Computer Graphics

Week 1

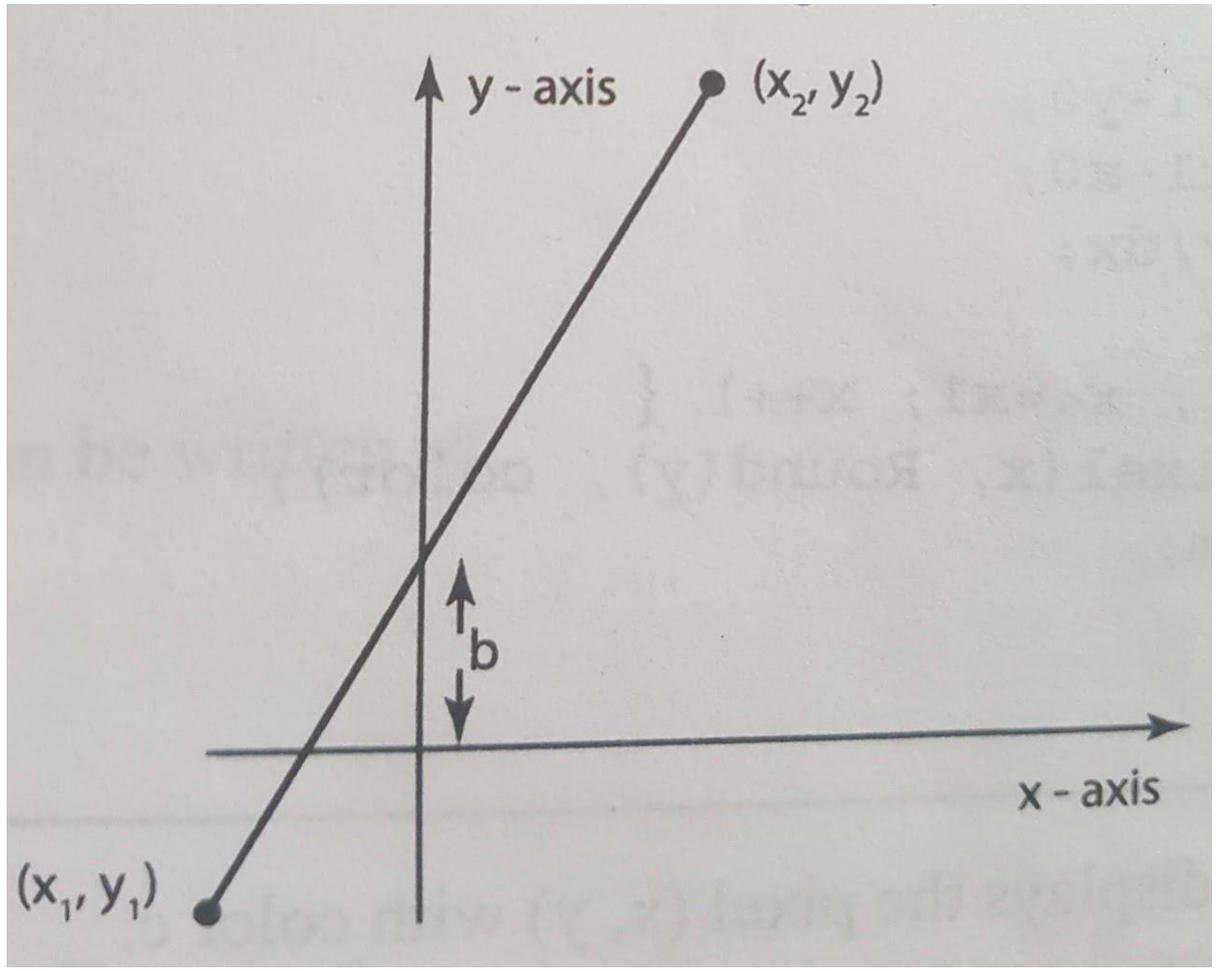
Lecture 2

Line Drawing Algorithms

Line Basics

Line equation: y=mx+b
or
f(x,y)=ax+by+c=0

By comparison
a=dy,b=-dx and c=B.dx



m=dy/dx
and
B= y-intercept

Line Basics

- For a point (xa,ya) above the line f(x,y)
f(xa,ya) is -ve

- For a point (xb,yb) below the line f(x,y)
