prob6

October 11, 2016

1 HW6 Q6

1.0.1 EECS 16A: Designing Information Devices and Systems I, Fall 2016

2 Circuit solver

In this question we will write a program that solves circuits methodically, able to include both voltage and current souces.

```
In [29]: import numpy as np
         from numpy import linalg
         from __future__ import print_function
```

(i) Write the incidence matrix F for the graph, considering v_1 and v_4 as a combined "supernode".

```
In [30]: F = np.array([[-1,0,1],
                       [1,-1,0],
                       [0,-1,1],
                       [1,0,-1],
                       [0,1,-1]])
         print('\nF:\n',F)
```

```
F:
 [[-1  0  1]
 [ 1 -1  0]
 [ 0 -1  1]
 [ 1  0 -1]
 [ 0  1 -1]]
```

(ii) Specify the resistance matrix R and the vector of voltage sources $\vec{b}$.

```
In [31]: R1,R2,R3,R4,R5 = 100000, 200, 100, 100000, 100
         Rvec = np.array([R1,R2,R3,R4,R5])
         R = np.eye(5)*Rvec

         Vs = 10
         b = np.array([[Vs],[0],[Vs],[0],[0]])

         # For convenience, we will define the conductance matrix G as the inverse of R.
         G = np.linalg.inv(R)

         print('\nR:\n',R)
         print('\nb:\n',b)
```

```
R:
 [[ 100000.       0.       0.       0.       0.]
 [      0.     200.      0.       0.       0.]
 [      0.       0.     100.      0.       0.]
 [      0.       0.       0.  100000.       0.]
 [      0.       0.       0.       0.     100.]]

b:
 [[10]
 [ 0]
 [10]
 [ 0]
 [ 0]]
```

(iii) Write down the vector f so that KCL is satisfied as: $F^T i + f = 0$

```
In [32]: f = np.array([[0],[0],[0]])
         print('\nf:\n', f)
```

```
f:
 [[0]
 [0]
 [0]]
```

(iii) What is the rank of F? Does it have a null space? If so, what is it?

Row-reducing F, we see F has a linearly dependent column (see Problem 6 Part d). Therefore, rank(F) is 2.

Since it's not a full rank, F does have a null space.

The null space is spanned by [[1],[1],[1]].

```
In [33]: # any code you write to help you answer above

         from sympy import Matrix
         Fm = Matrix(F)

         # Frr = row-reduced F matrix
         Frr = Fm.rref()
         print(Frr)

         # Fns = nullspace of F
         Fns = Fm.nullspace()
         print(Fns)
```

```
(Matrix([
[1.0,   0, -1.0],
[  0, 1.0, -1.0],
[  0,   0,    0],
[  0,   0,    0],
[  0,   0,    0]]), [0, 1])
[Matrix([
[1.0],
[1.0],
[  1]])]
```

(iv) Setting a potential in v to 0 corresponds to deleting a column of F. Let $v_4 = 0$, and write down our new "grounded" matrix F: F_grounded

```
In [34]: F_grounded = F[:, :2]
         print('\nF_grounded:\n', F_grounded)
```

```
F_grounded:
 [[-1  0]
 [ 1 -1]
 [ 0 -1]
 [ 1  0]
 [ 0  1]]
```

(v) Implement your algebraic solution to compute v in terms of F, G, $\vec{f}$, and $\vec{b}$. You may also have to slice $\vec{f}$ and $\vec{b}$.

```
In [35]: A = np.dot(F_grounded.T,np.dot(G,F_grounded))
         print('\nA:\n', A)

         B = np.linalg.inv(A)
         print('\nB:\n', B)

         f_gr = f[0:2]

         v_gr = - B.dot(f_gr + np.dot(F_grounded.T,np.dot(G,b)))
         print('\nv:\n', v_gr)
```

```
A:
 [[ 0.00502 -0.005  ]
 [-0.005    0.025  ]]

B:
 [[ 248.75621891   49.75124378]
 [  49.75124378   49.95024876]]

v:
 [[ 5.]
 [ 5.]]
```

(vi) Compute $\vec{i}$ with your solution of $\vec{v}$.

```
In [36]: i = G.dot(np.dot(F_grounded, v_gr) + b)

         print('\ni:\n', i)
```

```
i:
 [[  5.00000000e-05]
 [  8.88178420e-18]
 [  5.00000000e-02]
 [  5.00000000e-05]
 [  5.00000000e-02]]
```

```
In [ ]:
```