Информатика

для направления130102

«Технологии геологической разведки»

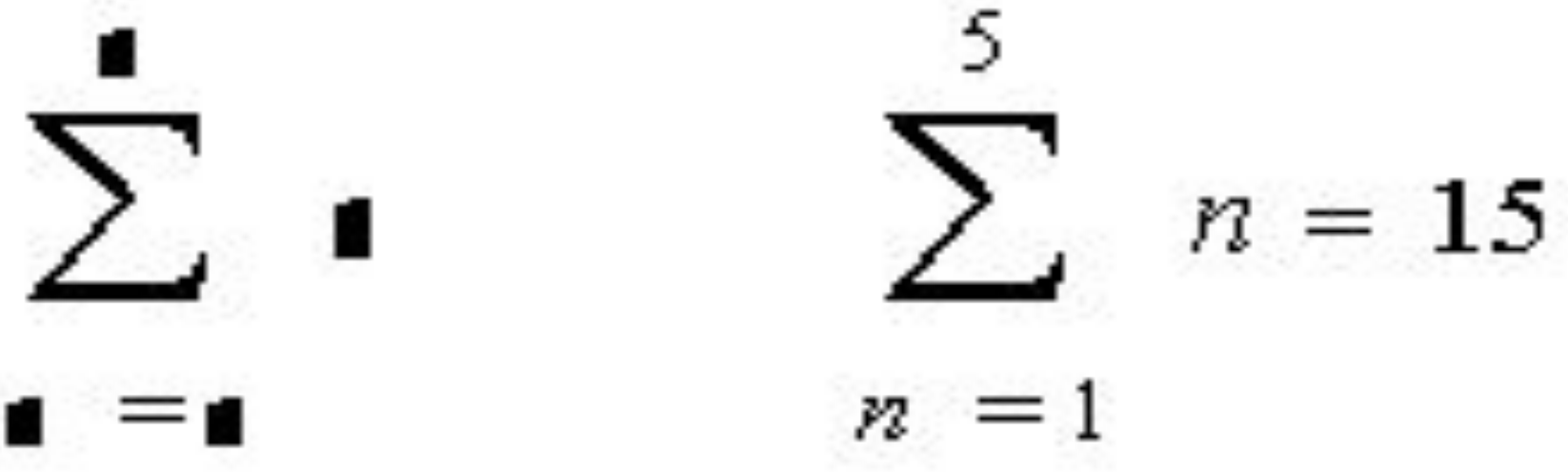
(двухсеместровая программа: семестр 2)

Автор-составитель: Хамухин Александр Анатольевич, 01.09.2010, aaxtpu@tpu.ru

Пакет MathCAD

- предназначен для выполнения инженерных расчетов **без составления программ** для формул, которые записываются в естественном (принятом в математике) виде;
- по форме ***MathCAD*** является **электронной таблицей** без предварительного разграничения ячеек, но с направлением описания **СЛЕВА-НАПРАВО** и **СВЕРХУ-ВНИЗ**;
- недостатком является «закрытость», что приводит к трудностям интеграции с другими приложениями;

- принцип заполнения **бланков** (**шаблонов**): ***MathCAD*** позволяет создать график или математическое выражение (интеграл, сумму и т.п.) путем заполнения свободных полей в вызываемых (определенными клавишами или меню) «бланках»:



- После щелчка в любом месте появляется крестик. Весь вводимый текст далее будет размещен в этом месте (ячейка электронной таблицы), пока курсор мыши не будет передвинут на новое место. Наберите, например, 15-8/2 и знак **=**. После знака **равно** ***MathCAD*** выполнит вычисления и выдаст результат.
- Порядок выполнения операций такой же, как в математике и **правило скобок** тоже действует аналогично

- Для определения любой переменной надо набрать ее имя и знак «**:**». ***MathCAD*** добавит знак «**=**», после чего можно ввести ее значение или выражение. Для переменной с индексом (массив, вектор, матрица) после имени набрать **квадратную скобку** и указать индексы через запятую.
- Переменные можно использовать после их определения только **правее** и/или **ниже**.

Дискретный аргумент принимает все целые значения в заданном диапазоне с заданным шагом

- Пример: указываем имя дискретного аргумента «**t**», затем знак двоеточие «**:**», затем начальное значение дискретного аргумента «**1**», затем знак запятая «**,**», затем следующее значение дискретного аргумента «**3**» (по нему вычислится ШАГ=3-1), затем ставим знак точка с запятой «**;**» (а вместо него *MathCAD поставит* две точки) и затем набираем конечное значение дискретного документа, например, «**8**».
- Для проверки того, какие значения будет принимать определенный таким образом дискретный аргумент, запишем «**t=**».

$t := 1,3..8$

$t =$

1
3
5
7

- Функции целесообразны в тех случаях, когда одна и та же формула применяется к различным, заранее не известным данным. Эти данные объявляются АРГУМЕНТАМИ функции (указываются в круглых скобках, если их несколько, то через запятую) и их не обязательно определять ранее как переменные. Другие имена в формуле, не объявленные аргументами, необходимо определить ранее как переменные.

$$a := 1.3 \quad b := 2.4 \quad c := 0.75$$

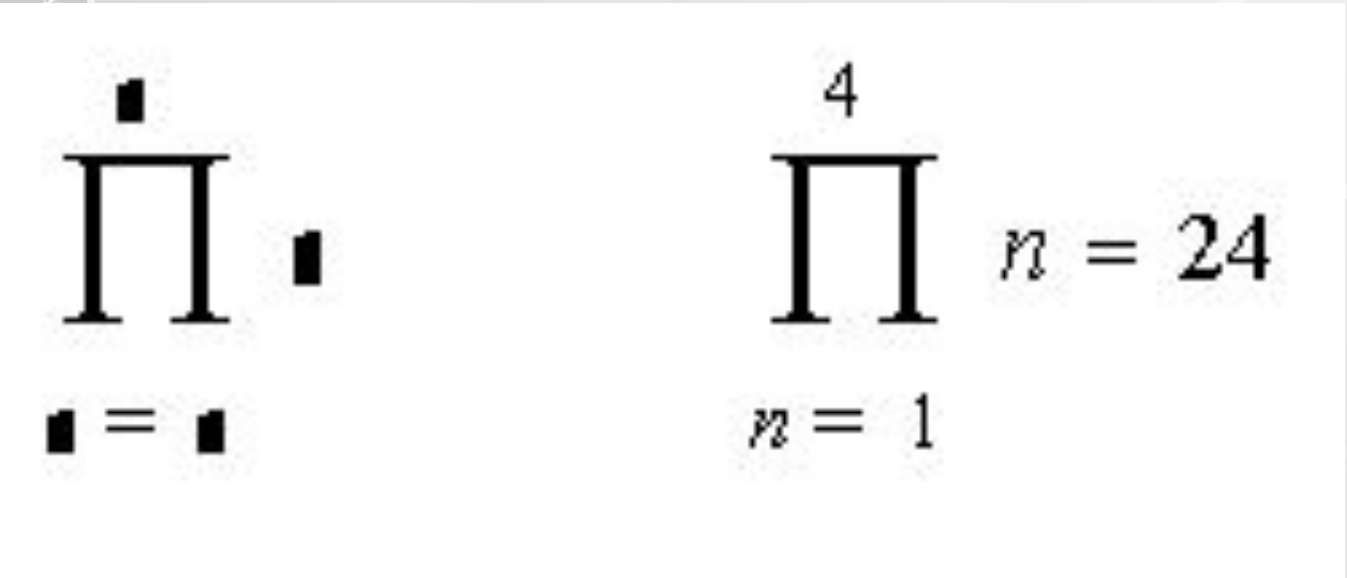
$$f1(x) := \left[\frac{\left(a \cdot b + c^2\right)}{|a - 2 \cdot b|} \right] + (\log(a \cdot x))^3$$

$$f1(13) = 2.903$$

$$f1(-1) = 0.417 - 2.487\mathrm{i}$$

$$f1(0) = \blacksquare$$

- Для вычисления суммы (произведения) одновременно нажать три клавиши *«ctrl» «shift»* «4» («3») или найти соответствующий значок на панели инструментов.
- Появится бланк, который нужно заполнить и нажать знак = для вычисления



- Для вычисления интеграла набрать знак &, или найти соответствующий значок на панели инструментов, заполнить на бланке пределы интегрирования, подынтегральную функцию и набрать =

$$\int_{0}^{1} x\,dx = 0.5$$

- В качестве пределов интегрирования можно использовать дискретный аргумент для вычисления множества значений интеграла

$$i := 0, 1 .. 2$$

$$\int_0^i x\,dx =$$

0
0.5
2

- Для работы с матрицами необходимо определить переменную с индексами. Для этого после имени переменной и двоеточия необходимо нажать клавиши *«ctrl» «M»*, указать размерность матрицы и далее использовать имя переменной с нижними индексами (переход на нижний индекс квадратная скобка)

$$x := \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{pmatrix} \qquad y := \begin{pmatrix} 1 & 2 & 10 \\ 2 & 6 & 7 \\ 3 & 2 & 5 \\ 4 & 6 & 7 \\ 5 & 2 & 6 \\ 6 & 6 & 3 \\ 7 & 2 & 4 \\ 8 & 6 & 3 \\ 9 & 2 & 2 \\ 10 & 5 & 3 \end{pmatrix}$$

- Выбрать пункт **Декартов график** из меню **Графика** или нажать знак на клавиатуре **@.** Появится бланк с шестью пустыми полями, которые нужно заполнить.
- Пустое поле в середине горизонтальной оси предназначено для независимой переменной, например, *x*. Введите туда дискретную переменную, переменную с индексом или любое выражение, содержащее дискретную переменную для того, чтобы на графике выводился целый ряд точек, координаты которых соответствуют таблице данных.
- Пустое поле в середине вертикальной оси предназначено для переменной, график которой нужно построить. Введите туда дискретную переменную, переменную с индексом или любое выражение, содержащее дискретную переменную, находящуюся на горизонтальной оси графика.
- Другие 4 поля предназначены для указания диапазонов отображения переменных по горизонтальной и вертикальной осям или вообще не заполняются (по умолчанию *MathCAD* сам определит диапазон).
- График не отображается пока не щелкнуть мышью вне его поля или не нажать *F9*

Чтобы представить несколько зависимостей на одном графике, введите первую переменную по оси ординат с запятой в конце. Ниже появится пустое поле для второй переменной (выражения), введите вторую переменную с запятой в конце, ниже появится третье поле и т.д.

