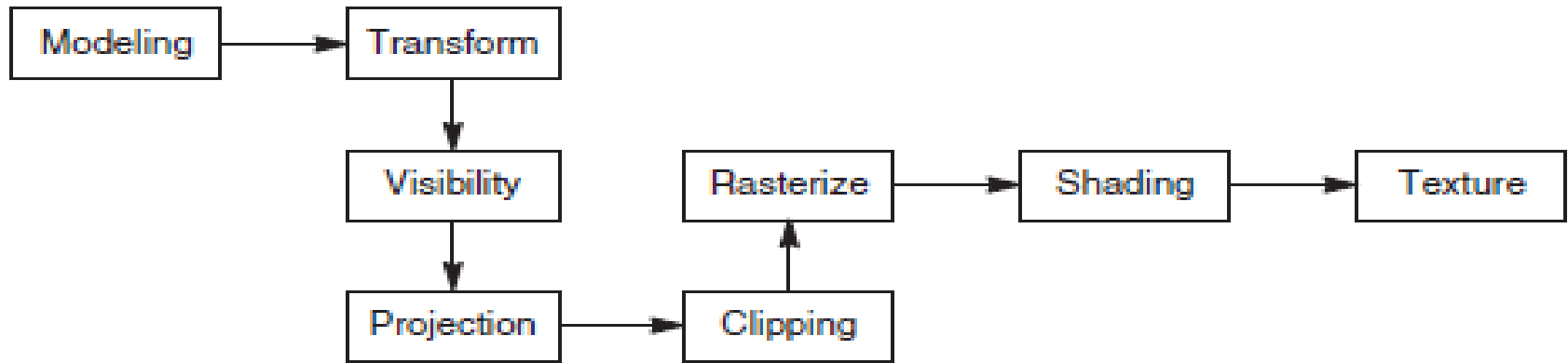


Компьютерная графика

Курс лекций

Графический конвейер



- отображение (rendering):
 - преобразования (transformation) – задание местоположения;
 - определение видимости (visibility) – область видимости (field of view) + нелицевые поверхности → отсечение (clipping);
 - проекция на картинную плоскость (projection);
 - растеризация (rasterization);
 - закрашка (shading);
 - текстурирование (texturing).

Понятие о моделировании

- модель – это абстрактное представление сущности реального мира
 - математическое моделирование физических, химических процессов и др.
- компьютерное моделирование:
 - данные о физических объектах не могут быть целиком введены в компьютер;
 - необходимо априори ограничить объем хранимой информации об объекте;
- задача моделирования:
 - найти вид модели, наилучшим образом отвечающий решаемой задаче.

Геометрическое моделирование

- в компьютерной графике используется геометрическое моделирование - моделирование объектов различной природы с помощью геометрических типов данных;
- методы:
 - получения модели;
 - представления модели;
 - обработки моделей;
- выбор модели:
 - соответствие поставленной задаче;
 - максимальное использование возможностей графической системы;
 - учет задач обработки и редактирования модели.

Свойства объектов, рассматриваемые в процессе моделирования

- свойства тела, связанные с пространством:
 - положение относительно других тел,
 - геометрическая форма,
 - размер тела,
 - ...
- информация о материале тела:
 - цвет,
 - текстура,
 - отражательная способность,
 - прозрачность,
 - плотность,
 - ...
- информация, характеризующая взаимодействие объектов:
 - масса,
 - возможность деформироваться,
 - ...

Характеристики представлений

Представление – класс (тип) цифровой модели трехмерных данных (один объект может иметь несколько представлений)

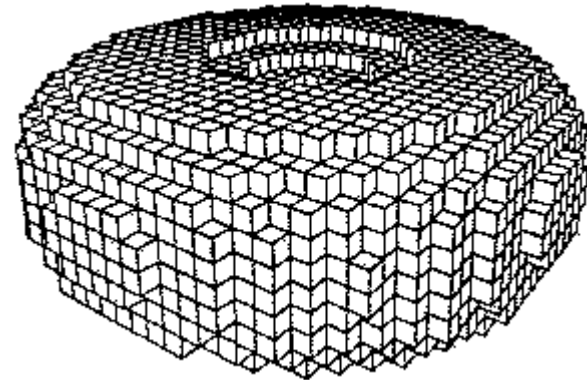
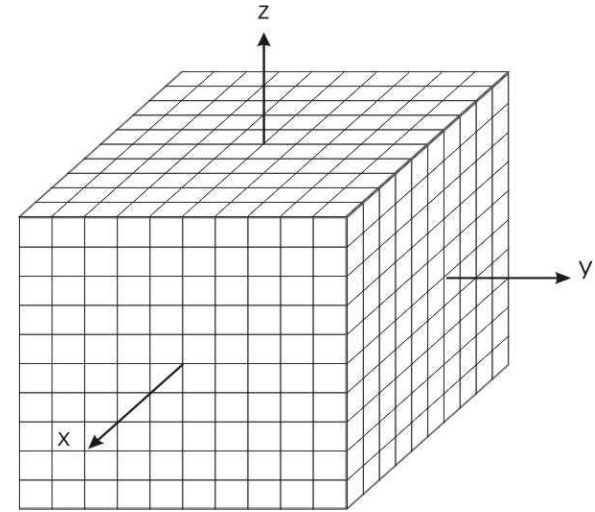
- структура данных;
- алгоритм построения (способ получения представления);
- количество памяти, необходимое для хранения модели;
- типичные свойства представления (алгоритмы, применяемые к данному классу представлений);
- область применения моделей в данном представлении;
- параметризация моделей.

Типы представлений объектов для последующей визуализации

- создание модели трехмерного объекта с последующим ее отображением:
 - объемное представление (разложение пространства на элементы, представление с разбиением пространства, space-partitioning representation):
 - воксельное представление;
 - октарные и бинарные деревья;
 - конструктивная геометрия тел (CSG, Constructive Solid Geometry);
 - граничное (поверхностное) представление:
 - неявное представление;
 - параметрическое задание, сплайны;
 - полигональные поверхности;
- синтез необходимого изображения трехмерных объектов по уже существующим изображениям.

Воксельное представление

- структура:
 - равномерная сетка (трехмерный растр), каждый элемент которой показывает, есть ли в нем часть тела;
 - ячейка называется воксель (voxel = volume element);
 - каждый воксель принимает значение 0 или 1;
 - позволяет также задавать плотность, прозрачность и другие свойства ячеек.
- способ получения:
 - дискретизация трехмерных данных на равномерной сетке

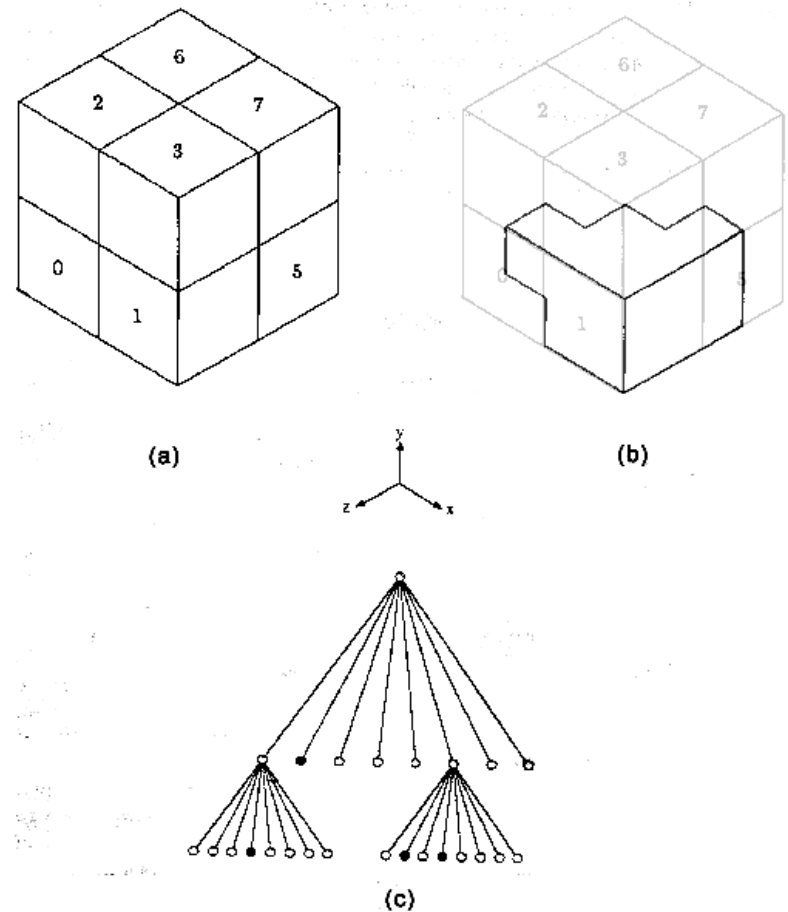


Воксельное представление: типичные алгоритмы

- пространственные алгоритмы
 - вычисление объема объекта;
 - нахождение центра масс;
 - ...
 - булевы операции (пересечение, объединение);
- плохо работают алгоритмы, требующие понятия поверхности (качество приближения поверхностей, не параллельных осям координат, зависит от относительного размера вокселей).

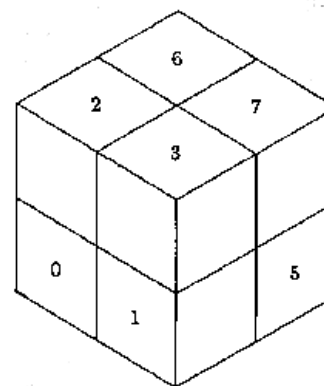
Воксельное представление: октодерево

- применяется для оптимизации воксельного представления;
- структура узла:
 - код узла
 - код = «черный» : пространство заполнено, лист;
 - код = «белый» : пространство пустое, лист;
 - код = «серый» : область пространства частично пуста и частично заполнена.
 - восемь указателей на потомков, пронумерованных от 0 до 7.

