

G V V Sharma\*

## CONTENTS

1	<b>Line</b>	1
2	<b>Medians of a Triangle</b>	3
3	<b>Altitudes of a Triangle</b>	4
4	<b>Angle Bisectors of a Triangle</b>	4
5	<b>Circle</b>	6
6	<b>Tangent and Derivative</b>	6
7	<b>Conic Sections</b>	8
8	<b>Area Within a Parabola</b>	10
8.1	Arithmetic Progression . . .	10
8.2	Geometric Progression . . .	11

*Abstract*—This manual shows how to generate figures encountered in high school geometry using python. The process provides simple applications of coordinate geometry.

## 1 LINE

**Problem 1.** Let

$$A = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, C = \begin{pmatrix} 4 \\ -1 \end{pmatrix}. \quad (1)$$

Draw  $\triangle ABC$ .

**Solution:** The following code yields the desired plot in Fig. 1

```
#This program draws the triangle
ABC
import numpy as np
import matplotlib.pyplot as plt
```

\*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

```
def slope_coeff(A,B):
    p = np.zeros((2,1))
    p[0] = (A[1]-B[1])/(A[0]-B
    [0])
    p[1] = (A[0]*B[1]-A[1]*B
    [0])/(A[0]-B[0])
    return p

A = np.matrix('-2;-2')
B = np.matrix('1;3')
C = np.matrix('4;-1')

x = np.linspace(np.asscalar(A[0]),
np.asscalar(B[0]),50)
p = slope_coeff(A,B)
y = p[0]*x + p[1]
plt.plot(x,y, label='$5x-3y+4=0$')

plt.plot(A[0], A[1], 'o')
plt.text(A[0] * (1 + 0.1), A[1] *
(1 - 0.1), 'A')

x = np.linspace(np.asscalar(B[0]),
np.asscalar(C[0]),50)
p = slope_coeff(B,C)
y = p[0]*x + p[1]
plt.plot(x,y, label='$4x+3y-13=0$')
plt.plot(B[0], B[1], 'o')
plt.text(B[0] * (1 - 0.2), B[1] *
(1), 'B')

x = np.linspace(np.asscalar(C[0]),
np.asscalar(A[0]),50)
p = slope_coeff(C,A)
y = p[0]*x + p[1]
plt.plot(x,y, label='$x-6y-10=0$')
plt.plot(C[0], C[1], 'o')
plt.text(C[0] * (1 + 0.03), C[1] *
```

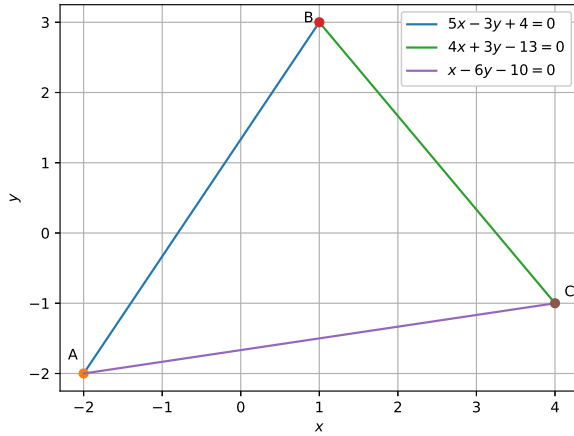


Fig. 1

```
(1 - 0.1) , 'C')
```

```
plt.grid()
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
#plt.savefig('../figs/triangle.eps')
plt.show()
```

**Problem 2.** Consider the line AB with

$$A = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad (2)$$

If AB is expressed by the equation

$$p_0x + p_1y + p_2 = 0, \quad (3)$$

find  $p_0, p_1$  and  $p_2$ .

**Solution:** Let

$$A = \begin{pmatrix} A_0 \\ A_1 \end{pmatrix}, B = \begin{pmatrix} B_0 \\ B_1 \end{pmatrix}, \quad (4)$$

The equation of AB is given by

$$\frac{y - A_1}{x - A_0} = \frac{A_1 - B_1}{A_0 - B_0} \quad (5)$$

$$\implies (A_1 - B_1)x + (B_0 - A_0)y + A_0B_1 - A_1B_0 = 0 \quad (6)$$

after some algebra. Thus,

$$p_0 = A_1 - B_1, p_1 = B_0 - A_0, p_2 = A_0B_1 - A_1B_0. \quad (7)$$

The following python code computes the numerical values and the equation for AB is

$$5x - 3y + 4 = 0 \quad (8)$$

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

```
A = np.matrix(' -2; -2 ')
B = np.matrix(' 1; 3 ')
```

```
p = np.zeros((3,1))
p[0] = A[1]-B[1]
p[1] = B[0]-A[0]
p[2] = A[0]*B[1]- A[1]*B[0]
```

```
print (p)
```

