$ Supplement Note 12: Linear Operator

$1: Linear Operator

An operator A which maps a vector in a vector space $\mathcal{V}$ to another: $\mathbf{x} \in \mathcal{V} \Rightarrow A\mathbf{x} \in \mathcal{V}$ is called a *linear operator* if the following *linearity conditions* are satisfied:

$$A(\mathbf{x}+\mathbf{y}) = A\mathbf{x} + A\mathbf{y}; \quad \mathbf{x},\mathbf{y} \in \mathcal{V} \tag{1a}$$

$$A(\alpha\,\mathbf{x}) = \alpha A\mathbf{x}; \quad \mathbf{x} \in \mathcal{V}, \alpha \text{ is a scalar} \tag{1b}$$

Note that *the product AB of two linear operators A and B is also a linear operator* since the linearity conditions (1) are satisfied:

$$AB(\mathbf{x}+\mathbf{y}) \equiv A(B(\mathbf{x}+\mathbf{y})) = A(B\mathbf{x}+B\mathbf{y}) = AB\mathbf{x} + AB\mathbf{y} \tag{2a}$$

$$AB(\alpha\,\mathbf{x}) = A(B(\alpha\,\mathbf{x})) = A(\alpha B\mathbf{x}) = \alpha AB\mathbf{x} \tag{2b}$$

The *identity operator* $\mathbf{I}$ which maps every vector $\mathbf{x} \in \mathcal{V}$ to itself ($\mathbf{I}\mathbf{x} = \mathbf{x}$) is definitely a linear operator. The following relation is satisfied for any linear operator A:

$$\mathbf{I}A = A\mathbf{I} = A \tag{3}$$

As an example, the *differentiation operator* $\mathcal{D}f \equiv \frac{df}{dx}$ is a *linear operator* in the vector space composed of differentiable functions with the ordinary addition taken as the addition operator in this space. Further, the product of $\mathcal{D}$ with itself $\mathcal{D}^2 \equiv \mathcal{D}\mathcal{D} \equiv \frac{d^2}{dx^2}$ is also a linear operator.

$2: Linear Equations

Let us consider the set of simultaneous linear equations for x_1, x_2:

$$\begin{array}{rl} 4x_1 + 5x_2 = & 1 \\ 3x_1 - 2x_2 = & 3 \end{array} \Leftrightarrow \begin{array}{rl} a_1^{\ 1}x_1 + a_1^{\ 2}x_2 = & c_1 \\ a_2^{\ 1}x_1 + a_2^{\ 2}x_2 = & c_2 \end{array} \tag{4}$$

where $c_1 \equiv 1, c_2 \equiv 3$ and $a_1^{\ 1} \equiv 4, a_1^{\ 2} \equiv 5, a_2^{\ 1} \equiv 3, a_2^{\ 2} \equiv -2$. The linear equations (4) may be seen as a *matrix equation*

$$A\mathbf{x} = \mathbf{c} \quad \Leftrightarrow \quad \sum_{j=1}^{2} a_i^{\ j}\, x_j = c_i\,, \quad i = 1, 2 \tag{5}$$

for the column vectors $\mathbf{x} \equiv \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and $\mathbf{c} \equiv \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \equiv \begin{pmatrix} 1 \\ 3 \end{pmatrix}$ with the matrix A given by

$$A \equiv \begin{pmatrix} a_1^{\ 1} & a_1^{\ 2} \\ a_2^{\ 1} & a_2^{\ 2} \end{pmatrix} \equiv \begin{pmatrix} 4 & 5 \\ 3 & -2 \end{pmatrix} \tag{6}$$

