Chapter 16 e^At: Eigenvalues and linear equations related to Euler's number e.

16.1 Find $e^{\mathbf{A}t}$ for

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 9 & 1 \end{bmatrix}$$

In this case the matrix $\mathbf{A}$ has two rows and two columns and, under these circumstances the equation

$$e^{\mathbf{A}\,t} = \alpha_1\,\mathbf{A}\,t + \alpha_0\,\mathbf{I}$$

applies.

$$\alpha_1 \begin{bmatrix} 1 & 1 \\ 9 & 1 \end{bmatrix} t + \alpha_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Building the matrix consists of simply summing the expressions for each of the element positions within the matrix.

$$\begin{bmatrix} \alpha_1 t + \alpha_0 & \alpha_1 t \\ 9\alpha_1 t & \alpha_1 t + \alpha_0 \end{bmatrix}$$

It is necessary to get the eigenvalues of the matrix, which W|A can give:

!| {{1, 1}, {9, 1}} |!

it turning out that they are equal to $\lambda_1 = 4$ and $\lambda_2 = -2$.

Throughout this chapter, whenever an eigenvalue is determined, it is presented as a dual pair, the second half constituting the variable t. With due respect, I think this is an abuse of the concept of eigenvalue. Eigenvalues, from what I can gather at a few sites, are constant scalar quantities. I will understand the current use as attaching a variable which, within the cases examined in this chapter, appears unfailingly with the appearance of the eigenvalue.

When the matrix $n = 2$, and the eigenvalues distinct, the elements of the final matrix are governed by the equation:

$$e^{\lambda\,t} = \lambda\,t\,\alpha_1 + \alpha_0$$

which in this case results in the equations:

$$e^{4t} = 4t\alpha_1 + \alpha_0$$

and

$$e^{-2t} = -2t\alpha_1 + \alpha_0$$

Wolfram Alpha was applied to for the simultaneous solution, but its standard computation time was not sufficient for the job. Instead, sympy did it in the wink of an eye. The code, and some scratch pad grafitti, are to be seen below.

After reconciling the values for α_0 and α_1, the matrix of $e^{\mathbf{A}t}$ morphs into:

$$e^{\mathbf{A}t} = \frac{1}{6}\begin{bmatrix} 3e^{4t} + 3e^{-2t} & e^{4t} - e^{-2t} \\ 9e^{4t} - 9e^{-2t} & 3e^{4t} + 3e^{-2t} \end{bmatrix}$$

The coefficients which were native to the symbolic form of the matrix, above, have an essential function in contributing to its final appearance.

In [11]:
```
from sympy import *

a0, a1, t = symbols('a0, a1, t')
eq1 = Eq(exp(4*t), 4*t*a1 + a0)
eq2 = Eq(exp(-2*t), -2*t*a1 + a0)
res = solve([eq1, eq2], (a0, a1))
print(res)
```

```
{a0: exp(4*t)/3 + 2*exp(-2*t)/3, a1: exp(4*t)/(6*t) - exp(-2*t)/(6*t)}
```

In [ ]:
```
(e^(6t) +2)*(e^(-2t)/3= (e^4t + 2*e^(-2t))/3 = alpha-0
```

In [ ]:
```
(e^(6t) - 1)*e^(-2t)/6t = (e^4t - e^(-2t))/6t = alpha-1 and after cancelling t factors ->
```

In [ ]:
```
(2*e^4t + 4*e^(-2t))/6 = alpha-0
```

In [ ]:
```
e^4t - e^(-2t)/6 = alpha-1
```

In [ ]:
```
1/6(3*e^4t + 3*e^(-2t)) combined at [1,1] and at [2,2]
```

16.2 Find $e^{\mathbf{A}t}$ for

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 8 & -2 \end{bmatrix}$$

In this case the matrix $\mathbf{A}$ has two rows and two columns and, under these circumstances the equation

$$e^{\mathbf{A}\,t} = \alpha_1\,\mathbf{A}\,t + \alpha_0\,\mathbf{I}$$

applies. The given matrix and the outline of the identity matrix together determine the entries, which in this case come together as

$$\alpha_1 \begin{bmatrix} 0 & 1 \\ 8 & -2 \end{bmatrix} t + \alpha_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

applies. Building the matrix consists of simply summing the expressions for each of the element positions within the matrix.

$$\begin{bmatrix} \alpha_0 & \alpha_1 t \\ 8\alpha_1 t & -2\alpha_1 t + \alpha_0 \end{bmatrix}$$

It is necessary to get the eigenvalues of the matrix, which W|A can give:

!| {{0, 1}, {8, -2}} |!

it turning out that they are equal to $\lambda_1 = -4$ and $\lambda_2 = 2$.