Модели решения функциональных и вычислительных задач

- **Модель** – это объект или описание объекта для замещения (при определенных условиях предложениях, гипотезах) одной системы (оригинала) другой системой (моделью) для лучшего изучения оригинала и/или воспроизведения каких-либо его свойств.
- По содержанию модели подразделяются на:

- **эмпирические** – созданные на основе экспериментальных фактов, опыта и профессионализма разработчика;
- **теоретические** – на основе математических теорий, научных законов;
- **статистические** – на основе накопленных данных об объекте в совокупности со статистическими методами анализа и обработки
- **когнитивные** – на основе составления схемы и оценки степени влияния одних параметров на другие

По форме модели подразделяются на:

- **физические** – на основе создания подобного объекту уменьшенного (упрощенного) экземпляра
- **электрические** – на основе подобия процессов в электросхеме и процессов, протекающих в объекте
- **математические** – на основе математических описаний процессов, протекающих в объекте
- **компьютерные** – на основе преобразования описаний объекта в программный код

- Любая из моделей может быть реализована в виде **компьютерной программы**.
- Наиболее правдоподобные объекту модели называют **компьютерными симуляторами**

Основные свойства модели:

- **целенаправленность** – модель всегда отображает некоторый объект с определенными целями
- **полнота** – в модели должны быть учтены все основные связи и отношения, необходимые для обеспечения цели моделирования
- **упрощенность** – модель отображает только существенные стороны объекта, чтобы быть доступной для исследования и воспроизведения
- **адекватность** – модель должна успешно описывать моделируемый объект в известных нам состояниях (реперных точках)
- **информативность** – модель должна выдавать достаточную новую информацию об объекте (в рамках гипотез, принятых при построении модели) для достижения целей моделирования
- **управляемость** – модель должна иметь хотя бы один параметр, изменениями которого можно настраивать поведение модели в соответствии с известным поведением моделируемого объекта в различных условиях.

Погрешности

- **Погрешность измерения** – оценка отклонения величины измеренного значения величины от её истинного значения.
- Поскольку выяснить с абсолютной точностью истинное значение любой величины невозможно, то невозможно и указать величину отклонения измеренного значения от истинного.
- При этом за истинное значение принимается значение эталонов мер, на практике используются показания эталонных приборов (поверка, тарировка, градуировка).

Погрешности - 2

- Погрешность измерения является мерой точности измерения.
- В измерениях необходимо указывать, какова их точность. Для этого вместе с полученным результатом указывается погрешность измерений. Например, запись **T=2,8±0,1** с. означает, что истинное значение величины **T** лежит в интервале от **2,7 с.** до **2,9 с.** некоторой оговорённой вероятностью (доверительная вероятность)
- Погрешность, указанная в единицах измерения величины, называется абсолютной

Погрешности - 3

- **Точность средства измерений** - степень совпадения показаний измерительного прибора, откалиброванного по эталону, с истинным значением измеряемой величины
- **Точность результата измерений** — одна из характеристик качества измерения, отражающая близость к нулю погрешности результата измерения. Следует отметить, что о повышении качества измерений всегда говорят термином «увеличить точность» - притом, что величина, характеризующая точность, при этом должна уменьшиться.

Погрешности - 4

- **Относительная погрешность** – это разность между истинным и наблюдаемым значениями, разделенная на истинное значение и умноженная на 100 %. Это безразмерная величина.
- Международные единицы измерения безразмерных величин:
  1 pm = 1 ‰ = 0,001 - промилле
  1 ppm = 0,000001 - *Parts Per Million*
  1 % = 10000 ppm
  1 ‰ = 1000 ppm
- Есть 1 ppb (1 миллиардная доля)
- и 1 ppt (1 триллионная доля)

Погрешности - 5

- **Методические погрешности** – это те погрешности, которые могут возникнуть после измерений в результате применения методов обработки данных, в том числе – расчетов.
- **Методические погрешности обусловлены:**

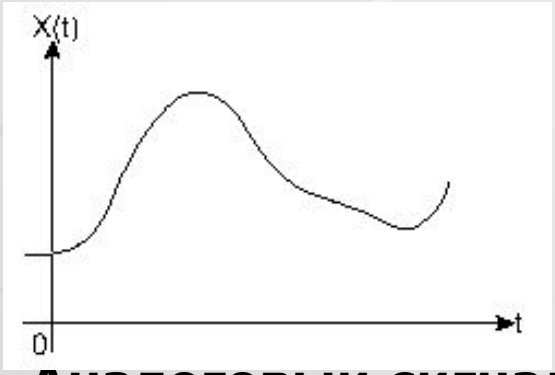
❖ Несовершенством метода обработки;

❖ Упрощениями, принятыми осознанно для снижения сложности обработки данных;

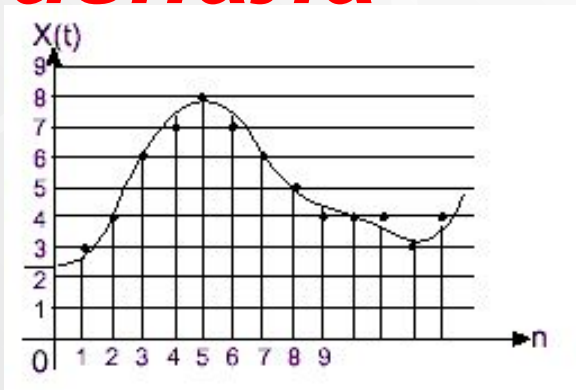
❖ Округлением при оцифровке данных;

❖ Ограниченной разрядностью вычислительного устройства (компьютера).