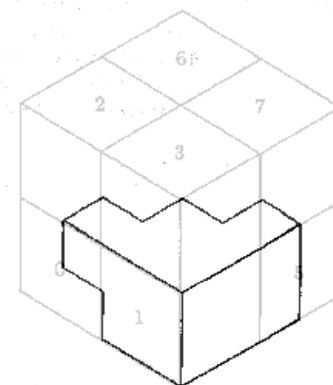


Воксельное представление: октодерево

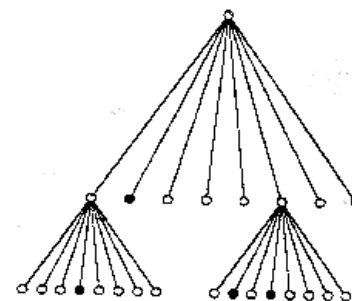
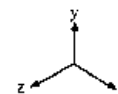
- способ получения:
 - из воксельного представления или непосредственно через дискретизацию:
 - разбиение на октанты;
 - октанты первого уровня соответствуют октантам локальной системы координат объекта;
- свойства:
 - позволяет хранить информацию только о блоках, относящихся к объекту;
 - число элементов пропорционально площади поверхности объекта, т.е. квадрату разрешения (для разреженных моделей позволяет уменьшить размер);
 - поддерживает эффективное выполнение булевых операций;
 - усложняются операции, требующие информации о смежных ячейках.



(a)



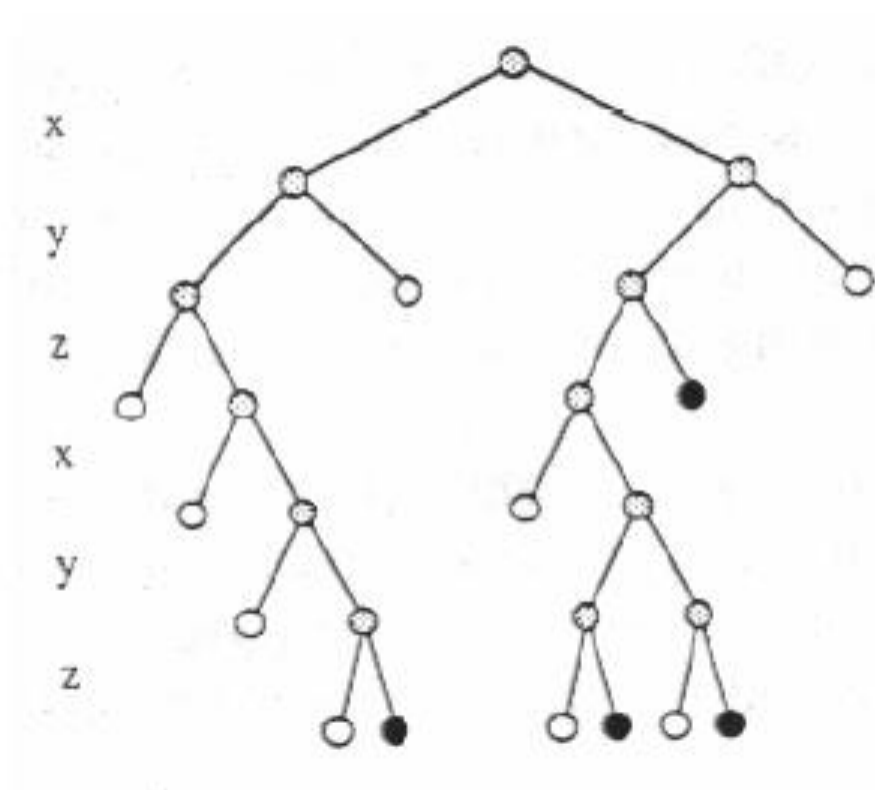
(b)



(c)

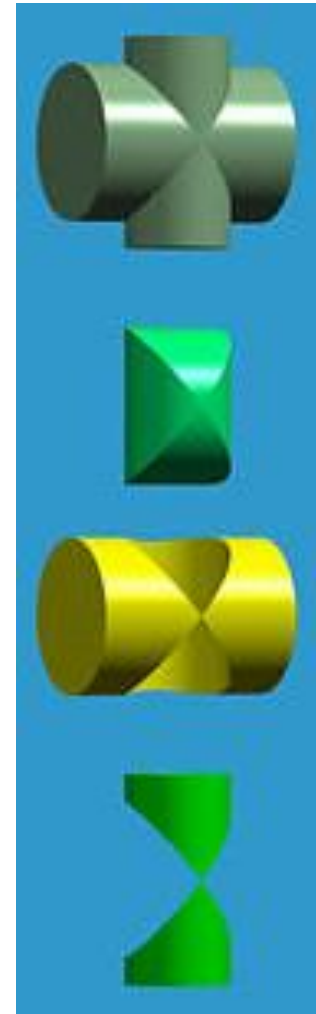
Воксельное представление: бинарное дерево

- бинарное разбиение последовательно в направлении осей **X**, **Y**, **Z** как альтернатива октарному делению



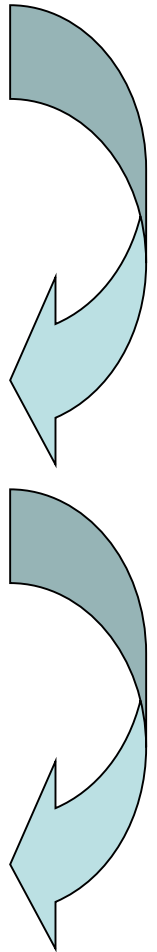
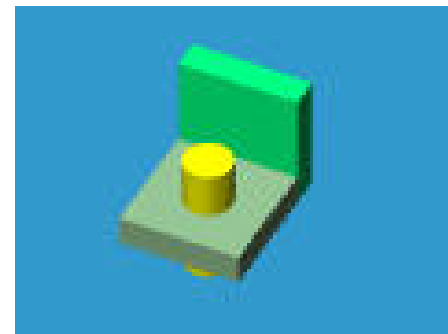
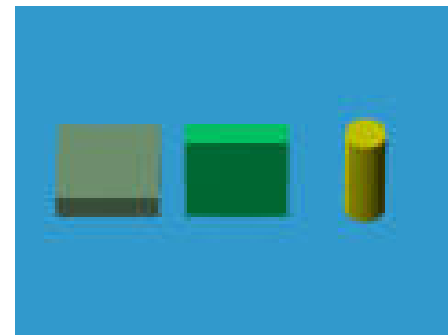
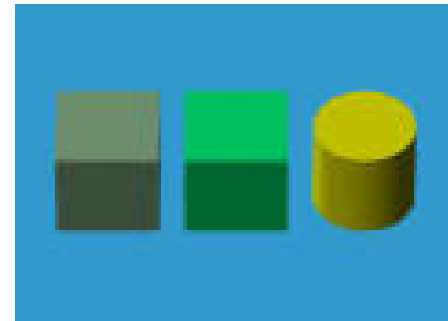
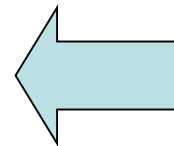
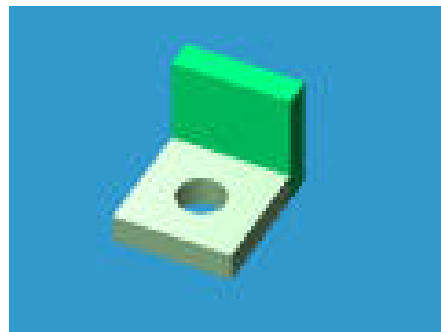
Конструктивная геометрия

- структура:
 - набор базовых примитивов:
 - сфера;
 - куб;
 - цилиндр;
 - ...
 - операции по их комбинированию.
- способ получения:
 - ручное моделирование.
- свойства:
 - описывает объем и поверхность;
 - непрерывное представление;
 - явное представление.



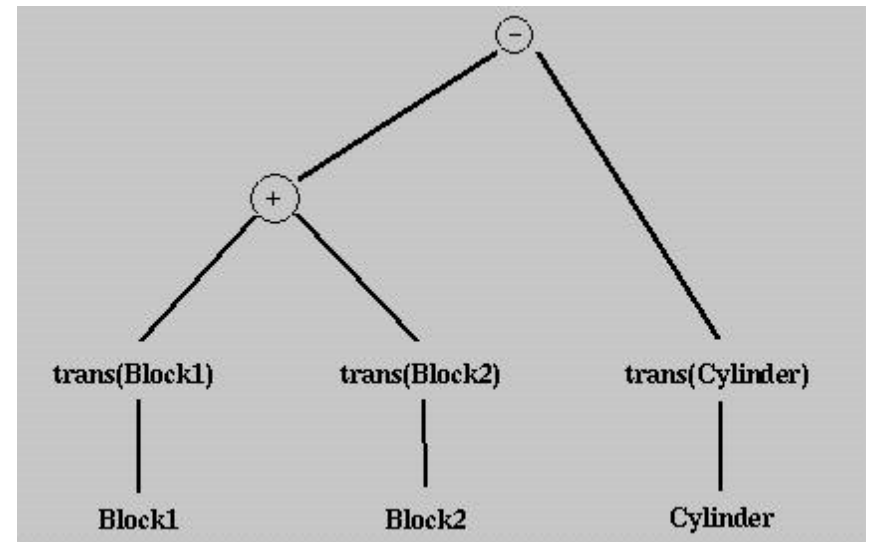
Конструктивная геометрия: операции с телами

- операции по комбинированию базовых примитивов:
 - теоретико-множественные:
 - объединение;
 - разность;
 - пересечение;
 - ...
 - геометрические преобразования:
 - перенос;
 - поворот;
 - масштабирование.

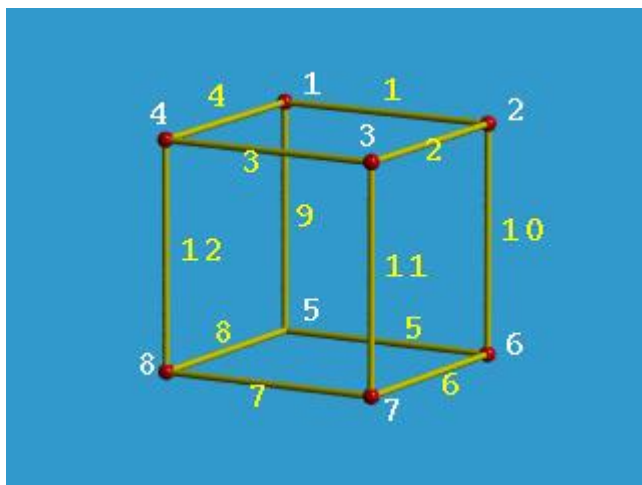


Конструктивная геометрия: структура данных

- дерево из операций и базовых объектов;
- корень — результирующий объект;
- листья — базовые примитивы;
- число потомков равно числу операндов операции;
- из-за повторного использования превращается в направленный ациклический граф.



