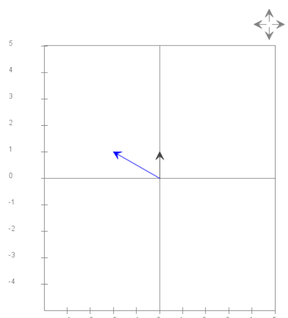
Computational Linear Algebra, Module 11

Maya Shende

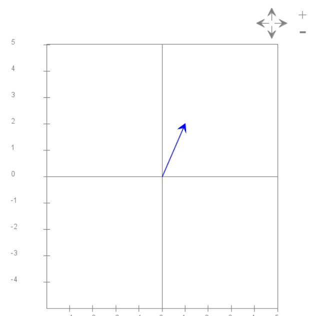
Due: April 18th, 2018

1. output:

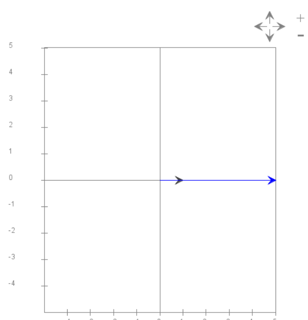
$u = (0, 1)$:



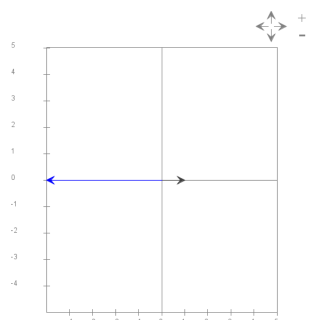
$u = (1, 2)$:



$u = (1, 0)$:

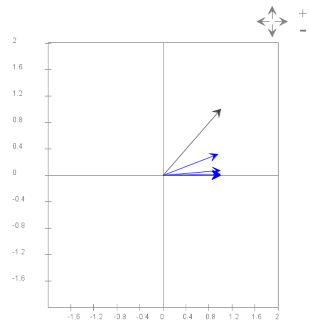


$u = (1, 0)$ and $\mathbf{A}[0][0] = -5$:

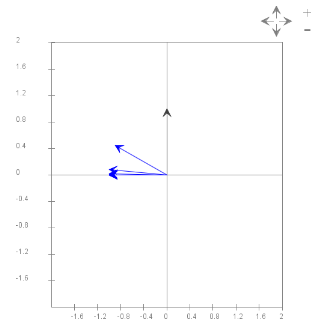


2. output:

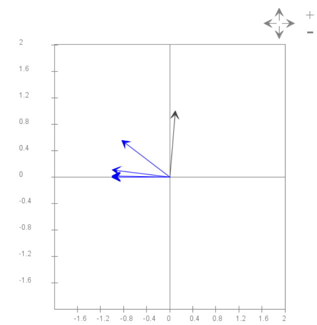
$u = (1, 1)$:



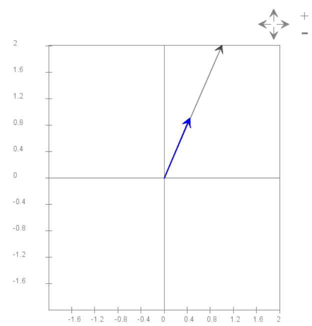
$u = (0, 1)$:



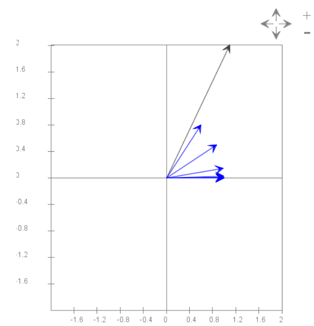
$u = (0.1, 1)$:



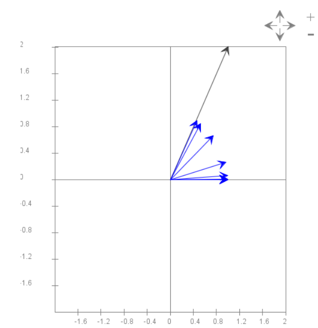
$u = (1, 2)$:



$u = (1.1, 2)$:



$u = (1.01, 2)$:

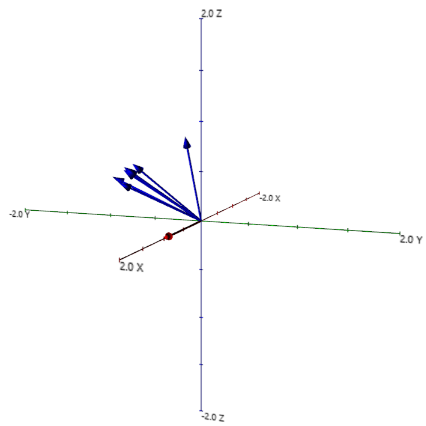


3.
$$\begin{bmatrix} 5 & -2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

   The eigenvalue is $\lambda = 1$.

4. Let $\mathbf{y} = -2\mathbf{x}$ then, $-\frac{1}{2}\mathbf{y} = \mathbf{x}$. Plugging back into the original formula $\mathbf{Ax} = \lambda\mathbf{x}$ gives you:
$$\mathbf{A}(-\frac{1}{2}\mathbf{y}) = \lambda(-\frac{1}{2}\mathbf{y})$$
$$\mathbf{A}(-\frac{1}{2}\mathbf{y}) = (-\lambda\frac{1}{2})\mathbf{y}$$

5. output:



6.

$$\mathbf{Ax}_1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{3}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} \end{bmatrix} = 3 \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\mathbf{Ax}_2 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = 1 \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

7. Let x_i be an eigenvector of A. Then, by definition, $Ax_i = \lambda_i x_i$. Then, $A^2 x_i = A(Ax_i) = A(\lambda_i x_i) = A\lambda_i x_i = \lambda_i(Ax_i) = \lambda_i \lambda_i x_i = \lambda_i^2 x_i$. Now, we can use this fact and assume that for all $n \leq k-1$, $A^n x_i = \lambda_i^n x_i$ to inductively prove this. So, we have $A^k x_i = A(A^{k-1} x_i) = A(\lambda_i^{k-1} x_i) = \lambda_i^{k-1}(Ax_i) = \lambda_i^{k-1} \lambda_i x_i = \lambda_i^k x_i$. ■

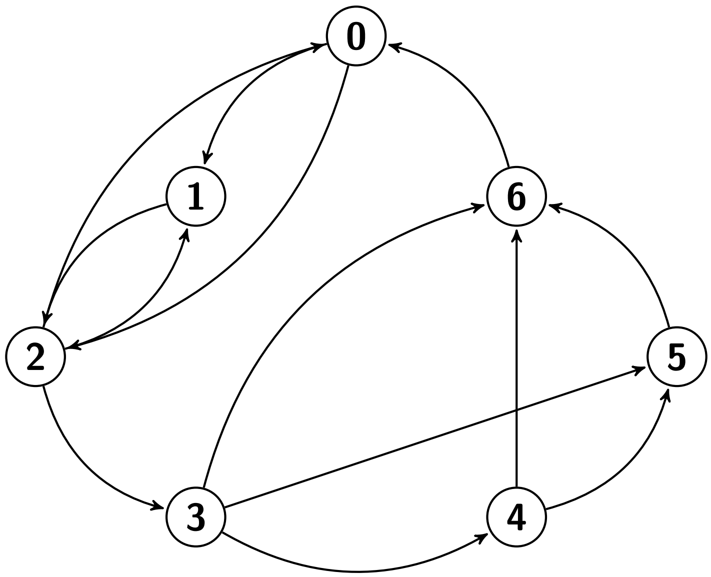
8. $\lambda_2 < \lambda_1$, and so we can say that $\frac{\lambda_2}{\lambda_1} < 1$, and so if we raise any number less than 1 to a power that approaches infinity, the denominator will grow much faster than the numerator, thus approaching 0.