When the matrix $n = 2$, and the eigenvalues distinct, the elements of the final matrix are governed by the equation:

$$e^{\lambda\,t} = \lambda\,t\,\alpha_1 + \alpha_0$$

which in this case results in the equations:

$$e^{-4t} = -4t\alpha_1 + \alpha_0$$

and

$$e^{2t} = 2t\alpha_1 + \alpha_0$$

Wolfram Alpha was applied to for the simultaneous solution, but its standard computation time was again not sufficient for the job. Instead, sympy did it again in a snap. The code, and some scratch pad grafitti, are to be seen below.

After reconciling the values for α_0 and α_1, the matrix of $e^{\mathbf{A}t}$ morphs into:

$$e^{\mathbf{A}t} = \frac{1}{6}\begin{bmatrix} 4e^{2t} + 2e^{-4t} & e^{2t} - e^{-4t} \\ 8e^{2t} - 8e^{-4t} & 2e^{2t} + 4e^{-4t} \end{bmatrix}$$

The coefficients which were native to the symbolic form of the matrix, above, have an essential function in contributing to its final appearance.

In [3]:
```
from sympy import *

a0, a1, t = symbols('a0, a1, t')
eq1 = Eq(exp(-4*t), -4*t*a1 + a0)
eq2 = Eq(exp(2*t), 2*t*a1 + a0)
res = solve([eq1, eq2], (a0, a1))
print(res)
```

```
{a0: 2*exp(2*t)/3 + exp(-4*t)/3, a1: exp(2*t)/(6*t) - exp(-4*t)/(6*t)}
```

In [ ]:
```
2e^(2t)/3 + e^(-4t)/3 = alpha-0
```

In [ ]:
```
e^(2t)/6t -e^(-4t)/6t = alpha-1
```

In [ ]:
```
4e^(2t)/6 + 2e^(-4t)/6 = alpha-0
```

16.3 Find $e^{\mathbf{A}t}$ for

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

In this case the matrix $\mathbf{A}$ has two rows and two columns and, under these circumstances the equation

$$e^{\mathbf{A}\,t} = \alpha_1\,\mathbf{A}\,t + \alpha_0\,\mathbf{I}$$

applies. The given matrix and the outline of the identity matrix together determine the entries, which in this case come together as shown in Problem 1:

$$\begin{bmatrix} \alpha_0 & \alpha_1 t \\ -\alpha_1 t & \alpha_0 \end{bmatrix}$$

It is necessary to get the eigenvalues of the matrix, which W|A can give:

!| {{0, 1}, {-1, 0}} |!

it turning out that they are equal to $\lambda_1 = i$ and $\lambda_2 = -i$.

When the matrix $n = 2$, and the eigenvalues distinct, the elements of the final matrix are governed by the equation:

$$e^{\lambda\,t} = \lambda\,t\,\alpha_1 + \alpha_0$$

which in this case results in the equations:

$$e^{it} = it\alpha_1 + \alpha_0$$

and

$$e^{-it} = -it\alpha_1 + \alpha_0$$

Wolfram Alpha was applied to for the simultaneous solution, and its standard computation time was sufficient for the job, but the answers were too general to be useful. Instead, sympy did it again in a snap. The code, and some scratch pad grafitti, are to be seen below.

On the web was found a couple of versions of Euler's relation which will come in handy:

$$\sin x = \frac{e^{ix} - e^{-ix}}{2\,i} \quad \text{and} \quad \cos x = \frac{e^{ix} + e^{-ix}}{2}$$

After reconciling the values for α_0 and α_1, the matrix of $e^{\mathbf{A}t}$ morphs into:

$$e^{\mathbf{A}t} = \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix}$$

The coefficients which were native to the symbolic form of the matrix, above, have an essential function in contributing to its final appearance.