f(xb,yb) is +ve

Given +ve a

DDA Algorithm

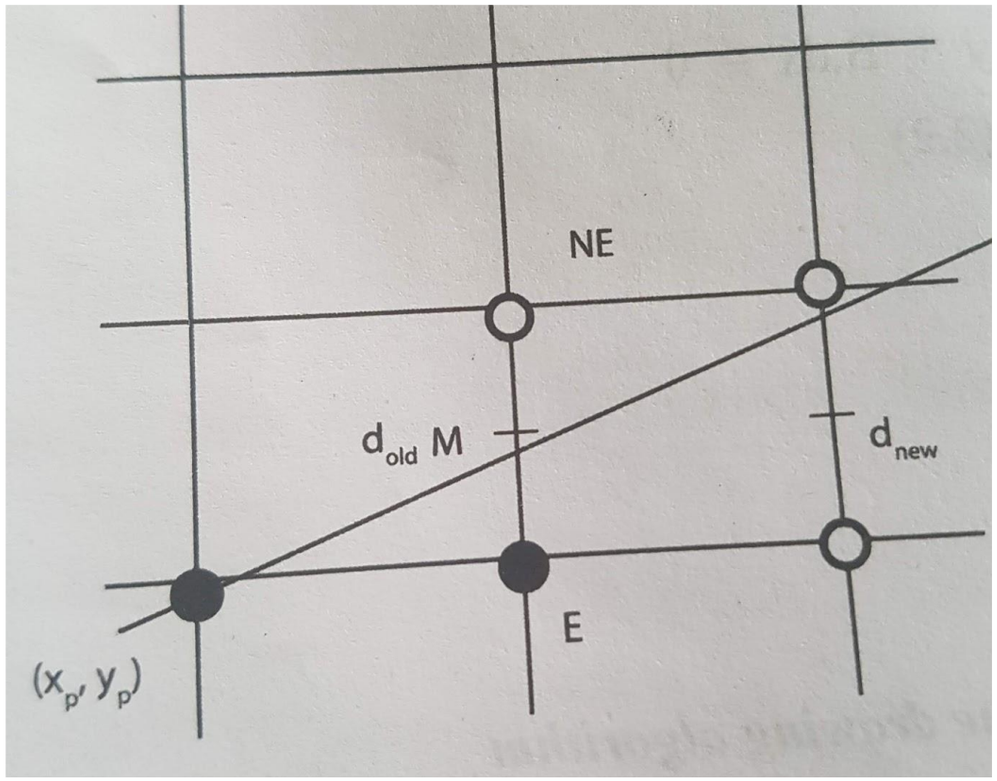
GRADIENT: Increase in y for a unit increase in x

ALGO:

Suppose line starts at pixel (xi,yi)
To reach next point on line:

- Increase x by 1, and
- Calculate the next y by Round(yi + m)
- Repeat the process with the new x and y

