Lecture-26

Date Nov. 24, 2008

Matrix multiplications

$$A = \begin{bmatrix} 2 & 4 & 6 \\ 1 & 2 & 4 \\ 3 & 2 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 5 & 2 \\ 3 & 6 & 1 \end{bmatrix}$$

We know

$$A.*B = \begin{bmatrix} 2*1 & 4.4 & 6.3 \\ 1.2 & 2.5 & 4.2 \\ 3.3 & 6.2 & 1.1 \end{bmatrix}$$

(Scalar multiplication)

Note that $B.*A$ is also the same as $A.*B$.

Matrix multiplication

$$A = \begin{bmatrix} 2 & 4 & 6 \\ 1 & 2 & 4 \\ 3 & 2 & 1 \end{bmatrix} \quad \& \quad B = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 5 & 2 \\ 3 & 6 & 1 \end{bmatrix}$$

$$A*B = \begin{bmatrix} (2\times1+4\times2+6\times3) & (2\times4+4\times5+6\times6) & (2\times3+4\times2+6\times1) \\ (1\times1+2\times2+4\times3) & (1\times4+2\times5+4\times6) & (1\times3+2\times2+4\times1) \\ (3\times1+2\times2+1\times3) & (3\times4+2\times5+1\times6) & (3\times3+2\times2+1\times1) \end{bmatrix}$$

$$= \begin{bmatrix} 28 & 64 & 20 \\ 19 & 38 & 11 \\ 10 & 28 & 14 \end{bmatrix} \quad \text{— (1)}$$

>> A*B in Matlab for A & B defined above will produce the ~~[illegible]~~ Matrix (1)

>> B*A ? Try at home; you will notice that $B*A \neq A*B$ in general.

- Suppose we are multiplying two matrices A & B where A has m_A rows & n_A columns and B has m_B rows & n_B columns.

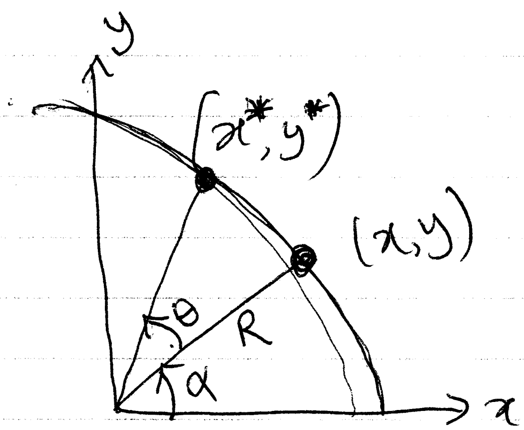
- We must make sure that for A*B matrix multiplication, $n_A = m_B$

- For B*A matrix multiplication $n_B = m_A$.

Rotating Coordinates

- Common use of matrix multiplication

2-D rotation



$$x^* = x\cos\theta - y\sin\theta$$

$$y^* = x\sin\theta + y\cos\theta.$$

i.e.

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

If $u = \begin{bmatrix} x^* \\ y^* \end{bmatrix}$, $A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, $v = \begin{bmatrix} x \\ y \end{bmatrix}$

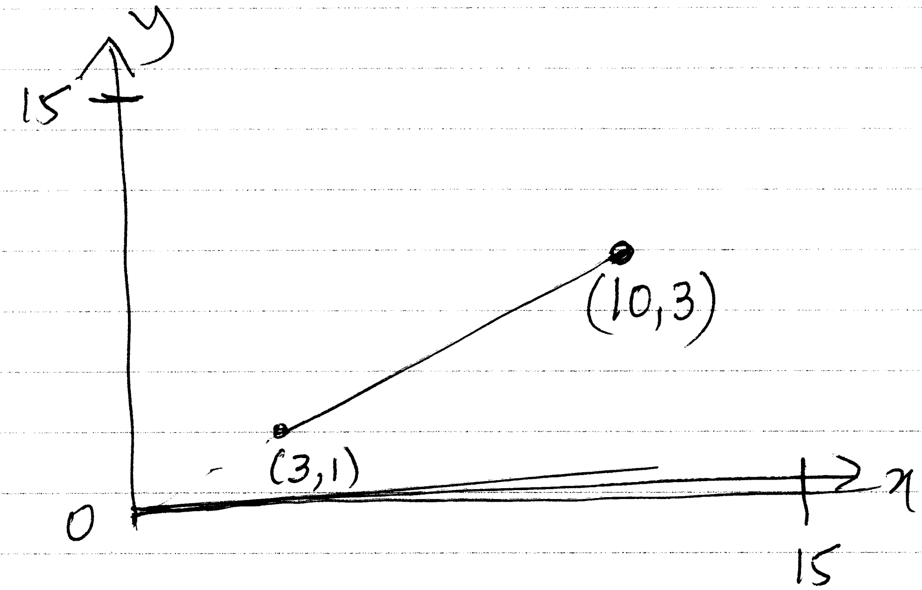
$\therefore\ u = A * v$: A has 2 rows & 2 cols
v has 2 rows & 1 col.

(Dimensions match for multiplication)

Function to compute a rotation matrix

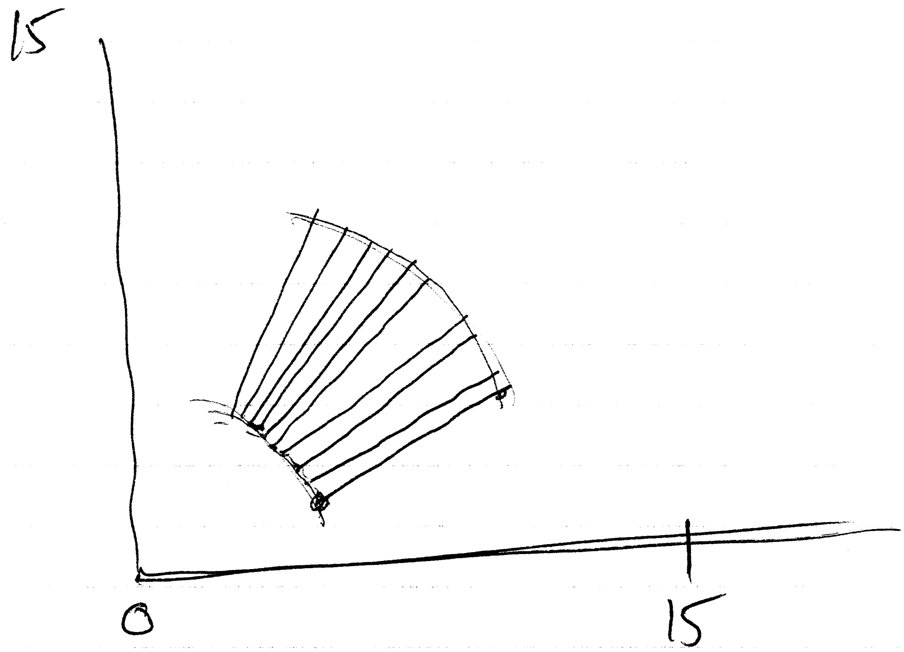
```
function transformation = rotation(angle)
% rotate by this angle (radians)
transformation = [cos(angle)  -sin(angle)
                  sin(angle)   cos(angle)];
```

Rotating a line around origin



Script

```
pts = [3, 10
       1  3];
plot(pts(1,:), pts(2,:))
axis([0 15 0 15]
hold on
for angle = 0.05 :0.05:1        % iterates across
                                % a selection of angles
    A = rotation(angle)         % obtain the rotation
                                  matrix
    pr = A*pts                  % rotates the original line
    plot(pr(1,:), pr(2,:))      % plot the rotated line
end
```



Output of script file