The matrix A with the rules (5) *is a linear operator* since the *linearity conditions* (1) are satisfied. Explicitly, with the column vectors $\mathbf{x} \equiv \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $\mathbf{y} \equiv \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ (1a) is expressed as:

$$\begin{pmatrix} a_1^{\ 1} & a_1^{\ 2} \\ a_2^{\ 1} & a_2^{\ 2} \end{pmatrix} \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \end{pmatrix} \equiv \begin{pmatrix} a_1^1\,(x_1 + y_1) + a_1^2\,(x_2 + y_2) \\ a_2^1\,(x_1 + y_1) + a_2^2\,(x_2 + y_2) \end{pmatrix} = \tag{7a}$$

$$\begin{pmatrix} a_1^{\ 1}\, x_1 + a_1^{\ 2}\, x_2 \\ a_2^{\ 1}\, x_1 + a_2^{\ 2}\, x_2 \end{pmatrix} + \begin{pmatrix} a_1^{\ 1}\, y_1 + a_1^{\ 2}\, y_2 \\ a_2^{\ 1}\, y_1 + a_2^{\ 2}\, y_2 \end{pmatrix} \equiv \begin{pmatrix} a_1^{\ 1}\, x_1 + a_1^{\ 2}\, x_2 + a_1^{\ 1}\, y_1 + a_1^{\ 2}\, y_2 \\ a_2^{\ 1}\, x_1 + a_2^{\ 2}\, x_2 + a_2^{\ 1}\, y_1 + a_2^{\ 2}\, y_2 \end{pmatrix} \tag{7b}$$

while (1b) is expressed as (where α is a scalar):

$$\begin{pmatrix} a_1^{\ 1} & a_1^{\ 2} \\ a_2^{\ 1} & a_2^{\ 2} \end{pmatrix} \alpha \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \equiv \begin{pmatrix} a_1^{\ 1} & a_1^{\ 2} \\ a_2^{\ 1} & a_2^{\ 2} \end{pmatrix} \begin{pmatrix} \alpha x_1 \\ \alpha x_2 \end{pmatrix} = \alpha \begin{pmatrix} a_1^{\ 1} & a_1^{\ 2} \\ a_2^{\ 1} & a_2^{\ 2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{8}$$

It is straightforward to generalize (7) and (8) to higher dimensions. This leads to the conclusion that *a matrix is a linear operator*.

The rule of product of two matrices $A \equiv \begin{pmatrix} a_1^{\ 1} & a_1^{\ 2} \\ a_2^{\ 1} & a_2^{\ 2} \end{pmatrix}$, $B \equiv \begin{pmatrix} b_1^{\ 1} & b_1^{\ 2} \\ b_2^{\ 1} & b_2^{\ 2} \end{pmatrix}$ follows from (5). For any column vector $\mathbf{x} \equiv \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ we have

$$\begin{aligned} AB\mathbf{x} \equiv A(B\mathbf{x}) \equiv A \begin{pmatrix} b_1^{\ 1} & b_1^{\ 2} \\ b_2^{\ 1} & b_2^{\ 2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} a_1^{\ 1} & a_1^{\ 2} \\ a_2^{\ 1} & a_2^{\ 2} \end{pmatrix} \begin{pmatrix} \sum_{j=1}^{2} b_1^{\ j} x_j \\ \sum_{j=1}^{2} b_2^{\ j} x_j \end{pmatrix} \\ = \begin{pmatrix} \sum_{j=1}(a_1^{\ 1} b_1^{\ j} x_j + a_1^{\ 2} b_2^{\ j} x_j) \\ \sum_{j=1}(a_2^{\ 1} b_1^{\ j} x_j + a_2^{\ 2} b_2^{\ j} x_j) \end{pmatrix} &\equiv \begin{pmatrix} \sum_k \sum_j a_1^{\ k} b_k^{\ j} x_j \\ \sum_k \sum_j a_2^{\ k} b_k^{\ j} x_j \end{pmatrix} \end{aligned} \tag{9}$$

It follows from (5) that the matrix elements of AB is given by

$$(AB)_i^{\ j} = \sum_k a_i^{\ k} b_k^{\ j},\ i, j = 1, \ldots \tag{10}$$

The set of simultaneous linear equations (4) can be easily solved by using linalg.solve package in numpy as is done in *LinEq.py*.