Bresenham line algo

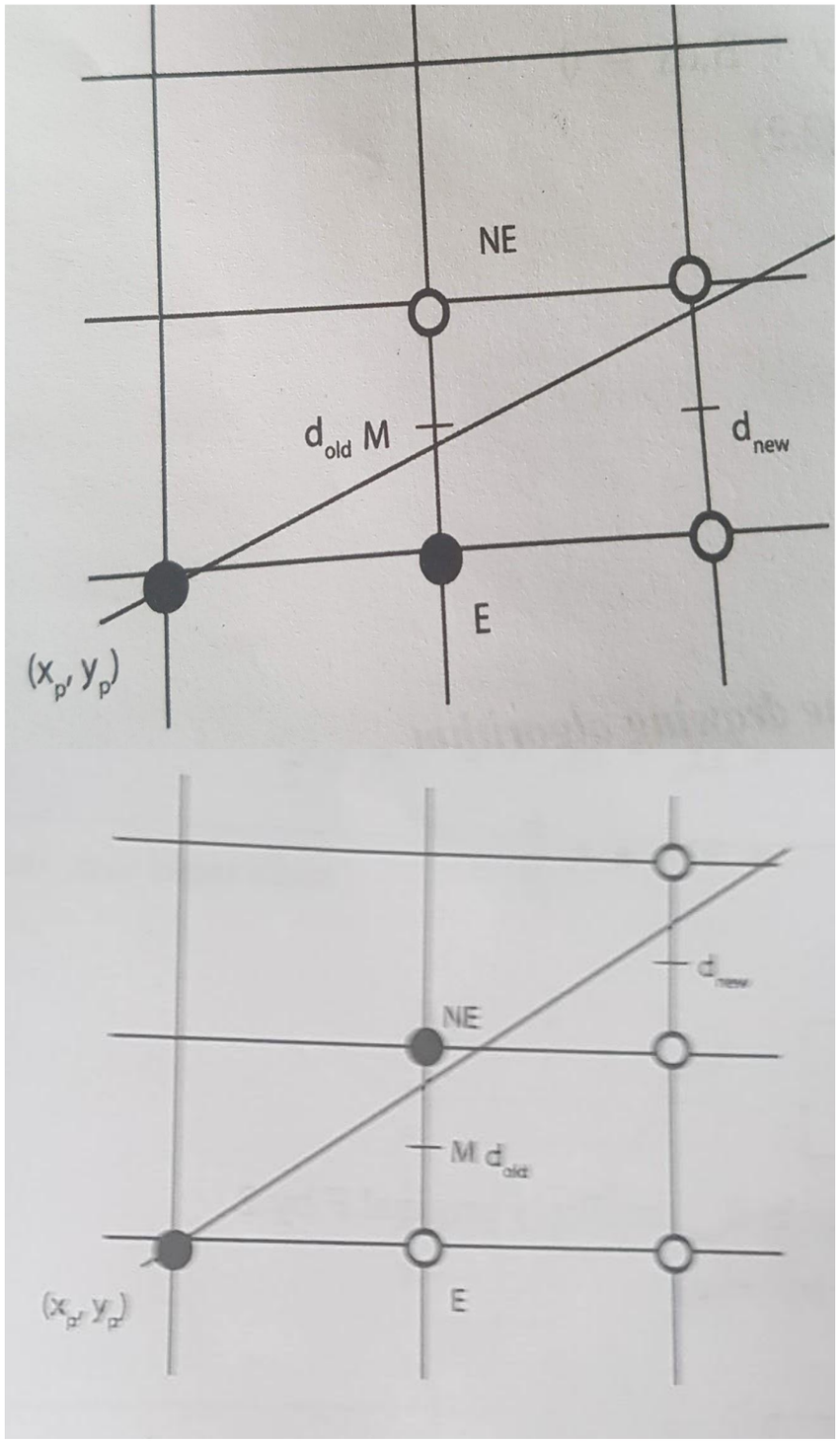


For lines in 1st quadrant with gradient between 0 and 1 (0<m<1)

Line starting point (xp,yp)
Then next point is either point E or point NE

choosing b/w E and NE

- Mid point b/w E and NE has coordinates (xp+1,yp+1/2)
- If midpoint is above the line choose E (i.e. f(xp+1,yp+1/2) will be -ve)
- If midpoint is below the line choose NE (i.e. f(xp+1,yp+1/2) will be +ve)



Decision variable

Let's say $f(xp+1,yp+1/2) = d_{start}$

- What's the value of d_{start} in terms of dy and dx ?
- If $d_{start} <= 0$ then E is next point else NE is next point on line
- If E is chosen how do we choose the next point?
- If NE is chosen how do we choose the next point?

Complete Bresenham Line Algo for m < 1

1. Input the twoline endpoints and store the left endpoint in (x0, y0)
2. Plot (x0,y0)
3. Calculate constants dx, dy, 2dy, and 2dy - dx, and obtain the starting value for the decision parameter as d_{start} or d_0= 2dy - dx
4. At each x_k along the line, starting at k = 0, perform the following test:
   a. If $d_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and $d_{k+1} = d_k + dy$ (**or $d_{k+1} = d_k + dE$**)
   b. Otherwise, the next point to plot is $(x_k + 1, y_k + 1)$ and $d_{k+1} = d_k + 2\,dy - 2\,dx$ (**or $d_{k+1} = d_k + dNE$**)
5. Repeat step 4 dx times.

Pseudo code

```
void MidpointLine(int x0, int y0, int x1, int y1, int color)
//Assuming the lines slope is between 0 and 1
  // (x0, y0) is lower left-end point
  // (x1, y1) is upper right-end point
{
   int dx, dy, deltaE, deltaNE, d, x, y;
   dx = x1 - x0;
   dy = y1 = y0;
   d = 2 * dy - dx;          // Initial value of d
   deltaE = 2 * dy;          // Increment used to move to East pixel
   deltaNE = 2 * (dy - dx);
   x = x0;
   y = y0;
      WritePixel(x, y, color) // The start pixel
   while (x<x1){
      if (d <=0) {
         d+=deltaE;
         x++;
      }
      else {
         d+=deltaNE;
         x++;
         y++;
      }
      WritePixel(x, y, color);
   }
}
```