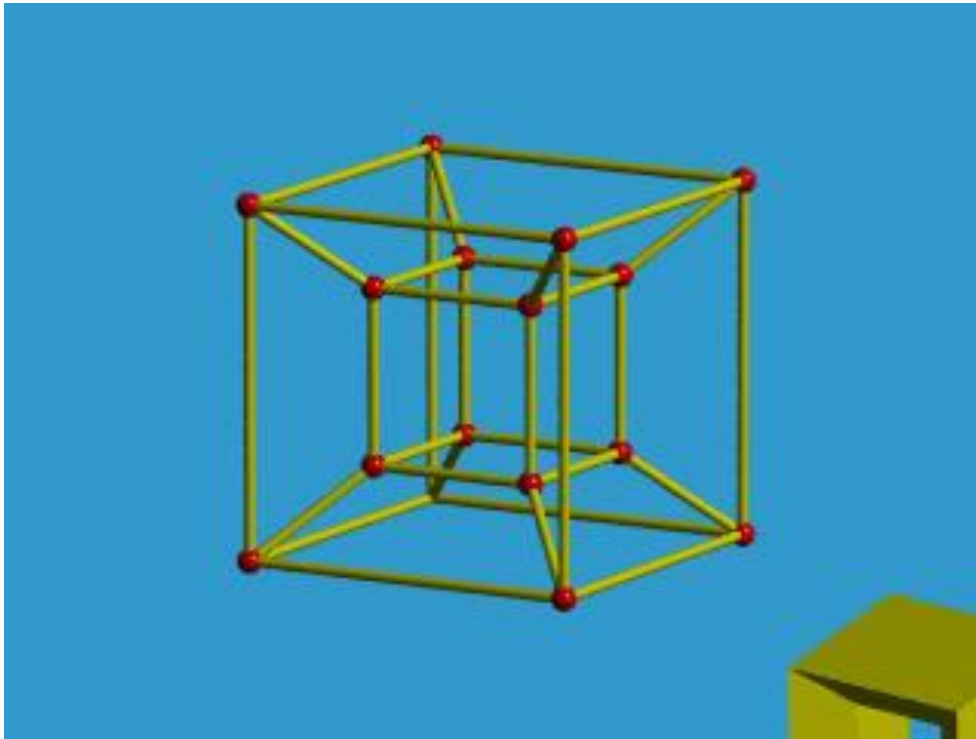
Каркасное представление (wireframe)



i	X	Y	Z
1	1	1	1
2	1	-1	1
3	-1	-1	1
4	-1	1	1
5	1	1	-1
6	1	-1	-1
7	-1	-1	-1
8	-1	1	-1

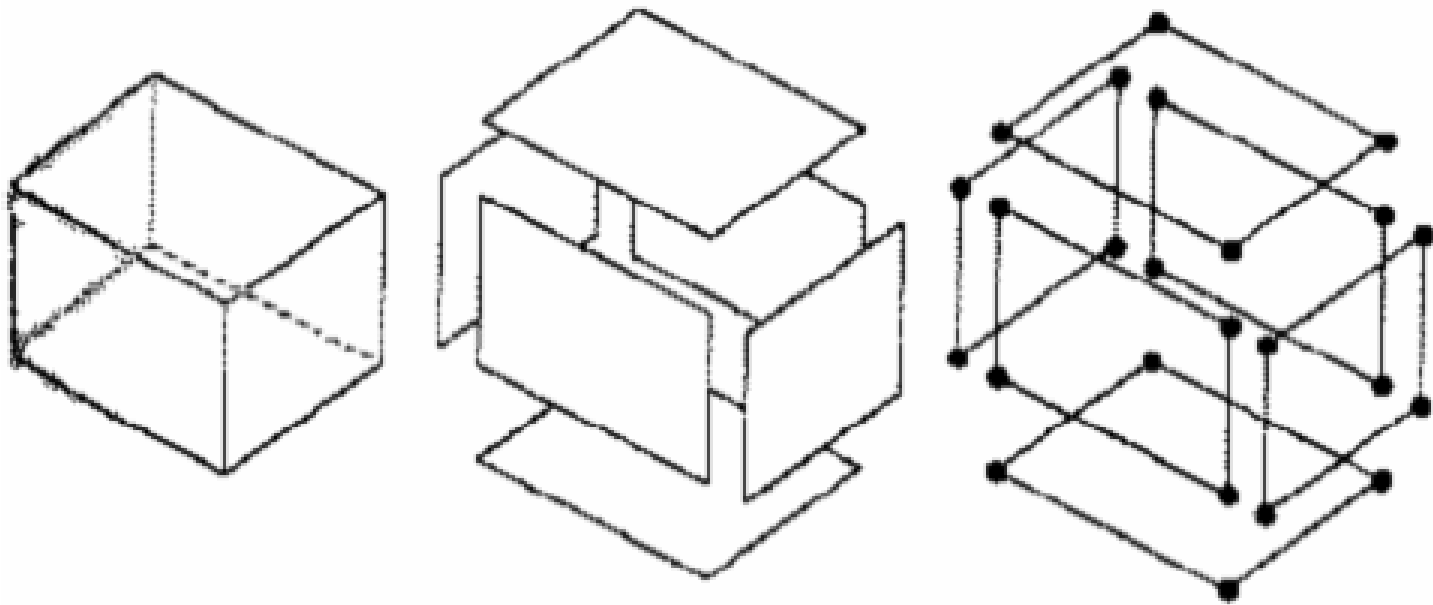
1	(1,2)	2	(2,3)	3	(3,4)
4	(5,6)	5	(5,6)	6	(6,7)
7	(7,8)	8	(8,5)	9	(1,5)
10	(2,6)	11	(3,7)	12	(4,8)

Каркасное представление: неоднозначная интерпретация



Граничное представление (boundary representation, B-REP)

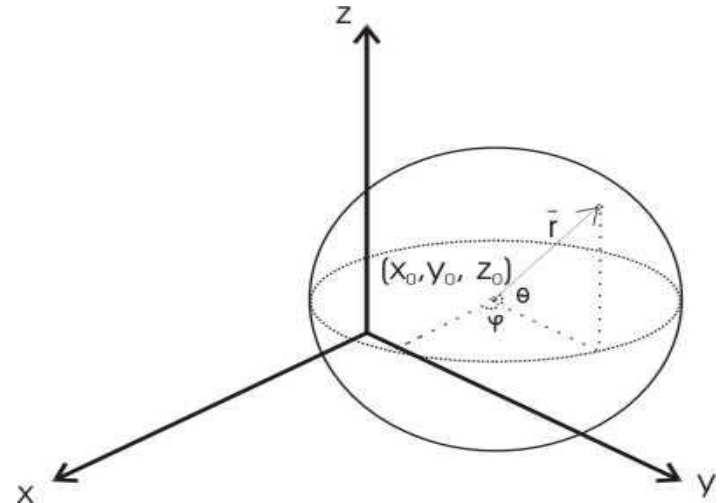
- неявное представление сплошного тела путем описания ограничивающей его поверхности.



Задание поверхности с помощью неявной функции

Неявная форма (implicit form) : $F(x, y, z) = 0$

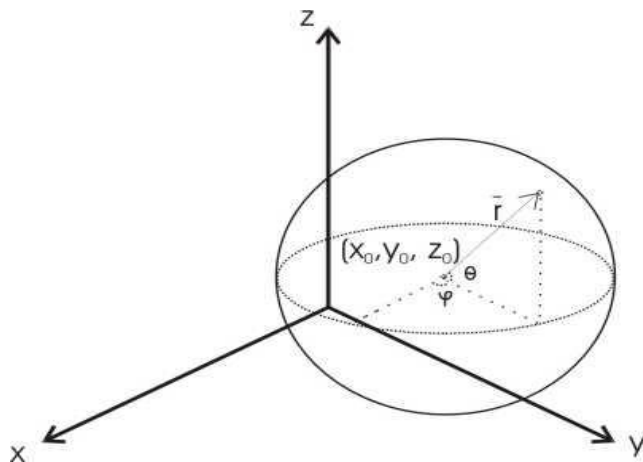
- неявная форма может задавать внутреннюю и внешнюю стороны поверхности:
 - $F(x, y, z) < 0$ – точка внутри поверхности (inside);
 - $F(x, y, z) = 0$ – точка на поверхности;
 - $F(x, y, z) > 0$ – точка вне поверхности (outside);
- влияние аффинного преобразования:
 $F'(P) = F(M^{-1}P)$;
- частные случаи:
 - плоскость:
 - $F(x, y, z) = n_x x + n_y y + n_z z - D$
 - $F(P) = \mathbf{n}(P - B)$
 - сфера:
 - $F(x, y, z) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2$



Параметрическое задание поверхности

- координаты точек поверхности рассматриваются как некие функции от двух параметров, пробегающих некоторый набор значений;
- используются различного рода особые параметрические поверхности, например, поверхности Безье, B-сплайны, в частности, NURBS (Non-Uniform Rational B-Spline, неоднородный рациональный B-сплайн).

$$\begin{cases} x = x(u, v) \\ y = y(u, v) \\ z = z(u, v) \end{cases}$$



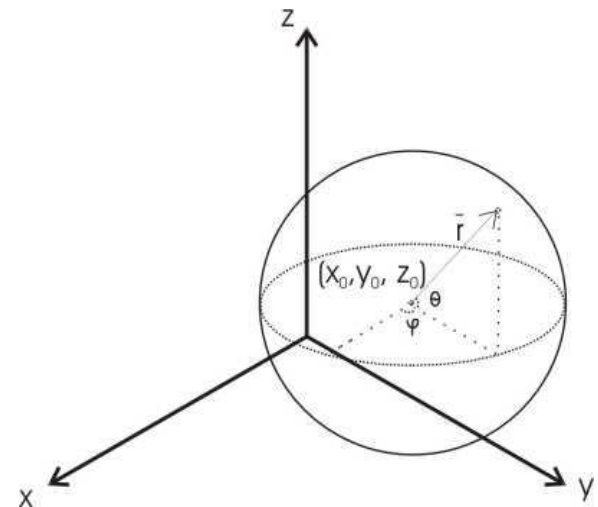
$$\begin{cases} x = x_0 + r \cos \theta \cos \varphi \\ y = y_0 + r \cos \theta \sin \varphi, & 0 \leq \varphi < 2\pi, \quad -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \\ z = z_0 + r \sin \theta \end{cases}$$