**Problem 3.** Let

$$C = \begin{pmatrix} 4 \\ -1 \end{pmatrix}. \quad (9)$$

Find the equations of BC and CA

**Solution:** The following code yields the coefficients resulting in the respective equations

$$4x + 3y - 13 = 0 \quad (10)$$

$$x - 6y - 10 = 0 \quad (11)$$

```
#This program calculates the
equations of
#all the sides of the triangle
import numpy as np
import matplotlib.pyplot as plt
```

```
def line_coeff(A,B):
    p = np.zeros((3,1))
    p[0] = A[1]-B[1]
    p[1] = B[0]-A[0]
    p[2] = A[0]*B[1]- A[1]*B
    [0]
    return p
```

```
A = np.matrix(' -2; -2 ')
B = np.matrix(' 1; 3 ')
C = np.matrix(' 4; -1 ')
```

```
print (line_coeff(A,B))
print (line_coeff(B,C))
print (line_coeff(C,A))
```

**Problem 4.** An alternative equation of the line  $AB$  is

$$y = p_0x + p_1 \quad (12)$$

Find the equations of  $AB, BC$  and  $CA$ .

**Solution:** From (5),

$$p_0 = \frac{A_1 - B_1}{A_0 - B_0} \quad (13)$$

$$\begin{aligned} p_1 &= A_1 - A_0 \frac{A_1 - B_1}{A_0 - B_0} \\ &= \frac{A_0 B_1 - A_1 B_0}{A_0 - B_0} \end{aligned} \quad (14)$$

The following python code calculates the above coefficients resulting in the equations for  $AB, BC, CA$

$$y = 1.67x + 1.33 \quad (15)$$

$$y = -1.33x + 4.33 \quad (16)$$

$$y = 0.16x - 1.67 \quad (17)$$

**Problem 5.** Draw the lines  $AB, BC$  and  $CA$ .

## 2 MEDIANS OF A TRIANGLE

**Problem 6.** Find the coordinates of  $D, E$  and  $F$  of the mid points of  $AB, BC$  and  $CA$  respectively for  $\triangle ABC$ .

**Solution:** The coordinates of the mid points are given by

$$D = \frac{B+C}{2}, E = \frac{C+A}{2}, F = \frac{A+B}{2} \quad (18)$$

The following code computes the values resulting in

$$D = \begin{pmatrix} 2.5 \\ 1 \end{pmatrix}, E = \begin{pmatrix} 1 \\ -1.5 \end{pmatrix}, F = \begin{pmatrix} -0.5 \\ 0.5 \end{pmatrix}, \quad (19)$$

```
#This program calculates the mid
point between
#any two coordinates
import numpy as np
import matplotlib.pyplot as plt

def mid_pt(B,C):
    D = (B+C)/2
```

```
return D
```

```
A = np. matrix(' -2; -2 ')
B = np. matrix(' 1; 3 ')
C = np. matrix(' 4; -1 ')

print (mid_pt(B,C))
print (mid_pt(C,A))
print (mid_pt(A,B))
```

**Problem 7.** Find the equations of  $AD, BE$  and  $CF$ . These lines are the medians of  $\triangle ABC$

**Solution:** Using the code in Problem 2 and simplifying, the respective equations are

$$2x - 3y - 2 = 0 \quad (20)$$

$$x - 1 = 0 \quad (21)$$

$$x + 3y - 1 = 0 \quad (22)$$

**Problem 8.** Find the point of intersection of  $AD$  and  $CF$ .

**Solution:** Let the respective equations be

$$p_0x + p_1 + p_2 = 0 \quad (23)$$

$$q_0x + q_1 + q_2 = 0 \quad (24)$$

This can be written as the matrix equation

$$\begin{pmatrix} p_0 & p_1 \\ q_0 & q_1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = - \begin{pmatrix} p_2 \\ q_2 \end{pmatrix} \quad (25)$$

The following code yields the point of intersection

$$G = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (26)$$

```
#This program calculates the
#intersection of two lines
#given their equations
import numpy as np
import matplotlib.pyplot as plt

def line_intersect(p,q):
    P = np.column_stack((p
    [0:2],q[0:2])).transpose
    ()
    c = -np.row_stack((p[2],q
    [2]))
    return np.matmul(np.linalg
    .inv(P),c)
```

```
p = np.matrix('2;-3;-2')
q = np.matrix('1;3;-1')
r = np.matrix('1;0;-1')

print(line_intersect(p,r))
```

**Problem 9.** Using the code in Problem 8, verify that  $G$  is the point of intersection of  $BE, CF$  as well as  $AD, BE$ .  $G$  is known as the centroid of  $\triangle ABC$ .

**Problem 10.** Graphically show that the medians of  $\triangle ABC$  meet at the centroid.

**Problem 11.** Verify that

$$G = \frac{A + B + C}{3} \quad (27)$$

### 3 ALTITUDES OF A TRIANGLE

**Definition 12.** In  $\triangle ABC$ , Let  $P$  be a point on  $BC$  such that  $AP \perp BC$ . Then  $AP$  is defined to be an altitude of  $\triangle ABC$ .

