



ШУТИС, Мэдээлэл Холбооны Технологийн Сургууль

F.CS209 Компьютерийн график

Лекц 6 – Тойрог зурах алгоритмууд

Боловсруулсан багш: Х.Хулан

2022 он

Өнөөдрийн хичээлээр бид дараах зүйлийг
судлах болно:

- Bresenham-ийн шулууныг зурах алгоритм
- Тойрог зурах алгоритм
 - Энгийн арга
 - mid-point тойрог алгоритм

Bresenham Line Algorithm

Bresenham алгоритм нь өсөн нэмэгдэх scan conversion алгоритм юм. Энэ алгоритмын хамгийн том давуу тал нь зөвхөн бүхэл тооны тооцоолол ашигладагт оршино.



Жак Брезенхэм академид орохоосоо өмнө IBM-д 27 жил ажилласан. Брезенхам өөрийн алдартай алгоритмуудаа 1960-аад оны эхээр IBM-д ажиллаж байхдаа боловсруулсан.

Тойрог зурах энгийн алгоритм

Тойргийн томъёо нь:

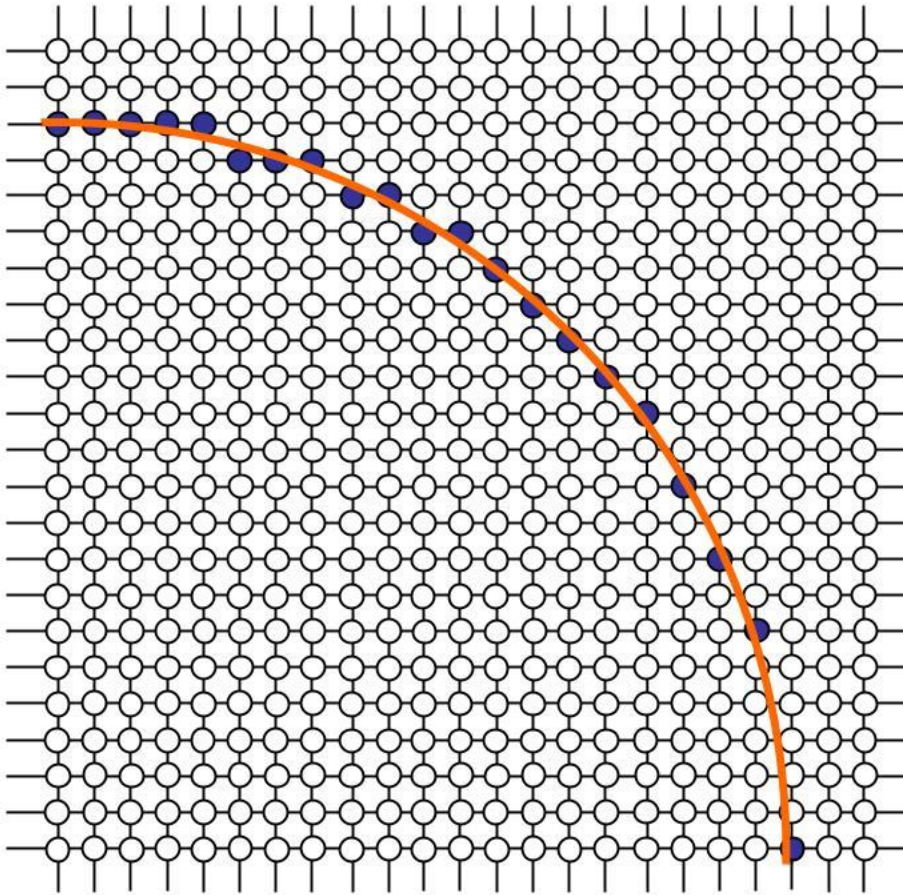
$$x^2 + y^2 = r^2$$

r нь тойргийн радиус

Иймд бид нэгж x интервал дээрх y
томъёог шийдэх замаар энгийн тойрог
зурах алгоритмыг бичих боломжтой:

$$y = \pm \sqrt{r^2 - x^2}$$

Тойрог зурах энгийн алгоритм(cont..)



$$y_0 = \sqrt{20^2 - 0^2} \approx 20$$

$$y_1 = \sqrt{20^2 - 1^2} \approx 20$$

$$y_2 = \sqrt{20^2 - 2^2} \approx 20$$

⋮

$$y_{19} = \sqrt{20^2 - 19^2} \approx 6$$

$$y_{20} = \sqrt{20^2 - 20^2} \approx 0$$

Тойрог зурах энгийн алгоритм

Энэ нь тийм ч сайн шийдэл биш юм!