9. Console Output:
0.9486832980505138 0.31622776601683794
0.9970544855015815 0.07669649888473705
0.9998740474835989 0.015871016626723796
0.9999948963838328 0.0031948718734307767
0.999999795331072 6.397951345688243E-4
0.9999999918090485 1.279918074758798E-4
0.9999999996723283 2.5599672315805973E-5
0.9999999999868928 5.119986892766448E-6
0.9999999999994756 1.023999475711732E-6
0.999999999999979 2.04799979028478E-7

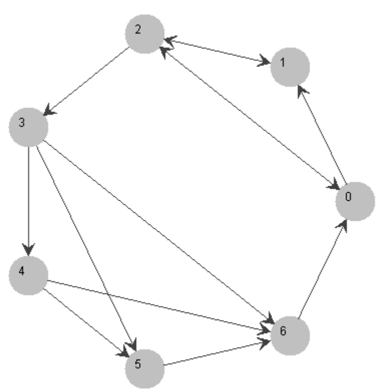
lambda=4.999999590399874

$\mathbf{A}^k\mathbf{u}^{(0)}$ =2.4414063E7 1.0

The two vectors are not the same.

10. 

$N(0) = \{(0,1),(0,2)\}$
$N(1) = \{(1,2)\}$
$N(2) = \{(2,0),(2,1),(2,3)\}$
$N(3) = \{(3,4),(3,5),(3,6)\}$
$N(4) = \{(4,5),(4,6)\}$
$N(5) = \{(5,6)\}$
$N(6) = \{(6,0)\}$

11. 

12. $v_0(10) = 3$
    $v_1(10) = 1$
    $v_2(10) = 3$
    $v_3(10) = 1$
    $v_4(10) = 0$
    $v_5(10) = 1$
    $v_6(10) = 1$

13. $\frac{v_0(t)}{t} = 0.1987$
    $\frac{v_1(t)}{t} = 0.2091$
    $\frac{v_2(t)}{t} = 0.3092$
    $\frac{v_3(t)}{t} = 0.0993$
    $\frac{v_4(t)}{t} = 0.0324$
    $\frac{v_5(t)}{t} = 0.0519$
    $\frac{v_6(t)}{t} = 0.0993$

14. You find the subset of pages relevant to the search query by examining the probabilities of page as it relates to the current page or search term and then sort the different probabilities, π_i.

15.

$$\mathbf{P} = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0.33 & 0.33 & 0 & 0.33 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
Transition probability matrix
 0.0000 0.5000 0.5000 0.0000 0.0000 0.0000 0.0000
 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000
 0.3333 0.3333 0.0000 0.3333 0.0000 0.0000 0.0000
 0.0000 0.0000 0.0000 0.0000 0.3333 0.3333 0.3333
 0.0000 0.0000 0.0000 0.0000 0.0000 0.5000 0.5000
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000
 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

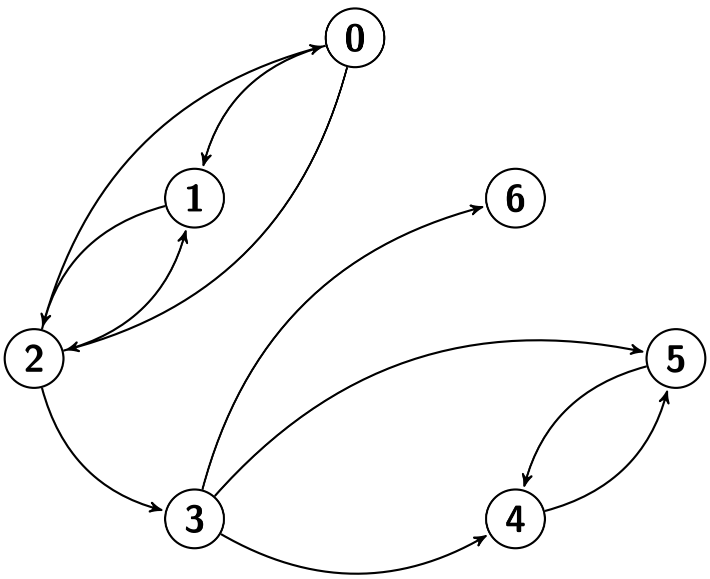
The relationship between the entries of the matrix and $N(i)$ is the entry in the matrix is zero unless there is an coordinate in the $N(i)$ and the value is the 1 divided by the size of the set $N(i)$.

