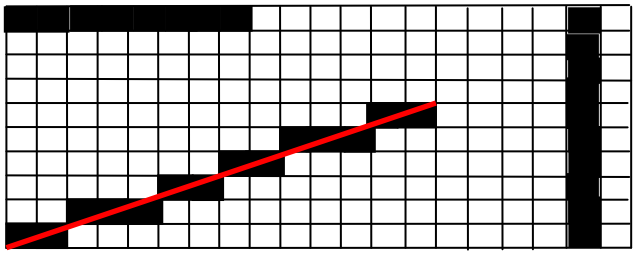
Generarea segmentelor

Fie **A(a,b)** şi **B(c,d)** două puncte (doi pixeli) de pe ecran şi presupunem că trebuie să se genereze punctele de pe segmentul cu extremităţile în aceste puncte. Deoarece ecranul este format dintr-o reţea de puncte (pixeli), în cazul în care segmentul nu este orizontal, sau vertical, atunci acesta apare "**zimţat**".



Ecuaţia dreptei care trece prin punctele A(a,b) şi B(c,d) este:

$$\frac{x-a}{c-a}=\frac{y-b}{d-b},$$

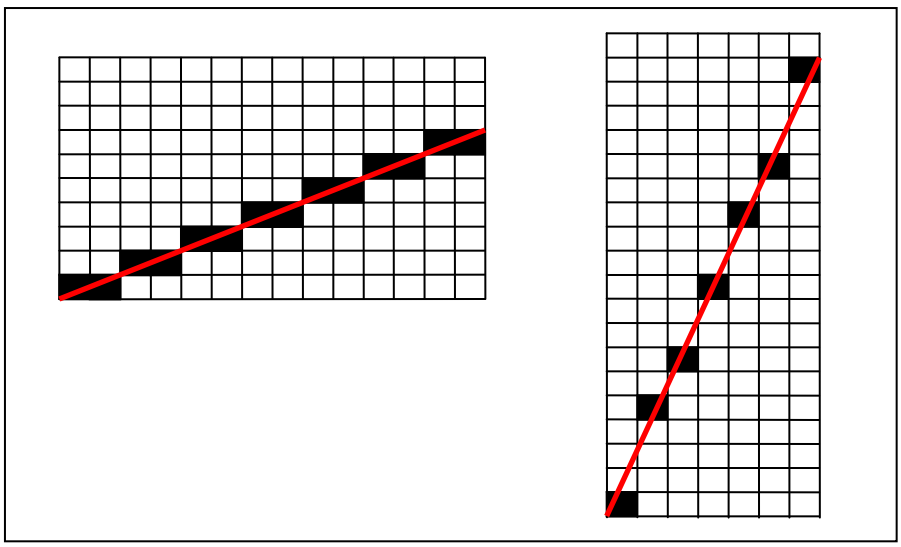
de unde obţinem:

(1) $$y=b+\frac{d-b}{c-a}(x-a)$$

Dacă c>a, atunci pentru **x=a, a+1, ..., c** se determină pixelul de pe coloana **x** (coloană de pixeli), aflat cel mai aproape de dreapta (1), deci:

(2) $$y=\left[b+\frac{d-b}{c-a}(x-a)+0.5\right]$$

Pentru perechea (x,y) astfel determinată se apelează funcţia **ScriePixel** (funcţie pentru pentru desenare, sau altă funcţie care foloseşte punctul generat)**.** Pentru ca formula (2) să se poată folosi şi în cazul în care c<a, atunci se determină S=sign(c-a) şi se modifică pasul lui x cu S (în loc de 1).
In figura următoare se arată modul de marcare a pixelilor în două situaţii.



In al doilea exemplu segmentul apare zimţat şi cu o anumită distanţă între doi pixeli consecutivi generaţi (pentru două coloane de pixeli), din cauză că segmentul este mai apropiat de verticală (în algoritm pe fiecare coloană de pixeli se alege un singur pixel). In acest caz trebuie să luăm: **y=b, b+S, ... , d**, unde S=sign(d-b), iar pentru fiecare **y** trebuie să se determine:

$$x = \left[a + \frac{c-a}{d-b}(y-b) + 0.5 \right].$$

Cu un astfel astfel de pixel **(x,y)** determinat se apelează funcţia **ScriePixel.**