Параметрическое задание поверхности

- координаты точек поверхности рассматриваются как некие функции от двух параметров, пробегающих некоторый набор значений: $P(u,v) = (X(u,v), Y(u,v), Z(u,v))$;
- v – *контур* (v-contour) – кривая, образованная изменением параметра u при фиксированном значении параметра v (аналогично вводится понятие u – контура);
- полигональная сетка строится путем разбиения диапазонов значений u и v на интервалы и соединения ребрами вершин, образованных для соседних значений u и v ;
- влияние аффинного преобразования:
 $P'(u,v) = MP(u,v)$
- частные случаи:
 - плоскость:
 - $P(u,v) = C + au + bv$
 - сфера:

$$\begin{cases} x = r \cos \theta \cos \varphi \\ y = r \cos \theta \sin \varphi, & 0 \leq \varphi < 2\pi, -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \\ z = r \sin \theta \end{cases}$$

$$\begin{cases} x = x(u, v) \\ y = y(u, v) \\ z = z(u, v) \end{cases}$$



Нормальный вектор к поверхности

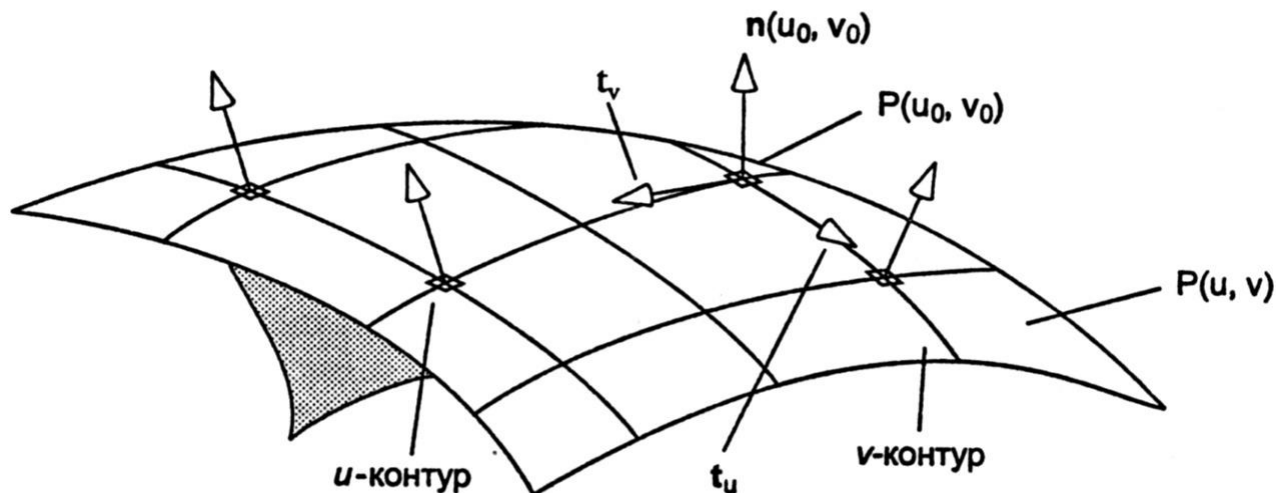
- для поверхности, заданной параметрически:

$$\mathbf{n}(u_0, v_0) = \left(\frac{\partial \mathbf{p}}{\partial u} \times \frac{\partial \mathbf{p}}{\partial v} \right) \Big|_{u=u_0, v=v_0},$$

$$\frac{\partial \mathbf{p}(u, v)}{\partial u} = \left(\frac{\partial X(u, v)}{\partial u}, \frac{\partial Y(u, v)}{\partial u}, \frac{\partial Z(u, v)}{\partial u} \right).$$

- для поверхности, заданной неявно:

$$\mathbf{n}(x_0, y_0, z_0) = \nabla F \Big|_{x=x_0, y=y_0, z=z_0} = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z} \right) \Big|_{x=x_0, y=y_0, z=z_0},$$

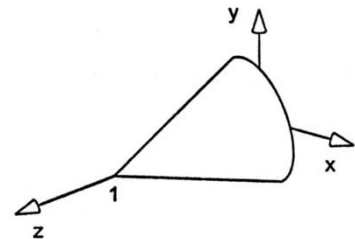
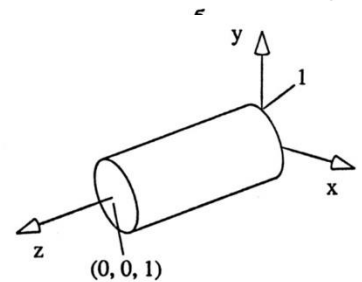
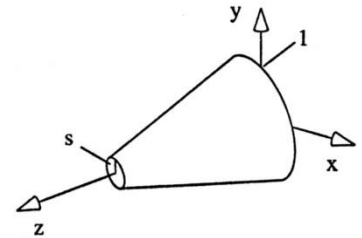
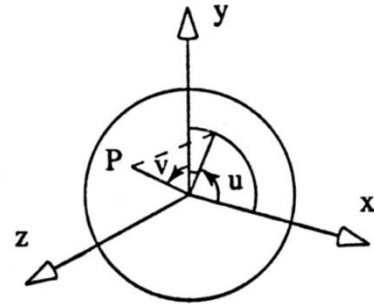


Базовые формы: сфера, цилиндр, конус

- базовая сфера (сфера единичного радиуса):
 - в неявной форме:

$$F(x,y,z) = x^2 + y^2 + z^2 - 1$$
 - в параметрической форме (u соответствует долготе (азимуту), v - широте):

$$P(u,v) = (\cos(v) \cdot \cos(u), \cos(v) \cdot \sin(u), \sin(v))$$
- базовый конический цилиндр:
 - в неявной форме:
 - стенка $F(x,y,z) = x^2 + y^2 - (1 + (s - 1) \cdot z)^2, 0 < z < 1$
 - основание $x^2 + y^2 < s^2 (z = 1)$
 - в параметрической форме:
 - стенка $P(u,v) = ((1 + (s-1) \cdot v) \cdot \cos(u), (1 + (s-1) \cdot v) \cdot \sin(u), v)$
 - основание $P(u,v) = (v \cdot \cos(u), v \cdot \sin(u), 1), v \in [0, s]$
 - частные случаи
 - при $s = 0$: конус
 - при $s = 1$: цилиндр



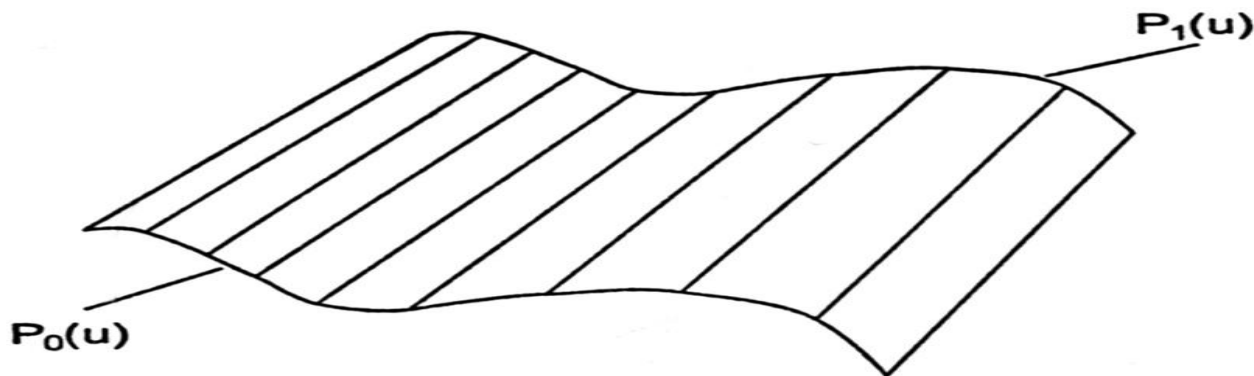
Поверхность	$p(u, v)$ в точке $p(u, v)$	$F(x, y, z)$
Сфера	$p(u, v)$	(x, y, z)
Конический цилиндр	$(\cos(u), \sin(u), 1-s)$	$(x, y, -(s-1)(1 + (s-1)z))$
Цилиндр	$(\cos(u), \sin(u), 0)$	$(x, y, 0)$
Конус	$(\cos(u), \sin(u), 1)$	$(x, y, 1-z)$

Линейчатые поверхности

- поверхность называется *линейчатой*, если через каждую ее точку проходит хотя бы одна прямая, целиком лежащая на данной поверхности;
- параметрическое представление ($P_0(u)$ и $P_1(u)$ определены в одном интервале):

$$P(u,v) = (1-v)*P_0(u) + v*P_1(u)$$

- прямая при $u = u'$ называют *образующей* (ruling) в точке u' ;
- v -контур при $v = v'$ является аффинной комбинацией кривых $P_0(u)$ и $P_1(u)$;
- *линейчатый лоскут* (ruled patch) формируется путем ограничения диапазонов значений u и v .