16. The eigenvalue for $\mathbf{A}\pi = \pi$ is 1.

17. They are roughly equal.

```
Comparison: simulation vs. theory:
i=  0: v_i(t)/t=0.2026  Powermethod=0.2034
i=  1: v_i(t)/t=0.2059  Powermethod=0.2034
i=  2: v_i(t)/t=0.3059  Powermethod=0.3051
i=  3: v_i(t)/t=0.1016  Powermethod=0.1017
i=  4: v_i(t)/t=0.0323  Powermethod=0.0339
i=  5: v_i(t)/t=0.0501  Powermethod=0.0508
i=  6: v_i(t)/t=0.1016  Powermethod=0.1017
```

18. The matrix **P** is a probability matrix, where each row is the probability of transitioning between any other node and the node associated with that row. Therefore, the sum of all probabilities across a single row must equal 1.



19.

Node 6 does not link to anything and nodes 4 and 5 form an endless loop.

20.

21. It solves the dangling problem because now the surfer always has the option of jumping to a random node that may or may not have any links to any other nodes.

22. $\frac{1}{n}hh^T = \frac{1}{n}\begin{bmatrix}1 & 1 & 1 & \dots & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\\\vdots\\1\end{bmatrix} = \frac{1}{n}\begin{bmatrix}1 & 1 & \dots & 1\\1 & 1 & \dots & 1\\\vdots & \vdots & \ddots & \vdots\\1 & 1 & \dots & 1\end{bmatrix} = \begin{bmatrix}\frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n}\\\frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n}\\\vdots & \vdots & \ddots & \vdots\\\frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n}\end{bmatrix}$

23. In the last step, we are basically just summing the terms in x^{k-1} because h^T is a row of 1's.
    We know

$$\mathbf{h}^T\mathbf{x}^{(k-1)} = \begin{bmatrix}1 & 1 & ... & 1\end{bmatrix}\begin{bmatrix}p_1\\p_2\\\vdots\\p_n\end{bmatrix}$$

    where $p_1 + p_2 + ... + p_n = 1$. From vector-vector multiplication, we get:

$$\begin{bmatrix}(1)p_1 + (1)p_2 + ... + (1)p_n\end{bmatrix} = 1$$

24. It is sparse because it is a transition matrix, and we are not assuming that the graph is fully connected or mostly connected, therefore we call it sparse.

25. Pseudocode

```
function:  matrix-vector multiplication {
     Let u be the vector
     Let A be a collection of linked lists
     for each linked list in the collection:
          for each entry in u, i:
               result[i] += A.currentNode.getValue * u[i]
               A.currentNode.getValue = A.getNextNode
}
```

26. The eigenvalues were $\lambda = 3$ for $\mathbf{x}_1$ and $\lambda = 1$ for $\mathbf{x}_2$.

```
2x2 Matrix:
2.8 0.5999999999999999
0.6 1.2
2x2 Matrix:
2.975609756097561 0.21951219512195086
0.21951219512195125 1.024390243902439
2x2 Matrix:
2.9972602739726026 0.07397260273972558
0.07397260273972606 1.0027397260273974
2x2 Matrix:
2.999695214873514 0.024687595245351565
0.024687595245352044 1.000304785126486
2x2 Matrix:
2.9999661303979672 0.00823031329381832
0.008230313293818806 1.0000338696020323
2x2 Matrix:
2.9999962366542343 0.0027434790626253546
0.0027434790626258403 1.000003763345765
2x2 Matrix:
2.999999581849771 9.144945504566586E-4
9.144945504571443E-4 1.0000004181502289
2x2 Matrix:
2.9999999535388544 3.0483157346984744E-4
3.048315734703336E-4 1.0000000464611452
2x2 Matrix:
2.9999999948376503 1.0161052658782133E-4
1.0161052658830741E-4 1.0000000051623497
2x2 Matrix:
2.9999999994264055 3.387017560666067E-5
3.387017560714678E-5 1.0000000005735945
```