In [8]:
```
from sympy import *

a0, a1, t, i = symbols('a0 a1 t i')
eq1 = Eq(exp(i*t), a1*i*t + a0)
eq2 = Eq(exp(-i*t),-a1*i*t + a0)
res = solve([eq1, eq2], (a0, a1))
print(res)
```

```
{a0: exp(i*t)/2 + exp(-i*t)/2, a1: exp(i*t)/(2*i*t) - exp(-i*t)/(2*i*t)}
```

In [ ]:
```
e^(it/2) + e^(-it)/2 alpha-0
```

In [ ]:
```
e^(it)/(2it) - e^(-it)/(2it) alpha-1
```

In [ ]:
```
1/2(e^it + e^-it) alpha-0
```

In [ ]:
```
1/2it(e^it - e^-it)  alpha-1
```

16.4 Find $e^{\mathbf{A}t}$ for

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -9 & 6 \end{bmatrix}$$

In this case the matrix $\mathbf{A}$ has two rows and two columns and, under these circumstances the equation

$$e^{\mathbf{A}\,t} = \alpha_1\,\mathbf{A}\,t + \alpha_0\,\mathbf{I}$$

applies.

The the present case the $e^{\mathbf{A}}$ matrix appears as:

$$\begin{bmatrix} \alpha_0 & \alpha_1 t \\ -9\alpha_1 t & 6\alpha_1 + \alpha_0 \end{bmatrix}$$

It is necessary to get the eigenvalues of the matrix, which W|A can give:

!| {{0, 1}, {-9, 6}} |!

it turning out that they are equal to $\lambda_1 = 3$ only. Evidently there is a duplicate eigenvalue for this size 2 matrix.

When the matrix $n = 2$, and the eigenvalues not distinct, the elements of the final matrix are governed by the equations:

$$e^{\lambda\,t} = \lambda\,t\,\alpha_1 + \alpha_0$$

$$e^{\lambda\,t} = \alpha_1$$

which in this case results in the equations:

$$e^{3t} = 3t\alpha_1 + \alpha_0$$

and

$$e^{3t} = \alpha_1$$

Wolfram Alpha was applied to for the simultaneous solution, and its standard computation time was sufficient only for and integer solution to the job. So, sympy did it again. The code, and some scratch pad grafitti, are to be seen below.

$$e^{\mathbf{A}t} = e^{3t}\begin{bmatrix} 1-3t & t \\ -9t & 7-3t \end{bmatrix}$$

The coefficients which were native to the symbolic form of the matrix, above, have an essential function in contributing to its final appearance. Note: text answer for matrix position [2][2] does not agree with the above.

In [1]:
```
from sympy import *

a0, a1, t = symbols('a0 a1 t')
eq1 = Eq(exp(3*t), a1*3*t + a0)
eq2 = Eq(exp(3*t),a1 )
res = solve([eq1, eq2], (a0, a1))
print(res)
```

```
{a0: -3*t*exp(3*t) + exp(3*t), a1: exp(3*t)}
```

In [ ]:
```
-3te^(3t) + e(3t) alpha-0
```

In [ ]:
```
e^(3t) alpha-1
```

In [ ]:
```
6a1 + a0 = 6e^(3t)-3te^(3t) + e^(3t) = e^(3t)[6 + 1 - 3t)] = e^(3t)[7 -3t] NB position [2
```

16.5 Find $e^{\mathbf{A}t}$ for

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

In this case the matrix **A** has three rows and three columns and, under these circumstances the equation

$$e^{\mathbf{A}\,t} = \alpha_2 \mathbf{A}^2\, t^2 + \alpha_1 \mathbf{A} t + \alpha_0 \mathbf{I} =$$

$$= \alpha_2 \begin{bmatrix} 9 & 6 & 1 \\ 0 & 9 & 6 \\ 0 & 0 & 9 \end{bmatrix} t^2 + \alpha_1 \begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix} t + \alpha_0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

applies. Building the matrix consists of simply summing the expressions for each of the element positions within the matrix.