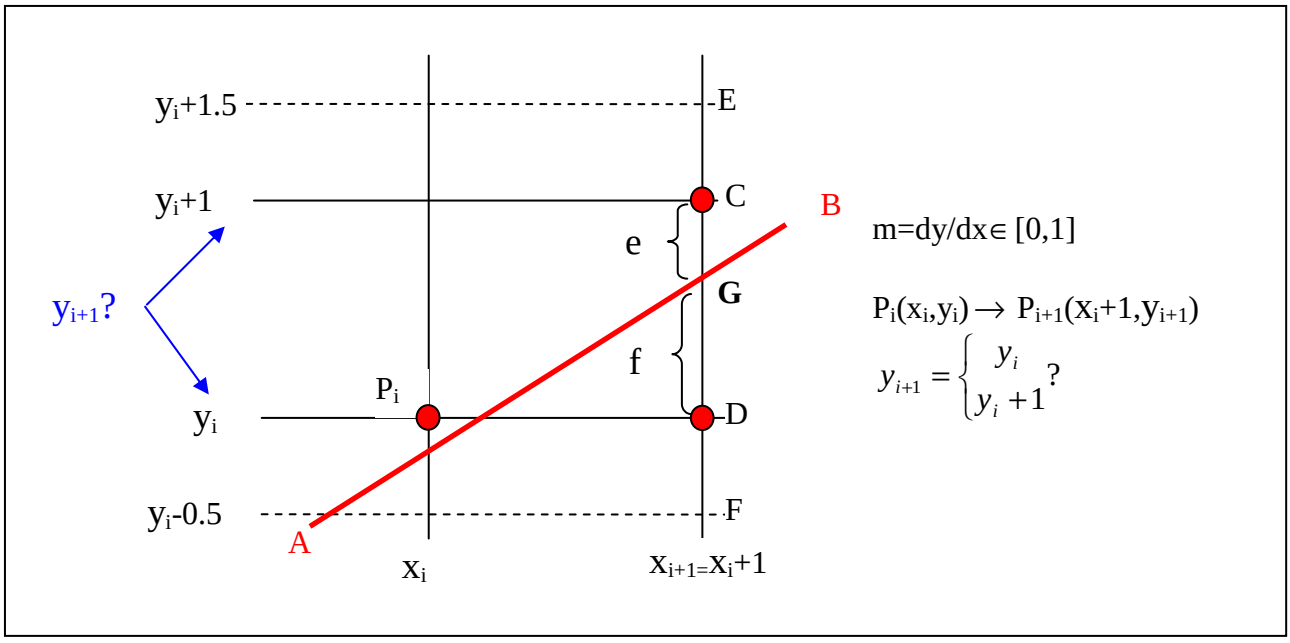
Algoritmul sugerat mai sus se poate sistematiza astfel:

```
1. Dacă a=c şi b=d, atunci ScriePunct(a,b);
2. Dacă |c-a|>=|d-b|, atunci:
     2.1. S=Sign(c-a);
     2.2. Pentru x=a, a+S, ..., c, execută:
        2.2.1. y=round(b+(d-b)/(c-a)*(x-a));
        2.2.2. ScriePixel(x,y);
   altfel:
     2.3. S=Sign(d-b);
     2.4. Pentru y=b, b+S, ..., d, execută:
        2.4.1. x=round(a+(c-a)/(d-b)*(y-b));
        2.4.2. ScriePixel(x,y);
```

Următorul algoritm, numit **algoritmul lui Bresenham**, efectuează operaţii numai cu numere întregi (deci este mult mai rapid).

Pentru a explica acest algoritm vom presupune, pentru început, că panta dreptei care trece prin punctele A şi B (coeficientul unghiular al acestei drepte) este în intervalul [0,1], iar la un moment dat (de exemplu la pasul **i** din algoritm) s-a marcat pixelul $\mathbf{P_i(x_i,y_i)}$, iniţial $\mathbf{P_0=A}$.

Din presupunerea făcută ($m \in [0,1]$) se deduce că dreapta (AB) va intersecta dreapta $x=x_i+1$ (deci la pasul următor, **i+1**) în punctul G aflat între E şi F, unde $C(x_i+1,y_i+1)$, $D(x_i+1,y_i)$, $E(x_i+1,y_i+1.5)$, $F(x_i+1,y_i-0.5)$, după cum se vede în figură.



Trebuie să marcăm pixelul C sau pixelul D. Alegerea între C şi D se face în funcţie de valorile **e** şi **f** (cu semnificaţia din figură). Dreapta (AB) va avea ecuaţia:

y=b+m(x-a),

unde: dx=c-a, dy=d-b, m=dy/dx.

