problem5

February 12, 2020

1 Problem 5. (Assignment 1)

1.1 Introduction

In this solution I assume that $x = \begin{bmatrix} x \\ x' \\ \dots \\ x^{n-1} \end{bmatrix}$

1.2 Functions

Here I define a function that gets an array of coefficients as an input an returns matrix A for SS model.

[1]:
```
import numpy as np
```

[2]:
```
def ode2matrix(a):
    """
    Return A obtained from mult. a0...an
    Test1:
    >>> ode2matrix(np.array([0, 2, 4, 3, 1]))
    array([[ 0.,  1.,  0.,  0.],
           [ 0.,  0.,  1.,  0.],
           [ 0.,  0.,  0.,  1.],
           [ 0., -2., -4., -3.]])
    """
    size = len(a)
    # creating zeros for the first column
    zeros = np.zeros((size - 2, 1))
    # the last row of A. normalization happens here
    last_row = [-a[:-1] / a[-1]]
    return np.block([[zeros, np.eye(size - 2)], last_row])
```

A function that returns a vector b for a given ODE.

[3]:
```
def ode2b(a, b0):
    """
    Return vector b for state space from a0...an and b0
    Test1:
    >>> ode2b(np.array([0, 2, 4, 3, 4]), 2)
    array([[0. ],
           [0. ],
           [0. ],
           [0.5]])
    """
    return np.concatenate((np.zeros((len(a) - 2, 1)), np.array([[b0 / a[-1]]])))
```

1.3 ODE2SS

This is a function that returns a pair (A, b) for a given ODE. Note that a should be a numpy array that begins with a_0 and ends with a_n. b_0 is right-hand constant. ODE:

$$a_k y^{(k)} + a_{k-1} y^{(k-1)} + ... + a_2 y'' + a_1 y' + a_0 y = b_0$$

[4]:

```
def ode2ss(a, b0):
    return ode2matrix(a), ode2b(a, b0)
```

1.4 Example

$$-x''' + 5x'' + 3x' + 7x = 10$$

[5]:

```
#ode2ss(np.array([0, 2, 4, 3, 1]), 6)
ode2ss(np.array([7, 3, 5, -1]), 10)
```

[5]:

```
(array([[0., 1., 0.],
        [0., 0., 1.],
        [7., 3., 5.]]), array([[  0.],
        [  0.],
        [-10.]]))
```

1.5 Doctest

This is a test section. Run it if you want to check if the code is working properly.

[6]:

```
import doctest
doctest.testmod(verbose=True)
```

```
Trying:
    ode2b(np.array([0, 2, 4, 3, 4]), 2)
Expecting:
    array([[0. ],
           [0. ],
           [0. ],
           [0.5]])
ok
Trying:
    ode2matrix(np.array([0, 2, 4, 3, 1]))
Expecting:
    array([[ 0.,  1.,  0.,  0.],
           [ 0.,  0.,  1.,  0.],
           [ 0.,  0.,  0.,  1.],
           [ 0., -2., -4., -3.]])
ok
2 items had no tests:
    __main__
    __main__.ode2ss
2 items passed all tests:
   1 tests in __main__.ode2b
   1 tests in __main__.ode2matrix
2 tests in 4 items.
2 passed and 0 failed.
Test passed.
```

[6]:
```
TestResults(failed=0, attempted=2)
```

[7]:
```
#ode2matrix(np.array([7, 3, 5, -1]))
#ode2matrix(np.array([0, 2, 4, 3, 1]))
```

[8]:
```
ode2b(np.array([0, 2, 4, 3, 4]), 2)
```

[8]:
```
array([[0. ],
       [0. ],
       [0. ],
       [0.5]])
```