In the present case the $e^{\mathbf{A}}$ matrix appears as:

$$\begin{bmatrix} 9\alpha_2 t^2 + 3\alpha_1 t + \alpha_0 & 6\alpha_2 t^2 + \alpha_1 t & \alpha_2 t^2 \\ 0 & 9\alpha_2 t^2 + 3\alpha_1 t + \alpha_0 & 6\alpha_2 t^2 + \alpha_1 t \\ 0 & 0 & 9\alpha_2 t^2 + 3\alpha_1 t + \alpha_0 \end{bmatrix}$$

It is necessary to get the eigenvalues of the matrix, which W|A can give:

!| {{3, 1, 0}, {0, 3, 1}, {0, 0, 3}} |!

it turning out that they are equal to $\lambda_1 = 3$ only. Evidently there is a triplicate eigenvalue for this size 3 matrix.

The text author chooses to introduce an eigenvalue related equation,

① $$r(\lambda) = \alpha_2 \lambda^2 + \alpha_1 \lambda + \alpha_0$$

and the purpose of this equation is to be the determination of other equations for manufacturing the entries into the final $e^{\mathbf{A}t}$ matrix. To this end the r equation is differentiated twice with respect to "λ", the two resultant equations being:

② $$\frac{dr(\lambda)}{d\lambda} = 2\alpha_2 \lambda + \alpha_1$$

③ $$\frac{d^2 r(\lambda)}{d\lambda^2} = 2\alpha_2$$

The three numbered equations above are based on $r(\lambda)$, and can be defined clearly by substituting the value of the eigenvalue and its sidekick t:

$$e^{3t} = \alpha_2\, 9t^2 + \alpha_1\, 3t + \alpha_0$$

$$e^{3t} = 6\alpha_2\, t + \alpha_1$$

$$e^{3t} = 2\alpha_2$$

Based on its insufficient processing time available for the last few sets of simultaneous equations, Wolfram Alpha is not applied to for the simultaneous solution. So, sympy does it again. The code, and considerable scratch pad grafitti, are to be seen below.

$$e^{\mathbf{A}t} = e^{3t} \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$$

The coefficients which were native to the symbolic form of the matrix, above, have an essential function in contributing to its final appearance.

In [1]:
```
from sympy import *

a0, a1, a2, t = symbols('a0 a1 a2 t')
eq1 = Eq(exp(3*t), a2*9*t**2 + a1*3*t + a0)
eq2 = Eq(exp(3*t), 6*a2*t + a1)
eq3 = Eq(exp(3*t), 2*a2)
res = solve([eq1, eq2, eq3], (a0, a1, a2))
print(res)
```

```
{a0: 9*t**2*exp(3*t)/2 - 3*t*exp(3*t) + exp(3*t), a1: -3*t*exp(3*t) + exp(3*t), a2: exp(
3*t)/2}
```