De aici deducem coordonatele punctului **G**:

G(x_i+1, b-ma+m(x_i+1)).

Succesiv obţinem:

e = y_i + 1 - [b - m∗a+m∗(x_i+1]; f=[b - m∗a+m∗(x_i +1)] - y_i;

f-e = 2 ∗ [b-dy/dx∗a+dy/dx∗(x_i +1)] - 2∗y_i -1;

(f-e)∗dx = 2*(x_i∗dy-y_i∗dx)+2b∗dx-2a∗dy+2∗dy-dx.

Vom nota cu **d_i=(f-e)∗dx**, valoare pe care o asociem punctului **P_i**, deci:

(3) **d_i = 2*(x_i∗dy-y_i∗dx)+2b∗dx-2a∗dy+2∗dy-dx.**

Deoarece formula de mai sus este adevărată oricare ar fi **i**, deducem relaţia:

d_{i+1} = 2∗(x_{i+1}∗dy-y_{i+1}∗dx)+2b∗dx-2a∗dy+2∗dy-dx.

Pentru generarea punctelor putem alege oricum ordinea extremităţile segmentului AB, putem presupune că **dx>0** (deci a<c). In acest caz semnul lui **(f-e)** este dat de semnul lui **d_i**. Pentru a determina o formulă de recurenţă în calculul lui **d_i**, vom evalua diferenţa dintre două valori consecutive **(d_{i+1}-d_i)**:

d_{i+1}-d_i = 2[(x_{i+1}-x_i)∗dy-(y_{i+1}-y_i) ∗dx].

Deoarece: **x_{i+1} = x_i + 1**, **y_{i+1} = y_i sau y_i + 1**, se obţine:

(2) $$d_{i+1} = d_i + \begin{cases} 2*dy, & daca\ y_{i+1} = y_i, (sau\ daca\ d_i < 0) \\ 2*(dy-dx), & daca\ y_{i+1} = y_i + 1, (sau\ daca\ d_i \geq 0) \end{cases}$$

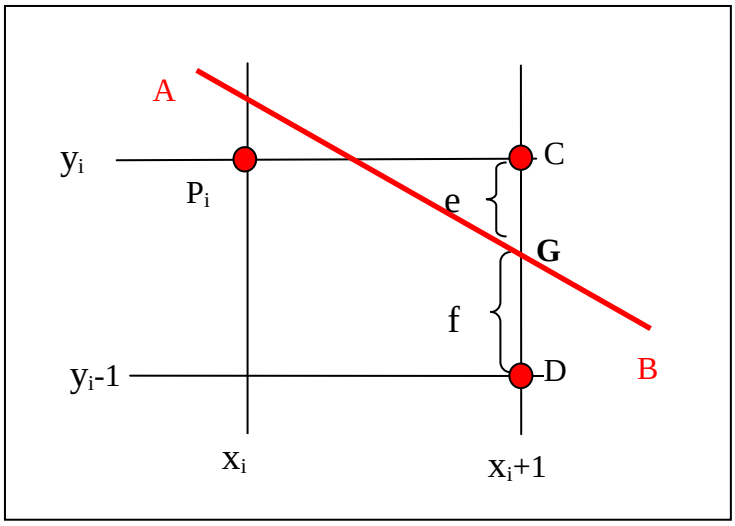
Dacă luăm **P_0 = A(a,b)**, atunci x_0=a, y_0=b, iar din (3) obţinem:

d_0 = 2∗(x_0∗dy-y_0∗dx)+2b∗dx-2a∗dy+2∗dy-dx,
d_0 = 2∗dy-dx.

La pasul **i**,

- dacă **d_i=(f-e)∗dx >=0**, atunci se marchează **P_{i+1} = C**, şi **y_{i+1} = y_i + 1, d_{i+1}=d_i+2(dy-dx)**,
- iar dacă **d_i<0 (deci f-e<0)**, atunci se va marca punctul **P_{i+1} = D, şi y_{i+1} = y_i, d_{i+1}=d_i+2dy**.

Dacă **m=dy/dx ∈ [-1,0)**, atunci apare următoarea figură.