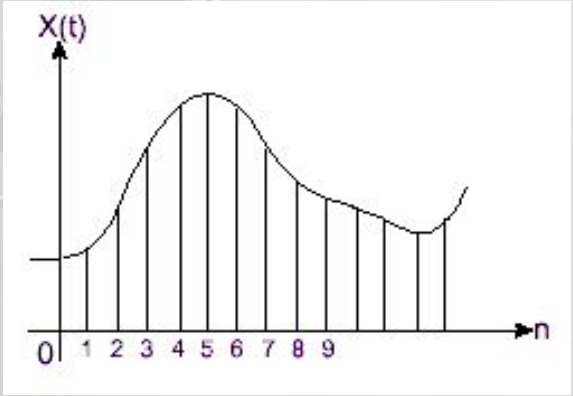
Цифровая модель аналогового сигнала



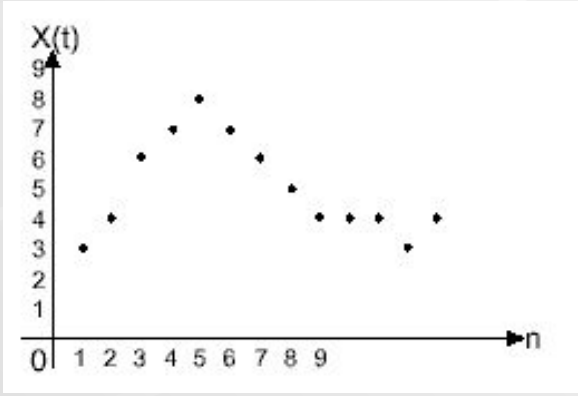
1. Аналоговый сигнал



3. Квантование по уровню



2. Дискретизация по времени

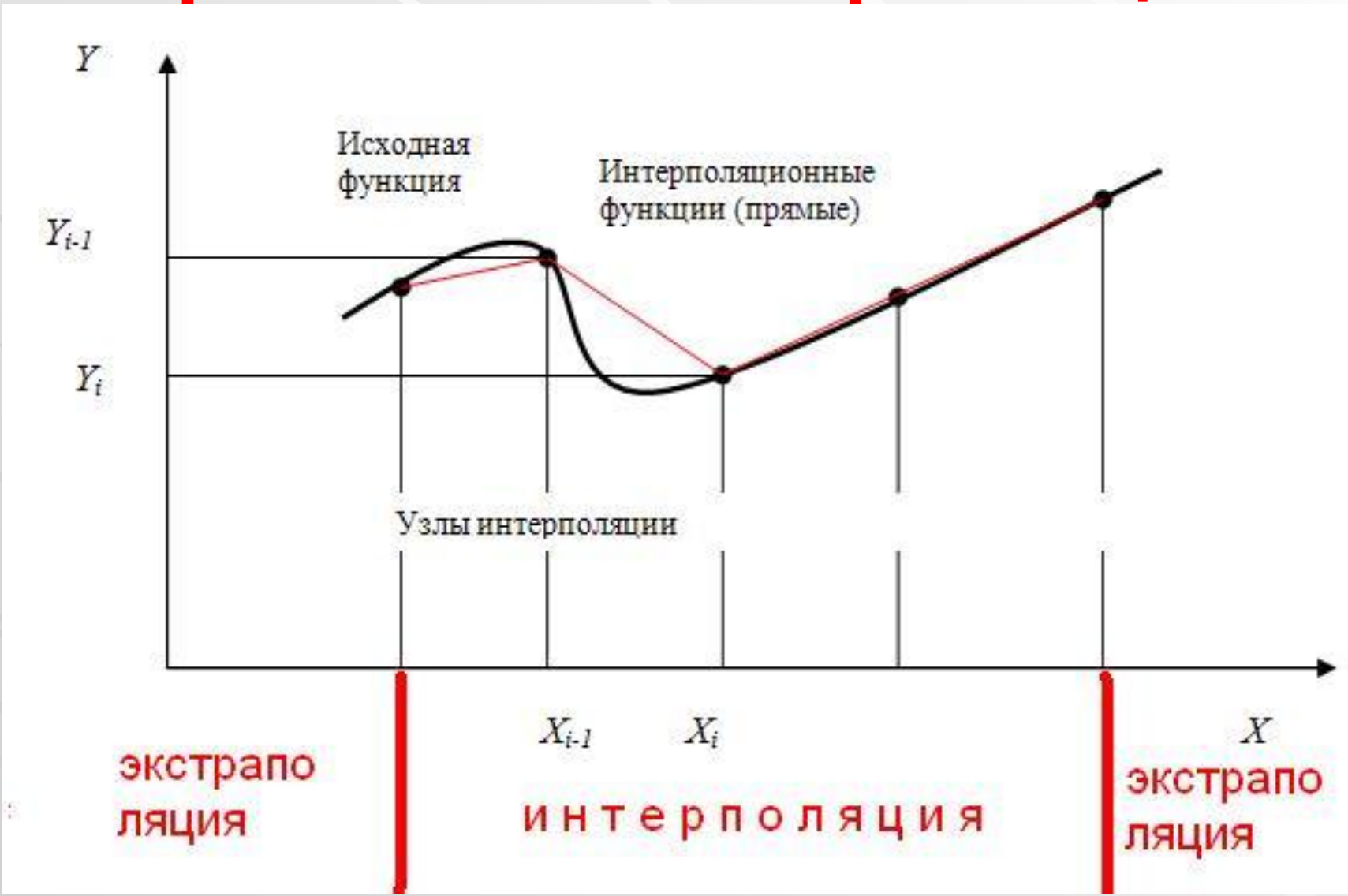


4. Цифровая модель: 3,4,6,7,8,7,6,5, и т.д.

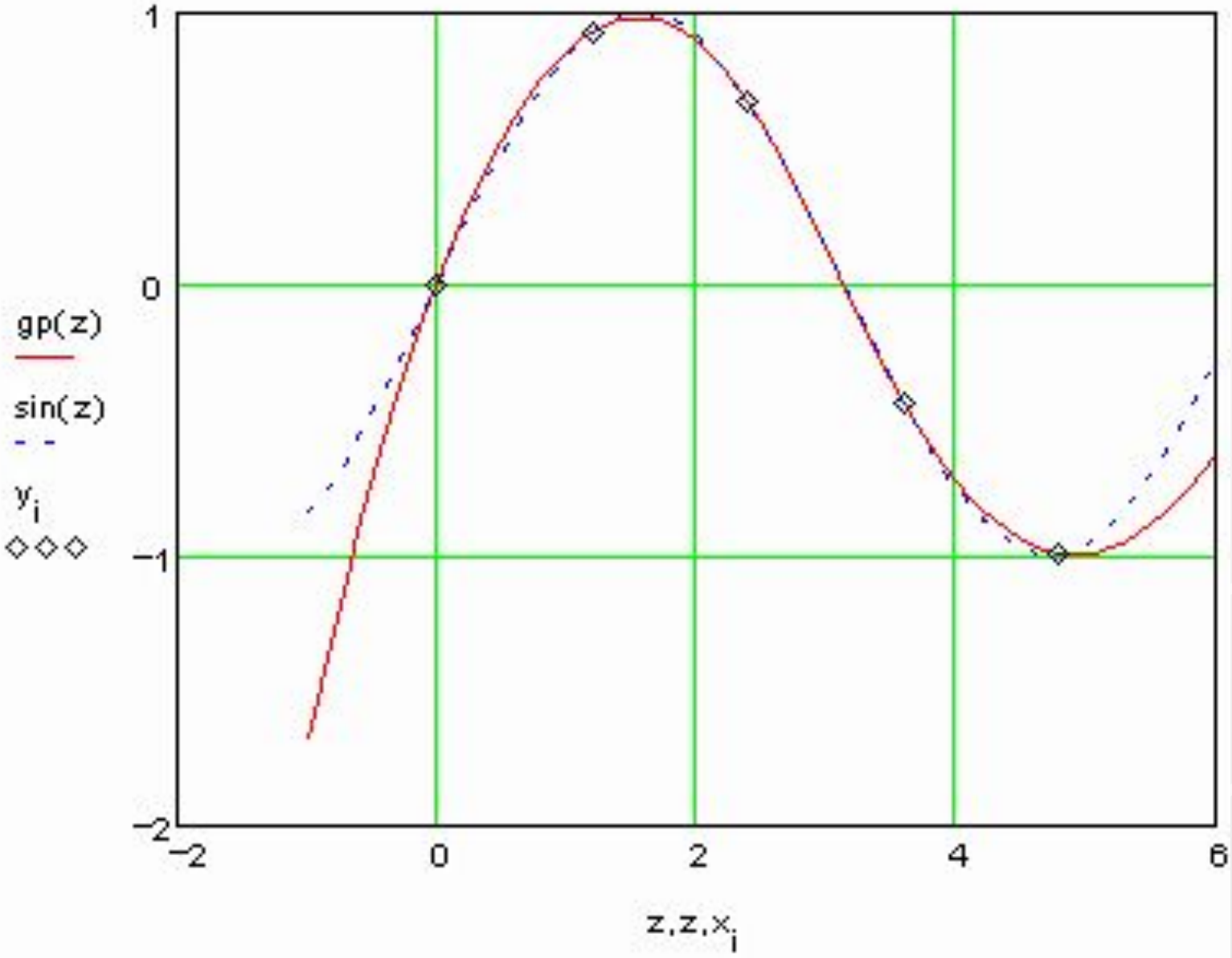
Интерполяция и аппроксимация

- **Интерполяция** – расчет промежуточных значений заданного ряда данных
- **Аппроксимацией** (приближением) функции ***f(x)*** называется нахождение такой функции ***g(x)*** (**аппроксимирующей функции**), которая была бы близка заданной.
- Интерполяция – частный случай аппроксимации, когда критерием близости является точное совпадение ***g(x)*** *и* ***f(x)*** *в заданных точках, которые называются* ***узлами интерполяции***
- **Экстраполяция** – расчет значений, выходящих за пределы заданного ряда данных

Простейшая интерполяция



Кусочно-линейная интерполяция



Интерполяция полиномом 4-й степени

Полином Лагранжа

- Применяется для интерполяции (экстраполяции) по *N* заданным точкам

$$Y(x) = \sum_{i=1}^{N} Y(x_i) * \prod_{\substack{j=1 \\ j \neq i}}^{N} \frac{(x - x_j)}{(x_i - x_j)};$$

$$\delta = \frac{\left| Y_{точное} - Y_{расчетн} \right|}{Y_{Точное}} \%.$$

Метод наименьших квадратов

- **Применяется для аппроксимации** (приближении) функции *y(x)* такой функцией *f(x)* (**аппроксимирующей функцией)**, которая была бы близка заданной по критерию:
- **минимальности функционала** *Ф*, являющегося суммой квадратов отклонений аппроксимирующей функции *f(x)* от экспериментальных точек y_i:

$$\Phi = \sum_{i=1}^{N} (y_i - f(x_i))^2 \rightarrow \min$$

$$\frac{\partial \Phi}{\partial a} = 0;$$

$$\frac{\partial \Phi}{\partial b} = 0.$$

$$a = \frac{N * \sum_{i=1}^{N} X_i * Y_i - \sum_{i=1}^{N} X_i * \sum_{i=1}^{N} Y_i}{N * \sum_{i=1}^{N} (X_i)^2 - (\sum_{i=1}^{N} X_i)^2}$$

$$b = \frac{1}{N} \sum_{i=1}^{N} (Y_i - a * X_i)$$

Численное интегрирование

- Пусть требуется вычислить определенный интеграл на интервале [*a*,*b*].

$$\int_a^b f(x)dx$$

- Для численного интегрирования подынтегральную функцию **аппроксимируют** какой-либо более простой функцией, интеграл от которой может быть вычислен. Обычно в качестве аппроксимирующей функции используют полином.
- В случае **полинома нулевой степени** мы получим **формулу прямоугольников**,
- в случае **полинома первой степени** – **формулу трапеций**,
- в случае **полинома второй степени** – **формулу Симпсона**. Все эти методы являются частными случаями **квадратурных формул Ньютона-Котеса**.