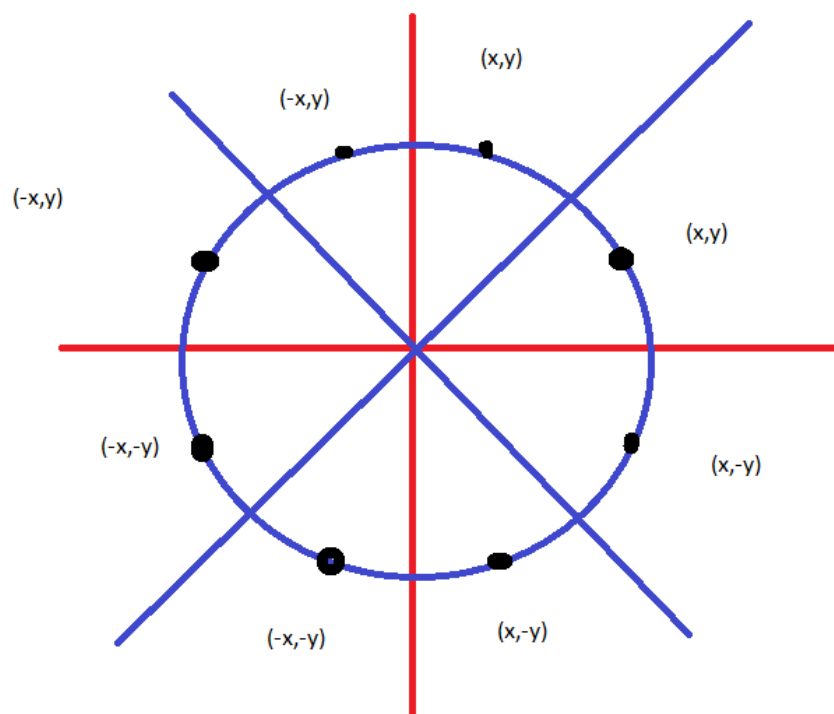
Нэгдүгээр, тойргийн үр дүн налуу нь
босоо тэнхлэгт ойртох том зайтай байдаг
Хоёрдугаарт, тооцоолол тийм ч үр дүнтэй
биш байна

- квадрат (үржүүлэх) үйлдлүүд
- Квадрат язгуур авах үйлдлүүд– эдгээрээс
зайлсхийх хэрэгтэй!

Бидэнд илүү үр дүнтэй, илүү нарийвчлалтай
шийдэл хэрэгтэй.

Найм-талт тэгш хэм

Тойрог зурах алгоритмаа илүү үр дүнтэй болгохын тулд бидний анзаарч байгаа хамгийн эхний зүйл бол $(0, 0)$ төвтэй тойрог нь найман талын тэгш хэмтэй байх явдал юм.



Дундаж цэгийн тойргийн алгоритм (Mid-Point Circle Algorithm)

Шулууны тохиолдлууд ижил
нөхцөлд тойрог зурах өсөлтийн
алгоритм бий
mid-point circle algorithm

Уг алгоритмын хувьд:

Найман-талт тэгш хэмийг
ашиглах тул зөвхөн тойргийн
баруун дээд цэгийг тооцоолох
ба дараа нь тэгш хэмийг
ашиглан үлдсэн цэгүүдийг
авна.