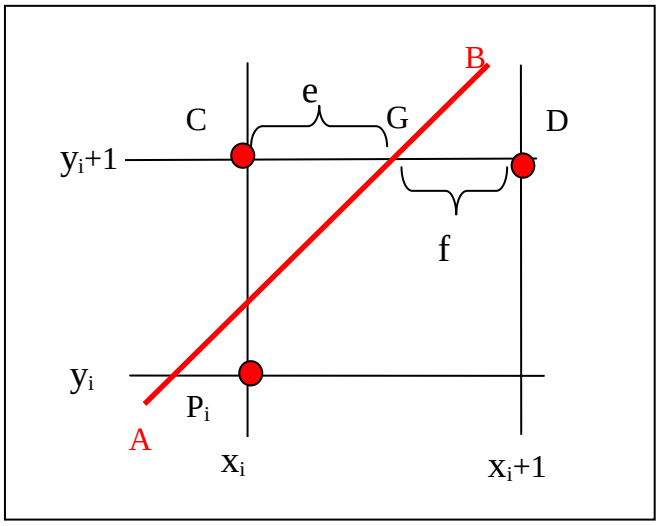


Refăcând toate calculele precedentele, în **ipoteza dx>0**, obţinem relaţia de recurenţă:

$$d_{i+1} = d_i + \begin{cases} 2*dy, & daca\ y_{i+1} = y_i, (sau\ daca\ d_i \geq 0) \\ 2*(dx+dy), & daca\ y_{i+1} = y_i + 1, (sau\ daca\ d_i < 0) \end{cases}$$

iar valoarea iniţială este $\mathbf{d_0 = dx + 2*dy}$.

Pentru cazul **m=dy/dx>1**, considerăm figura următoare:



Fără explicaţii, relaţiile deduse în cazurile precedente devin:

(AB): y-b=m(x-a),

x=a+$\frac{1}{m}$ (y-b), sau: x=a+$\frac{dx}{dy}$ (y-b),

G(a+$\frac{dx}{dy}$ (y_i+1-b), y_i+1),

e-f = 2∗[a+dx/dy∗(y_i+1-b)]-2∗x_i -1;

(e-f)∗dy = 2*(y_i∗dx-x_i∗dy)+2a∗dy-2b∗dx+2∗dx-dy = d_i,

d_{i+1} = 2∗(y_{i+1}∗dx-x_{i+1}∗dy)+2a∗dy-2b∗dx+2∗dx-dy,

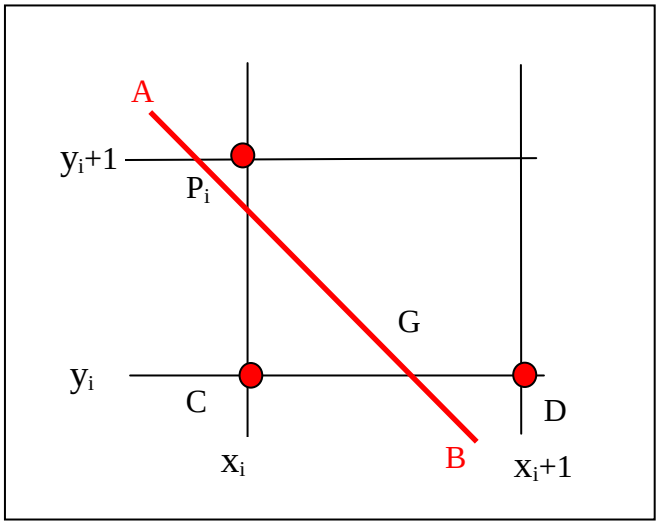
$d_{i+1}-d_i = 2[(y_{i+1}-y_i)*dx-(x_{i+1}-x_i)*dy] = \begin{cases} 2*dx, & daca\ x_{y+1}=x_i \\ 2*(dx-dy), & daca\ x_{i+1}=x_i+1 \end{cases}$

deoarece $(y_{i+1}-y_i)=1$ şi $(x_{i+1}-x_i)=0$ sau 1.

Presupunând **dy>0** (b<d), atunci semnul lui (e-f) este dat de semnul lui $\mathbf{d_i}$, deci:

$$d_{i+1} = d_i + \begin{cases} 2*dx, & daca\ x_{i+1}=x_i, (sau\ daca\ d_i<0) \\ 2*(dx-dy), & daca\ x_{i+1}=x_i+1, (sau\ daca\ d_i \geq 0) \end{cases}$$

Pentru P_0=A(a,b), obţinem $\mathbf{d_0 = 2*dx-dy}$.
Pentru cazul **m=dy/dx<-1**, avem nevoie de figura următoare:



In ipoteza dy>0 obţinem relaţia de recurenţă:

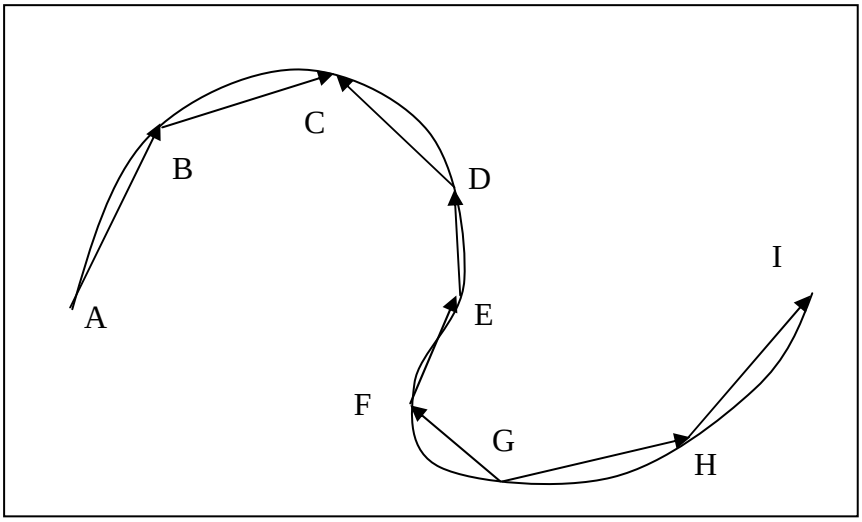
$$d_{i+1} = d_i + \begin{cases} 2*dx, & daca\ x_{i+1}=x_i \\ 2*(dx+dy), & daca\ x_{i+1}=x_i+1 \end{cases}$$

şi valoarea de plecare este $\mathbf{d_0 = -2*dx-dy}$.
Prin **sistematizarea** acestor calcule obţinem tabelul următor. In partea de jos a tabelului se află valorile iniţiale ale variabilelor şi relaţiile de modificare necesare în cadrul procedurii de generare.

m	**[0,1]**	**[-1,0)**	**(1,∞)**	**(-∞,-1)**
Condiţie	dx>0	dx>0	dy>0	dy>0
d_0	2*dy-dx	2*dy+dx	2*dx-dy	-2*dx-dy
d_i>=0	$P_{i+1}(x_i+1, y_i+1)$ $d_{i+1}=d_i+2*(dy-dx)$	$P_{i+1}(x_i+1, y_i)$ $d_{i+1}=d_i+2*dy$	$P_{i+1}(x_i+1, y_i+1)$ $d_{i+1}=d_i+2*(dx-dy)$	$P_{i+1}(x_i, y_i+1)$ $d_{i+1}=d_i+2*dx$
d_i<0	$P_{i+1}(x_i+1, y_i)$ $d_{i+1}=d_i+2*dy$	$P_{i+1}(x_i+1, y_i-1)$ $d_{i+1}=d_i+2*(dx+dy)$	$P_{i+1}(x_i, y_i+1)$ $d_{i+1}=d_i+2*dx$	$P_{i+1}(x_i-1, y_i+1)$ $d_{i+1}=d_i+2*(dx+dy)$
Valori iniţiale	dd=2*dy-dx S1=1 p1=2*(dy-dx) S2=0 p2=2*dy	dd=2*dy+dx S1=0 p1=2*dy S2=-1 p2=2*(dx+*dy)	dd=2*dx-dy S1=1 p1=2*(dx-dy) S2=0 p2=2*dx	dd=2*dx+dy S1=0 p1=2*dx S2=-1 p2=2*(dx+dy)
dd>=0	**P(x+1,y+S1) x=x+1, y=y+S1 dd=dd+p1**		**P(x+S1,y+1) x=x+S1, y=y+1 dd=dd+p1**	
dd<=0	**P(x+1,y+S2) x=x+1, y=y+S2 dd=dd+p2**		**P(x+1,y+S2) x=x+1, y=y+S2 dd=dd+p2**	