$$I_{прлев} \approx h * \sum_{i=1}^{N-1} Y(x_i) \qquad I_{прправ} \approx h * \sum_{i=2}^{N} Y(x_i)$$

$$I_{трап} \approx \frac{1}{2} \sum_{i=2}^{N} (X_i - X_{i-1}) * \left[f(X_i) + f(X_{i-1}) \right]$$

$$I_{трап} \approx \frac{Y_1 + Y_N}{2} h + \sum_{i=2}^{N-1} h * Y_i$$

$$I_{Симп} = \frac{1}{3} \left[4 I_{трап}(h) - I_{трап}(2h) \right]$$

Поиск особых точек

- Одной важных функциональных задач является поиск так называемых «особых точек» в экспериментальных данных или на моделях. Это могут быть минимумы или **максимумы** в профилях контуров нефтяных залежей, **точки перегиба**, **точки пересечения различных линий уровня** и **точки разрыва.**
- В качестве «особых точек» мы будем рассматривать только точки пересечения некоторой функцией (заданной таблично или аналитически) оси *OX* (**нули функции**). К такой постановке могут быть сведены задачи поиска и других видов особых точек

Метод последовательных приближений

- Метод последовательных приближений (**итераций**) базируется на известном в математике (теория множеств) принципе сжатых отображений. Этот принцип определяет так называемый оператор сжатия ***F***, для которого выполняется условие:

$$\left|F(x_2)-F(x_1)\right| \le a\left|x_2-x_1\right| \qquad a \subset (0,1)$$

- Метод дает только приближенное решение с некоторой погрешностью, которая **уменьшается на каждом шаге** (итерации). Метод выполняется пока погрешность не станет меньше заданной.

- **Сходимость** – это свойство алгоритма обеспечивать приближение к точному решению
- **Условия сходимости** – это математические соотношения величин, участвующих в итерационном процессе, которые обеспечивают алгоритму свойство сходимости.
- **Скорость сходимости** – это степень приближения к точному решению на каждой итерации, обычно она бывает линейной или квадратичной.

Метод дихотомии

- Первым этапом метода дихотомии является **табуляция** исследуемой функции (вычисления таблицы значений с заданным шагом) для разделения ее на такие отрезки, на концах которых ее значения противоположны.
- Затем каждый из найденных отрезков поочередно подвергается **делению пополам**.
- После разделения из двух вновь полученных отрезков **выбирается тот, на концах которого значения исследуемой функции противоположны.**
- Выбранный из двух отрезок **вновь делится пополам.**
- Эта процедура повторяется до тех пор, пока значение функции на середине отрезка **не станет меньше заранее заданного сколь угодно малого числа.**

Выбираем

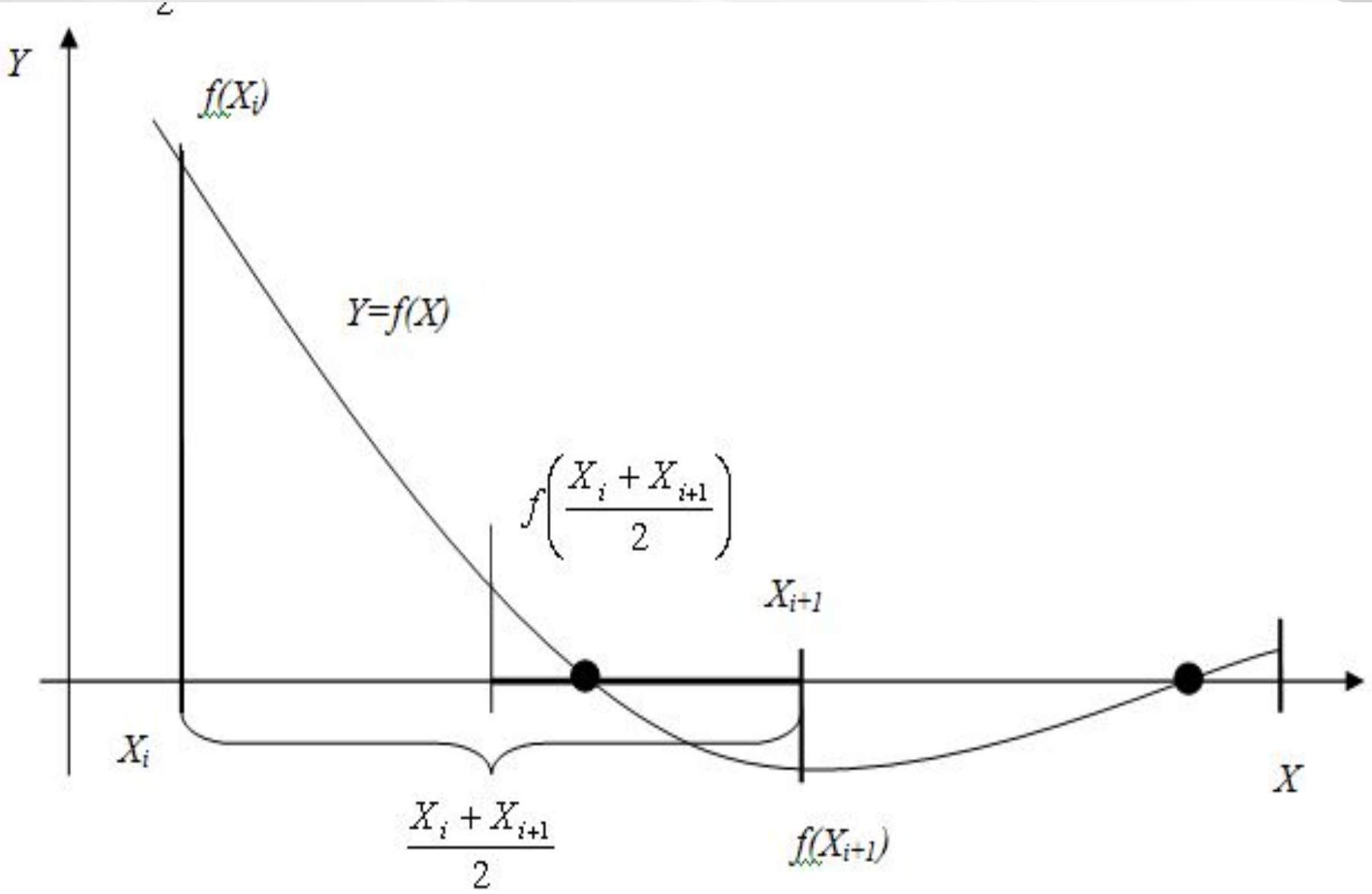
$$\{X_i, X_{i+1}\}: f(X_i) * f(X_{i+1}) < 0;$$

левый, если :

$$f(X_i) * f(\frac{X_i + X_{i+1}}{2}) < 0;$$

правый, если :

$$f(\frac{X_i + X_{i+1}}{2}) * f(X_{i+1}) < 0.$$



Пример последовательного приближения к точке пересечения функцией y=f(x) оси X методом дихотомии

Метод Ньютона

- Действительный корень *x* уравнения *f(x) = 0* вычисляется методом Ньютона по итерационному уравнению:

$$x_{k+1} = x_k - f(x_k)/f'(x_k).$$

- Недостатком этого метода является необходимость вычислять производную на каждой итерации, поэтому на практике чаще используют модификации этого метода, которые заменяют производную ее разностными приближениями (см. след. слайд)