Линейчатые поверхности: конус, цилиндр, билинейный лоскут

- конус:

- $P_0(u)$ представляет собой единственную точку:

$$P(u,v) = (1-v)*P_0 + v*P_1(u)$$

- частные случаи: круговой конус, прямой круговой конус;

- цилиндр:

- кривая $P_1(u)$ представляет собой смещенную кривую $P_0(u)$ ($P_1(u) = P_0(u) + \mathbf{d}$)

$$P(u,v) = P_0(u) + \mathbf{d}*v$$

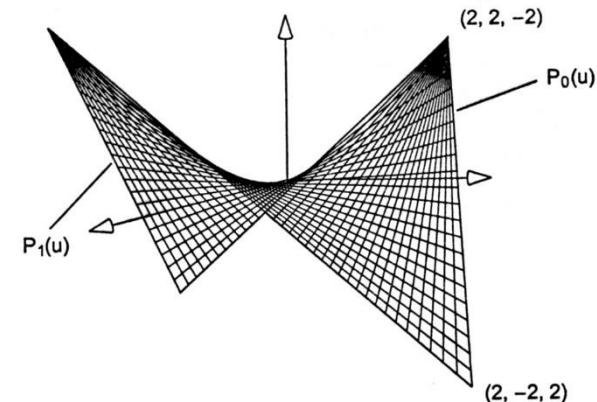
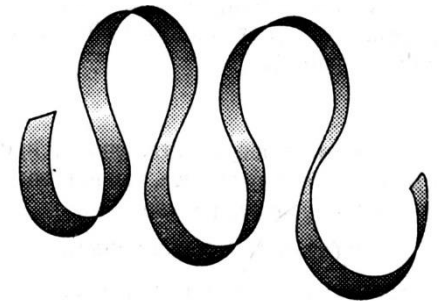
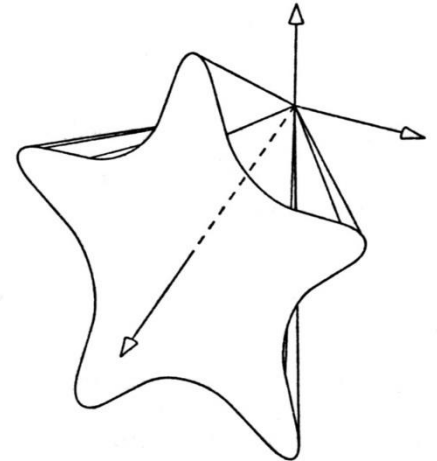
- $P_0(u)$ называют *направляющей* кривой, \mathbf{d} – *образующей*;

- частные случаи: круговой цилиндр, прямой цилиндр;

- билинейный лоскут (bilinear patch):

- $P_0(u)$, $P_1(u)$ являются отрезками прямых линий (для одного интервала u)

$$P(u,v) = (1-v)*(1-u)*P_{00} + (1-v)*u*P_{01} + v*(1-u)*P_{10} + u*v*P_{11}$$



Поверхности вращения

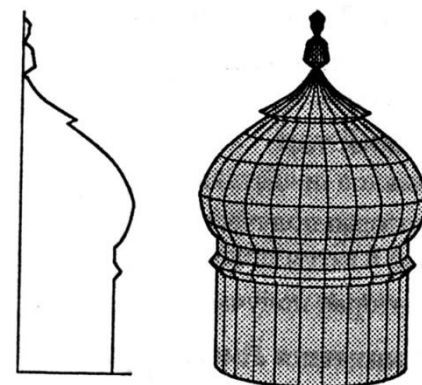
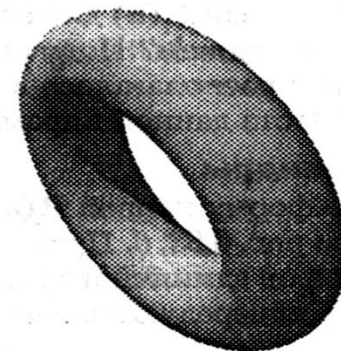
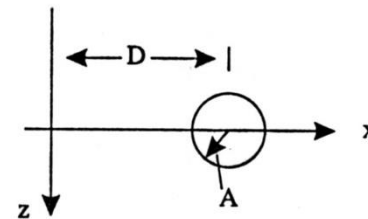
- поверхность вращения образуется посредством *вращательной развертки с заметанием* (rotational sweep) профильной кривой $C(v) = (X(v), Z(v))$ вокруг некоторой оси (профиль помещается в плоскость XOZ и вращается вокруг оси Z , параметр u определяет угол поворота);
- пересечение поверхности вращения с плоскостью, перпендикулярной оси вращения, называется *параллелью*;
- пересечение поверхности вращения с плоскостью, содержащей ось вращения, называется *меридианом*;
- параметрическое задание поверхности и нормали:

$$P(u,v) = (X(v)*\cos(u), X(v)*\sin(u), Z(v))$$

$$n(u,v) = X(v)*(Z'(v)*\cos(u), Z'(v)*\sin(u), -X'(v))$$

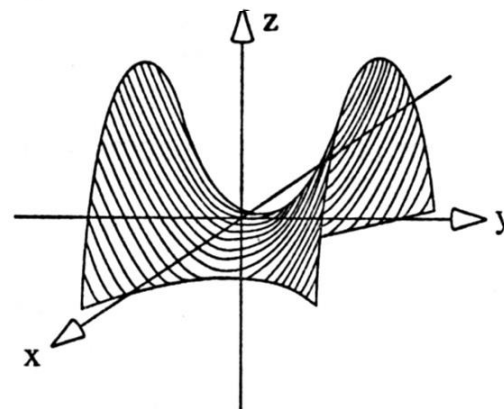
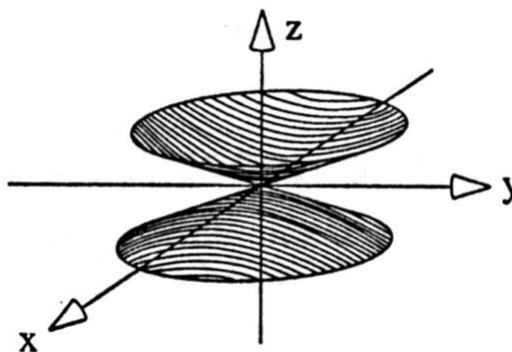
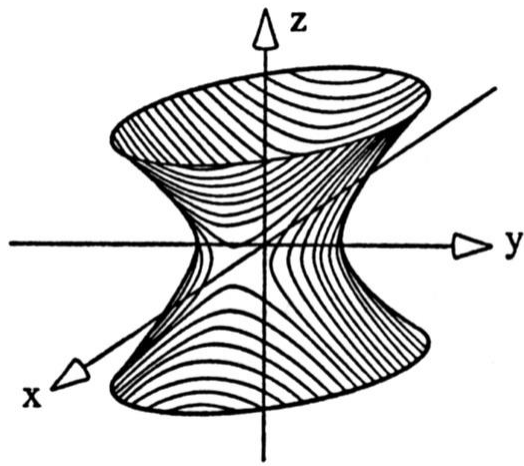
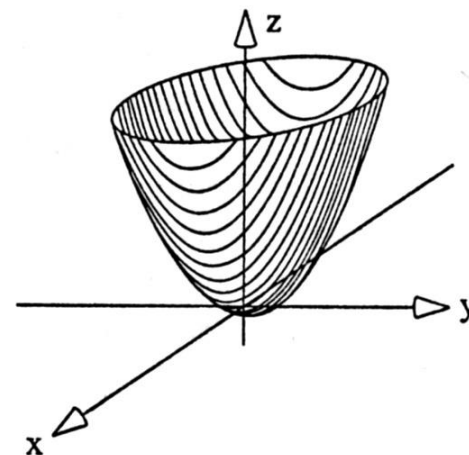
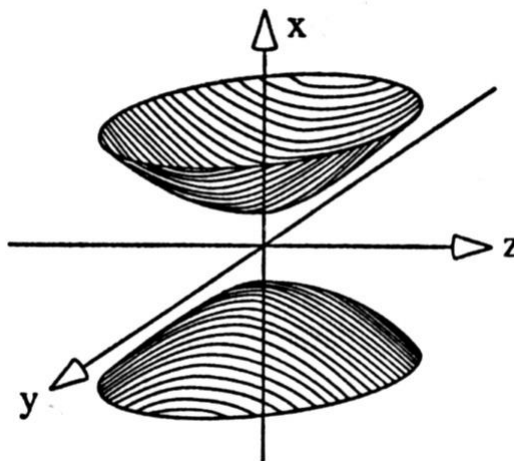
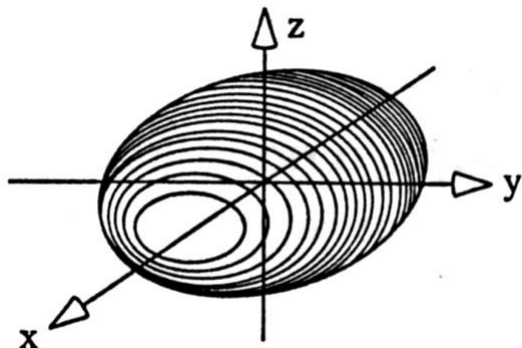
- частные случаи: сфера, конический цилиндр, конус, тор;
- тор:

$$P(u,v) = ((R+r*\cos(v))*\cos(u), (R+r*\cos(v))*\sin(u), r*\sin(v))$$



Поверхности второго порядка (квадрики)

Название квадрики	Неявная форма	Параметрическая форма	Интервал изменения по v и по u
Эллипсоид	$x^2 + y^2 + z^2 = 1$	$(\cos(v) \cos(u), \cos(v) \sin(u), \sin(v))$	$(-\pi/2, \pi/2), (-\pi, \pi)$
Однополостный гиперболоид	$x^2 + y^2 - z^2 = 1$	$(\sec(v) \cos(u), \sec(v) \sin(u), \operatorname{tg}(v))$	$(-\pi/2, \pi/2), (-\pi, \pi)$
Двуполостный гиперболоид	$x^2 - y^2 - z^2 = 1$	$(\sec(v) \cos(u), \sec(v) \operatorname{tg}(u), \operatorname{tg}(v))$	$(-\pi/2, \pi/2)^1$
Эллиптический конус	$x^2 + y^2 - z^2 = 0$	$(v \cos(u), v \sin(u), v)$	Любые вещ. числа, $(-\pi, \pi)$
Эллиптический параболоид	$x^2 + y^2 + z = 0$	$(v \cos(u), v \sin(u), v^2)$	$v \geq 0, (-\pi, \pi)$
Гиперболический параболоид	$-x^2 + y^2 - z = 0$	$(v \operatorname{tg}(u), v \sec(u), v^2)$	$v \geq 0, (-\pi, \pi)$