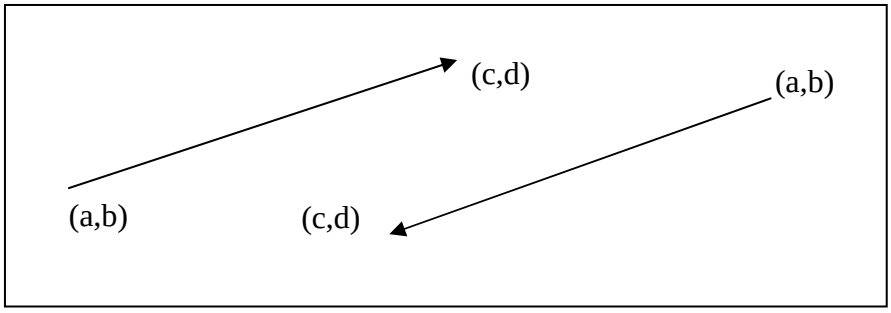
Algoritmul lui Bresenham, descris mai sus, se poate folosi şi pentru desenarea unui segment de dreaptă conform unui "**model**". Un astfel de model este precizat printr-un şir de biţi: $\mathbf{b_0b_1...b_i...b_{p-1}}$. Bitul $\mathbf{b_i}$ egal cu **1** precizează că al **i**-lea pixel de pe segment se desenează, iar valoarea **0** pentru acest bit va cere ca punctul corespunzător să nu se deseneze. Modelul definit prin p biţi se va repeta dacă pe segment sunt mai mult de **p** puncte..

O curbă se desenează, în modul cel mai simplu, printr-o linie poligonală cu vârfurile pe curbă. Algoritmul lui Bresenham se poate folosi foarte bine pentru desenarea fiecărui segment de dreaptă, dacă desenul se face fără un model de biţi. Pentru desenarea curbei (printr-o reuniune de segnente) cu ajutorul unui model, metoda amintită nu mai poate fi folosită deoarece poate produce modele neregulate, aşa cum se vede din următorul exemplu. Deoarece pentru fiecare segment avem o conţiţie de respectat (dx>0 sau dy>0),



rezultă că segmentele se vor desena plecând de la o extremitate (pe figură este marcat sensul de generare a punctelor pentru fiecare segment). Se observă că la desenarea curbei ar trebui ca modelul segmentului AB să fie continuat în segmentul CD. La fel ar trebui să se facă o continuare a modelului în fiecare punct intermediar. Analizând ordinea de desenare a segmentelor se observă că această regulă nu este respectată în punctele C şi G. Pentru a reuşi desenarea propusă avem nevoie de o modificare a algoritmului lui Bresenham astfel încât să nu mai apară condiţia dx>0 sau dy>0, deci desenul să se facă prin generarea pixelilor de la punctul iniţial la cel final.Refăcând toate calculele anterioare obţinem formulele ce vor fi date în continuare. Din rezultatele obţinute se observă că noul algoritm efectuează acelaşi număr de operaţii ca varianta descrisă anterior.

0<=m<=1



Fie S=1 dacă a<c şi S=-1 dacă a>c

$d_i=(f-e)*dx$

$d_i=2*(x_i*dy-y_i*dx)+2b*dx-2a*dy+2S*dy-S*dx$

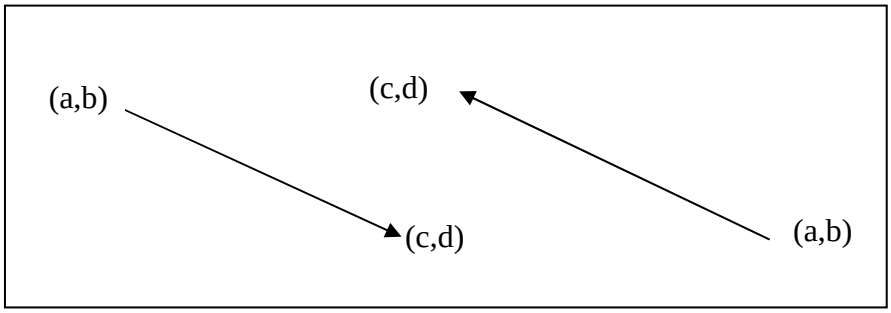
$d_{i+1}=d_i+2*[dy*(x_{i+1}-x_i)-dx*(y_{i+1}-y_i)]$

$d_0=2S*dy-S*dx$

Dacă $d_i>=0$, atunci $P_{i+1}(x_i+S, y_i+S)$, $d_{i+1}=d_i+2*(dy*S-dx*S)=d_i+2S*(dy-dx)$