**Problem 13.** Find the equation of  $AP$ .

**Solution:** Let the equation for  $BC$  and  $AP$  be

$$y = p_0x + p_1 \quad (28)$$

$$y = q_0x + q_1 \quad (29)$$

respectively. Since,  $AP \perp BC$ ,

$$p_0q_0 = -1 \quad (30)$$

The equation for  $AP$  is then obtained as

$$y - A_1 = q_0(x - A_0) \quad (31)$$

$$\implies y = q_0x + A_1 - q_0A_0 \quad (32)$$

From the following python code,  $AP$  can be expressed as

$$y = 0.75x - 0.5 \quad (33)$$

```
#This program calculates the
#equation of the altitude
import numpy as np
import matplotlib.pyplot as plt

def slope_coeff(A,B):
    p = np.zeros((2,1))
    p[0] = (A[1]-B[1])/(A[0]-B
    [0])
    p[1] = (A[0]*B[1]-A[1]*B
    [0])/(A[0]-B[0])
```

```
return p

def alt_coeff(p,A):
    q = np.zeros((2,1))
    q[0] = -1/p[0]
    q[1] = A[1] - q[0]*A[0]
    return q
```

```
A = np.matrix('-2;-2')
B = np.matrix('1;3')
C = np.matrix('4;-1')
```

```
p = slope_coeff(B,C)
q = alt_coeff(p,A)
print(q)
```

**Problem 14.** Find the equations of the altitudes  $BQ$  and  $CR$ .

**Solution:** Using the code in Problem 13, the respective equations are

$$y = -6x + 9 \quad (34)$$

$$y = -0.6x + 1.4 \quad (35)$$

**Problem 15.** Find the point of intersection of  $AP$  and  $BQ$ .

**Solution:** Using the code in Problem 8, the desired point of intersection is

$$H = \begin{pmatrix} 1.407 \\ 0.56 \end{pmatrix} \quad (36)$$

Interestingly,  $BQ$  and  $CR$  also intersect at the same point. Thus, the altitudes of a triangle meet at a single point known as the *orthocentre*

**Problem 16.** Find  $P, Q, R$ .

**Solution:**  $P$  is the intersection of  $AP$  and  $BC$ . Thus, the code in Problem 8 can be used to find  $P$ . The desired coordinates are

$$P = \begin{pmatrix} 2.32 \\ 1.24 \end{pmatrix}, Q = \begin{pmatrix} 1.73 \\ -1.38 \end{pmatrix}, R = \begin{pmatrix} 0.03 \\ 1.38 \end{pmatrix} \quad (37)$$

**Problem 17.** Draw  $AP, BQ$  and  $CR$  and verify that they meet at  $H$ .

### 4 ANGLE BISECTORS OF A TRIANGLE

**Definition 18.** In  $\triangle ABC$ , let  $U$  be a point on  $BC$  such that  $\angle BAU = \angle CAU$ . Then  $AU$  is known as the angle bisector.

**Problem 19.** Find the length of  $AB$ ,  $BC$  and  $CA$

**Solution:** The length of  $CA$  is given by

$$CA = \sqrt{(C_0 - A_0)^2 + (C_1 - A_1)^2} \quad (38)$$

The following code calculates the respective values as

$$AB = 5.83, BC = 5, CA = 6.08 \quad (39)$$

```
#This program calculates the
  distance between
#two points
import numpy as np
import matplotlib.pyplot as plt

def side_length(A,B):
    return np.sqrt((A[0]-B[0])
        **2 + (A[1]-B[1])**2)

A = np. matrix(' -2; -2 ')
B = np. matrix(' 1; 3 ')
C = np. matrix(' 4; -1 ')

print (side_length(A,B))
print (side_length(B,C))
print (side_length(C,A))
```

**Problem 20.** If  $AU$ ,  $BV$  and  $CW$  are the angle bisectors, find the coordinates of  $U$ ,  $V$  and  $W$ .

**Solution:** Using the section formula,

$$W = \frac{AW.B + WB.A}{AW + WB} = \frac{\frac{AW}{WB}.B + A}{\frac{AW}{WB} + 1} \quad (40)$$

$$= \frac{\frac{CA}{BC}.B + A}{\frac{CA}{BC} + 1} \quad (41)$$

$$= \frac{CA \times B + BC \times A}{BC + CA} \quad (42)$$

since the angle bisector has the property that

$$\frac{AW}{WB} = \frac{CA}{AB} \quad (43)$$

The following code computes the coordinates as

$$U = \begin{pmatrix} 2.47 \\ 1.04 \end{pmatrix}, V = \begin{pmatrix} 1.23 \\ -1.46 \end{pmatrix} \approx \begin{pmatrix} -0.35 \\ 0.75 \end{pmatrix} \quad (44)$$

```
#This program calculates point
#where the angle bisector meets
the
```

```
#opposite side
```