As an application of linear equations, let us consider the following resistor circuit diagram. These problems are solved using *Kirchhoff's current* and *voltage rules*:

1. The *current rule*: The algebraic sum of all currents entering a node must be zero,
$$\sum I = 0 \tag{11}$$
The current rule is a direct result of the principle of *conservation of charge*.

2. The *voltage rule*: The algebraic sum of the potential differences (i.e. voltage changes) in any loop must be zero,
$$\sum \phi - \sum IR = 0 \tag{12}$$
where ϕ is the emf of the voltage sources, and R is the resistance of the resistor on the loop. Here we have used *Ohm's law*, which states that the voltage drop across a resistor is equal to the product of the current and the resistance IR. Kirchhoff's voltage rule is an expression of the *conservation of energy*.

Application of these rules to the circuit in Figure 1 now yields the following linear equations:

$$I_1 - I_2 - I_3 = 0 \tag{13a}$$
$$I_2 + I_3 - I_4 = 0 \tag{13b}$$
$$(20 + 5 + 10)I_2 - 2I_3 = 0 \tag{13c}$$
$$10I_1 + 2I_2 + 20I_4 = 110 - 0 \tag{13d}$$

which may be written as a matrix equation $A\ I = V$ with

$$A = \begin{pmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \\ 0 & 35 & -2 & 0 \\ 10 & 2 & 0 & 20 \end{pmatrix}, \quad I = \begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{pmatrix}, \quad \text{and } V = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 110 \end{pmatrix}$$

```
#LinEq.py
import numpy as np
## solve linear equations
#  4 x_1 + 5 x_2 = 1
#  3 x_1 - 2 x_2 = 3
AA = np.array([4,5,3, -2])
A = AA.reshape(2,2)
print('A=', A)
c = np.array([1,3])
print('c=', c)
x = np.linalg.solve(A,c)
print('solution of A x = c, x=', x)
cc = A @ x     # matrix product
print('Check: A x =',cc)
## solve resistor circuit
AA = np.array([1,-1,-1,0,0,1,1,-1,0,20+5+10,-2,0,10,2,0,20])
A = AA.reshape(4,4)
print('A=', A)
V = np.array([0,0,0,110-0])
print('V=', V)
I = np.linalg.solve(A,V)
print('I=', I)

## code for drawing Resistor Circuit
from matplotlib import patches
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_subplot(111, aspect='equal')
ax.set_xbound(0,14)
ax.set_ybound(0,8)
def ResistorH(x0,y0,ohm): #horizontal resistor
    ax.text(x0-0.5,y0+0.5,r'R = {:2d}$\Omega$'.format(ohm))
    dx, dy, dxh = 0.08, 0.24, 0.04
```

```
    x2, y1, y2=  x0+ dxh, y0+dy, y0-dy
    plt.plot([x0,x2],[y0,y1],color='k')
    for i in range(2):
        x0= x2
        x1,x2 = x0 +dx, x0 + 2*dx
        plt.plot([x0,x1],[y1,y2],color='k')
        plt.plot([x1,x2],[y2,y1],color='k')
    x0= x2
    x2 = x2+dx
    plt.plot([x0,x2],[y1,y2],color='k')
    plt.plot([x2,x2+dxh],[y2,y0],color='k')
def ResistorV(x0,y0,ohm): #vertical resistor
    ax.text(x0+0.4,y0+0.15,r'R = {:2d}$\Omega$'.format(ohm))
    #dx, dy =  0.24,0.08
    x1, x2=  x0 + 0.24, x0- 0.24
    xi, yi = x0, y0
    xf, yf = x1, y0+ 0.04
    plt.plot([xi,xf],[yi,yf],color='k')
    for i in range(3):
        xi, yi = xf,yf
        xf, yf = x2 , yi + 0.08
        plt.plot([xi,xf],[yi,yf],color='k')
        xi, yi = xf,yf
        xf, yf = x1 , yi + 0.08
        if i == 3:
            xf, yf= x0, yi + 0.04
        plt.plot([xi,xf],[yi,yf],color='k')
#start drawing
xx= [[0,1],[1.48, 7.08],[7.56,8.06]] # horizontal lines
yy= [[-2,1],[1.48, 4.48]]             # vertical lines
xH = [1,7.08]                         # horizontal resistors
ohm1 = [10,20]
ohm2 = [20,10]
```

```
y0,y1 = -2, 4.48
for x1 in xx: plt.plot(x1,[y0, y0], color='k')
for x1,oh in zip(xH, ohm1): ResistorH(x1,y0,oh)
for x1 in xx: plt.plot(x1,[y1, y1], color='k')
for x1,oh in zip(xH, ohm2): ResistorH(x1,y1,oh)
for y in yy: plt.plot([4,4],y,color='k')
for y in yy: plt.plot([0,0],y,color='k')
ResistorV(0,1,5)
ResistorV(4,1,2)
#circle and Voltages
ax.text(8.3,y0,r'$V_4=0$ volts')
xcenter,ycenter= 8.17, y0
c1=patches.Circle((xcenter,ycenter),radius=0.15,fill=False,linewidth=0.5)
ax.add_patch(c1)
ax.text(8.3,y1,r'$V_1= 110$ volts')
xcenter,ycenter= 8.17, y1
c2=patches.Circle((xcenter,ycenter),radius=0.15,fill=False,linewidth=0.5)
ax.add_patch(c2)
#currents
plt.arrow(3, y1, -0.5,0, width = 0.05)
ax.text(2.5,y1+0.5,r'$I_2 $')
plt.arrow(6, y1, -0.5,0, width = 0.05)
ax.text(5.5,y1+0.5,r'$I_1 $')
plt.arrow(4, 4, 0,-0.5, width = 0.05)
ax.text(4.5,3.5,r'$I_3 $')
plt.arrow(5, y0, 0.5,0, width = 0.05)
ax.text(5,y0+0.5,r'$I_4 $')

plt.axis('off')
figg = plt.gcf()
figg.savefig('resistos.eps', format='eps')
plt.show()
```