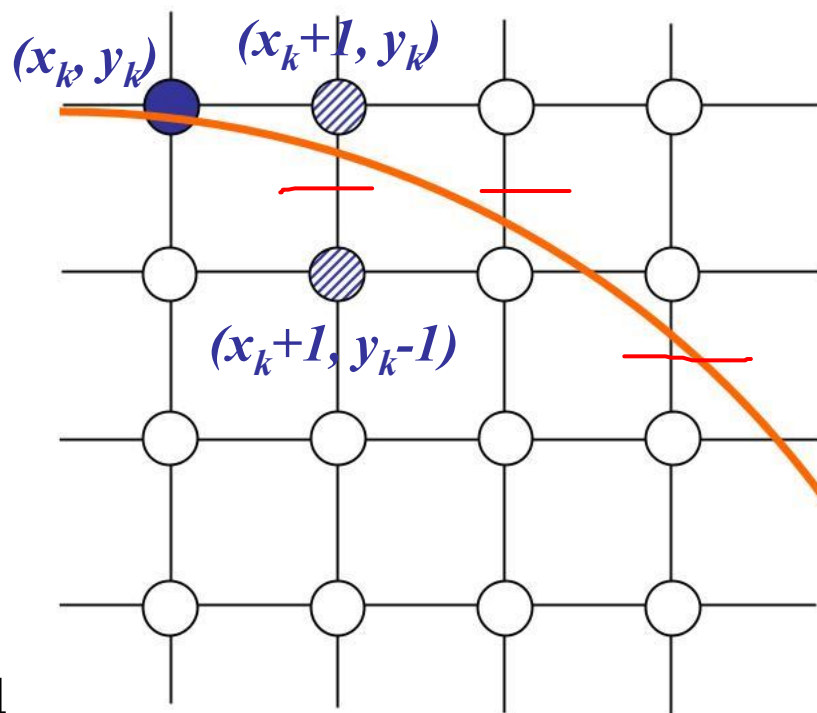
mid-point circle
algorithm-ийг бидний
өмнө сонсож байсан
Jack Bresenham
хөгжүүлсэн.

Дундаж цэгийн тойргийн алгоритм (cont..)

(x_k, y_k) цэгийг тэмдгэлсэн гэж үзье.

Дараагийн цэг бол (x_k+1, y_k)
 $y_k)$
ба (x_k+1, y_k-1) хоорондох
сонголт юм.

Бид тойрогтой хамгийн
ойрхон цэгийг сонгоно. Бид
энэ сонголтыг хэрхэн хийх
вэ?



Дундаж цэгийн тойргийн алгоритм (cont..)

Бидэнд өгөгдсөн тойргийн тэгшитгэлийг
бага зэрэг давтъя:

$$f_{circ}(x, y) = x^2 + y^2 - r^2$$

Тэгшитгэлийг дараах байдлаар үнэлнэ:


$$f_{circ}(x, y) \begin{cases} < 0, \text{ if } (x, y) \text{ тойргийн хил дотор} \\ = 0, \text{ if } (x, y) \text{ тойргийн хил дэр} \\ > 0, \text{ if } (x, y) \text{ тойргийн хилээс гадна} \end{cases}$$

Оролцогч цэгүүдийн хоорондох дундаж цэг дээр энэ
функцийг тооцоолсоноор бид шийдвэрээ гаргана.

Дундаж цэгийн тойргийн алгоритм (cont...)

Бид (x_k, y_k) дээрх цэгийг тэмдгэлсэн гэж үзвэл $(x_k + 1, y_k)$ ба $(x_k + 1, y_k - 1)$ хоорондох сонголтыг хийнэ.

Бидний шийдвэрийн хувьсагч:


$$p_k = f_{\text{circ}}(x_k + 1, y_k - \frac{1}{2})$$

$$x^2 + y^2 - r^2$$

$$= (x_k + 1)^2 + (y_k - \frac{1}{2})^2 - r^2$$

$p_k < 0$ бол дундаж цэг тойргийн дотор орших ба y_k нь тойрогтой ойр оршино.

Эсэргээрээ дундаж цэг гадна оршиж байвал $y_k - 1$ ойр байна

$$p_k > 0$$

Дундаж цэгийн тойргийн алгоритм (cont...)

Аливаа зүйлийг аль болох үр дүнтэй байлгахын тулд бүх тооцоогоо алхам алхмаар хийнэ

Эхлээд:

$$\begin{aligned} p_{k+1} &= f_{\text{circ}} \left(x_{k+1} + 1, y_{k+1} - \frac{1}{2} \right) \\ &= [(x_k + 1) + 1]^2 + \left(y_{k+1} - \frac{1}{2} \right)^2 - r^2 \end{aligned}$$

буюу:

$$p_{k+1} = p_k + 2(x_k + 1) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 1$$

y_{k+1} нь p_k хамаарч y_k эсвэл $y_k - 1$ байна.

Дундаж цэгийн тойргийн алгоритм (cont..)

Эхний шийдвэрийн хувьсагч дараах байдлаар
өгөгдсөн:

$$\begin{aligned} p_0 &= f_{circ}(1, r - 1/2) \\ &= 1 + (r - 1/2)^2 - r^2 \\ &= 5/4 - r \end{aligned}$$

Хэрэв $p_k < 0$ бол дараагийн шийдвэрийн
хувьсагч:

$$p_{k+1} = p_k + 2x_{k+1} + 1$$

$p_k > 0$ бол шийдвэрийн хувьсагч:

$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_k + 1$$

Дундаж цэгийн тойргийн алгоритм

MID-POINT CIRCLE ALGORITHM

- Input radius r and circle centre (x_c, y_c) , then set the coordinates for the first point on the circumference of a circle centred on the origin as:

$$(x_0, y_0) = (0, r)$$

- Calculate the initial value of the decision parameter as:

$$p_0 = \frac{5}{4} - r$$

- Starting with $k = 0$ at each position x_k , perform the following test. If $p_k < 0$, the next point along the circle centred on $(0, 0)$ is (x_{k+1}, y_k) and:

$$p_{k+1} = p_k + 2x_{k+1} + 1$$

Дундаж цэгийн тойргийн алгоритм (cont..)

Otherwise the next point along the circle is (x_k+1, y_k-1) and:

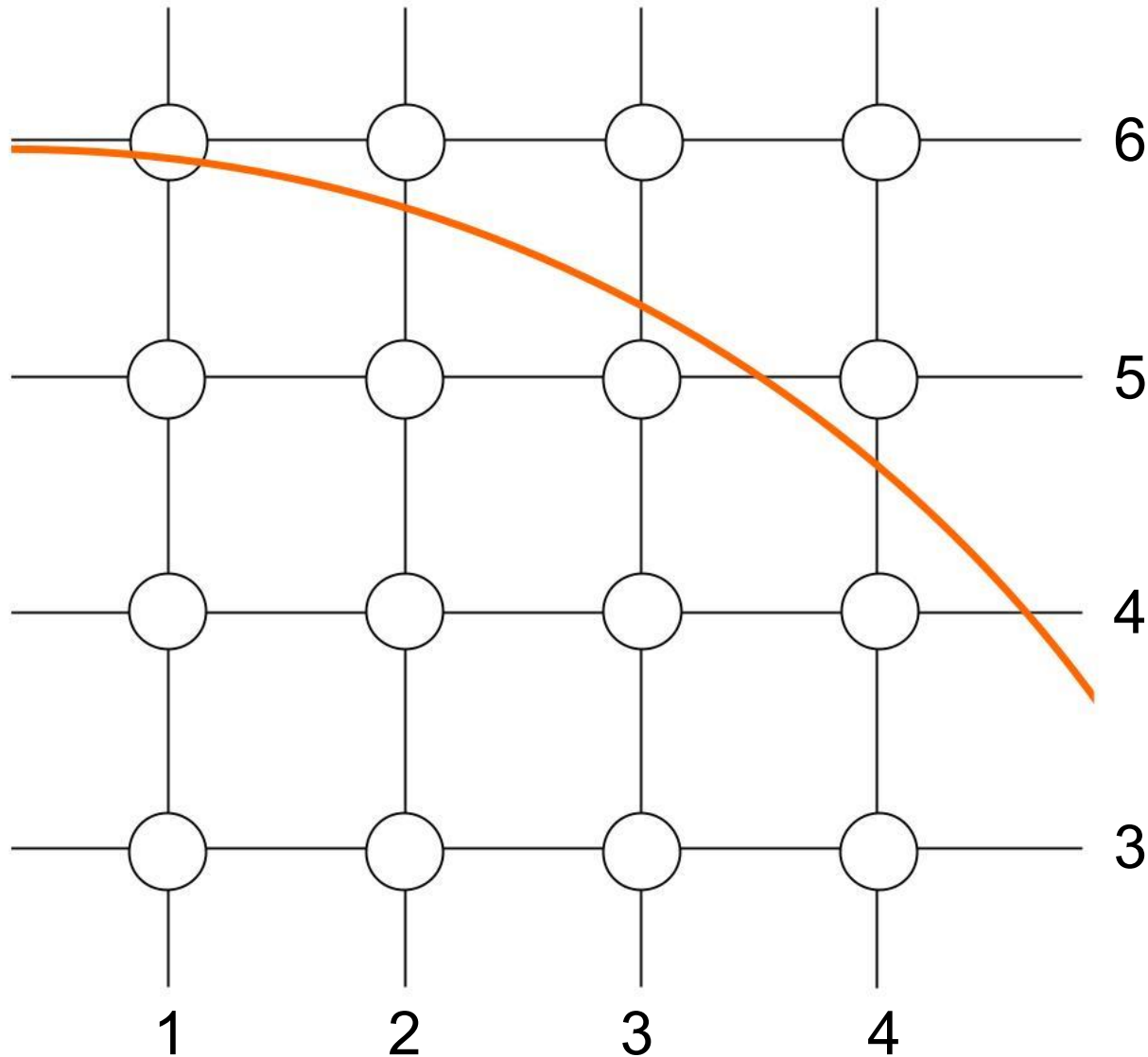
$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$

4. Determine symmetry points in the other seven octants
5. Move each calculated pixel position (x, y) onto the circular path centred at (x_c, y_c) to plot the coordinate values:

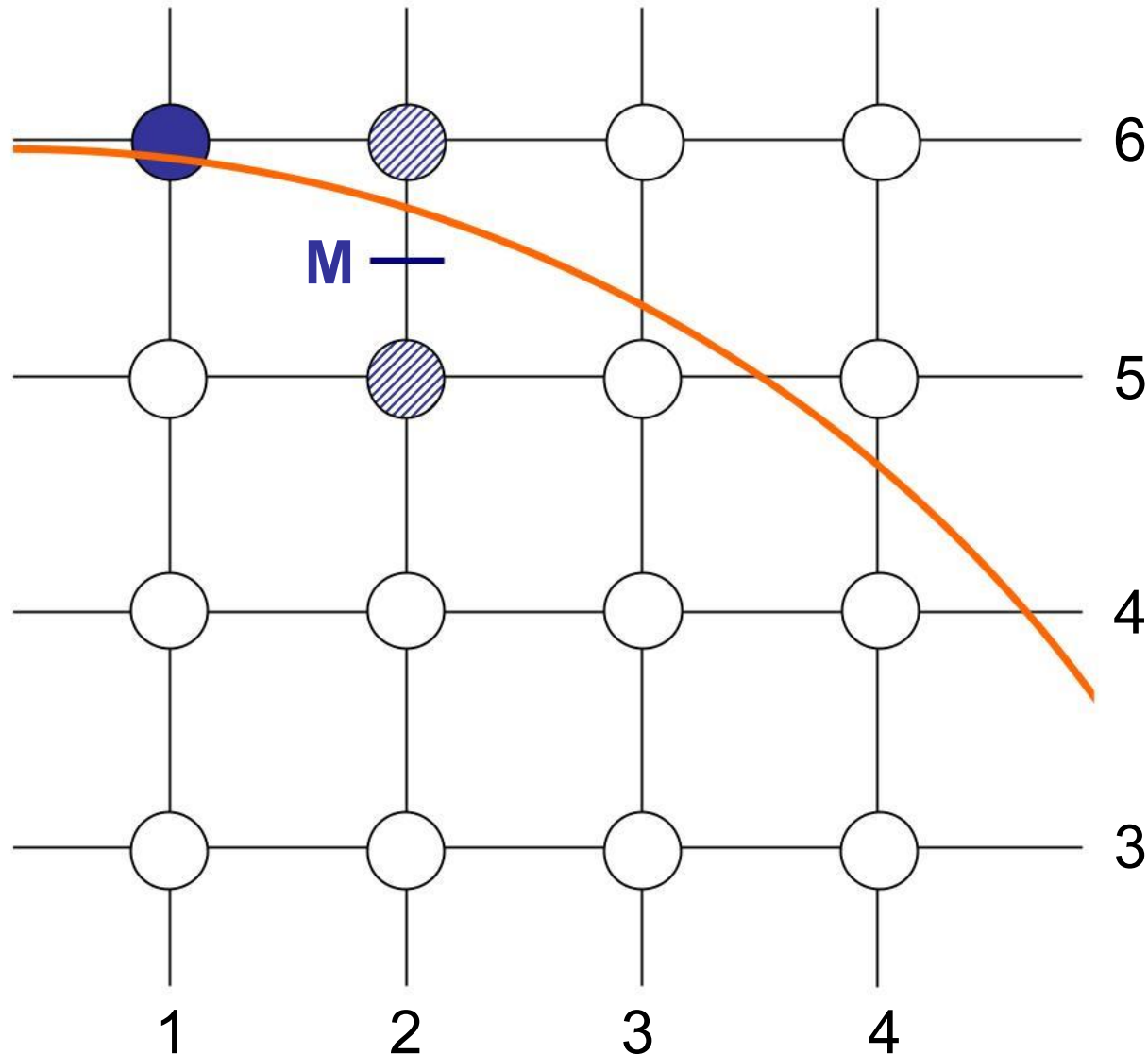
$$x = x + x_c \quad y = y + y_c$$

6. Repeat steps 3 to 5 until $x \geq y$

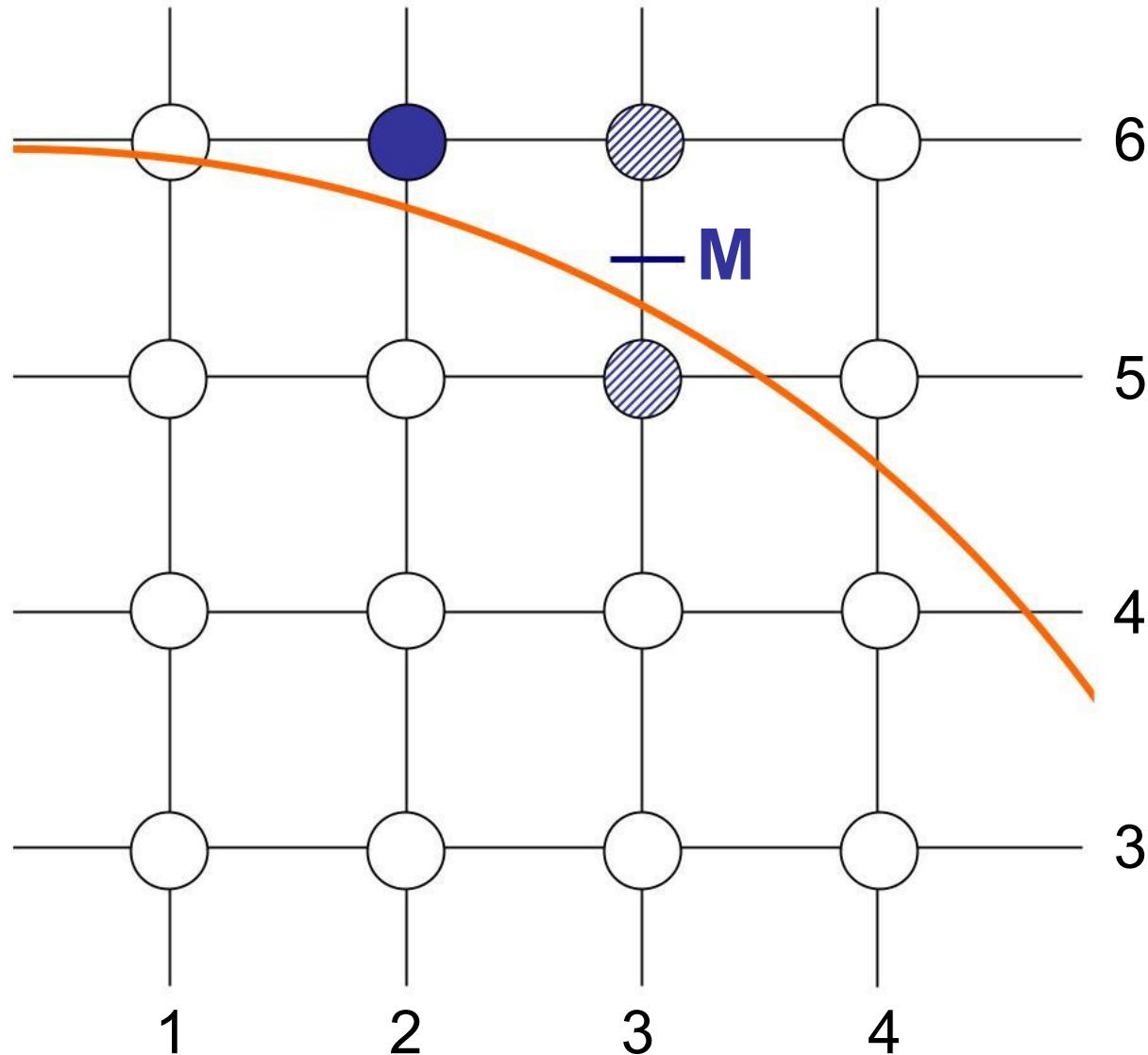
Mid-Point Circle Algorithm (cont..)



Mid-Point Circle Algorithm (cont..)



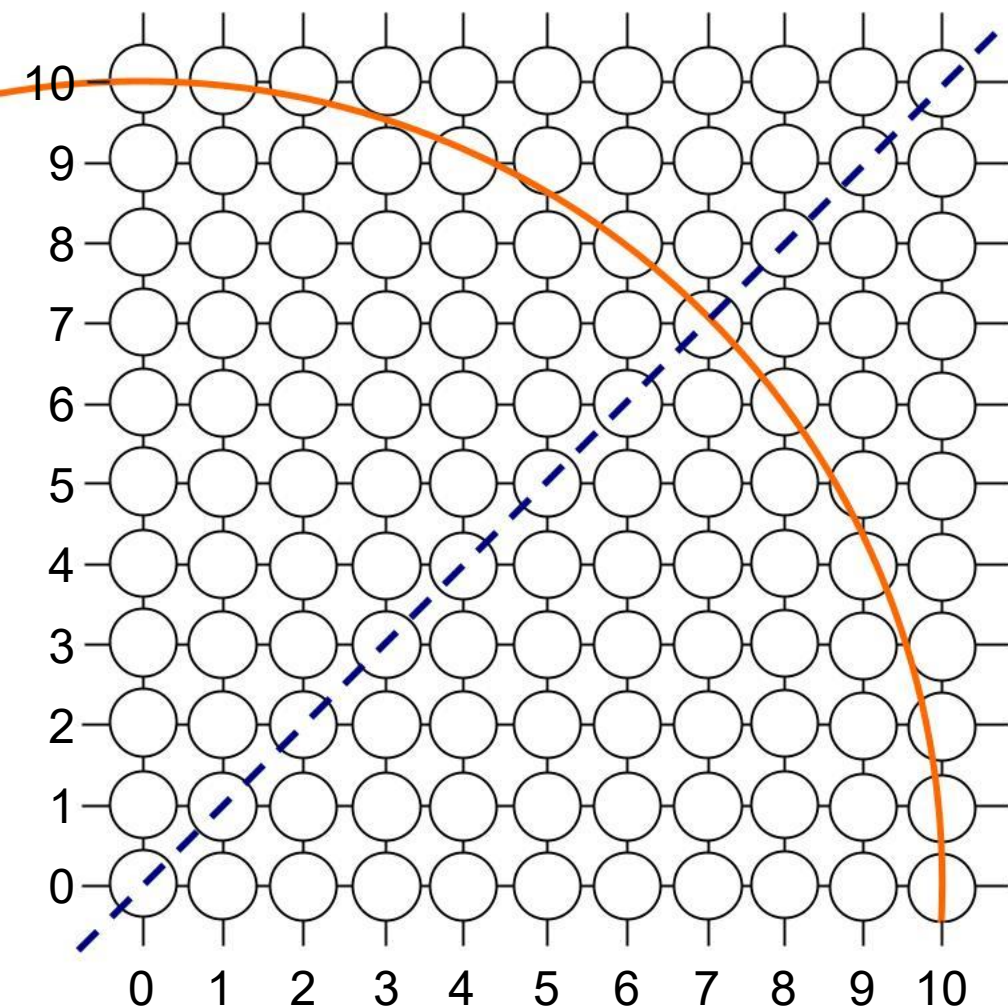
Mid-Point Circle Algorithm (cont..)



Дундаж цэгийн тойргийн алгоритмын жишээ

Дундаж цэгийн тойргийн алгоритмыг
үйлдлээр нь харахын тулд $(0,0)$ төвтэй, 8
радиустай тойрог зурахад ашиглана уу.

Дундаж цэгийн тойргийн алгоритмын жишээ(cont..)



k	p_k	(x_{k+1}, y_{k+1})	$2x_{k+1}$	$2y_{k+1}$
0				
1				
2				
3				
4				
5				
6				