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

```
def side_length(A,B):
    return np.sqrt((A[0]-B[0])
        **2 + (A[1]-B[1])**2)
```

```
def angle_bisect_coord(b,c,B,C):
    return np.multiply((np.
        multiply(b,B)+np.
        multiply(c,C)),1/(b+c))
```

```
A = np. matrix(' -2; -2 ')
B = np. matrix(' 1; 3 ')
C = np. matrix(' 4; -1 ')

a = side_length(B,C)
b = side_length(C,A)
c = side_length(A,B)
```

```
U = angle_bisect_coord(b,c,B,C)
V = angle_bisect_coord(c,a,C,A)
W = angle_bisect_coord(a,b,A,B)
```

```
print (U)
print (V)
print (W)
```

**Problem 21.** Find the intersection of  $AU$  and  $BV$ .

**Solution:** Using the code in Problem 8, the desired point of intersection is

$$I = \begin{pmatrix} 1.15 \\ 0.14 \end{pmatrix} \quad (45)$$

It is easy to verify that even  $BV$  and  $CW$  meet at the same point.  $I$  is known as the *incentre* of  $\Delta ABC$ .

**Problem 22.** Draw  $AU$ ,  $BV$  and  $CW$  and verify that they meet at a point  $I$ .

**Problem 23.** Verify that

$$I = \frac{BC.A + CA.B + AB.C}{AB + BC + CA} \quad (46)$$

**Problem 24.** Let the perpendiculars from  $I$  to  $AB$ ,  $BC$  and  $CA$  be  $IX$ ,  $IY$ ,  $IZ$ . Verify that

$$IX = IY = IZ = r \quad (47)$$

$r$  is known as the inradius of  $\Delta ABC$ .

**Solution:** The distance of a point  $(a, b)$  from the line  $p_0x + p_1y + p_2 = 0$  is given by

$$\frac{|ap_0 + bp_1 + p_2|}{\sqrt{p_0^2 + p_1^2}} \quad (48)$$

The following code computes IX.

```
#This program calculates the
  inradius

import numpy as np
import matplotlib.pyplot as plt

def line_dist(I,p):
    return np.abs((I[0]*p[0]+I
        [1]*p[1]+p[2])/(np.sqrt(
            p[0]**2+p[1]**2)))

I = np. matrix('1.15;0.14')
A = np. matrix('-2;-2')
B = np. matrix('1;3')
C = np. matrix('4;-1')
AB = line_coeff(A,B)
BC = line_coeff(B,C)
CA = line_coeff(C,A)

print(line_dist(I,AB))
```

## 5 CIRCLE

**Definition 25.** From Problem 24, it is obvious that a circle with centre at  $I$  and radius  $r$  passing through  $X, Y, Z$  can be drawn. The incircle is defined as the circle with centre at the incentre  $I$  and radius equal to the inradius.

**Problem 26.** Obtain the equation of the incircle of  $\Delta ABC$  and draw it.

**Solution:** Letting  $I = (a, b)$ , the equation of the incircle is given by

$$(x - a)^2 + (y - b)^2 = r^2, \quad (49)$$

where  $r$  is the inradius. The following code plots this circle in Fig. 26

```
#This program plots the incircle
import numpy as np
import matplotlib.pyplot as plt
```

```
r=1.6
a=1.15
b=0.14
x=np.linspace(a-r, a+r, 1000)
y1=b+np.sqrt((r)**2-((x-a)**2))
y2=b-np.sqrt((r)**2-((x-a)**2))
plt.plot(x,y1)
plt.plot(x,y2)
plt.grid()
plt.axis("equal")
#plt.savefig('../figs/incircle.eps')
plt.show()
```

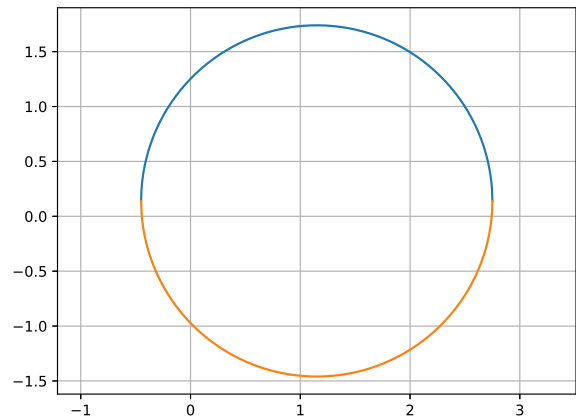


Fig. 26

## 6 TANGENT AND DERIVATIVE

**Definition 27.** A line that meets the circle at exactly one point is known as a tangent to the circle.

**Problem 28.** Draw  $\Delta ABC$  and its incircle in the same graph and verify that the lines  $AB, BC, CA$  are tangents to the incircle

**Solution:** Fig. 28 can be drawn using the codes in Problems 1 and 26. It is obvious from the figure that  $AB, BC$  and  $CA$  are tangents to the incircle.

**Problem 29.** Let the equation of  $AB$  be

$$p_0x + p_1y + p_2 = 0 \quad (50)$$

and the incircle be

$$(x - a)^2 + (y - b)^2 = r^2, \quad (51)$$

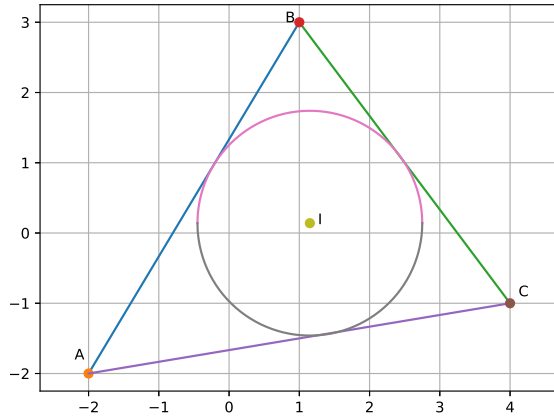


Fig. 28

Verify that

$$\begin{aligned} & \left[ p_0(p_2 + bp_1) - ap_1^2 \right]^2 \\ &= (p_0^2 + p_1^2) \left[ p_1^2 a^2 + (p_2 + bp_1)^2 - p_1^2 r^2 \right] \end{aligned} \quad (52)$$

and the point of contact

$$X = \begin{pmatrix} \frac{ap_1^2 - p_0(p_2 + bp_1)}{p_0^2 + p_1^2} \\ -\frac{p_2}{p_1} + \frac{p_0}{p_1} \frac{ap_1^2 - p_0(p_2 + bp_1)}{p_0^2 + p_1^2} \end{pmatrix} \quad (53)$$

**Solution:** The following code computes the point of contact

```
#This program computes the point
of contact between a circle
#and its tangent

import numpy as np
import matplotlib.pyplot as plt

def line_coeff(A,B):
    p = np.zeros((3,1))
    p[0] = A[1]-B[1]
    p[1] = B[0]-A[0]
    p[2] = A[0]*B[1]- A[1]*B
    [0]
    return p

A = np.matrix(' -2; -2 ')
B = np.matrix(' 1; 3 ')
p = line_coeff(A,B)
r=1.6
a=1.15
```

```
b=0.14
```

```
print((p[0]*(p[2]+b*p[1])-a*p
[1]**2)**2)
print((p[0]**2+p[1]**2)*(p[1]**2*a
**2+(p[2]+b*p[1])**2-p[1]**2*r
**2))
```

```
X = np.zeros((2,1))
X[0] = -(p[0]*(p[2]+b*p[1])-a*p
[1]**2)/(p[0]**2+p[1]**2)
X[1] = -(p[2]+p[0]*X[0])/p[1]
print(X)
```

**Problem 30.** Verify that

$$AX = AZ \quad (54)$$

$$BX = BY \quad (55)$$

$$CY = CZ \quad (56)$$

**Problem 31.** Devise a method for calculating the slope of the tangent at  $X$ , given the equation of the circle and the point  $X$ .

**Solution:** In Fig. 31, it can be seen that the tangent at  $X$  has the same slope as the chord  $BC$ . From the equation of the circle,

$$(p_1 - a)^2 + (q_1 - b)^2 = r^2 \quad (57)$$

$$(p_2 - a)^2 + (q_2 - b)^2 = r^2 \quad (58)$$

which, after simplification, leads to the slope of the chord as

$$\frac{q_1 - q_2}{p_1 - p_2} = -\frac{p_1 + p_2 - 2a}{q_1 + q_2 - 2b} \quad (59)$$

$$\Rightarrow \frac{\Delta y}{\Delta x} = -\frac{p_1 + p_2 - 2a}{q_1 + q_2 - 2b} \quad (60)$$

If we keep choosing smaller chords parallel to  $BC$ ,  $p_1$  and  $p_2$  come closer while  $q_1$  and  $q_2$  come closer, without any change in the slope on the LHS. The limiting behaviour results in  $p_1 = p_2 = p$  and  $q_1 = q_2 = q$ . This results in an expression for the slope of the tangent at  $X$

$$\frac{dy}{dx} = -\frac{p - a}{q - b} \quad (61)$$

where

$$\lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \frac{dy}{dx} \quad (62)$$

## 7 CONIC SECTIONS

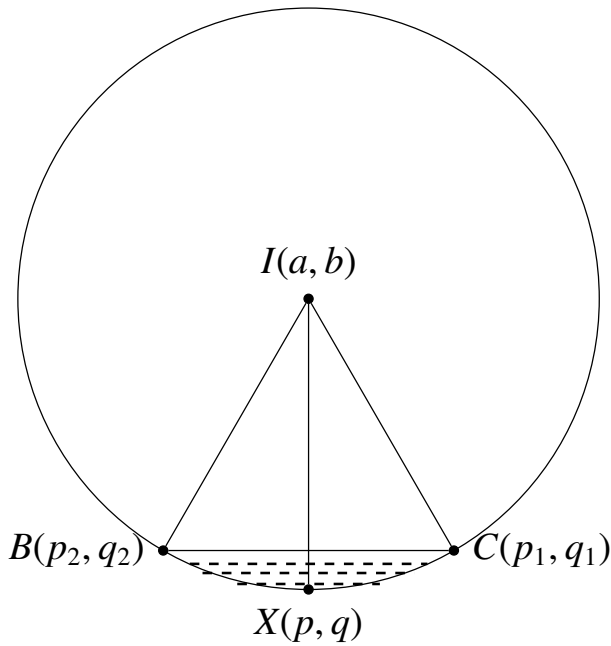


Fig. 31: Notion of the derivative.