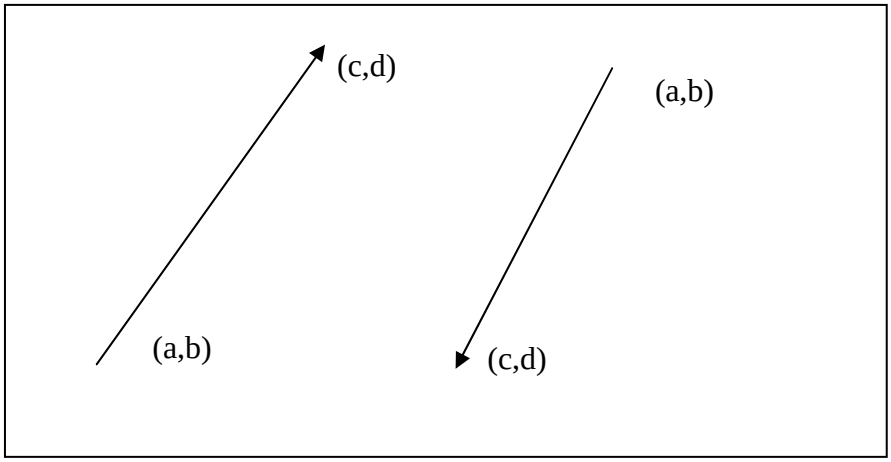
Dacă $d_i<0$, atunci $P_{i+1}(x_i+S, y_i)$, $d_{i+1}= d_i+2S*dy$.

-1<=m<0



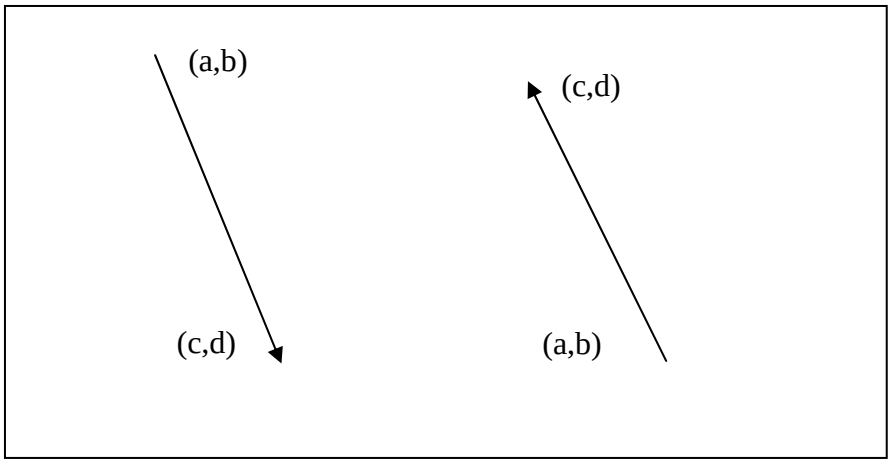
Fie S=1 dacă a<c şi S=-1 dacă a>c
$d_i=(f-e) * dx$
$d_i=2*(x_i*dy-y_i*dx)+2b*dx-2a*dy+2S*dy+S*dx$
$d_{i+1}=d_i+2*[dy*(x_{i+1}-x_i)-dx*(y_{i+1}-y_i)]$
$d_0=2S*dy+S*dx$
Dacă $d_i>=0$, atunci $P_{i+1}(x_i+S, y_i)$, $d_{i+1}= d_i+2S*dy$
Dacă $d_i<0$, atunci $P_{i+1}(x_i+S, y_i-S)$, $d_{i+1}= d_i+2S*(dx+dy)$.

1<m<∞



Fie S=1 dacă b<d şi S=-1 dacă b>d
$d_i=(e-f) * dy$
$d_i=2*(y_i*dx-x_i*dy)+2a*dy-2b*dx+2S*dx-S*dy$
$d_{i+1}=d_i+2*[dx*(y_{i+1}-y_i)-dy*(x_{i+1}-x_i)]$
$d_0=2S*dx-S*dy$
Dacă $d_i>=0$, atunci $P_{i+1}(x_i+S, y_i+S)$, $d_{i+1}= d_i+2S*(dx-dy)$
Dacă $d_i<0$, atunci $P_{i+1}(x_i, y_i+S)$, $d_{i+1}= d_i+2S*dx$.

