Solution Review: Setting Up an Optical System

Solution review to the tasks in setting up an optical system.

WE'LL COVER THE FOLLOWING

Solution 1

```
import numpy as np

C1 = np.array([[1, -1],[1, 0]])
C2 = np.array([[1, 1],[1, 0]])
C3 = np.array([[1, -1],[1, 1]])
C4 = np.array([[1, 0],[0, 1]])
C5 = np.array([[0, 1],[1, 1]])
C6 = np.array([[1, 0],[0, 1]])

arrs = [C1, C2, C3, C4, C5, C6]

system = np.array([[1, 0],[0, 1]])    # unit martrix
for i in range(len(arrs)):
  system = np.dot(system, arrs[i])

print(system)
```

Explanation

- In lines 3 - 8, we have defined the matrices $C_1 - C_6$.

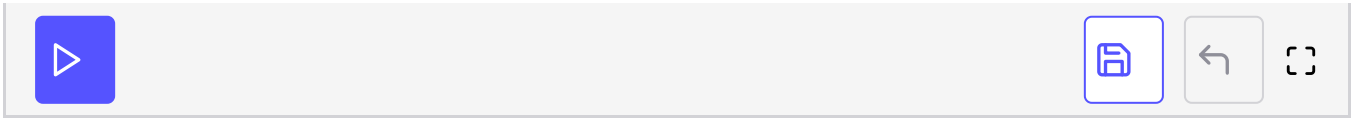
- We have defined a unit matrix in line 12 as the initial system matrix. A matrix multiplied with the unit matrix results in the matrix itself.
- In lines 13 - 14, we have defined a `for` loop to multiply all the subsequent matrices.

Solution 2

```
import numpy as np

system = np.array([[1, 2],[0, 2]])

evals, evecs = np.linalg.eig(system)
print("Eigenvalues:", evals)
print("Eigenvectors:", evecs)
```



Explanation

- We use the `eig` function from the `linalg` submodule from `numpy`. It returns two kinds of values: eigenvalues and eigenvectors.

Solution 3

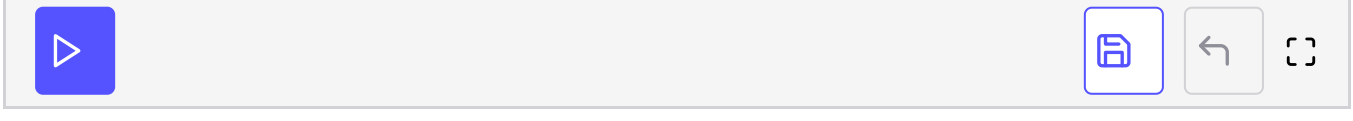
We have to solve the following system of linear equations to get the input Jones vector:

$$\begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

```
import numpy as np

system = np.array([[1, 2],[0, 2]])
out_vec = np.array([-1, 1])

in_vec = np.linalg.solve(system, out_vec)
print(in_vec)
```



Explanation

- In line 3, we have declared the system matrix `system` based on our answer from solution 1.
- In line 4, we have declared the matrix on the right-hand side of the equation.
- In line 6, we use the `solve` function from the `linalg` submodule to solve the system of linear equations. This will give us the input Jones vector.

The next lesson gives you a preview of another new concept: transfer functions.