In [2]:
```
a0 = 9*t**2*exp(3*t)/2 - 3*t*exp(3*t) + exp(3*t)
a0
```

Out[2]: $\frac{9t^2 e^{3t}}{2} - 3te^{3t} + e^{3t}$

In [3]:
```
a1 = -3*t*exp(3*t) + exp(3*t)
a1
```

Out[3]: $-3te^{3t} + e^{3t}$

In [4]:
```
a2 = exp(3*t)/2
a2
```

Out[4]: e^{3t}

In [18]:
```
one1 = 9*a2*t**2 + 3*a1*t + a0
one1
```

Out[18]: $9t^2 e^{3t} + t\left(-9te^{3t} + 3e^{3t}\right) - 3te^{3t} + e^{3t}$

In [19]:
```
simplify(one1)
```

Out[19]: e^{3t}

In [11]:
```
one2 = 6*a2*t**2 + a1*t
one2
```

Out[11]: $3t^2 e^{3t} + t\left(-3te^{3t} + e^{3t}\right)$

In [16]:
```
simplify(one2)
```

Out[16]: te^{3t}

In [12]:
```
one3 = a2*t**2
one3
```

Out[12]: $\frac{t^2 e^{3t}}{2}$

In [13]:
```
two2 = 9*a2*t**2 + 3*a1*t + a0
two2
```

Out[13]: $9t^2 e^{3t} + t\left(-9te^{3t} + 3e^{3t}\right) - 3te^{3t} + e^{3t}$

In [17]:
```
simplify(two2)
```

Out[17]: e^{3t}

In [14]:
```
two3 = 6*a2*t**2 + a1*t
two3
```

Out[14]: $3t^2 e^{3t} + t\left(-3te^{3t} + e^{3t}\right)$

In [20]:
```
simplify(two3)
```

Out[20]: te^{3t}

In [15]:
```
three3 = 9*a2*t**2 + 3*a1*t + a0
three3
```

Out[15]: $9t^2e^{3t} + t\left(-9te^{3t} + 3e^{3t}\right) - 3te^{3t} + e^{3t}$

In [21]:
```
simplify(three3)
```

Out[21]: e^{3t}

16.6 Find $e^{\mathbf{A}t}$ for

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 2 \end{bmatrix}$$

In this case the matrix $\mathbf{A}$ has three rows and three columns and, under these circumstances the equation

$$e^{\mathbf{A}\,t} = \alpha_2 \mathbf{A}^2 t^2 + \alpha_1 \mathbf{A} t + \alpha_0 \mathbf{I} =$$

$$= \alpha_2 \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 2 \\ 0 & -2 & 3 \end{bmatrix} t^2 + \alpha_1 \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 2 \end{bmatrix} t + \alpha_0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

applies. Building the matrix consists of simply summing the expressions for each of the element positions within the matrix.

The the present case the $e^{\mathbf{A}}$ matrix appears as:

$$\begin{bmatrix} \alpha_0 & \alpha_1 t & \alpha_2 t^2 \\ 0 & -\alpha_2 t^2 + \alpha_0 & 2\alpha_2 t^2 + \alpha_1 t \\ 0 & -2\alpha_2 t^2 - \alpha_1 t & 3\alpha_2 t^2 + 2\alpha_1 t + \alpha_0 \end{bmatrix}$$

It is necessary to get the eigenvalues of the matrix, which W|A can give. However, scipy has an advantage over W|A in the matter of eigenvalues, because when there are duplicates, scipy will tell which they are and how many of each. In this case (see cell below) there are three eigenvalues, all real, 2 of which equal 1 and one of which equals 0.

The text author chooses to introduce an eigenvalue related equation,

① $$r(\lambda) = \alpha_2 \lambda^2 + \alpha_1 \lambda + \alpha_0$$

and the purpose of this equation is to be the determination of other equations for manufacturing the entries into the final $e^{\mathbf{A}t}$ matrix. To this end the r equation is differentiated twice with respect to "λ", the two resultant equations being:

② $$\frac{dr(\lambda)}{d\lambda} = 2\alpha_2 \lambda + \alpha_1$$

③ $$\frac{d^2 r(\lambda)}{d\lambda^2} = 2\alpha_2$$

The three numbered equations above are based on $r(\lambda)$, and can be defined by substituting the value of the eigenvalue and its sidekick t. The first two equations take as their operative the eigenvalues equal to 1. The third equation utilizes the eigenvalue equal to 0, which appears when the tricky t variable equals 0, and inhabits the $\mathbf{I}$, matrix, not the $\mathbf{A}$ matrix:

$$e^t = \alpha_2 t^2 + \alpha_1 t + \alpha_0$$

$$e^t = 2\alpha_2 t + \alpha_1$$

$$e^0 = \alpha_0$$

Based on its insufficient processing time available for the last few sets of simultaneous equations, Wolfram Alpha is not applied to for the simultaneous solution. So, sympy does it again. The code, and scratch pad grafitti, are to be seen below.