-∞<m<-1



Fie S=1 dacă b<d şi S=-1 dacă b>d

$d_i=(e-f) * dy$

$d_i=2*(y_i*dx-x_i*dy)+2a*dy-2b*dx+2S*dx+S*dy$

$d_{i+1}=d_i+2*[dx*(y_{i+1}-y_i)-dy*(x_{i+1}-x_i)]$

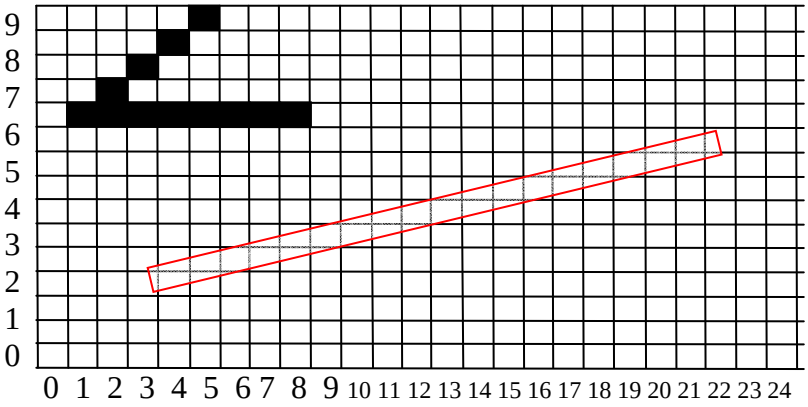
$d_0=2S*dx+S*dx$

Dacă $d_i>=0$, atunci $P_{i+1}(x_i, y_i+S)$, $d_{i+1}= d_i+2S*dx$

Dacă $d_i<0$, atunci $P_{i+1}(x_i-S, y_i+S)$, $d_{i+1}= d_i+2S*(dx+dy)$.

	dx=c-a; dy=d-b;			
	S=1 daca a<c, altfel S=-1		**S=1 daca b<d, altfel S=-1**	
m	**[0,1]**	**[-1,0)**	**(1,∞)**	**(-∞,-1)**
d_0	**2S*dy-S*dx**	**2S*dy+S*dx**	**2S*dx-S*dy**	**- 2S*dx-S*dy**
$d_i>=0$	$P_{i+1}(x_i+S, y_i+S)$ $d_{i+1}=d_i+2S*(dy-dx)$	$P_{i+1}(x_i+S, y_i)$ $d_{i+1}=d_i+2S*dy$	$P_{i+1}(x_i+S, y_i+S)$ $d_{i+1}=d_i+2S*(dx-dy)$	$P_{i+1}(x_i, y_i+S)$ $d_{i+1}=d_i+2S*dx$
$d_i<0$	$P_{i+1}(x_i+S, y_i)$ $d_{i+1}=d_i+2S*dy$	$P_{i+1}(x_i+S, y_i-S)$ $d_{i+1}=d_i+2S*(dx+dy)$	$P_{i+1}(x_i, y_i+S)$ $d_{i+1}=d_i+2S*dx$	$P_{i+1}(x_i-S, y_i+S)$ $d_{i+1}=d_i+2S*(dx+dy)$
Valori iniţiale	dd=S*(2*dy-dx) S1=S p1=2S (dy-dx) S2=0 p2=2S*dy	dd=S*(2*dy+dx) S1=0 p1=2S dy S2=-S p2=2S*(dx+*dy)	dd=S*(2*dx-dy) S1=S p1=2S*(dx-dy) S2=0 p2=2S*dx	dd=S*(2*dx+dy) S1=0 p1=2S*dx S2=-S p2=2S*(dx+dy)
dd	**2S*dy-S*dx**	**2S*dy+S*dx**	**2S*dx-S*dy**	**- 2S*dx-S*dy**
dd>=0	**P(x+S,y+S1) x=x+S, y=y+S1 dd=dd+p1**		**P(x+S1,y+S) x=x+S1, y=y+S dd=dd+p1**	
dd<=0	**P(x+S,y+S2) x=x+S, y=y+S2 dd=dd+p2**		**P(x+S,y+S2) x=x+S, y=y+S dd=dd+p2**	

Dacă segmentul care se desenează nu este orizontal sau veritical, atunci el apare "zimţat" pe ecranul unui calculator. Pentru a elimina, în parte, acest defect, se pot folosi diverse tehnici de "netezire" (antialiasing) a segmentului. Pentru aceasta vom presupune că un punct (pixel) de pe ecran are o arie nenulă, iar segmentul ce se desenează are o grosime nenulă.



In exemplul de mai sus apare desenat un segment orizontal, unul înclinat la 45^0, şi un segment suprapus peste o reţea de pixeli. Fiecare pixel acoperit, total sau parţial, de acest segment, ar trebui să aibă o intensitate a culorii proporţională cu porţiunea sa acoperită de segment. Dacă desenăm segmentul cu culoarea neagră, atunci pixelul (4,2) este aproximativ 75% negru, (5,2) este aproximativ 30% negru, (5,3) aprox. 70% negru, etc.