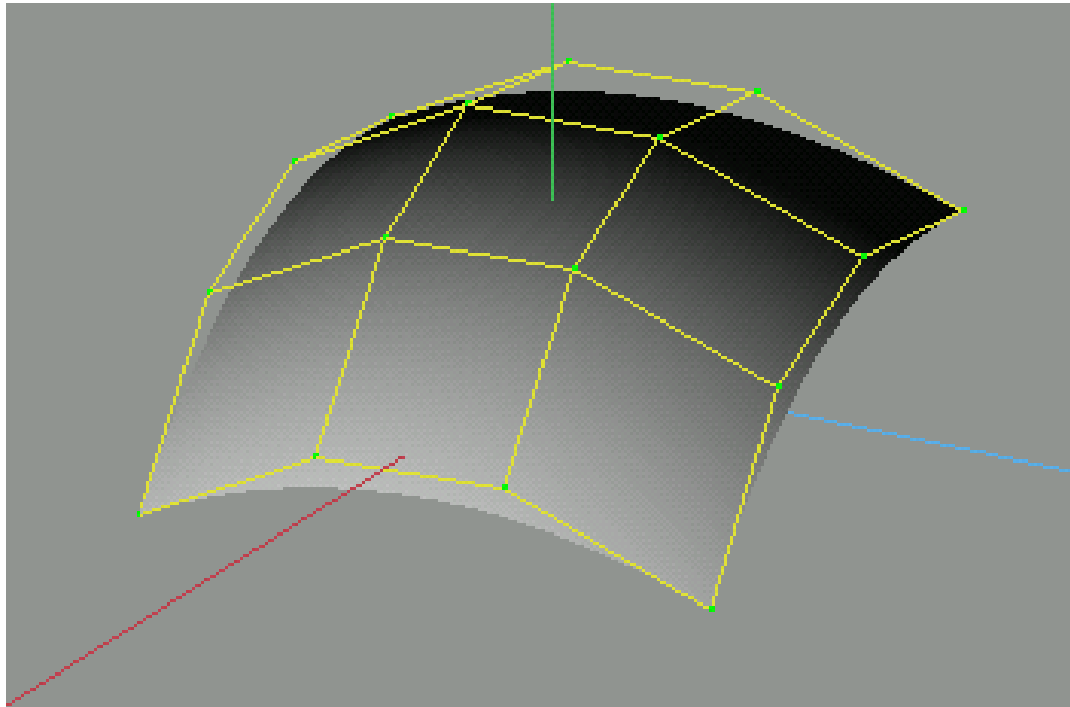


Граничное представление: разбиение на грани

- глобальная параметризация объекта произвольной формы является трудновыполнимой задачей: поверхность приближается набором *граней (face)*, каждая из которых имеет компактное математическое представление;
- границы граней представляются *ребрами (edge)*;
- часть кривой, формирующей ребро, заканчивается *вершинами (vertex)*;
- поверхностная модель, которая состоит исключительно из плоских граней, называется *полигональной моделью (полигональной сеткой, polygonal mesh)*;
- полигональная модель может задавать:
 - многогранники;
 - аппроксимации гладких объектов (линейная интерполяция).

Граничные представления высших порядков

- контрольные точки + способ интерполяции 2-го порядка и выше (например, полиномы Эрмита)

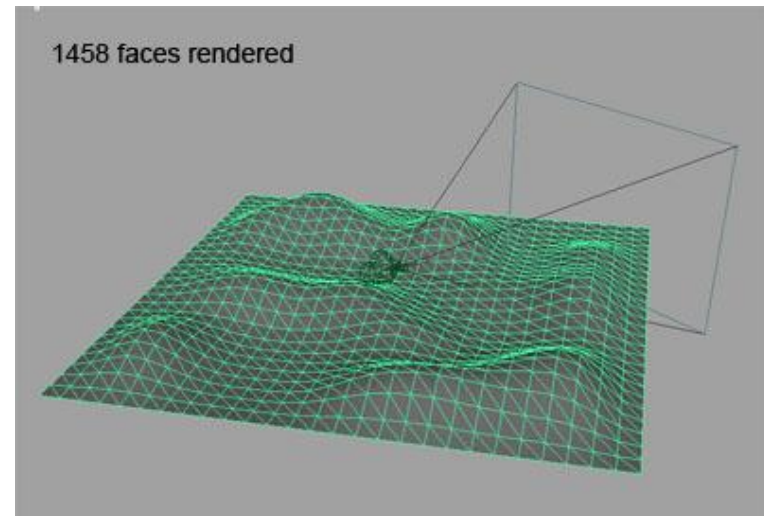
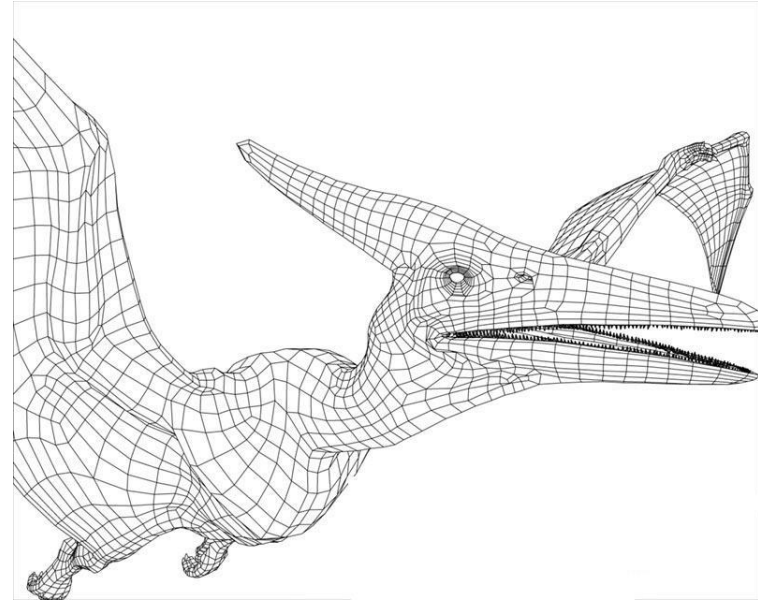


Граничное представление: типичные алгоритмы

- проверка правильности задания;
- вычисление габаритного объема;
- вычисление нормали в точке;
- вычисление кривизны поверхности;
- нахождение точки пересечения с лучом или кривой;
- определение положения точки относительно поверхности.

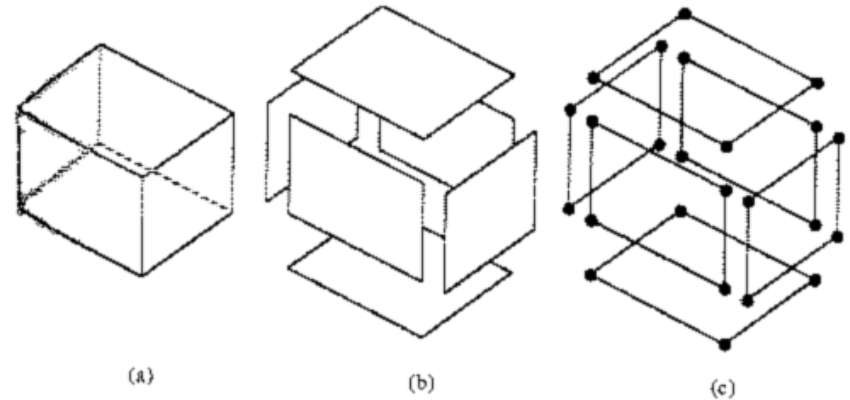
Полигональное представление

- для построения требует информации о поверхности объекта;
- рассматриваются представления первого порядка (линейная интерполяция);
- структура данных:
 - вершины: геометрия (координаты);
 - грани: топология (связность);
 - нормали: ориентация;
- свойства полигональных сеток:
 - монолитность (ограничивает некоторое конечное пространство);
 - связность (существует непрерывный путь вдоль ребер полигона);
 - простота (монолитность и отсутствие отверстий);
 - плоскостность (каждая грань задается плоским многоугольником);
 - выпуклость.



Полигональное представление: явное представление

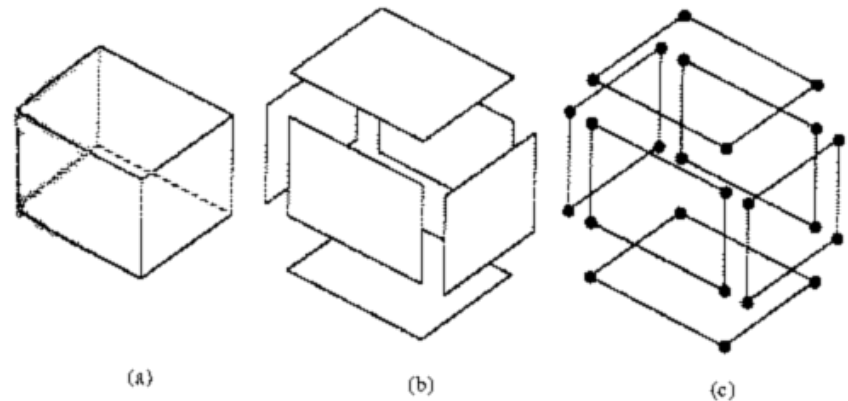
- каждая грань – многоугольник, состоящий из последовательности координат вершин;
- объект состоит из набора граней;
- недостатки:
 - взаимоотношения граней заданы неявно;
 - координаты каждой вершины появляются столько раз, сколько граней имеют эту вершину;
 - алгоритмы поиска инцидентных ребер требуют полного перебора.



Полигональное представление: индексированное по вершинам

- выделение координат вершин в отдельную структуру;
- с гранями ассоциируются не координаты вершин, а индексы в массиве координат вершин;
- недостатки: аналогично явному представлению.

Вершины	Координаты	Грани	Вершины
v1	x1 y1 z1	f1	v1 v2 v3 v4
v2	x2 y2 z2	f2	v6 v2 v1 v5
v3	x3 y3 z3	f3	v7 v3 v2 v6
v4	x4 y4 z4	f4	v8 v4 v3 v7
v5	x5 y5 z5	f5	v5 v1 v4 v8
v6	x6 y6 z6	f6	v8 v7 v6 v5
v7	x7 y7 z7		
v8	x8 y8 z8		



Полигональное представление: индексированное по ребрам

- грани определяются через ребра (последовательность индексов в массиве ребер);
- ребра задаются вершинами (последовательность индексов в массиве вершин);
- вершины задаются положением в пространстве.