**Problem 32.** Verify that the derivative of the circle at  $X$  is actually the slope of  $AB$ .

**Solution:** The verification is done by the following program.

```
import numpy as np

def slope_coeff(A,B):
    p = np.zeros((2,1))
    p[0] = (A[1]-B[1])/(A[0]-B[0])
    p[1] = (A[0]*B[1]-A[1]*B[0])/(A[0]-B[0])
    return p

A = np.matrix(' -2; -2')
B = np.matrix(' 1; 3')
p = slope_coeff(A,B)
print(p[0])
X=np.matrix(' -0.22;0.96')
I=np.matrix(' 1.15;0.14')
print(-(X[0]-I[0])/(X[1]-I[1]))

#C=np.matrix(' 2.43;1.09')
```

**Problem 33.** Plot the circle

$$x^2 + y^2 = 1 \quad (63)$$

**Solution:**

```
#This program draws the unit
circle
import numpy as np
import matplotlib.pyplot as plt

r = 1
theta = np.linspace(-np.pi,np.pi
,50)
x = r*np.cos(theta)
y = r*np.sin(theta)

plt.plot(x,y)
plt.grid()
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.axis('equal')
plt.show()
```

**Problem 34.** Show that (63) can be expressed as

$$(x \ y \ 1) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \quad (64)$$

**Problem 35.** Show that

$$(x \ y \ 1) M^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} M \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \quad (65)$$

for

$$M = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 2 & 1 & 1 \end{pmatrix} \quad (66)$$

can be expressed as

$$x^2 + 2xy + y^2 - 4x - 2y - 1 = 0 \quad (67)$$

**Problem 36.** Show that (67) results in the curve in Fig. 36. This is known as a parabola.

**Problem 37.** Show that using

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (68)$$



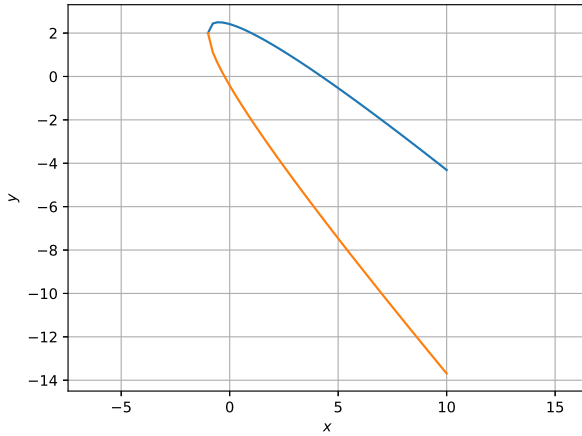


Fig. 36

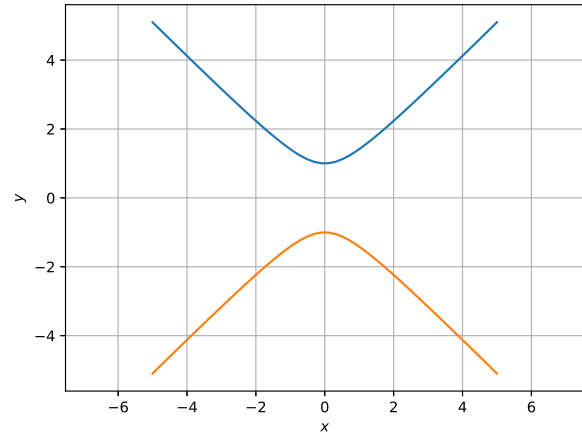


Fig. 38

in Problem 35 results in

$$y^2 - x^2 = 1 \quad (69)$$

**Problem 38.** Sketch (69) to obtain Fig. 38. This curve is known as a hyperbola

**Solution:**

```
#This program draws a hyperbola
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-5,5,50)
y1 = np.sqrt(1+x**2)
y2 = -np.sqrt(1+x**2)

plt.plot(x,y1,x,y2)
plt.grid()
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.axis('equal')
plt.savefig('../figs/circle_
hyperbola.eps')
plt.show()
```

**Problem 39.** Generate the points

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = M \begin{pmatrix} x \\ y \end{pmatrix} \quad (70)$$

where  $x, y$  are the points generated in Problem 33. Plot  $y_1$  with respect to  $x_1$ . The figure that you obtain in Fig. 39 is known as an ellipse.

**Solution:**

```
#This program draws the triangle
ABC
import numpy as np
import matplotlib.pyplot as plt

r = 1
theta = np.linspace(-np.pi,np.pi
,50)
x = r*np.cos(theta)
y = r*np.sin(theta)
X = np.row_stack((x,y))
M = np.matrix('3,0;0,2')
Y = M*X
x1 = np.array(Y)[0]
y1 = np.array(Y)[1]

plt.plot(x1,y1)
plt.grid()
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.axis('equal')
#plt.savefig('../figs/ellipse_
transform.eps')
plt.show()
```

**Problem 40.** Draw the curve

$$\frac{x^2}{9} + \frac{y^2}{4} = 1 \quad (71)$$

Comment.

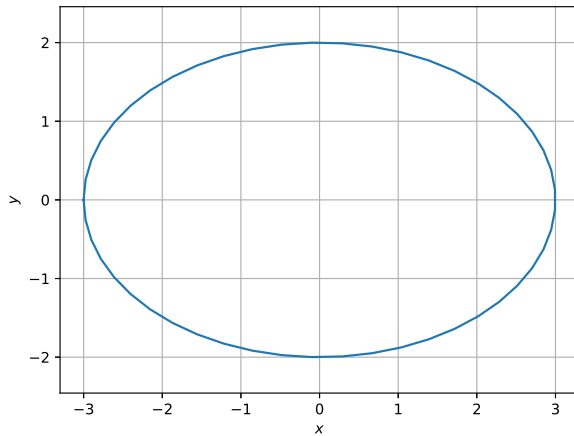


Fig. 39

## 8 AREA WITHIN A PARABOLA

**Problem 41.** Sketch the parabola

$$y^2 = x \quad (72)$$

**Problem 42.** Using  $n$  rectangles of equal width as shown in Fig. 42, find the limiting area of the parabola in  $x \in (0, 1)$ .

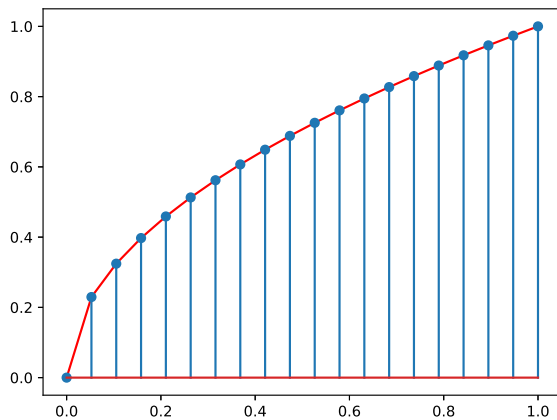


Fig. 42

**Solution:** Considering the width of the rectangle as  $h = \frac{1}{n}$ ,  $n = 100$ , the approximate area of the parabola can be computed as

$$A = h * (\sqrt{h} + \sqrt{2h} + \dots + \sqrt{100h}) \quad (73)$$

$$\approx 0.67 \quad (74)$$

using the following program

```
#Area under the parabola
import numpy as np
import matplotlib.pyplot as plt

n = 100
h = 1/n
x = np.linspace(1,n,n)
y = np.sqrt(h*x)
A = h*np.sum(y)
print(A)
```

## 8.1 Arithmetic Progression

**Problem 43.** Plot

$$y = x^2 \quad (75)$$

and verify that the area under this parabola for  $x \in (0, 1)$  is  $A_0 = 1 - A$ .

**Solution:** The following code

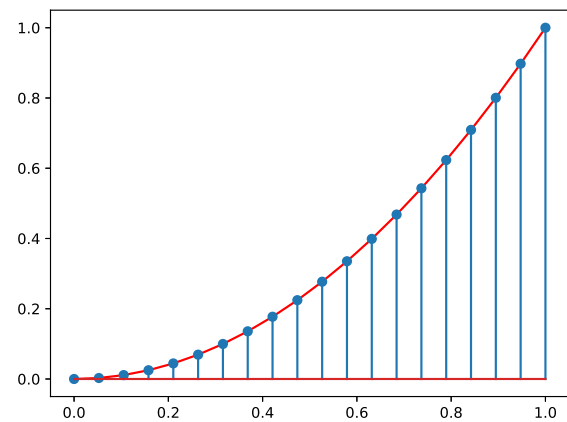


Fig. 43

```
#Area under the parabola
import numpy as np
import matplotlib.pyplot as plt

n = 100
h = 1/n
x = np.linspace(1,n,n)
y = (h*x)**2
A_1 = h*np.sum(y)
print(A_1)
```

yields the area in Fig.43 as  $A_0 \approx 0.33 = 1 - A = 1 - 0.67$ .

**Problem 44.** Show that the limiting area in Problem 43 is

$$A_0 = \lim_{n \rightarrow \infty} \left(\frac{1}{n}\right)^3 \sum_{k=1}^n k^2 = \frac{1}{3} \quad (76)$$

**Solution:** We have

$$k^3 - (k-1)^3 = 3k^2 - 3k + 1 \quad (77)$$

$$\Rightarrow n^3 = 3 \sum_{k=1}^n k^2 - 3 \sum_{k=1}^n k + n \quad (78)$$

$$\Rightarrow \sum_{k=1}^n k^2 = \frac{1}{3} \left[ n^3 + 3 \sum_{k=1}^n k - n \right] \quad (79)$$

Letting

$$S_n = 1 + 2 + \dots + n \quad (80)$$

$$S_n = n + n - 1 + \dots + 1 \quad (81)$$

$$\Rightarrow 2S_n = n(n+1) \quad (82)$$

$$\Rightarrow S_n = \frac{n(n+1)}{2} \quad (83)$$

Thus,

$$\sum_{k=1}^n k^2 = \frac{1}{3} \left[ n^3 + 3 \frac{n(n+1)}{2} - n \right] \quad (84)$$

$$= \frac{n}{6} [2n^2 + 3n + 1] \quad (85)$$

$$= \frac{n(n+1)(2n+1)}{6} \quad (86)$$

and

$$A_0 = \lim_{n \rightarrow \infty} \left(\frac{1}{n}\right)^3 \sum_{k=1}^n k^2 \quad (87)$$

$$= \lim_{n \rightarrow \infty} \frac{1}{6} \left(1 + \frac{1}{n}\right) \left(2 + \frac{1}{n}\right) \quad (88)$$

$$= \frac{1}{3} \quad (89)$$

This process of finding the area under a curve is known as *integration*. Integration is the opposite of differentiation. The sequence that is summed in  $S_n$  is known as an *Arithmetic Progression*.

## 8.2 Geometric Progression

**Problem 45.** Plot the parabola

$$y = x^2 \quad (90)$$

for  $x \in (0, 1)$  with points  $(r^k, 0)$ ,  $k = 0, 1, \dots, n$  for  $r = 0.8$ ,  $n = 10$ .

**Solution:** The following code

```
#Area under the parabola
import numpy as np
import matplotlib.pyplot as plt

n = 100
h = 1/n
x = np.linspace(1, n, n)
y = (h*x)**2
A_1 = h*np.sum(y)
print(A_1)
```

plots Fig. 45

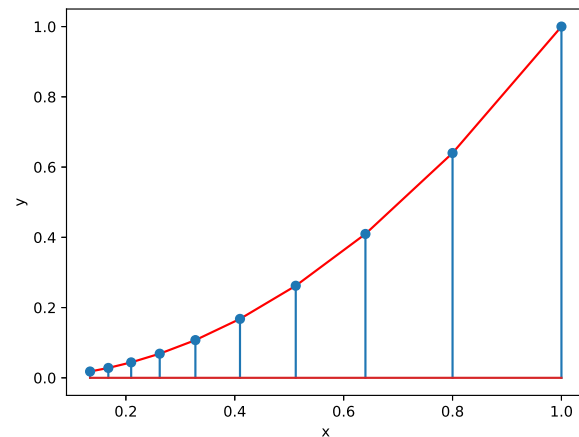


Fig. 45

**Problem 46.** Calculate the area of the parabola using Fig. 45 with  $n = 100$ ,  $r = 0.98$ .

**Solution:** The intervals are of width  $r^{k-1}(1-r)$ ,  $k = 1, \dots, n$ . The corresponding heights are  $r^{2k-2}$ . Thus, the area is

$$A_0 = \sum_{k=1}^n r^{k-1} (1-r) r^{2k-2} \quad (91)$$

$$= (1-r) \sum_{k=1}^n r^{3k-3} \quad (92)$$

The following code calculates the desired area as 0.33

```
#Area under the parabola
import numpy as np
import matplotlib.pyplot as plt
```

```

n = 100
r = 0.98
k = np.linspace(1, n, n)
y = r**(3*k-3)
A = (1-r)*np.sum(y)
print(A)

```

**Problem 47.** Obtain the area in the previous problem as the limit of a sum.

**Solution:** Let

$$A_0 = \lim_{\substack{r \rightarrow 1 \\ n \rightarrow \infty}} (1-r) \sum_{k=1}^n r^{3k-3}, \quad r < 1 \quad (93)$$

If  $p = r^3$ ,

$$S_n = \sum_{k=1}^n p^{k-1} \quad (94)$$

$$\Rightarrow pS_n = \sum_{k=1}^n p^k \quad (95)$$

$$\Rightarrow (1-p)S_n = 1 - p^n \quad (96)$$

$$\Rightarrow S_n = \frac{1 - p^n}{1 - p} \quad (97)$$

The sequence  $p^{k-1}, k = 1, \dots, n$  is known as a *Geometric Progression*. Substituting in (93),

$$A_0 = \lim_{\substack{r \rightarrow 1 \\ n \rightarrow \infty}} (1-r) \frac{1 - r^{3n}}{1 - r^3} \quad (98)$$

$$= \lim_{\substack{r \rightarrow 1 \\ n \rightarrow \infty}} \frac{1 - r^{3n}}{1 + r + r^2} \quad (99)$$

$$= \frac{1}{3} \quad (\because r^{3n} \rightarrow 0) \quad (100)$$