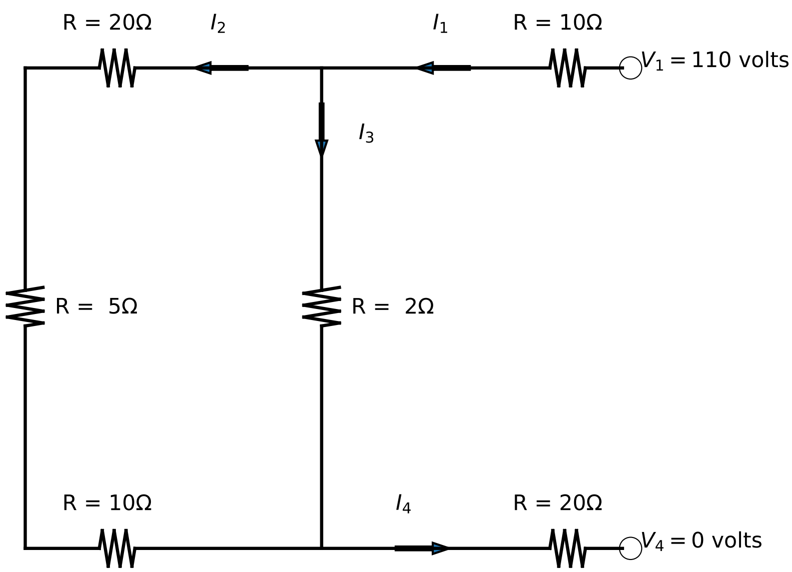

Figure 1: Resistor Circuit