The coefficients which were native to the symbolic form of the matrix, above, have an essential function in contributing to its final appearance.

$$e^{\mathbf{A}t} = \begin{bmatrix} 1 & -te^t + 2e^t - 2 & te^t - e^t + 1 \\ 0 & (1-t)e^t & te^t \\ 0 & -te^t & (t+1)e^t \end{bmatrix}$$

In [4]:
```
import numpy as np
from scipy.linalg import eigvals as eig

a = np.array([[0, 1, 0],[0, 0, 1],[0, -1, 2]])
eig(a)
```

Out[4]:
```
array([0.+0.j, 1.+0.j, 1.+0.j])
```

In [3]:
```
from sympy import *

a0, a1, a2, t = symbols('a0 a1 a2 t')
eq1 = Eq(exp(t), a2*t**2 + a1*t + a0)
eq2 = Eq(exp(t), 2*a2*t + a1)
eq3 = Eq(exp(0), a0)
res = solve([eq1, eq2, eq3], (a0, a1, a2))
print(res)
```

```
{a0: 1, a1: -exp(t) + 2*exp(t)/t - 2/t, a2: exp(t)/t - exp(t)/t**2 + t**(-2)}
```

In [5]:
```
a0 =1
a0
```

Out[5]: 1

In [6]:
```
a1 = -exp(t) + 2*exp(t)/t -2/t
a1
```

Out[6]: $-e^t + \frac{2e^t}{t} - \frac{2}{t}$

In [18]:
```
a2 = exp(t)/t -exp(t)/t**2 + t**(-2)
a2
```

Out[18]: $\frac{e^t}{t} - \frac{e^t}{t^2} + \frac{1}{t^2}$

In [20]:
```
three3 = 3*a2*t**2+2*a1*t+a0
three3
```

Out[20]: $t^2 \cdot \left(\frac{3e^t}{t} - \frac{3e^t}{t^2} + \frac{3}{t^2}\right) + t\left(-2e^t + \frac{4e^t}{t} - \frac{4}{t}\right) + 1$

In [21]:
```
simplify(three3)
```

Out[21]: $(t+1)\,e^t$

In [23]:
```
three2 = -2*a2*t**2-a1*t
three2
```

Out[23]: $t^2\left(-\frac{2e^t}{t} + \frac{2e^t}{t^2} - \frac{2}{t^2}\right) - t\left(-e^t + \frac{2e^t}{t} - \frac{2}{t}\right)$

In [24]:
```
simplify(three2)
```

Out[24]: $-te^t$

In [28]:
```
two3 = 2*a2*t**2+a1*t
two3
```

Out[28]: $t^2 \cdot \left(\frac{2e^t}{t} - \frac{2e^t}{t^2} + \frac{2}{t^2}\right) + t\left(-e^t + \frac{2e^t}{t} - \frac{2}{t}\right)$

In [29]:
```
simplify(two3)
```

Out[29]: te^t

In [30]:
```
two2 = -a2*t**2+a0
two2
```

Out[30]: $t^2\left(-\frac{e^t}{t} + \frac{e^t}{t^2} - \frac{1}{t^2}\right) + 1$

In [31]:
```
simplify(two2)
```

Out[31]: $(1-t)\,e^t$

In [33]:
```
one3 = a2*t**2
one3
```

Out[33]: $t^2\left(\frac{e^t}{t} - \frac{e^t}{t^2} + \frac{1}{t^2}\right)$

In [34]:
```
simplify(one3)
```

Out[34]: $te^t - e^t + 1$

16.7 Find $e^{\mathbf{A}t}$ for

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & -5 \\ 0 & 1 & 2 \end{bmatrix}$$

In this case the matrix **A** has three rows and three columns and, under these circumstances the equation

$$e^{\mathbf{A}\,t} = \alpha_2 \mathbf{A}^2\, t^2 + \alpha_1 \mathbf{A} t + \alpha_0 \mathbf{I} =$$

$$= \alpha_2 \begin{bmatrix} 0 & -2 & -5 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} t^2 + \alpha_1 \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & -5 \\ 0 & 1 & 2 \end{bmatrix} t + \alpha_0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

applies. Building the matrix consists of simply summing the expressions for each of the element positions within the matrix.

In the present case the $e^{\mathbf{A}}$ matrix appears as:

$$\begin{bmatrix} \alpha_0 & -2\alpha_2 t^2 + \alpha_1 t & -5\alpha_2 t^2 \\ 0 & -\alpha_2 t^2 - 2\alpha_1 t + \alpha_0 & -5\alpha_1 t \\ 0 & \alpha_1 t & -\alpha_2 t^2 + 2\alpha_1 t + \alpha_0 \end{bmatrix}$$

It is necessary to get the eigenvalues of the matrix, which W|A can give. However, scipy has an advantage over W|A in the matter of eigenvalues, because when there are duplicates, scipy will tell which they are and how many of each. In this case, shown in a code cell below, there are three eigenvalues:

$$\lambda_0 = 0 \quad \lambda_1 = i \quad \lambda_2 = -i$$

In keeping with the style of the text, each eigenvalue has a t variable yoked to it. The subscripting of scipy matches the subscripting of the text.