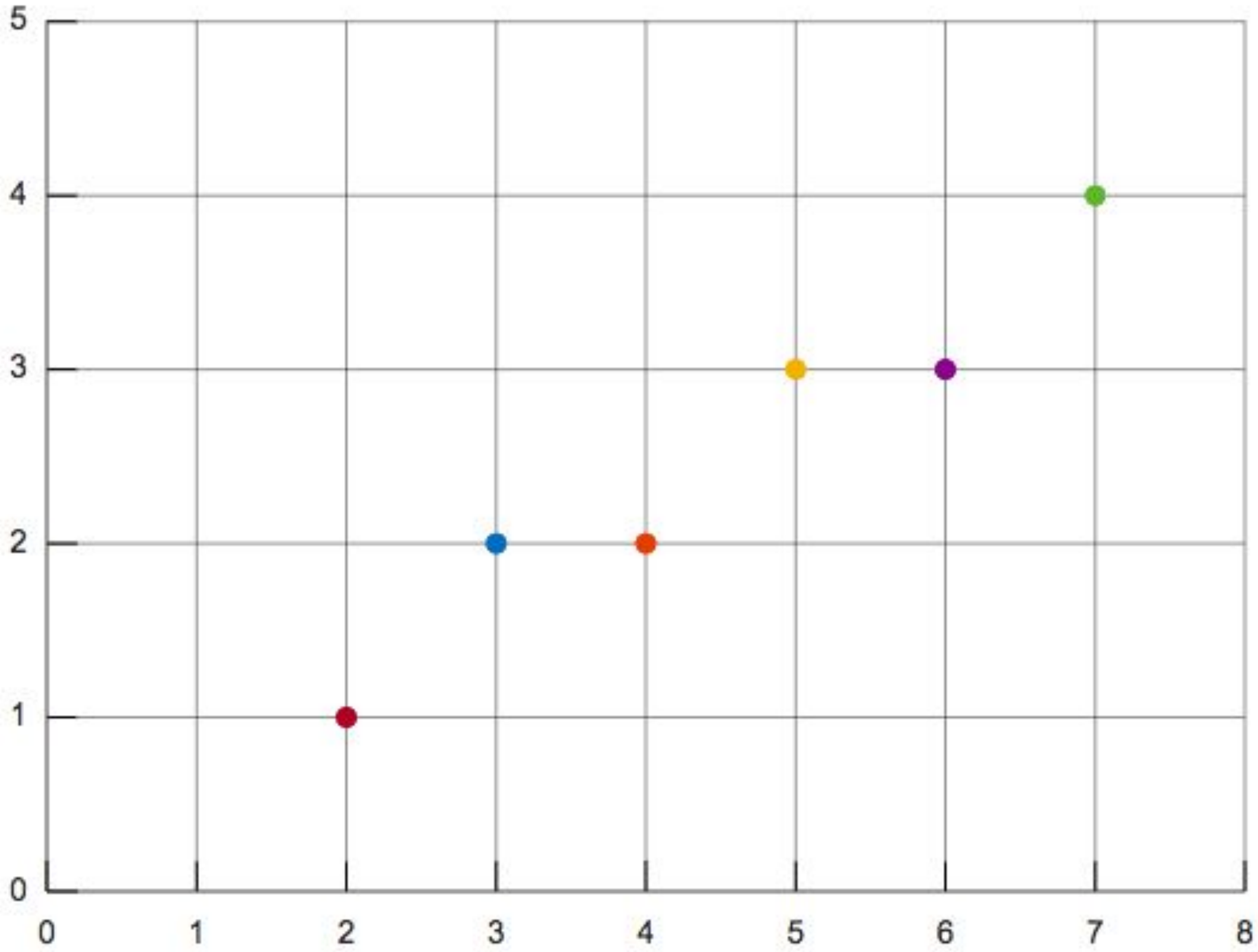
Example on board

- Plot a line between (2,1) and (7,4)
- how to make lines whose gradient are >1 or -ve

By Matlab Code

```
% testing Bresenham line algo
% Line between (2,1) and (7,4)

x0=2;y0=1;x1=7;y1=4;

x=x0;
y=y0;

dy = 4-1;
dx = 7-2;

d = 2 * dy - dx;

deltaE  = 2 * dy;
deltaNE = 2* (dy - dx);

scatter (x,y,60,"filled")
axis([0 8 0 5])
hold on
grid on
```

```
while (x < x1)
    if(d<=0)
        d=d+deltaE;
        x++;
    else
        d=d+deltaNE;
        x++;
        y++;
    end

    scatter(x,y,60,"filled");
end
```


5
4
3
2
1
0
0
1
2
3
4
5
6
7
8

The End

Slide 1

Text

Text

Text

–Johnny Appleseed