을 찾아내는 최적화 과정을 학습(learning) 또는 훈련(training)이라 한다.
- Signoid 그래프는 0과 1에 한없이 가까워진다.
- ReLU 그래프는 입력값이 0보다 작으면 항상 0을 0보다 크면, 입력값을 그대로 출력한다
- tanh 그래프는 1과 -1에 한없이 가까워진다

3주 / 2 차시
Part One Modeling, Computers, and Error Analysis,
교재 Ch.1: 번지 점프 시의 체감 속도 수식 유도 (Mathematical Deviation)

3주 2차시: 수학적인 모델링 기법을 통한 체감 속도 구하기

Falling Velocity of Bungee Jumper (Chapter 1)

Force(F) is composed of downward force due to gravity and upward force due to air resistance

$$F = F_d + F_u = mg - cdv^2$$

$$F_d = +mg \quad (downward \;\; force)$$

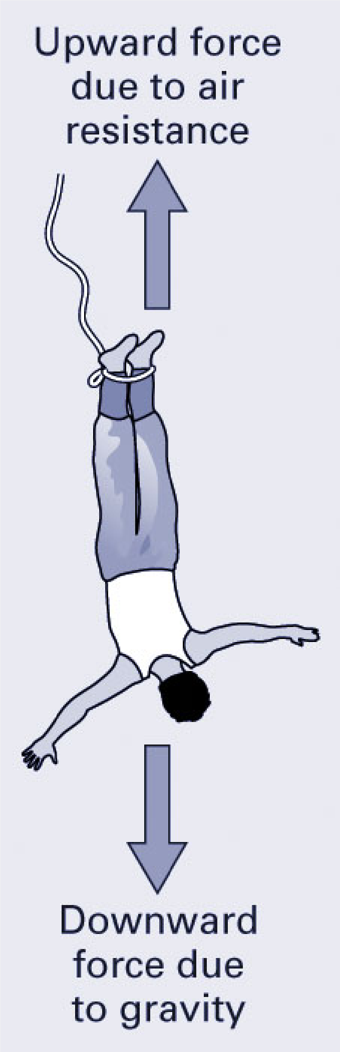
$$F_u = -cdv^2 \quad (upward \;\; force)$$

We can get

$$mg - c_d v^2 = m \cdot \frac{dv}{dt}$$

since velocity has the derivative based on time, $\frac{dv}{dt}$ is called as a $differential\ equation$

$$g - \frac{c_d}{m} v^2 = \frac{dv}{dt}$$



$$\frac{mg}{c_d} - v^2 = \frac{m}{c_d} \cdot \frac{dv}{dt}$$

put $\frac{mg}{c_d} = k^2$ $\quad therefore,\ k = \sqrt{\frac{mg}{c_d}}$

$$k^2 - v^2 = \frac{m}{c_d} \cdot \frac{dv}{dt}$$

When we take the inverse on both sides, we get $\frac{1}{k^2 - v^2} = \frac{1}{\frac{m}{c_d} \cdot \frac{dv}{dt}}$

https://en.wikipedia.org/wiki/Hyperbolic_function

$$\frac{c_d}{m} \cdot \frac{dt}{dv} = \frac{1}{k^2 - v^2}$$

$$\int \frac{du}{a^2 - u^2} = a^{-1} \tanh^{-1}\left(\frac{u}{a}\right) + C; u^2 < a^2$$

Take integration on both sides, we get $\int \frac{c_d}{m} \cdot dt = \int \frac{1}{k^2 - v^2} \cdot dv$

Use $\int \frac{1}{k^2 - v^2} \cdot dx = \frac{1}{k} tanh^{-1} \frac{v}{k} + c$ to the right side of equation,

Which comes $\int \frac{1}{a^2 - u^2} \cdot du = \frac{1}{a} tanh^{-1} \frac{u}{a} + c$ from the property of

Tan hyperbolic function

we get $\frac{c_d}{m} \cdot t + c_1 = \frac{1}{k} tanh^{-1} \frac{v}{k} + c_2$

$$\frac{c_d}{m} \cdot t = \frac{1}{k} tanh^{-1} \frac{v}{k} + c_2 - c_1$$

Since the velocity equals zero when the time is zero, we can say that

$t = 0 , v = 0$

$$0 = 0 + c_2 - c_1$$

$$c_2 = c_1$$

Since c_1 and c_2 are same, the c_1 and c_2 can be cancelled out.

$$\therefore \ \frac{c_d}{m} \cdot t + c_1 = \frac{1}{k} tanh^{-1} \frac{v}{k} + c_2$$

$$\therefore \ \frac{c_d}{m} \cdot t = \frac{1}{k} tanh^{-1} \frac{v}{k}$$

When k is moved to left side, we can get

$$\frac{kc_d}{m} \cdot t = tanh^{-1} \frac{v}{k}$$

tanh inverse function can be moved to the left side and the inverse is no more existed in the right side.

$$tanh\left(\frac{kc_d}{m} \cdot t\right) = \frac{v}{k}$$

$$v = k \cdot tanh\left(\frac{kc_d}{m} \cdot t\right)$$

Falling Velocity by Differential Equation

$v = k \cdot tanh\left(\frac{kc_d}{m} \cdot t\right)$, since $k = \sqrt{\frac{mg}{c_d}}$

We can get

$$v = \sqrt{\frac{mg}{c_d}}\ tanh\left(\sqrt{\frac{mg}{c_d}} \cdot \frac{c_d}{m} \cdot t\right)$$

$$= \sqrt{\frac{mg}{c_d}}\ tanh\left(\frac{\sqrt{g}}{\sqrt{c_d}} \frac{\sqrt{c_d}}{\sqrt{m}} \cdot t\right)$$

$$= \sqrt{\frac{mg}{c_d}}\ tanh\left(\sqrt{\frac{gc_d}{m}} \cdot t\right)$$

Therefore, we get the falling velocity $v(t)$, where v is dependent on time t.

$$\boxed{v(t) = \sqrt{\frac{mg}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} \cdot t\right)}$$

Falling Velocity by Euler's method

$$g - \frac{c_d}{m}v^2 = \frac{dv}{dt} \cong \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}$$

$$\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} = g - \frac{c_d}{m}v(t_i)^2$$

$$\boxed{v(t_{i+1}) = v(t_i) + \left[g - \frac{c_d}{m}v(t_i)^2\right](t_{i+1} - t_i)}$$

Falling Velocity by Euler's method

```
g = 9.8
cd = 0.25
m = 68

% euler
's method
v0 = 0
v1 = (1 - 0) * (g - cd / m * v0 ^ 2) + v0
v2 = (2 - 1) * (g - cd / m * v1 ^ 2) + v1
v3 = (3 - 2) * (g - cd / m * v2 ^ 2) + v2
v4 = (4 - 3) * (g - cd / m * v3 ^ 2) + v3

vel_eulor = [v0 v1 v2 v3 v4]
```

Original method (Falling Velocity by Differential Equation)

```
% original method
vel = [];
for (t=0:4)
v = sqrt(g * m / cd) * tanh(sqrt(g * cd / m) * t)
vel = [vel v]
end

time = 0:4

plot(time, vel, '-ro', time, vel_eulor, '-b>')
grid on

legend('original', 'eulor')
```