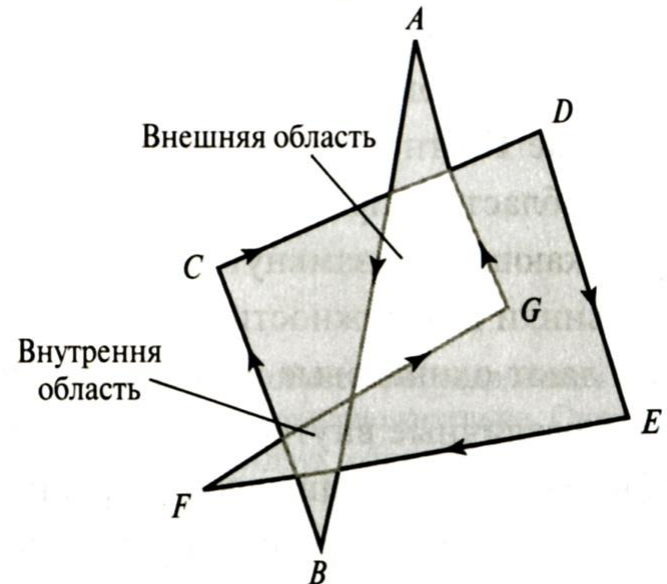
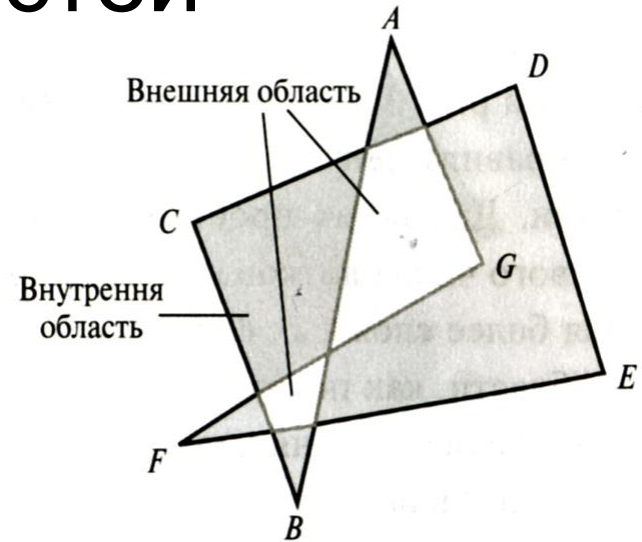
Реб	Верш	Верш	Коорд	Грн	Ребра
e1	v1 v2	v1	x1 y1 z1	f1	e1 e2 e3 e4
e2	v2 v3	v2	x2 y2 z2	f2	e9 e6 e1 e5
e3	v3 v4	v3	x3 y3 z3	f3	e10 e7 e2 e6
e4	v4 v1	v4	x4 y4 z4	f4	e11 e8 e7 e3
e5	v1 v5	v5	x5 y5 z5	f5	e12 e5 e4 e8
e6	v2 v6	v6	x6 y6 z6	f6	e12 e11 e10 e9
e7	v3 v7	v7	x7 y7 z7		
e8	v4 v8	v8	x8 y8 z8		
e9	v5 v6				
e10	v6 v7				
e11	v7 v8				
e12	v8 v5				

Многоугольники

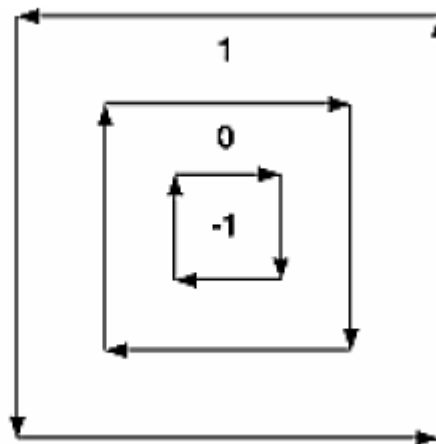
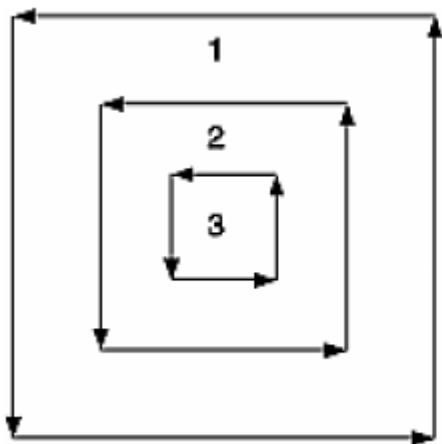
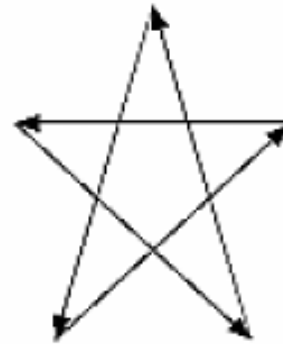
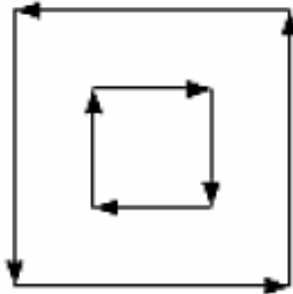
- многоугольник — это плоская геометрическая фигура:
 - задается с помощью набора точек (*вершин*);
 - вершины последовательно соединены прямолинейными отрезками, которые называются *сторонами* (*ребрами*) многоугольника (таким образом, стороны многоугольника образуют замкнутую ломаную);
- вершины многоугольника называются *соседними*, если они являются концами одной из его сторон;
- *углом* (или внутренним углом) многоугольника при данной вершине называется угол, образованный его сторонами, сходящимися в этой вершине (при этом угол считается со стороны внутренней области многоугольника);
- многоугольник называется *правильным*, если у него все стороны равны и все углы равны (в пределе - окружность).

Задание многоугольников и их внутренних областей

- в общем случае многоугольник задается набором *контуров* (contour, замкнутая ломаная);
- по соглашению, для внешних контуров направление обхода задается против часовой стрелки (CCW, counter-clockwise), для внутренних контуров (отверстий) – по часовой стрелке (CW, clockwise);
- внутренней нормалью к ребру будет нормаль, направленная во внутреннюю область;
- задание внутренней области (interior):
 - правило «четного-нечетного» (подсчет числа пересечений с контурами отрезка, соединяющего тестируемую точку с точкой, лежащей вне области многоугольника);
 - правило ненулевого количества витков (подсчет числа витков контуров вокруг тестируемой точки, положительным считается виток в направлении против часовой стрелки).



Задание контуров и подсчет количества витков

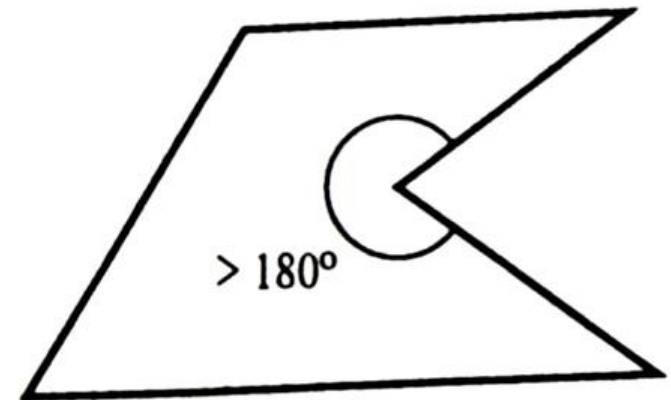
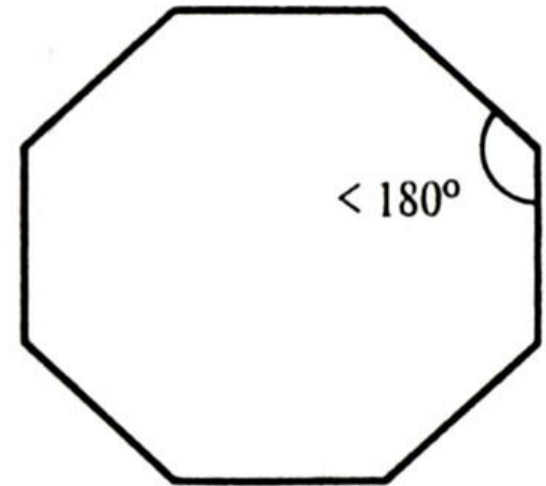


Классификация многоугольников

- пересекающиеся;
- взаимопересекающиеся (self - intersecting);
- вырожденные (degenerate):
 - коллинеарные вершины;
 - совпадающие вершины;
 - количество вершин менее трех;
- простые (непересекающиеся, единственный контур - simple);
- выпуклые (convex);
- вогнутые (concave).

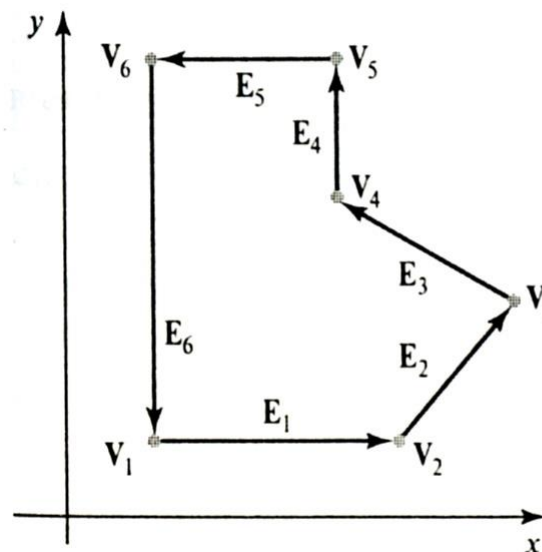
Критерии выпуклости многоугольников

- многоугольник называют *выпуклым*, если выполнено одно из следующих (эквивалентных) условий:
 - все внутренние углы многоугольника меньше 180° градусов;
 - многоугольник лежит по одну сторону от любой прямой, содержащей его ребро (т. е. продолжения сторон многоугольника не пересекают других его сторон);
 - любой отрезок с концами в точках, принадлежащих многоугольнику, целиком ему принадлежит.