The text author chooses to introduce an eigenvalue related equation,

$$r(\lambda) = \alpha_2 \lambda^2 + \alpha_1 \lambda + \alpha_0$$

and the purpose of this equation is to be the determination of other equations for manufacturing the entries into the final $e^{\mathbf{A}t}$ matrix. It is further asserted that:

$$e^{\lambda i} = r(\lambda_i)$$

Setting out the representative equations, they are seen as:

$$e^0 = \alpha_2 (0)^2 + \alpha_1 (0) + \alpha_0$$

$$e^{it} = \alpha_2 (it)^2 + \alpha_1 (it) + \alpha_0$$

$$e^{-it} = \alpha_2 (-it)^2 + \alpha_1 (-it) + \alpha_0$$

Based on its habitual processing time problem for simultaneous equations, Wolfram Alpha is not applied to for the simultaneous solution. So, sympy does it again. The code, and scratch pad grafitti, are to be seen below.

The final form of the answer matrix is shown below. The text rendition of an answer was luckier in some areas of simplification, although there are minor points of substance in which sympy disagrees with the text answer.

$$e^{\mathbf{A}t} = \begin{bmatrix} 1 & \frac{\sinh(it)}{i} - \frac{e^{it}}{i^2} - \frac{e^{-it}}{i^2} & -5\frac{((e^{it}-2)e^{it}+1)e^{-it}}{2i^2} \\ 0 & 1 - \frac{2\sinh(it)}{i} - \frac{e^{it}}{2i^2} + \frac{1}{i^2} - \frac{e^{-it}}{2i^2} & -5\frac{\sinh(\mathrm{it})}{i} \\ 0 & \frac{\sinh(\mathrm{it})}{i} & 1 + \frac{2\sinh(it)}{i} - \frac{e^{it}}{2i^2} + \frac{1}{i^2} - \frac{e^{-it}}{2i^2} \end{bmatrix}$$

In [2]:
```
import numpy as np
from scipy.linalg import eigvals as eig

a = np.array([[0, 1, 0],[0, -2, -5],[0, 1, 2]])
eig(a)
```

Out[2]:
```
array([0.0000000e+00+0.j, 6.9388939e-17+1.j, 6.9388939e-17-1.j])
```

In [3]:
```
from scipy import linalg
import numpy as np

a = np.array([[0, 1, 0], [0, -2, -5], [0, 1, 2]])

#of the 3 ways to multiply arrays in python,
#"matmul" gives the "matrix product"
#"multipy" returns element-wise multiplication
#"dot" returns dot product

res1 = np.matmul(a, a)

#for line in res:
    #print ('  '.join(map(str, line)))
print(res1)
```

```
[[ 0 -2 -5]
 [ 0 -1  0]
 [ 0  0 -1]]
```

In [7]:
```
from sympy import *

a0, a1, a2, t, i = symbols('a0 a1 a2 t i')

eq1 = Eq(exp(0), a0)
eq2 = Eq(exp(i*t), a2*(i*t)**2 + a1*(i*t) + a0)
eq3 = Eq(exp(-i*t), a2*(-i*t)**2 + a1*(-i*t) + a0)
res = solve([eq1, eq2, eq3], (a0, a1, a2))
print(res)
```

```
{a0: 1, a1: exp(i*t)/(2*i*t) - exp(-i*t)/(2*i*t), a2: exp(i*t)/(2*i**2*t**2) - 1/(i**2*t
**2) + exp(-i*t)/(2*i**2*t**2)}
```

In [8]:
```
a0 = 1
a0
```

Out[8]: 1

In [19]:
```
a1 = exp(i*t)/(2*i*t) - exp(-i*t)/(2*i*t)
a1
```

Out[19]: $\frac{e^{it}}{2it} - \frac{e^{-it}}{2it}$

In [21]:
```
sa1 = simplify(a1)
sa1
```

Out[21]: $\frac{\sinh(it)}{it}$

In [11]:
```
a2 = exp(i*t)/(2*i**2*t**2) - 1/(i**2*t**2) + exp(-i*t)/(2*i**2*t**2)
a2
```

