Title: 8.2 Exercises

Author: Chad Wood

Date: 4 Feb 2022

Modified By: Chad Wood

Description: This program demonstrates the use of python to perform matrix calculations and transformations.

Solving a System of Linear Equations

Use Python to solve the following system of equations. Hint: You can write this system as a matrix equation in the form Ax = b, where the matrix A is invertible.

$$\begin{array}{rcrcrcrcr} 27x_1 & - & 10x_2 & + & 4x_3 & - & 29x_4 & = & 1 \\ -16x_1 & + & 5x_2 & - & 2x_3 & + & 18x_4 & = & -1 \\ -17x_1 & + & 4x_2 & - & 2x_3 & + & 20x_4 & = & 0 \\ -7x_1 & + & 2x_2 & - & x_3 & + & 8x_4 & = & 1 \end{array}$$

In [33]:

```
import numpy as np
from scipy.linalg import solve

# Creates matrix A
A = np.matrix('27 -10 4 -29;'
              '-16 5 -2 18;'
              '-17 4 -2 20;'
              '-7 2 -0 8')

# Defines result as matrix B
B = np.matrix('1; -1; 0; 1')

# Solves for x
x = solve(A, B)

# Verifies with True that matrices Ax and B are equal element wise
print(np.allclose(np.dot(A, x), B))

# Prints solution
x
```

```
True
```

Out[33]:

```
array([[-11. ],
       [ -6. ],
       [  1.5],
       [ -8. ]])
```

Visualizing a Linear Transformation

Let T be the linear transformation that takes in a vector v in R^2 and outputs the product of the 2 × 2 matrix A shown below and v.

$$\mathbf{A} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}$$

I.e., T is a function that takes in a vector v in R^2 and outputs the vector T(v) = Av, also in R^2.

```
(a)Find T(x), where x = [2/-2]

(b) Mimic the code in Section 7.1.3 of Essential Math for Data Science to visualize the linear transformation T.

(c) Use your visualization from (2) to describe geometrically how T transforms vectors in general.
```

In [42]:

```
# (a) Find T(x), where x = [2/-2]

from math import sqrt

# Defines √2/2 to create matrix A values
a = sqrt(2/2)

# Defines matrix A
A = np.matrix(f'{a} {-a};'
              f'{a} {a}')

# Performs T([2/-2]) using T(v) = Av
T = A * 2/-2 # Will be the transformation matrix

T
```

Out[42]:

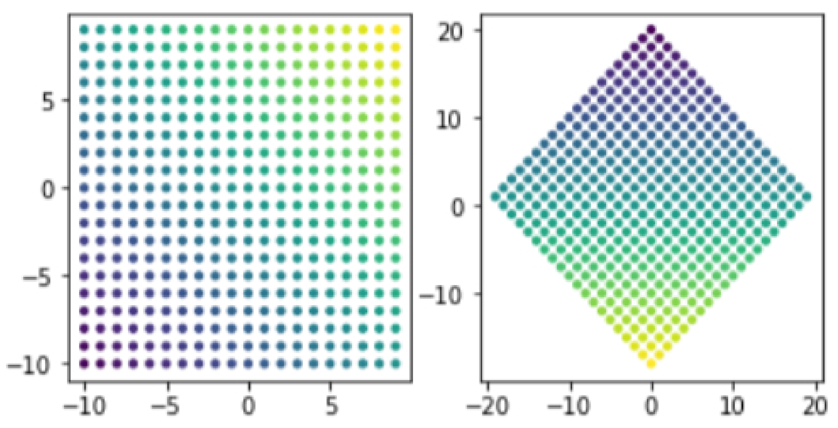
```
matrix([[-1.,  1.],
        [-1., -1.]])
```

In [55]:

```
# (b) Mimic the code in Section 7.1.3 of Essential Math for Data Science to visualize the linear transformation T.

import matplotlib.pyplot as plt

# Code from book in section 7.1.3
# Only changes are to T matrix used, comments, and convertion of transformed matrices to 1-D obj (array)

# Creates two vectors
x = np.arange(-10, 10, 1)
y = np.arange(-10, 10, 1)

# Calculates all combinations of x and y arrays
xx, yy = np.meshgrid(x, y)

# Restructures points to allow application of transformation matrix T
xy = np.vstack([xx.flatten(), yy.flatten()])

# Applies transformation matrix to all points
trans = T @ xy

# Reshapes transformed arrays for comparison to original
xx_transformed = np.array(trans[0].reshape(xx.shape)) # Adds np.array to make 1-D. Was not in book.
yy_transformed = np.array(trans[1].reshape(yy.shape)) # Adds np.array to make 1-D. Was not in book.

# Plots before and after transformation
f, axes = plt.subplots(1, 2, figsize=(6, 3))
axes[0].scatter(xx, yy, s=10, c=xx+yy)
axes[1].scatter(xx_transformed, yy_transformed, s=10, c=xx+yy)
```

Out[55]:

```
<matplotlib.collections.PathCollection at 0x263740a8cd0>
```



In [57]:

```
# (c) Use your visualization from (2) to describe geometrically how T transforms vectors in general.
('In general, its obvious that T geometrically transforms vectors by rotation.')
```

Out[57]:

```
'In general, its obvious that T geometrically transforms vectors by rotation.'
```