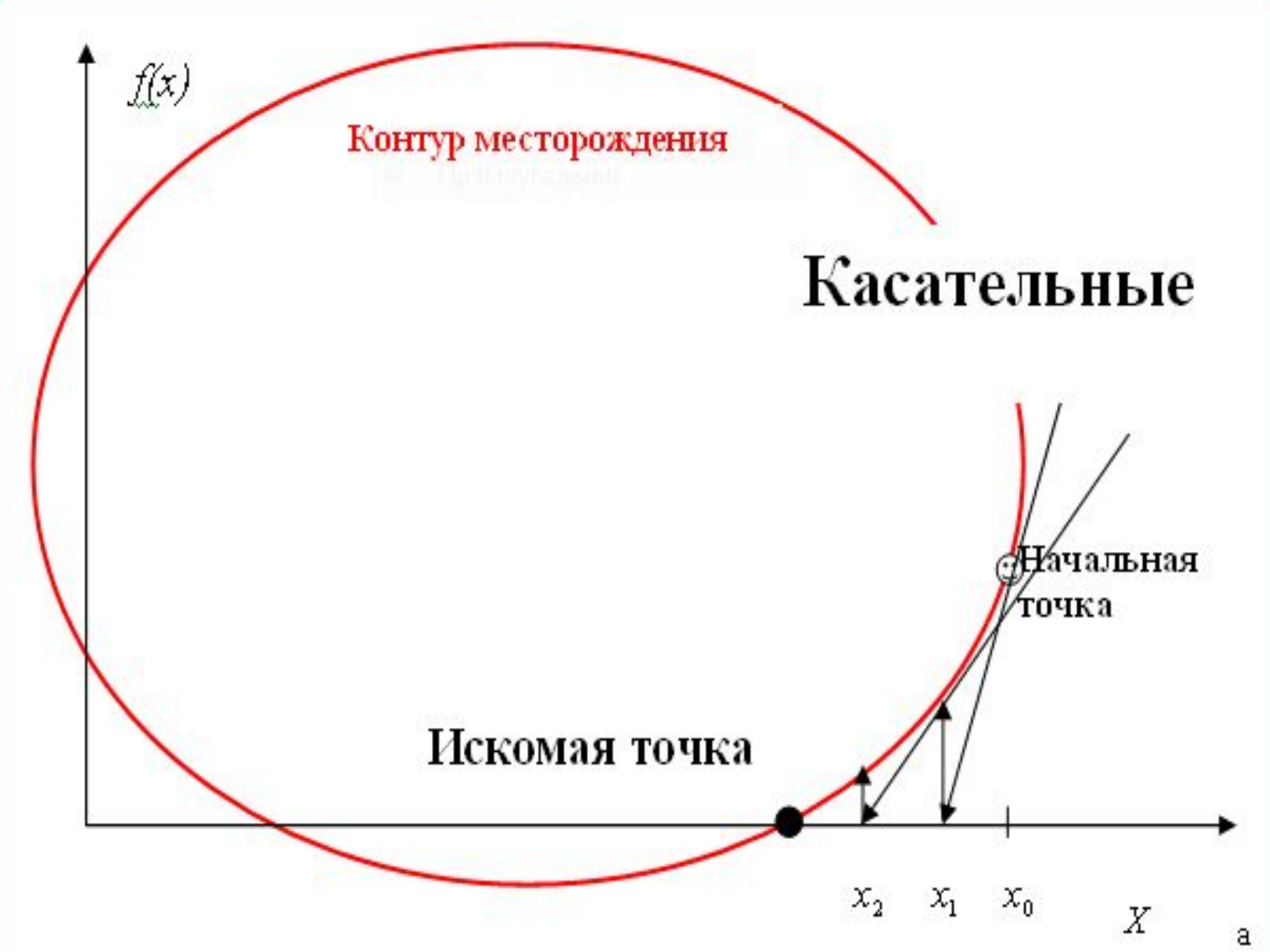
Варианты реализации метода Ньютона

- Метод хорд

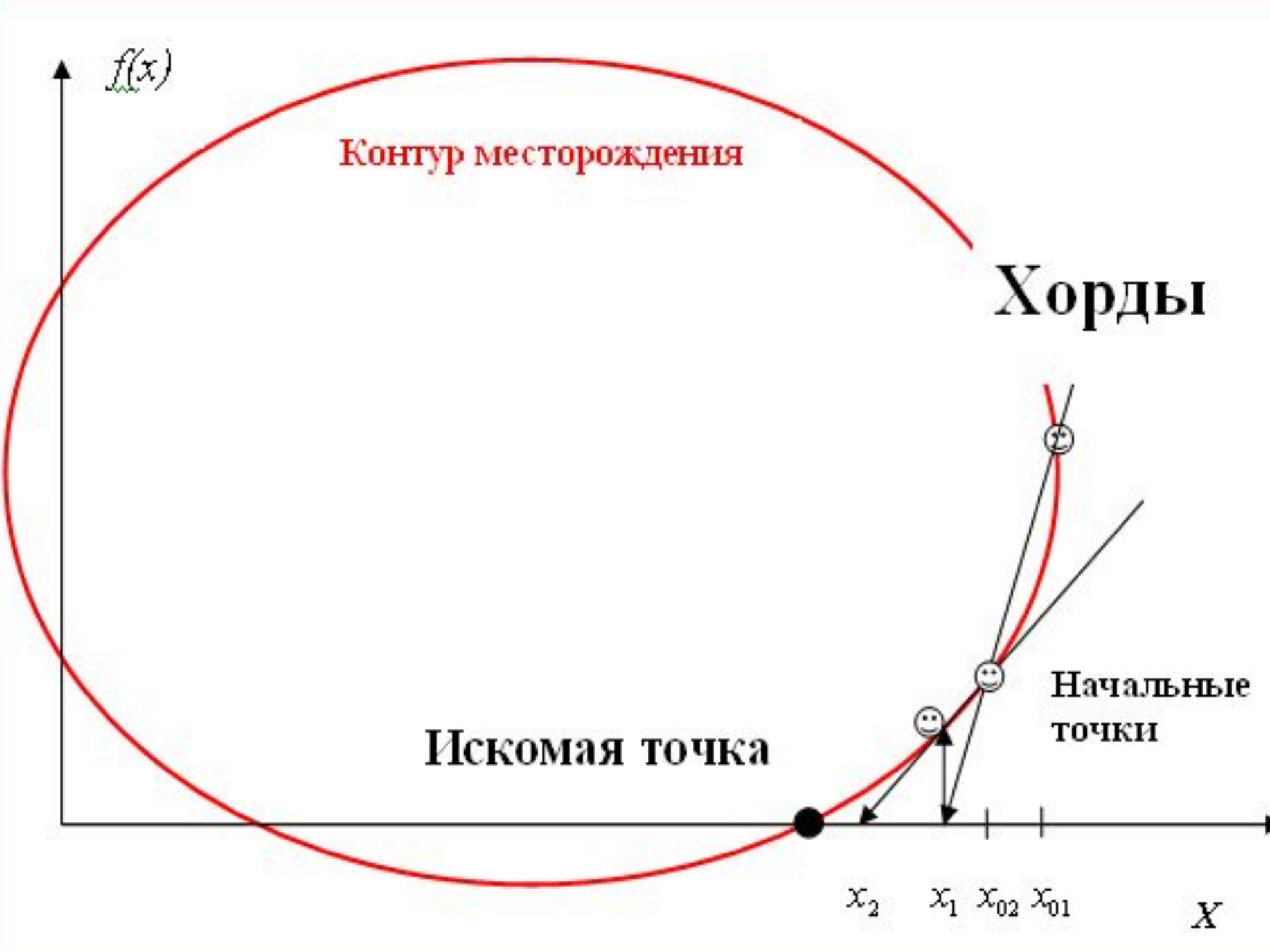
$$X_{j+1} = X_j - \frac{(X_j - X_{j-1}) * f(X_j)}{f(X_j) - f(X_{j-1})}$$

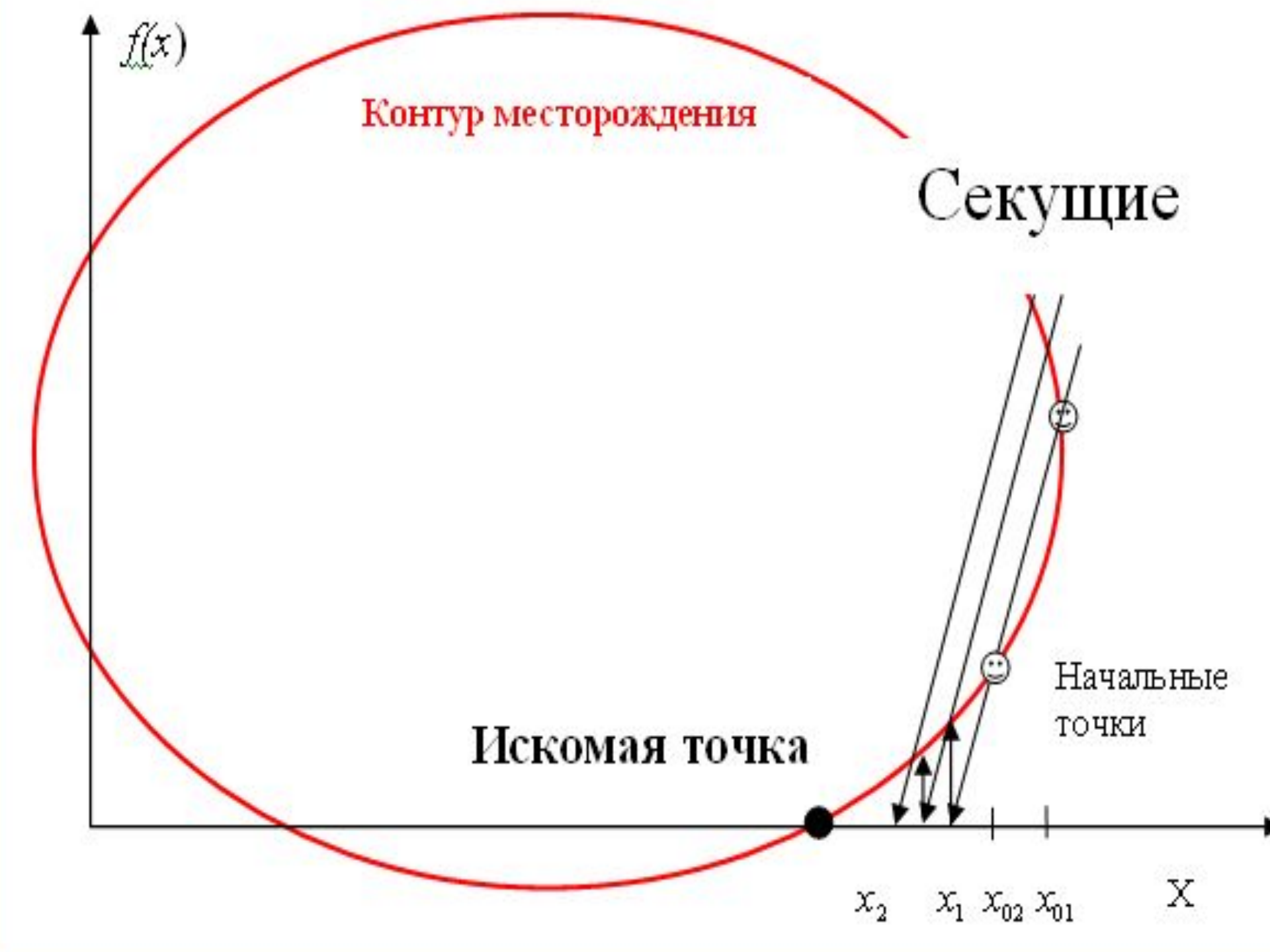
- Метод секущих

$$X_{j+1} = X_j - \frac{(X_1 - X_0) * f(X_j)}{f(X_1) - f(X_0)}$$


f(x)
Контур месторождения
Касательные
Начальная точка
Искомая точка
x_2
x_1
x_0
X

а


f(x)
Контур месторождения
Хорды
Начальные точки
Искомая точка
x_2
x_1
x_{02}
x_{01}
X


f(x)
Контур месторождения
Секущие
Начальные точки
Искомая точка
x_2
x_1
x_{02}
x_{01}
X

Пусть задана функциональная зависимость Y в виде ряда данных для равноотстоящих значений аргумента X.

$$X_k = X_0 + k\Delta X \quad (k = 0,1,2,\ldots) \quad \Delta X = X_{i+1} - X_i = h > 0.$$

Первая конечная разность (нисходящая, вперед)

$$\Delta Y_i = Y_{i+1} - Y_i$$

и (восходящая, назад)

$$\Delta Y_i = Y_i - Y_{i-1}.$$

Вторая конечная разность (нисходящая)

$$\Delta^2 Y_i = \Delta Y_{i+1} - \Delta Y_i$$

и (восходящая)

$$\Delta^2 Y_i = \Delta Y_i - \Delta Y_{i-1}.$$

Или, если подставить первую конечную разность нисходящую

$$\Delta^2 Y_i = Y_{i+2} - 2Y_{i+1} + Y_i.$$

Конечно-разностные приближения бесконечно-малых величин

Основы компьютерной графики

- Графические форматы представляют собой способ организации файла для хранения изображения и подразделяются на: **растровые**, **векторные** и **комплексные**.
- **Растровый** формат характеризуется тем, что все изображение по вертикали и горизонтали разбивается на достаточно мелкие прямоугольники – так называемые **пиксели**
- Минимально возможный размер пикселя называется **зерном экрана** и измеряется в мм (или в дюймах)
- максимально возможное количество пикселей по горизонтали на количество пикселей по вертикали экрана называется **разрешающей способностью** монитора

- Количество бит видеопамяти, отводимых под хранение каждого пикселя, называется **глубиной цвета**
- В черно-белых мониторах достаточно одного бита для хранения цвета пикселя, чтобы отражать на мониторе 256 цветов (если сам монитор это позволяет), уже требуется выделять по 8 бит ($2^8=256$ – формула Хартли) для хранения каждого пикселя.
- файлы, которые хранят данные в описанном выше виде, имеют формат ***BMP*** (*Bit MaP*).
- Все остальные графические форматы используют те или иные **методы сжатия** данных, так как *BMP*-файлы требуют слишком много памяти для хранения.

Базовые цвета

- Цветовые оттенки получают смешением базовых цветов **RGB**

Red = “#FF0000” **Green** = "#008000”

Blue = "#0000FF" ”

- В полиграфии для повышения качества применяют 4 базовых цвета: CMYK

Растровые графические форматы

- ***PCX***. Исторический графический формат эпохи «до *Windows*», используется только для совместимости со старыми программами.
- ***GIF***. Популярный графический формат эпохи появления Интернет. Поддерживает только до 256 цветов, использует метод «сжатие без потерь»(*LZW*), режим постепенного появления изображения (сначала каждая 8-я строка, затем каждая 4-я и т.д.) благодаря чему файлы получались небольшие и быстро открывались на web-страницах. Поддерживает анимацию (несколько кадров в одном файле).
- ***PNG***. Является плодом труда сообщества независимых программистов, как ответная реакция на переход формата *GIF* в разряд коммерческих продуктов.
- ***JPEG*** (или *JPG*). Самый популярный формат для хранения фотографических изображений, использует метод «сжатие с потерей» данных, но эти «потери» малозаметны для человеческого глаза. Степень сжатия *JPEG* и, соответственно, потери информации, можно регулировать. Не рекомендуется использовать формат *JPEG* для изображений, подлежащих последующей обработке, так как при каждом сохранении документа в этом формате происходит ухудшение качества изображения.
- ***TIFF***. Разработан специально для издательских систем, требующих изображения наилучшего качества.
- ***PSD***. Стандартный формат фирмы *Adobe,* отличается от большинства обычных растровых форматов поддержкой слоев, альфа-каналов, прозрачности, контуров, векторных надписей. Он содержит много дополнительных переменных (не уступает *TIFF* по их количеству) и сжимает изображения, используя алгоритм сжатия без потерь *RLE.*

Векторные графические форматы

- Изображение в **векторном формате** состоит из математических описаний простейших элементов (линия, ломаная, кривая Безье, эллипс, прямоугольник и т.д.), для каждого из которых определен ряд атрибутов (например, для замкнутого многоугольника - координаты угловых точек, толщина и цвет контурной линии, тип и цвета заливки и т.д.). Хранится и место расположения объектов на странице.
- Растровый позволяет передавать тонкие, едва уловимые детали образов, векторный же лучше применять, если оригинал имеет отчетливые геометрические очертания. Векторные файлы меньше по объему и не теряют качество при масштабировании.
- Многие фирмы разрабатывают собственные векторные форматы. Примеры: *AI (Adobe Illustrator), CDR (Corel Draw), DXF (AutoCAD), SVG* (стандарт web-консорциума), *WMF (Windows)*.

Пример масштабирования изображения в векторном формате

$$\Phi = \sum_{i=1}^{N} (y_i - f(x_i))^2 \to \min$$

$$\Phi = \sum_{i=1}^{N} (y_i - f(x_i))^2 \to \min$$

Пример масштабирования изображения в растровом формате

$$\Phi = \sum_{i=1}^{N} (y_i - f(x_i))^2 \to \min$$

$$\Phi = \sum_{i=1}^{N} (y_i - f(x_i))^2 \to \min$$

Комплексные графические форматы

- **PDF**. Разработан фирмой Adobe. Позволяет внедрять необходимые
шрифты, векторные и растровые изображения, формы и мультимедиа-вставки. Поддерживает несколько типов сжатия растровой информации. Имеет собственные технические форматы для полиграфии. Включает механизм электронных подписей для защиты и проверки подлинности документов. В этом формате распространяется деловая документация.
- **DjVu** – технология сжатия изображений с потерями, разработанная специально для **хранения сканированных документов** – книг, журналов, рукописей и прочее. DjVu-файл может содержать текстовый слой, что позволяет осуществлять полнотекстовый поиск по файлу. Кроме того, DjVu-файл может содержать встроенное интерактивное оглавление и активные области – ссылки, что позволяет реализовывать удобную навигацию в DjVu-книгах. Обеспечивает наибольшую степень сжатия (до 100).

Типы мониторов

- ЭЛТ – (*cathode ray tube, CRT*) на основе свечения люминофора от потока электронов
- ЖК – (*liquid crystal display, LCD*) на основе свойства пропускать или не пропускать свет
- Плазменный – на основе образования плазмы в газоразрядных микроячейках (неон, ксенон)
- OLED-монитор – (*organic light-emitting diode)* на основе органических светоизлучающих диодов
- Виртуальный ретинальный монитор – технология устройств вывода, формирующая изображение непосредственно на сетчатке глаза.
- Лазерный – на основе лазерной панели (пока только внедряется в производство)

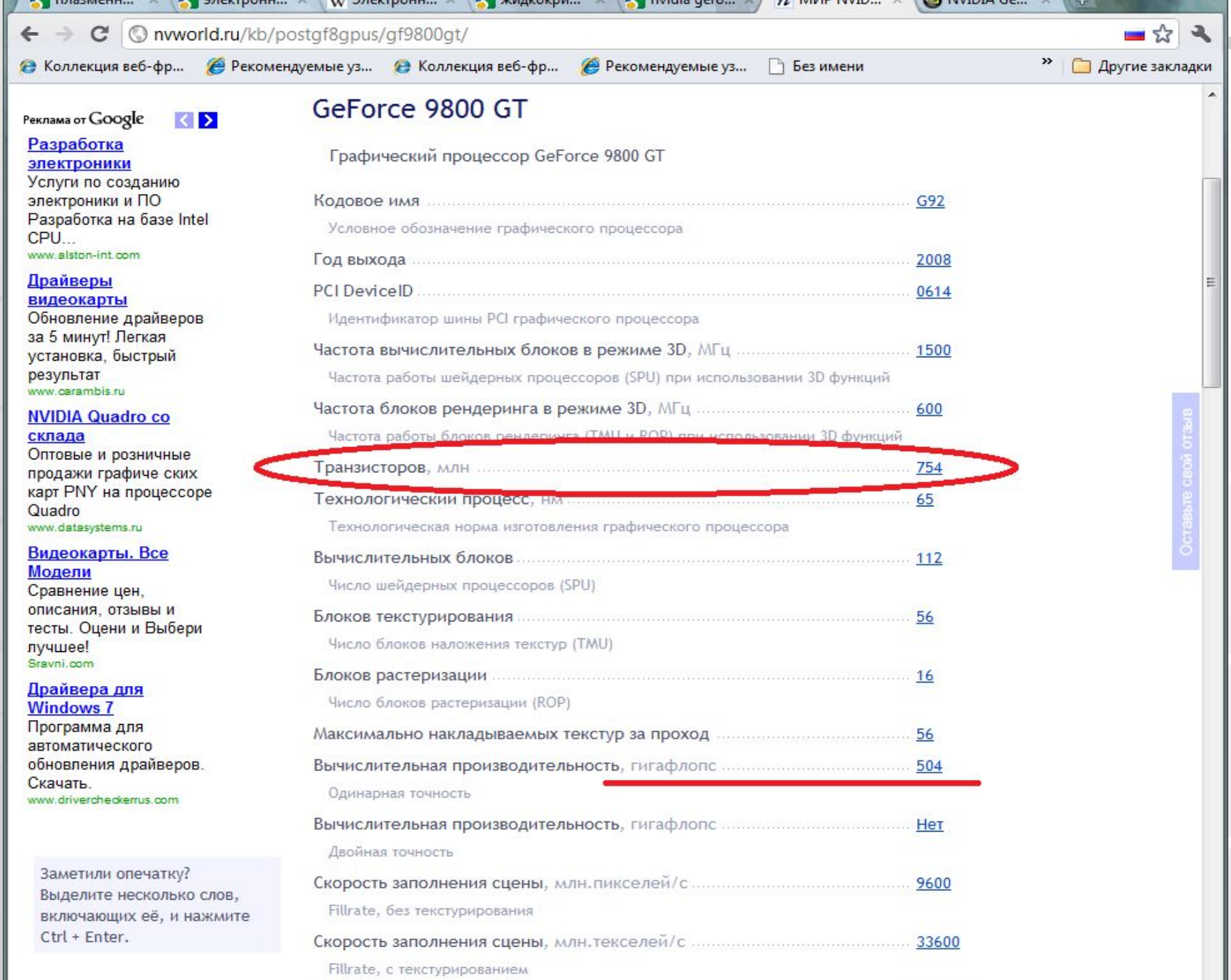
Аппаратно-программная поддержка компьютерной графики

Возможности для работы с компьютерной графикой определяются комплексом характеристик, включающим в себя:

- *производительность процессора*
- *объем оперативной памяти*
- *объем видеопамяти*
- *тактовую частоту системной шины*
- *характеристики монитора*
- *наличие графического ускорителя и специального программного обеспечения*.

Провал по одной из этих позиций делает невозможным эффективную работу с компьютерной графикой.

- **Графические ускорители** – это встраиваемые в видеокарты дополнительные микросхемы, в которых аппаратно реализованы типовые для *3D*-графики функции обработки данных.
- **Графические процессоры** – это специализированные процессоры для ускорения компьютерной графики.
- В качестве примера аппаратной поддержки компьютерной графики можно привести графический ускоритель компании *Gigabyte Technology* с процессором ***Nvidia GeForce 310M*** *: 16 ядер, встроенные OpenGL DirectX, 1,5 ГГц, 2560*1600*
- в 2009 г. передовые фирмы работали по 65-нм и начали переход на 45-нм, идут исследования 30-нм, а в Россию разрешили экспорт технологии 130-нм (РосНАНО).


nvworld.ru/kb/postgf8gpus/gf9800gt/
Коллекция веб-фр...
Рекомендуемые уз...
Коллекция веб-фр...
Рекомендуемые уз...
Без имени
Другие закладки
Реклама от Google
Разработка электроники
Услуги по созданию электроники и ПО Разработка на базе Intel CPU...
www.alston-int.com
Драйверы видеокарты
Обновление драйверов за 5 минут! Легкая установка, быстрый результат
www.carambis.ru
NVIDIA Quadro со склада
Оптовые и розничные продажи графиче ских карт PNY на процессоре Quadro
www.datasystems.ru
Видеокарты. Все Модели
Сравнение цен, описания, отзывы и тесты. Оцени и Выбери лучшее!
Sravni.com
Драйвера для Windows 7
Программа для автоматического обновления драйверов. Скачать.
www.drivercheckerrus.com
Заметили опечатку? Выделите несколько слов, включающих её, и нажмите Ctrl + Enter.
GeForce 9800 GT
Графический процессор GeForce 9800 GT
Кодовое имя G92
Условное обозначение графического процессора
Год выхода 2008
PCI DeviceID 0614
Идентификатор шины PCI графического процессора
Частота вычислительных блоков в режиме 3D, МГц 1500
Частота работы шейдерных процессоров (SPU) при использовании 3D функций
Частота блоков рендеринга в режиме 3D, МГц 600
Частота работы блоков рендеринга (TMU и ROP) при использовании 3D функций
Транзисторов, млн 754
Технологический процесс, нм 65
Технологическая норма изготовления графического процессора
Вычислительных блоков 112
Число шейдерных процессоров (SPU)
Блоков текстурирования 56
Число блоков наложения текстур (TMU)
Блоков растеризации 16
Число блоков растеризации (ROP)
Максимально накладываемых текстур за проход 56
Вычислительная производительность, гигафлопс 504
Одинарная точность
Вычислительная производительность, гигафлопс Нет
Двойная точность
Скорость заполнения сцены, млн.пикселей/с 9600
Fillrate, без текстурирования
Скорость заполнения сцены, млн.текселей/с 33600
Fillrate, с текстурированием
Оставьте свой отзыв

- ***DirectX*** – это библиотека программ компании *Microsoft:* средства *Windows API* для работы с аппаратурой напрямую или с минимальным количеством «посредников», что ускоряет работу программы. *DirectX* содержит компоненты для работы с 2*D*-графикой (*DirectDraw*), 3*D*-графикой (*Direct3D*), звуком (*DirectSound*), устройствами управления (*DirectInput*), сетями (*DirectPlay*) и т.д. Для написания программ, использующих *DirectX*, требуется *MS DirectX SDK* (для *C/C++*) или *SDK*, адаптированный под другие языки.
- ***OpenGL*** – это библиотека программ компании *Silicon Graphics*, которые реализованы как аппаратно-независимые, но используют ускорители графики, если они доступны. Большинство графических приложений требуют для своей работы наличие ***DirectX*** или ***OpenGL***

Основы *3D*-графики

Пространство на мониторе, предназначенное для размещения трехмерного изображения, называется **сценой.**

Каждая **сцена** состоит из:

- набора **объектов,**
- набора **источников света,**
- набора **текстур,**
- набора **камер** наблюдения(обычно используется одна).

Каждый **объект** задается:

- набором **вершин** (**вершина** определяется своими *3D* координатами и соответствующими ей координатами в текстуре),
- набором **граней** (**грань** определяется тремя вершинами и текстурой, кроме текстуры могут быть заданы, например, коэффициенты рассеивания и отражения света),
- **поведением** объекта (расположение, то есть смещение, ось поворота, угол поворота, коэффициент масштабирования, и т.д.) в зависимости от номера кадра; обычно задается в нескольких ключевых точках и интерполируется между ними с помощью сплайнов).

Каждый источник света должен иметь:

- **положение** (координаты на сцене или смещение координат относительно некоторой точки),
- **ориентацию** (точка, в которую направлен этот источник, *target*),
- **тип** (фоновый, направленный, ненаправленный),
- **цвет** (обычно ***RGB***).

Каждая **текстура** представляет собой прямоугольную 2*D* картинку, часто бывает фиксированных размеров (например, 64x64, 128x128, 256x256 пикселей).

Каждая камера задается следующим:

- **положением** (*location*),
- **направлением** (точнее, точкой, в которую направлена эта камера; *target*),
- **углом зрения** (*FOV*),
- **углом поворота** относительно своей оси (*roll*).

- **Альфа-канал** – специальный канал, входящий в описание цвета (*RGB*), который отвечает за прозрачность данного цвета. Таким образом, цвет с учетом альфа-канала описывается как *ARGB*.
- **Проецирование** – процесс отображение трехмерного объекта на плоскости с точки расположения камеры.
- **Растеризация** – разделение объекта на пиксели, то есть представление в виде растрового формата.
- **Рендеринг** – процесс воссоздания трехмерного изображения на экране, по его «скелетной» фигуре, требующий больших вычислительных ресурсов.
- **Текстурирование (*Texture Mapping*)** – вычислительный процесс наложения («натягивания») двухмерных текстур на трехмерный объект, для придания ему более реалистичного изображения без больших вычислительных затрат на рендеринг.

- **Интерполяция** – математический способ восстановления отсутствующей информации. Например, необходимо увеличить размер изображения в 2 раза, со 100 пикселей до 200. Недостающие пиксели генерируются с помощью интерполяции пикселей, соседних с тем, который необходимо восстановить. После восстановления всех недостающих пикселей получается 200 пикселей вместо 100 существовавших, и таким образом, изображение увеличилось вдвое.
- **Анти-алиасинг (*Anti-aliasing*)** – способ обработки (интерполяции) пикселов для получения более четких краев (границ) изображения. Наиболее часто используемая техника, для создания плавного перехода от цвета линии или края объекта к цвету фона. В некоторых случаях, результатом является смазывание (*blurring*) краев.

- **Фрейм-буфер** – буфер кадра. Специально отведенная область памяти компьютера или отдельной платы для временного хранения данных о пикселях, требуемых для отображения одного кадра (полного изображения) на экране монитора. Емкость буфера кадра определяется количеством битов, задействованных для определения каждого пикселя, который должен отображать изменяемую область или количество цветов и их интенсивность на экране.
- ***Z*-буфер** – часть графической памяти, в которой хранятся расстояния от точки наблюдения до каждого пикселя (значения *Z*). *Z*-буфер определяет, какая из многих перекрывающихся точек наиболее близка к плоскости наблюдения.
- С помощью *z*-буфера реализуется процедура **удаления невидимых поверхностей**. Обычно *z*-буфер имеет не менее 16 бит на пиксель для представления глубины цвета. Аппаратные акселераторы *3D*-графики могут иметь собственный *z*-буфер на графической карте.

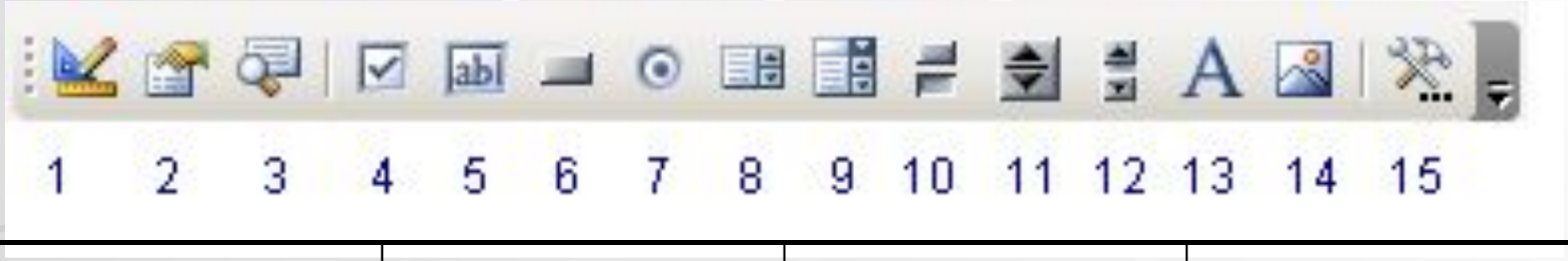
- **Триангуляция** – процесс деления изображения на более мелкие формы, в частности – треугольники, так как они легче обсчитываются и ими легче манипулировать для преобразования изображения.
- **Спот (*Spot*)** – световой источник, похожий на точечный. Он светит не во всех направлениях, а в пределах некого конуса. Освещаются только объекты, попадающие в этот конус.
- ***Ambient*** – световой источник, который светит одинаково во всех направлениях. Все объекты освещаются с равной интенсивностью.

- ***MIP-Mapping*** – дословно "много в одном". Метод улучшения качества текстурных изображений при помощи использования текстур с разным разрешением для различных объектов одного и того же изображения в зависимости от их размера и глубины. Таким образом, в памяти хранятся несколько копий текстурированного изображения в различных разрешениях. В результате этого изображение остается качественным при приближении к объекту и при удалении от него. При использовании этого метода Вы увидите изображение в высоком разрешении, находясь близко от объекта, и изображение в низком разрешении, при удалении от объекта. *MIP-Mapping* снижает мерцание и «зашумленность» изображения, возникающие при *texture mapping.*

- ***Bump Texture Mapping*** – в отличие от *texture mapping* , технология *bump mapping* подразумевает использование, как минимум, еще одной (обычно в оттеках серого) текстуры, которая служит в качестве карты для рельефа, который должен проявиться при визуализации. Эта технология разработана для придания дополнительной детализации и объемности объектам без изменения их геометрических размеров.
- **Трассировка лучей (*Ray Tracing*)** – один из самых сложных и качественных методов построения реалистических изображений. Наиболее распространен вариант «обратной трассировки лучей»: от глаза наблюдателя. Через пиксель строящегося изображения, проводят луч и, учитывая все его отражения от объектов, вычисляют цвет этого пикселя.

- ***Flat Shading* (*Flat*)** – метод затенения, называемый также постоянным затенением. Поверхность объекта, построенного с использованием этого метода, получается наиболее низкого качества, и изображение выглядит как бы поделенным на блоки.
- ***Gouraud Shading (Smooth shading)*** – **затенение методом Гуро** (или плавное затенение), один из наиболее популярных алгоритмов затенения, который обеспечивает прорисовку плавных теней вокруг изображаемого объекта, что позволяет изображать трехмерные объекты на плоском экране. *Gouraud Shading*, или цветовая интерполяция – процесс, с помощью которого цветовая информация интерполируется по поверхности многоугольника для определения цветов в каждом пикселе.

Элементы управления ActiveX



1 – режим Конструктор	**2** - Свойства	**3** – текст программы	**4** -флажок *CheckBox*
5 – поле *TextBox*	**6** – ?	**7** – *Option Button*	**8** - Список *ListBox*
9 -Поле со списком *ComboBox*	**10** –*Toggle Button*	**11** счетчик *SpinButton*	**12** – *ScrollBar*
13 – надпись	**14** – рисунок	15 – другие ЭУ:	Самостоя-тельно

Базы данных

- **База данных** (БД, *database*) – поименованная совокупность структурированных данных, относящихся к определенной предметной области.
- **Предметная область** – некоторая часть реально существующей системы, функционирующая как самостоятельная единица. Полная предметная область может представлять собой экономику страны или отрасли, однако на практике для информационных систем наибольшее значение имеет предметная область масштаба отдельного предприятия или корпорации.
- **Система управления базами данных** (СУБД) – комплекс программных и языковых средств, необходимых для: **добавления**, **модификации**, **удаления**, **поиска и отбора** данных, **представления** данных на экране и в печатном виде, **разграничения** прав доступа к данным.

Типы баз данных

- Реляционная БД. Состоит из таблиц – **отношений** (*relations*), в которых строка называется **кортежем (записью)**, а столбец называется **атрибутом (полем записи)**: **Oracle**, **DB2**(IBM), **Microsoft Access**, **SQL Server**, **MySQL**
- Иерархическая БД: **Win Explorer**
- Объектно-ориентированная БД: **Interbase**

- **Ключевой элемент таблицы** (ключ, *regular key*) – такое ее поле (простой ключ) или строковое выражение, образованное из значений нескольких полей (составной ключ), по которому можно определить значения других полей таблицы.
- **Первичный ключ** (*primary key*) – главный ключевой элемент, однозначно идентифицирующий строку в таблице. Могут также существовать альтернативный (*candidate key*) и уникальный (*unique key*) ключи, служащие также для идентификации строк в таблице.
- **Внешний ключ** (*foreign key*) – ключевой элемент подчиненной (внешней, дочерней) таблицы, значение которого совпадает со значением первичного ключа главной (родительской) таблицы.

- В реляционных базах данных между таблицами устанавливаются **связи по ключам**, один из которых в главной (***parent***, родительской) таблице – **первичный**, второй – **внешний** ключ – во внешней (***child***, дочерней) таблице, как правило, первичным не является и образует связь **«один ко многим»** (1:N). В случае первичного внешнего ключа связь между таблицами имеет тип **«один к одному»** (1:1).

- **Ссылочная целостность данных** (*referential integrity*) – набор правил, обеспечивающих соответствие ключевых значений в связанных таблицах.
- **Репликация базы данных** – создание копий базы данных (реплик), которые могут обмениваться обновляемыми данными в результате выполнения процесса синхронизации.
- **Транзакция** – изменение информации в базе в результате выполнения одной операции или их последовательности, которое должно быть выполнено полностью или не выполнено вообще.

- **Язык *SQL*** (*Structured Query Language*) – универсальный язык работы с базами данных, включающий возможности ее создания, модификации структуры, отбора данных по запросам, модификации информации в базе и прочие операции манипулирования базой данных.
- **Нормализация** – это формальный метод анализа отношений на основе их первичного ключа и существующих связей. Ее задача – это замена одной схемы (или совокупности отношений) БД другой схемой, в которой отношения имеют более простую и регулярную структуру.

Основные этапы разработки БД

1. Разработка информационно-логической модели предметной области (ИЛМ ПО)
2. Определение логической структуры БД
3. Конструирование объектов БД средствами СУБД
4. Заполнение (загрузка) БД с документов-источников
5. Тестирование работы БД

Подготовка к централизованному тестированию

- http://www.fepo.ru/
- http://ad.cctpu.edu.ru/ -ключ доступа к реальному тестированию:
- ***http://www.analiz-fepo.ru/***

- **22.03.2010г.**
- **ТПУ получил доступ к ТРЕНАЖЕРУ Федерального Интернет-экзамена на весенний семестр 2010г.**
- **Поготовиться к Интернет-экзамену в режиме обучения можно на сайте http://www.i-exam.ru/ Ключ доступа: 41618tt796 Само тестирование в этом году выполняется на сайте http://www.att.nica.ru/ *Ключ доступа: 43862dt796***
- **Вопросы по сравнению с прошлым годом изменились. В сторону современных информационных технологий.**

Как продолжить изучение информатики дальше?

- САМОСТОЯТЕЛЬНО: Intuit.ru, ru.wikipedia.org, Freesoft.ru и т.п.