Out[11]: $\frac{e^{it}}{2i^2t^2} - \frac{1}{i^2t^2} + \frac{e^{-it}}{2i^2t^2}$

In [22]:
```
sa2 = simplify(a2)
sa2
```

Out[22]: $\frac{\left(\left(e^{it} - 2\right)e^{it} + 1\right)e^{-it}}{2i^2t^2}$

In [23]:
```
three3 = -sa2*t**2 + 2*sa1*t + a0
three3
```

Out[23]: $1 + \frac{2\sinh(it)}{i} - \frac{\left(\left(e^{it} - 2\right)e^{it} + 1\right)e^{-it}}{2i^2}$

In [24]:
```
simplify(three3)
```

Out[24]: $1 + \frac{2\sinh(it)}{i} - \frac{e^{it}}{2i^2} + \frac{1}{i^2} - \frac{e^{-it}}{2i^2}$

In [25]:
```
two2 = -sa2*t**2 - 2*sa1*t + 1
two2
```

Out[25]: $1 - \frac{2\sinh(it)}{i} - \frac{\left(\left(e^{it} - 2\right)e^{it} + 1\right)e^{-it}}{2i^2}$

In [26]:
```
simplify(two2)
```

Out[26]: $1 - \frac{2\sinh(it)}{i} - \frac{e^{it}}{2i^2} + \frac{1}{i^2} - \frac{e^{-it}}{2i^2}$

In [27]:
```
one3 = -5*sa2*t**2
one3
```

Out[27]: $-\frac{5\left(\left(e^{it} - 2\right)e^{it} + 1\right)e^{-it}}{2i^2}$

In [29]:
```
one2 = -2*sa2*t**2+sa1*t
one2
```

Out[29]: $\frac{\sinh(it)}{i} - \frac{\left(\left(e^{it} - 2\right)e^{it} + 1\right)e^{-it}}{i^2}$

In [30]:
```
simplify(one2)
```

Out[30]: $\frac{\sinh(it)}{i} - \frac{e^{it}}{i^2} + \frac{2}{i^2} - \frac{e^{-it}}{i^2}$

16.9 Find $e^{\mathbf{A}t}e^{\mathbf{B}t}$ and $e^{(\mathbf{A}+\mathbf{B})t}$ for

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}$$

This problem is not well-suited for Python. Numpy can to matrix exponentiation, but not with a symbol like t. Sympy cannot do matrix exponentiation. Scipy can't do it either. Maxima can do the problem without sweat, but so can Wolfram Alpha. This time Wolfram Alpha will be chosen.

The matrices of interest are these:

$$\mathbf{A}t = \begin{bmatrix} 0 & t \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B}t = \begin{bmatrix} 0 & 0 \\ -t & 0 \end{bmatrix}$$

The entry for $e^{\mathbf{A}t}$:

!| MatrixExp[{{0, t}, {0, 0}}] |!

and the answer is received:

$$\begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

The entry for $e^{\mathbf{B}t}$:

!| MatrixExp[{{0, 0}, {-t, 0}}] |!

and the answer is received:

$$\begin{bmatrix} 1 & 0 \\ -t & 1 \end{bmatrix}$$

The entry for $e^{(\mathbf{A}+\mathbf{B})t}$:

!| MatrixExp[[{{0, t}, {0, 0}} + {{0, 0}, {-t, 0}}] |!

and the answer is received:

① $$\begin{bmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{bmatrix}$$

The entry for $e^{(\mathbf{A})t}\ e^{(\mathbf{B})t}$:

!| {{1, t}, {0, 1}} . {{1, 0}, {-t, 1}} |!

and the answer is received:

② $$\begin{bmatrix} 1 - t^2 & t \\ -t & 1 \end{bmatrix}$$

The inequality of ① and ② show what the problem was trying to establish: Just because multiplying two exponented quantities generally means adding the exponents, this method does not hold for matrices.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: