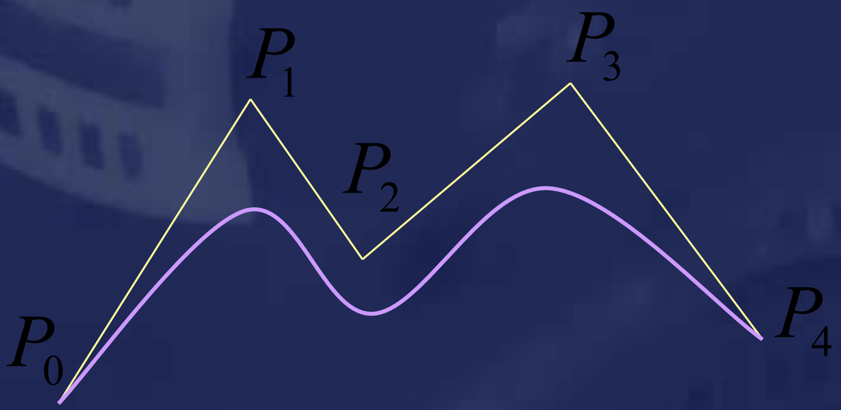
Кривые и поверхности высших порядков

Лекция 12

Астана 2004

Кривые высших порядков: постановка задачи

- Задача: построить параметрическую кривую “повторяющую” заданную ломаную (на плоскости или в пространстве)



$$P_i, i = 0,1,\ldots, N-1.$$

$$Q(t) = \sum_{i=0}^{N-1} P_i B_i(t), t_{min} \le t \le t_{max}.$$

$P_i \in R_2, R_3, R_4$ - контрольные точки

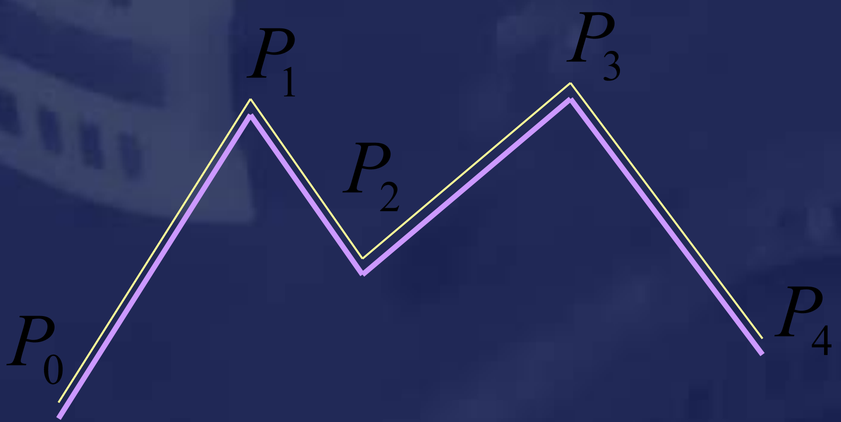
$B_i(t) \in R$ - базисные функции (обычно полиномы некоторой степени)

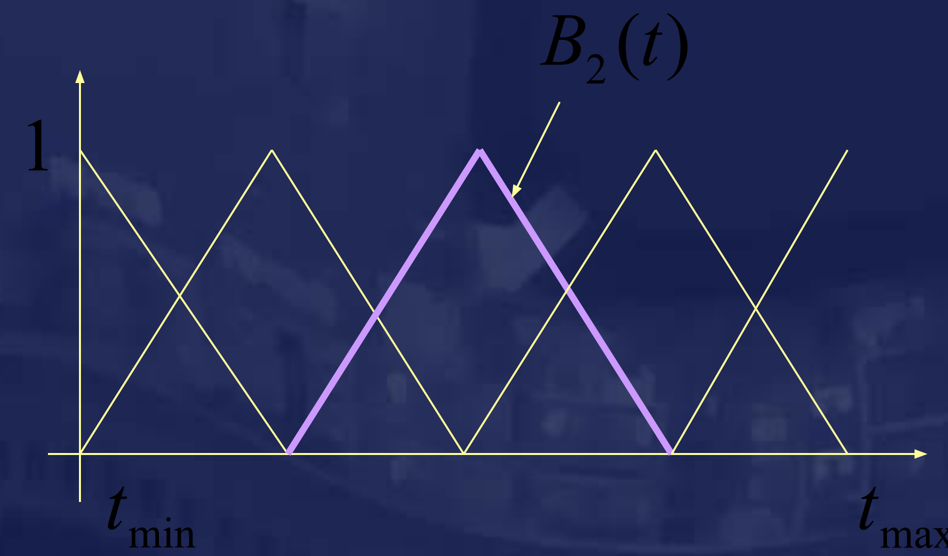
- Для рисования кривая обычно разбивается на *M* точек

$$Q_j = Q(u_j), t_{\min} = u_0 < u_1 < \ldots < u_{M-1}, j = 0,1,\ldots, M-1.$$

Кривые высших порядков: примеры базисов

✔ Базис первого порядка для 5 контрольных точек





✔ Кривые Безье. Степень кривой = *N* - 1

$$B_i = C_{N-1}^{i} t^{i} (1-t)^{N-1-i}, 0 \le t \le 1$$ - полиномы Бернштейна

✔ Любая контрольная точка P_i оказывает влияние на форму всей кривой

✔ Для большого количества точек степень кривой окажется тоже высокой.

Кривые высших порядков: B-сплайны

- Задача: построить параметрическую кривую, форма которой изменяется локально при изменении одной из контрольных точек.

$$Q(t) = \sum_{i=0}^{N-1} P_i B_{i,p}(t), t_{min} \le t \le t_{max}, 2 \le p \le N-1.$$

p - степень B-сплайна.

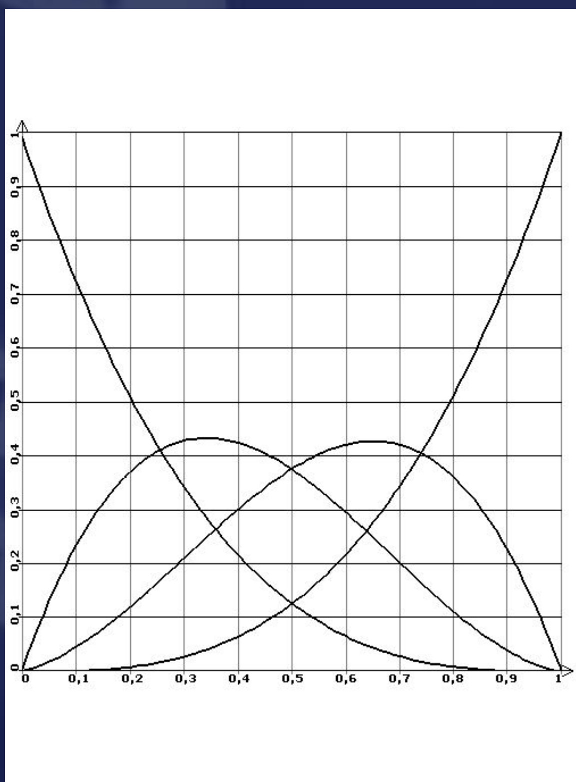
- Базисные функции рассчитываются по рекуррентным формулам Кокса-де Бура.

$$B_{i,0}(t) = \begin{cases} 1, t_i \le t \le t_{i+1}, \\ 0, \text{в противном случае.} \end{cases}, i = 0,1,\dots, N+p.$$

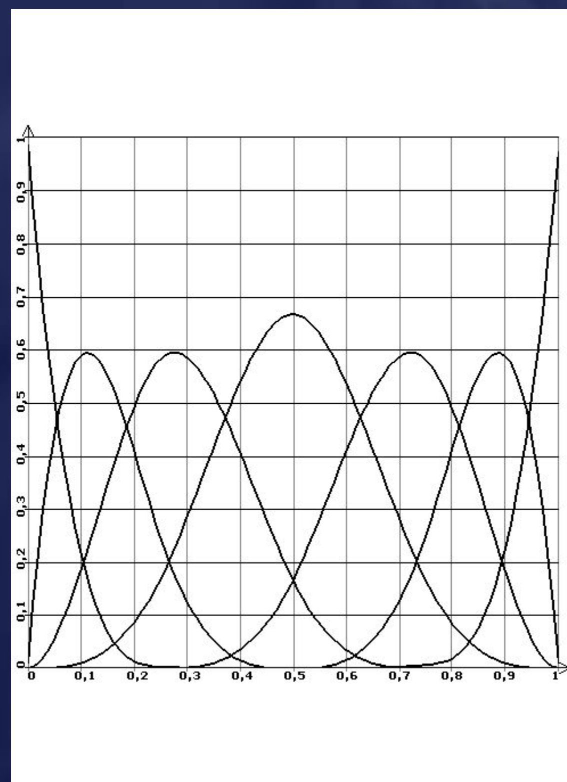
$$B_{i,k(t)} = \frac{(t-t_i)B_{i,k-1}(t)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-t)B_{i+1,k-1}(t)}{t_{i+k}-t_{i+1}}, i = 0,1,\dots, N+p-k.$$

Открытый узловой вектор

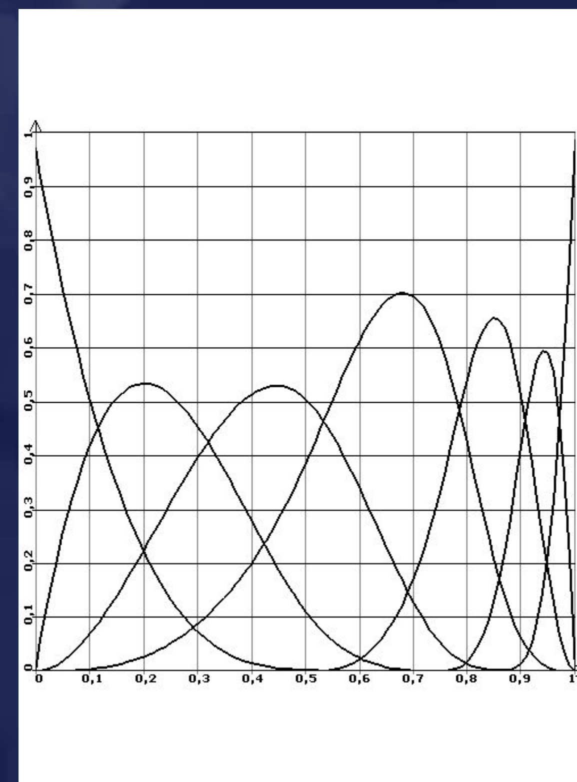
✔ Равномерный вектор: $t = [\,\overbrace{\underbrace{0\;0\;0\;0}_{p+1}\;\;0.25\;0.75\;\underbrace{1\;1\;1\;1}_{p+1}}^{N+p+1}\,]$



N = 4, p = 3



N = 7, p = 3

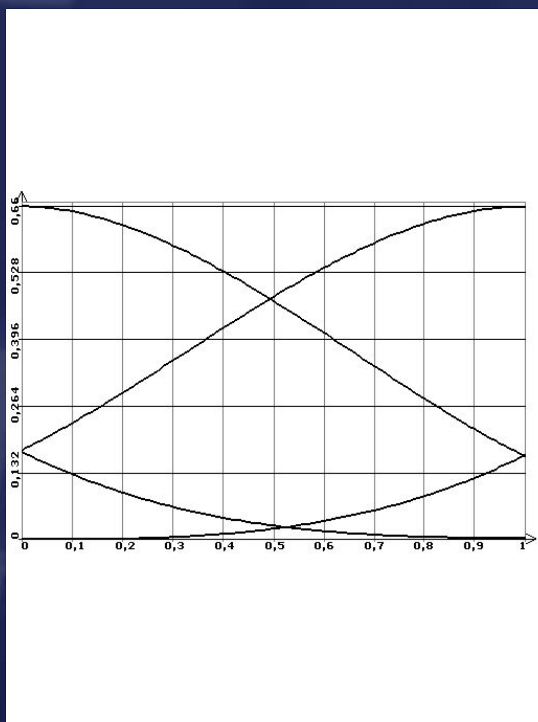


t = [0 0 0 0 0.25 0.75 0.875 1 1 1 1]

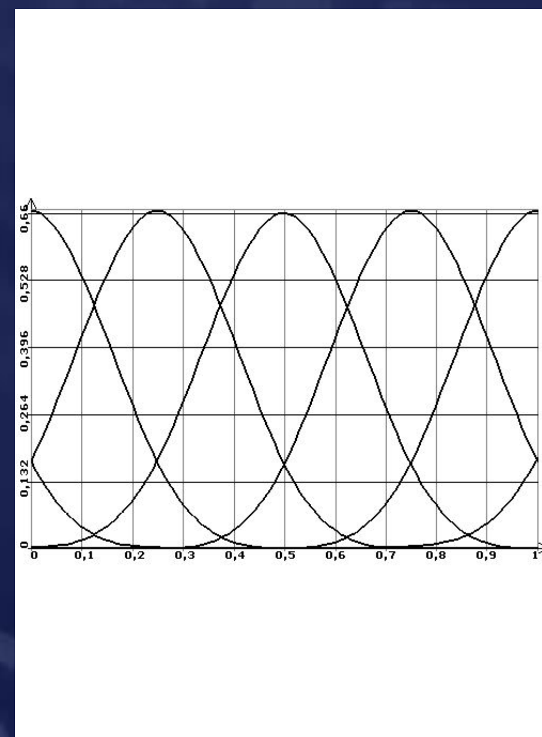
Периодический узловой вектор

✔ Равномерный вектор:

$$t = [\ \overbrace{\underbrace{-0.75\ \ -0.5\ \ -0.25\ \ 0}_{p+1}\ \ 0.25\ \ 0.75\ \ \underbrace{1.0\ \ 1.25\ \ 1.5\ \ 1.75}_{p+1}}^{N\,+\,p\,+\,1}\,]$$



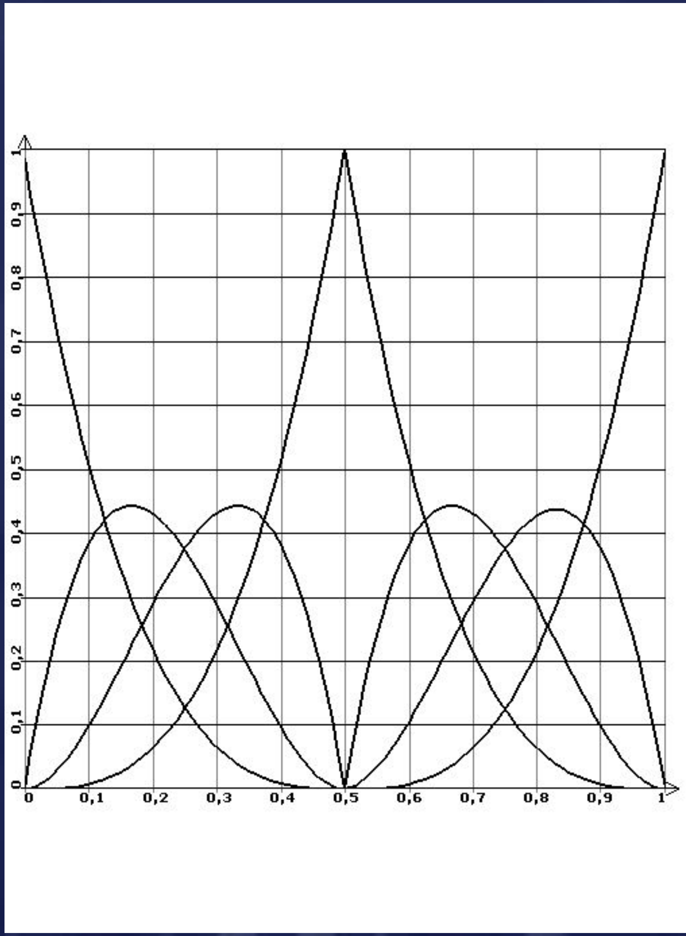
N = 4, p = 3



N = 7, p = 3

Повторяющиеся узлы

t = [0 0 0 0 0.5 0.5 0.5 1 1 1 1]



N = 7, p = 3

Расчет производных

- ✔ Коэффициенты при степенях постоянны на каждом из интервалов узлового вектора:

$$B_{i,n}(t) = \sum_{k=0}^{n} C_{i,n,k}(t).t^{k}$$

$$\frac{dB_{i,n}(t)}{dt} = \sum_{k=1}^{n} k.C_{i,n,k}(t).t^{k-1}$$

- ✔ Формулы для вычисления получаются из формул Кокса-де-Бура

$$C_{i,0,0}(t) = B_{i,0}(t)$$

$$C_{i,n,0}(t) = \frac{t_{i+n+1}}{t_{i+n+1} - t_{i+1}} C_{i+1,n-1,0}(t) - \frac{t_i}{t_{i+n} - t_i} C_{i,n-1,0}(t)$$

$$C_{i,n,n}(t) = \frac{1}{t_{i+n} - t_i} C_{i,n-1,n-1}(t) - \frac{1}{t_{i+n+1} - t_{i+1}} C_{i+1,n-1,n-1}(t)$$

$$\forall k \in \{1..n-1\}, C_{i,n,k}(t) = \frac{C_{i,n-1,k-1}(t) - t_i.C_{i,n-1,k}(t)}{t_{i+n} - t_i} - \frac{C_{i+1,n-1,k-1}(t) - t_{i+n+1}.C_{i+1,n-1,k}(t)}{t_{i+n+1} - t_{i+1}}$$

Рациональные сплайны

✔ Рациональный сплайн является проекцией обычного сплайна из пространства более высокой размерности (см. однородные координаты)

$$[x, y, z, w] \leftrightarrow [x/w, y/w, z/w] = [x', y', z']$$

$$Q(t) = \frac{\sum_{i=0}^{N-1} w_i P_i' B_{i,p}(t)}{\sum_{i=0}^{N-1} w_i B_{i,p}(t)}.$$

✔ Формулы для пересчета нормалей оказываются неверными.

✔ w > 0 является весом вершины. Чем больше вес, тем большее влияние вершина оказывает на форму кривой.

B-Spline поверхности

✔ Поверхность строится на основе двух наборов базисных функций

$$Q(u,v)=\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}B_{i,p}(u)B_{j,p}(v)P_{ij}.$$

$$\frac{\partial Q(u,v)}{\partial u}=\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}\frac{dB_{i,p}(u)}{du}B_{j,p}(v)P_{ij}.$$

$$\frac{\partial Q(u,v)}{\partial v}=\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}B_{i,p}(u)\frac{dB_{j,p}(v)}{dv}P_{ij}.$$

✔ Край поверхности является B-сплайном, который определяют граничные контрольные точки (для открытого базиса)

$$Q(0,v)=\sum_{i=0}^{N_1-1}B_{i,p}(0)\sum_{j=0}^{N_2-1}B_{j,p}(v)P_{ij}=\sum_{j=0}^{N_2-1}B_{j,p}(v)P_{0,j}.$$

Литература

- Роджерс Д., Адамс Дж. Математические основы машинной графики.
- vprat. ifrance.com - статься про NURBS
    - Копия этой статьи на сайте cg.cs.msu.su
- www.google.com :)

Вспомогательная библиотека GLU

- ✔ Входит в состав OpenGL и основана на командах OpenGL
- ✔ Функции GLU можно разделить на четыре класса
  - ✔ Вспомогательные функции (gluPerspective, gluLookAt, …)
  - ✔ Функции для рисования базовых геометрических объектов: сферы, цилиндра, круга и сектора круга.
  - ✔ Функции для разбиения невыпуклых многоугольников
  - ✔ Функции для работы с кривыми и поверхностями NURBS

Рисование геометрических объектов (1/2)

Перед началом рисования необходимо создать объект GLUQuadricObj, хранящий режимы рисования объектов GLU

```
GLUquadricObj *obj = gluNewQuadric();
```

Для управления режимами рисования предназначены следующие функции:

gluQuadricDrawStyle Каркасный или сплошной режим рисования

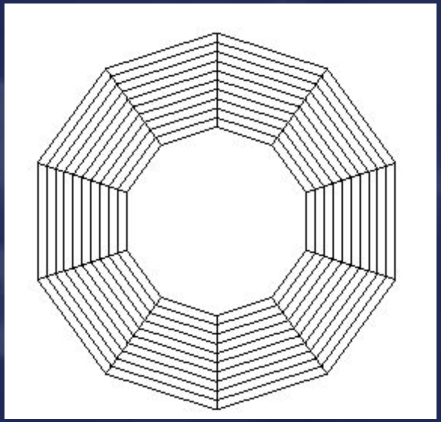
gluQuadricOrientation Направление нормалей

gluQuadricNormals Режим расчета нормалей

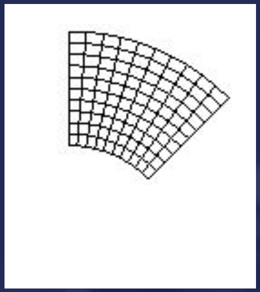
gluQuadricTexture Рассчитывать или нет текстурные координаты

```
gluQuadricNormals(obj, GL_FLAT);
```

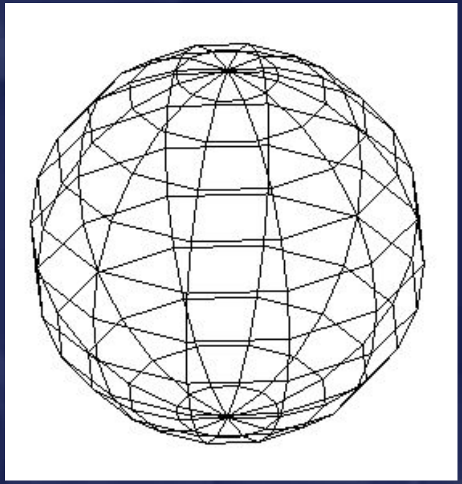
Рисование геометрических объектов (2/2)



disk



partial disk



sphere

Для рисования объектов предназначены функции gluSphere, gluCylinder, gluDisk и gluPartialDisk.

```
gluSphere(obj, 1.0, 20, 10);
```

Когда объект не нужен, память можно освободить

```
gluDeleteQuadric(obj);
```

Рисование кривых и поверхностей NURBS (1/2)

Перед началом рисования кривой или поверхности NURBS необходимо создать объект для хранения режимов построения NURBS

```
GLUnurbsObj *obj = new gluNewNurbsRenderer();
```

При помощи функции gluNurbsProperty можно задать режим разбиения, режим рисования и режимы отсечения кривых и поверхностей.

Предусмотрены следующие режимы разбиения кривых и поверхностей:

1. Постоянный шаг по параметрам u и v

2. Адаптивное разбиение в зависимости от длины кривой/площади поверхности на экране.

3. Адаптивное разбиение в зависимости от ошибки аппроксимации

Рисование кривых и поверхностей NURBS (2/2)

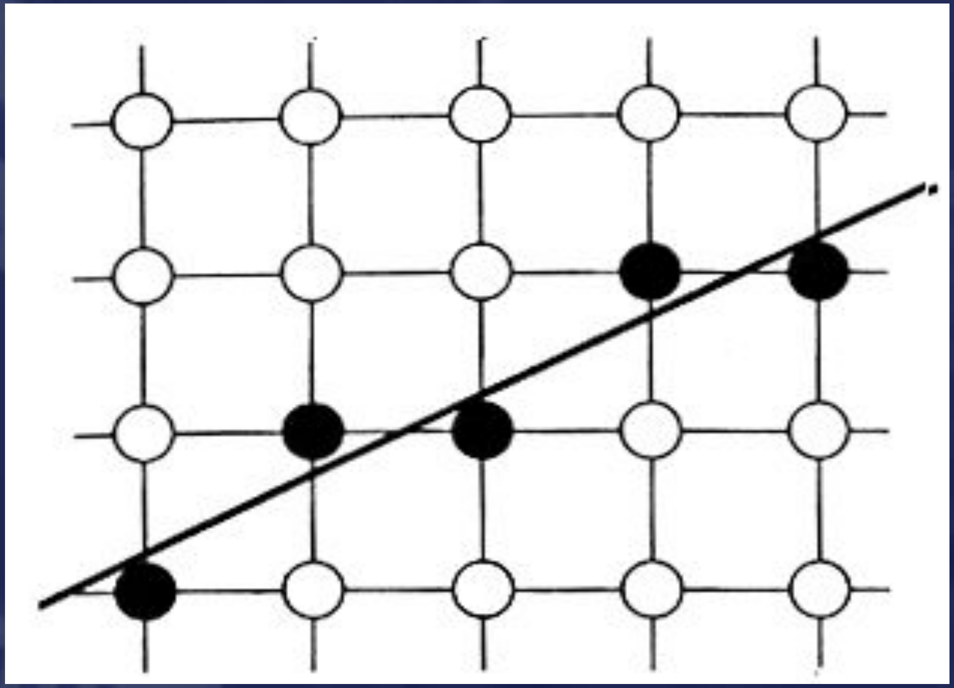
- ✔ Рисование кривой NURBS

  Между командами gluBeginCurve и gluEndCurve вызываются команды gluNurbsCurve для задания массивов контрольных точек, а также нормалей, цветов и текстурных координат вершин.

- ✔ Рисование поверхности NURBS

  Между командами gluBeginSurface и gluEndSurface вызываются команды gluNurbsSurface для задания массивов контрольных точек, а также нормалей, цветов и текстурных координат вершин.

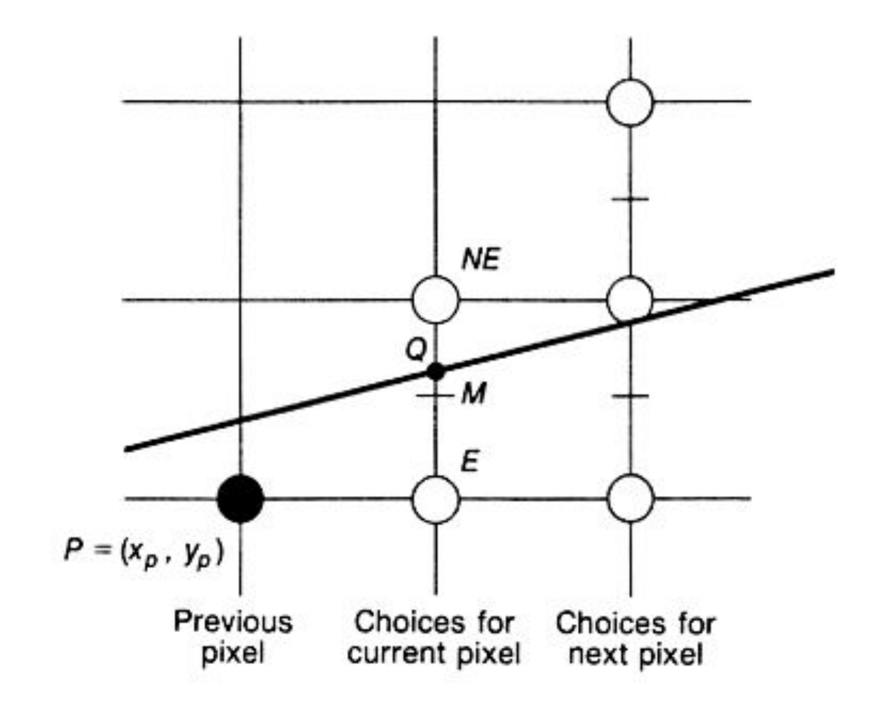
Алгоритм Брезенхема (1/4)



Отрезок, соединяющий P(x1, y1) и Q(x2, y2)

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1), x \in [x_1, x_2]$$

Алгоритм Брезенхема (2/4)



$$F(x,y) = (x - x_1)dy - (y - y_1)dx$$

$$dx = x_2 - x_1$$

$$dy = y_2 - y_1$$

$F(x,y) = 0$ -- точка на отрезке

$F(x,y) < 0$ -- точка выше

$F(x,y) > 0$ -- точка ниже

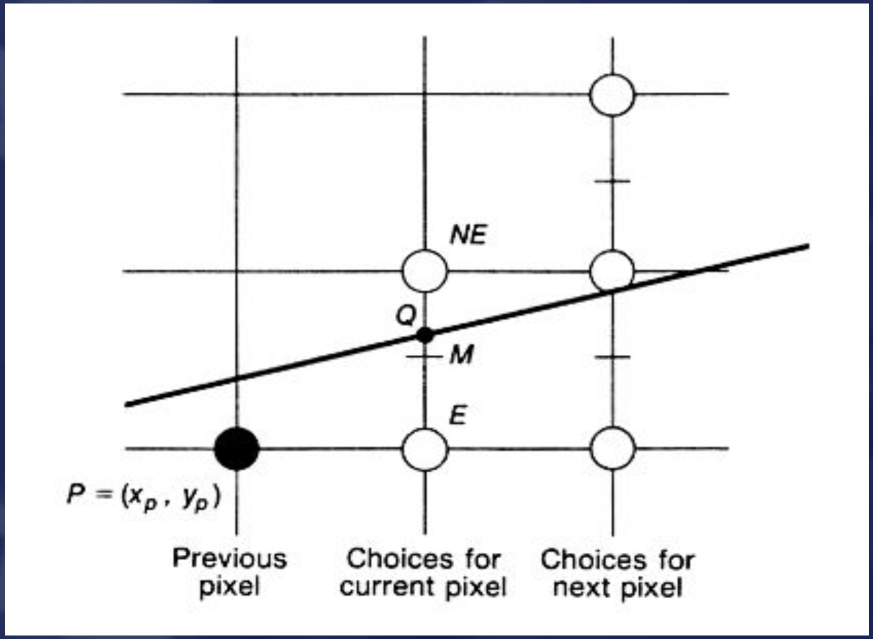
Точка P определена, тогда координаты срединной точки

$$(x_p + 1, y_p + 1/2)$$

и значение функции в этой точке

$$d = F(x_p + 1, y_p + 1/2)$$

Алгоритм Брезенхема (3/4)



Если $d < 0$, то выбирается E и

$$d_{new} = F(x_p + 2, y_p + \tfrac{1}{2})$$

$$d_{new} - d_{old} = F(x_p + 2, y_p + \frac{1}{2}) - F(x_p + 1, y_p + \frac{1}{2})$$

$$\Delta_E = d_{new} - d_{old} = dy = y2 - y1$$

Если $d \geq 0$, то выбирается NE

$$d_{new} = F(x_p + 2, y_p + \frac{3}{2})$$

$$\Delta_{NE} = dy - dx = (y_2 - y_1) - (x_2 - x_1)$$

В начальной точке

$$d_{start} = F(x_1 + 1, y_1 + 1/2) =$$

$$(x_1 + 1 - x_1)dy - (y_1 + 1/2 - y_1)dx =$$

$$= dy - dx/2$$

Алгоритм Брезенхема (4/4)

Одна неприятность -- деление на 2

Чтобы избежать вещественной арифметики, сделаем преобразование

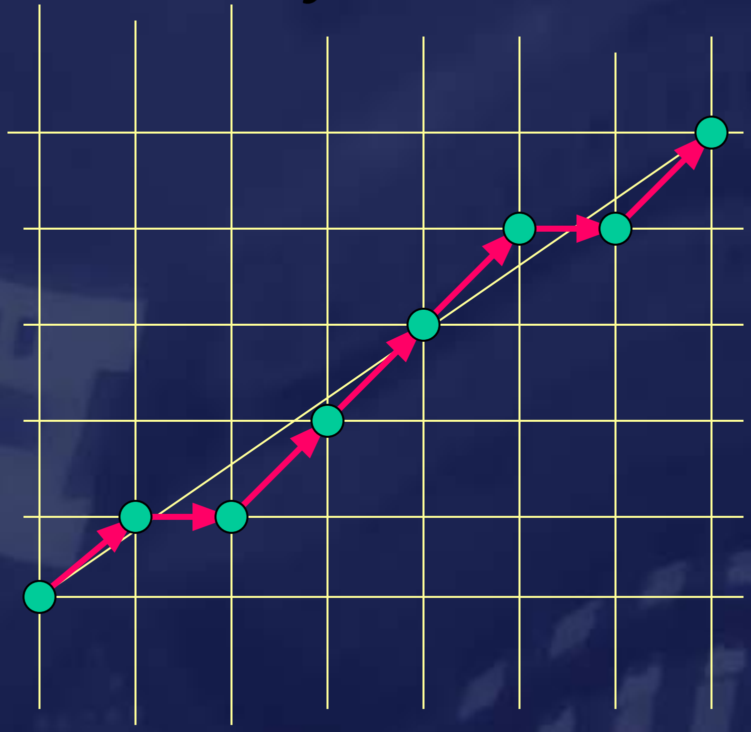
$$F'(x,y) = 2F(x,y)$$

$$d' = 2d$$

$$d_{start} = 2dy - dx$$

$$\Delta d' = 2\Delta d$$

$$d_{start} = 2dy - dx = 3; d_{NE} = -4; d_E = 10$$



$d_0 = 10 - 7 = 3 > 0$ (NE)

$d_1 = 3 - 4 = -1 < 0$ (E)

$d_2 = -1 + 10 = 9$ (NE)

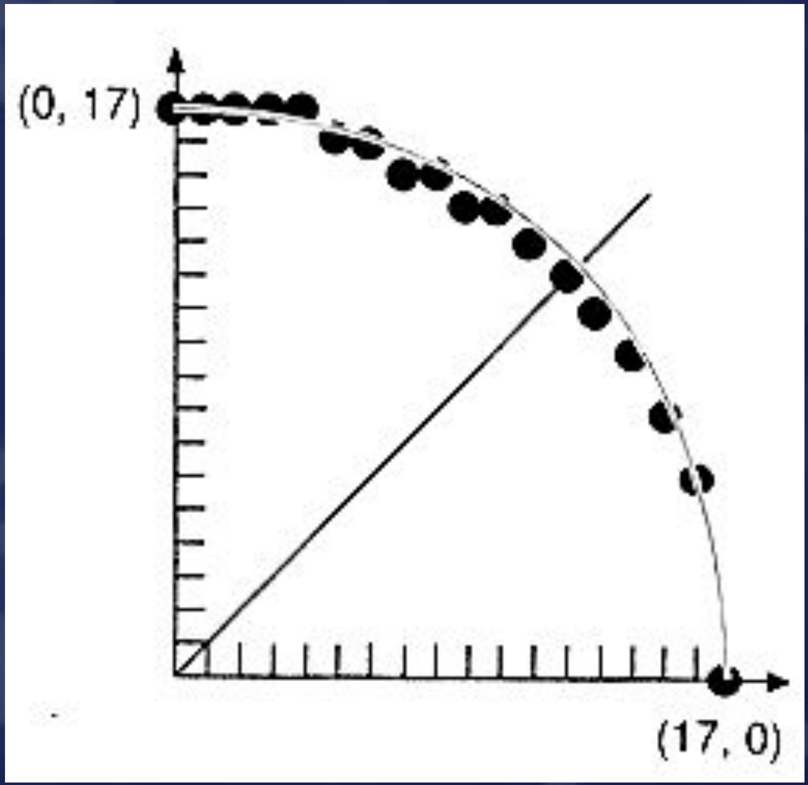
$d_3 = 9 - 4 = 5$ (NE)

$d_4 = 5 - 4 = 1$ (NE)

$d_5 = 1 - 4 = -3$ (E)

$d_6 = -3 + 10 = 7$ (NE)

Алгоритм Брезенхема (1/4)
(окружность)



Неявное и явное представление

$$x^2 + y^2 = R^2$$

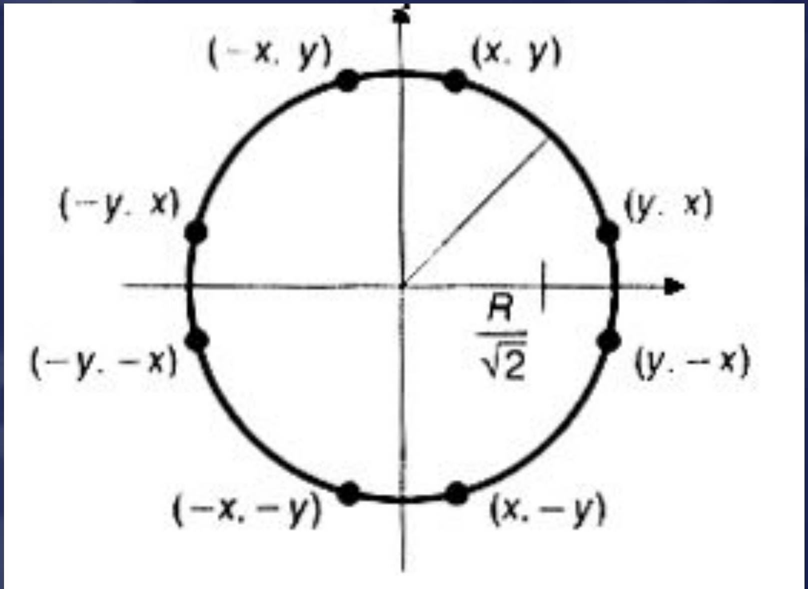
$$y = \pm\sqrt{R^2 - x^2}$$

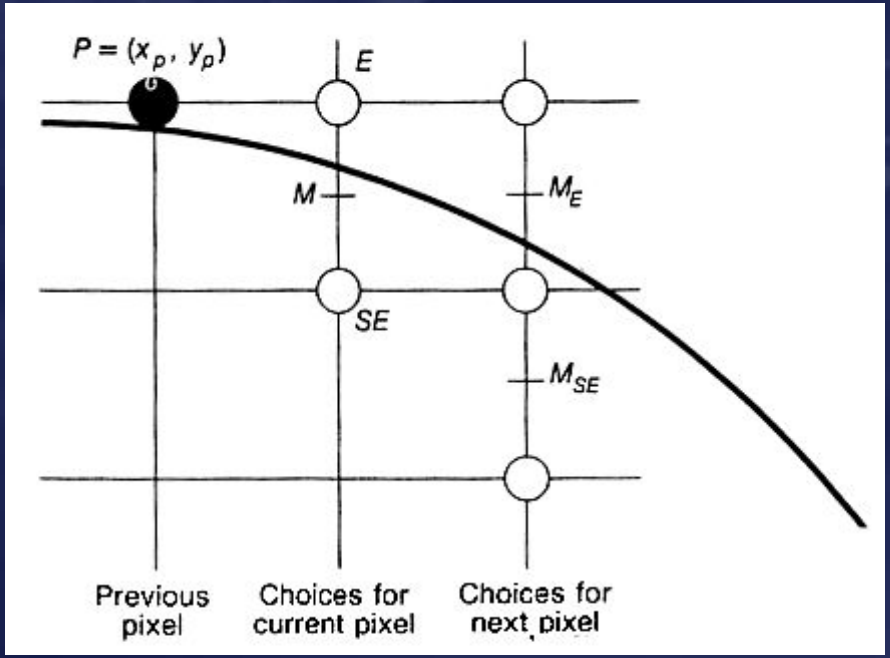
Параметрическое представление

$$x = R\cos\theta$$

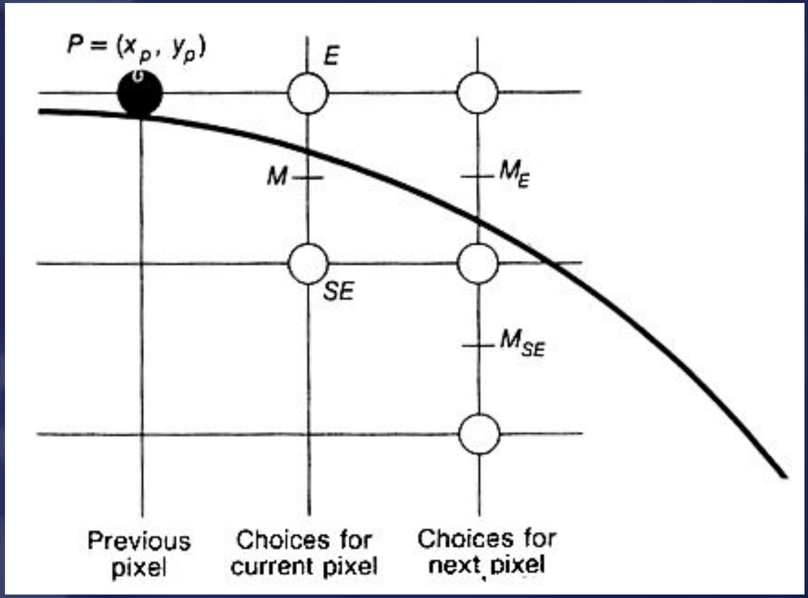
$$y = R\sin\theta$$

Алгоритм Брезенхема (2/4)
(окружность)





Алгоритм Брезенхема (3/4)
(окружность)



$$F(x,y)=x^2+y^2-R^2$$

Для точки P с коорд. (x_p, y_p)

$$d_{old}=F(x_p+1, y_p-\tfrac{1}{2})$$

Для пиксела E:

$$d_{new}=f(x_p+2, y_p-\frac{1}{2})=d_{old}+(2x_p+3)$$

$$\Delta d_E=(2x_p+3)$$

Для пиксела SE:

$$d_{new}=d_{old}+(2x_p-2y_p+5)$$

$$\Delta d_{SE}=(2x_p-2y_p+5)$$

Алгоритм Брезенхема (4/4)
(окружность)

В начальной точке5 (0, R)

$$F(1, R - \frac{1}{2}) = 1 + (R^2 - R + \frac{1}{4}) - R^2 = \frac{5}{4} - R$$

$$d_0 = \frac{5}{4} - R$$

И опять нужно исключить вещественные операции.

Сделав замену $h = d - 1/4$, получим $h = 1 - R$.

Тогда необходимо сравнивать h с $-1/4$, но так как приращения d – целые числа, то сравнивать можно с нулем.