Определение (тест) выпуклости многоугольников

1. все векторные произведения смежных сторон многоугольника:
 - одного знака - многоугольник выпуклый;
 - иначе – вогнутый;
2. расположение вершин многоугольника относительно каждой прямой, содержащей его сторону (анализ знаков скалярного произведения нормали к прямой и векторов, соединяющих начальную вершину ребра и все остальные вершины многоугольника):
 - по одну сторону (скалярные произведения одного знака) – многоугольник выпуклый;
 - иначе – вогнутый
3. метод вращения.



$$(E_1 \times E_2)_z > 0$$

$$(E_2 \times E_3)_z > 0$$

$$(E_3 \times E_4)_z < 0$$

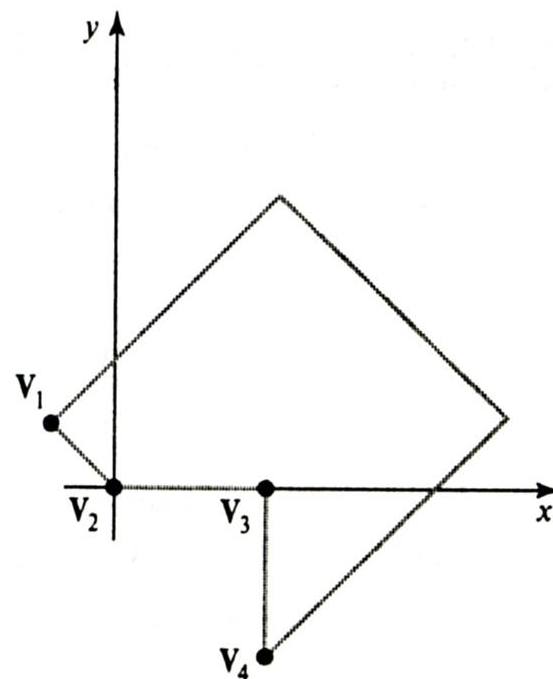
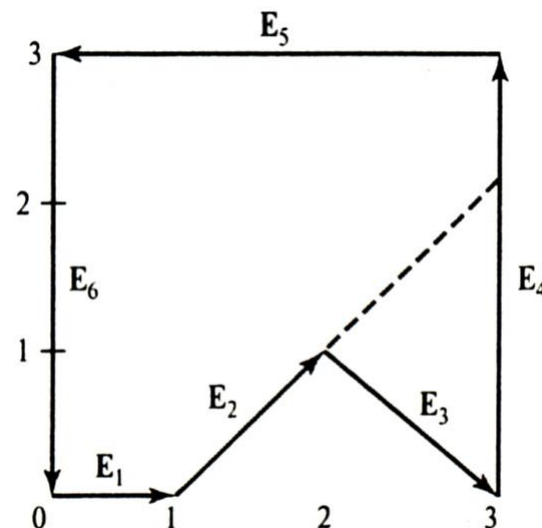
$$(E_4 \times E_5)_z > 0$$

$$(E_5 \times E_6)_z > 0$$

$$(E_6 \times E_1)_z > 0$$

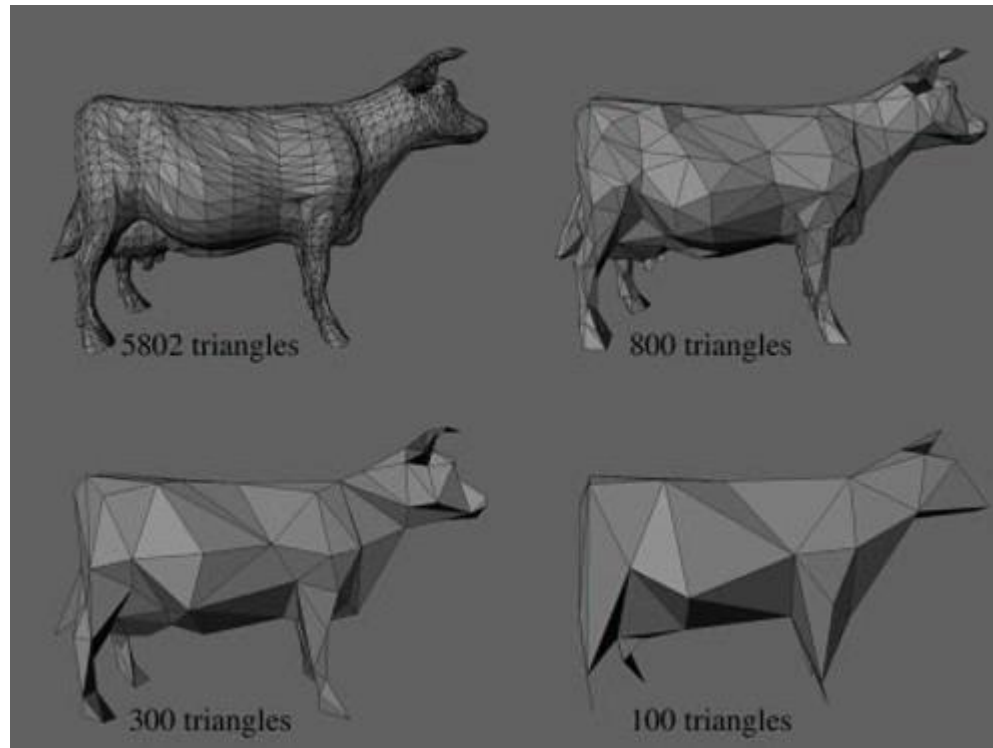
Разбиение невыпуклых многоугольников

- векторный метод: разбиение многоугольника относительно прямой, содержащей первую из пары соседних сторон, для которых нарушается знак векторного произведения (предполагается, что многоугольник лежит в плоскости XOY , а знак векторного произведения определяется компонентой z);
- метод вращения:
 - перемещаясь в направлении против часовой стрелки, вершины V_k последовательно совмещаются с началом координат, а ребра $V_k V_{k+1}$ – с осью OX ;
 - если вершина V_{k+2} оказывается ниже оси OX , то производится разбиение многоугольника по оси OX .



Триангуляция

- триангуляция – выделение троек вершин (~ tessellation / tiling) / аппроксимация поверхности треугольными гранями:
 - грани, образованные в результате триангуляции, заведомо задают плоскости;
 - упрощает выполнение многих функций графических пакетов;
 - для выпуклых многоугольников сводится к последовательному выделению троек вершин.

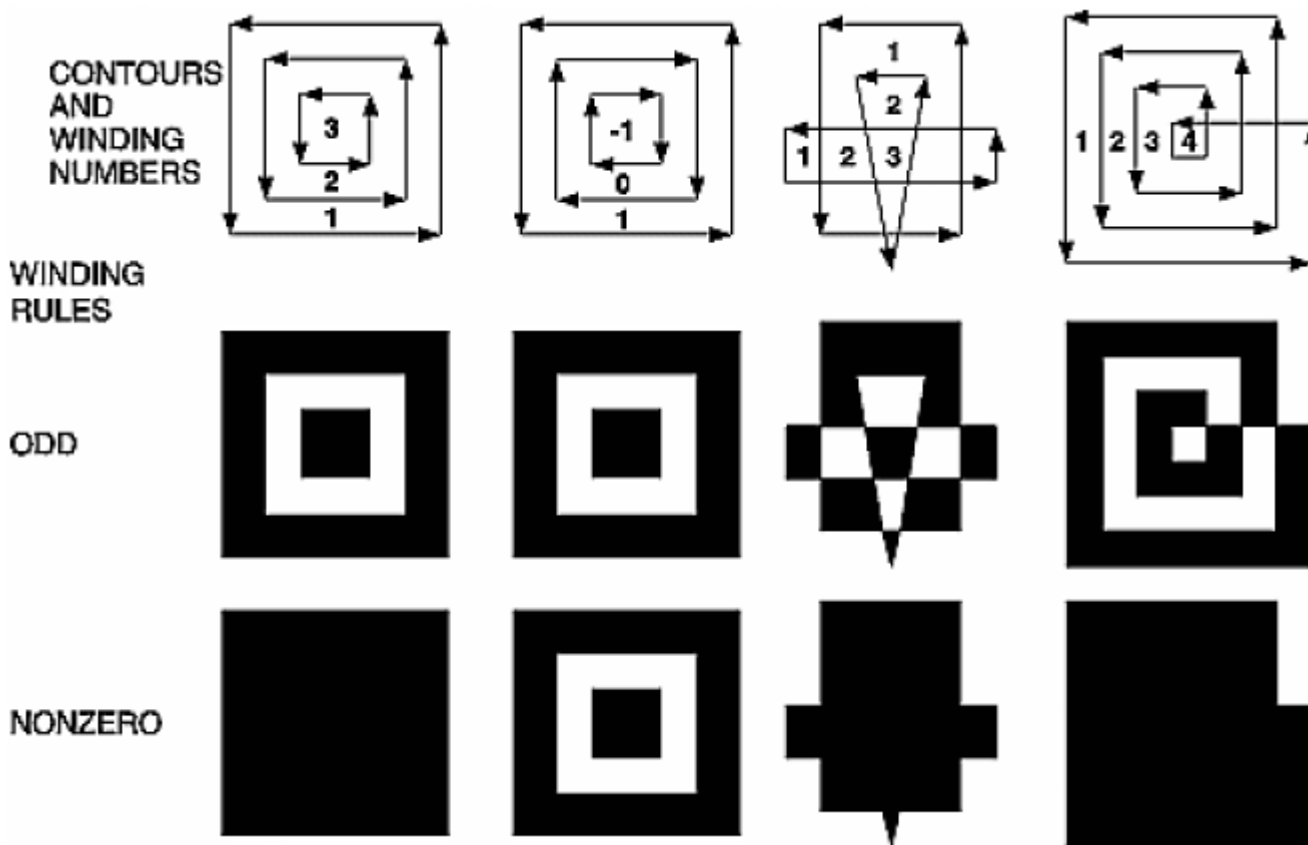


Разбиение многоугольников в OpenGL

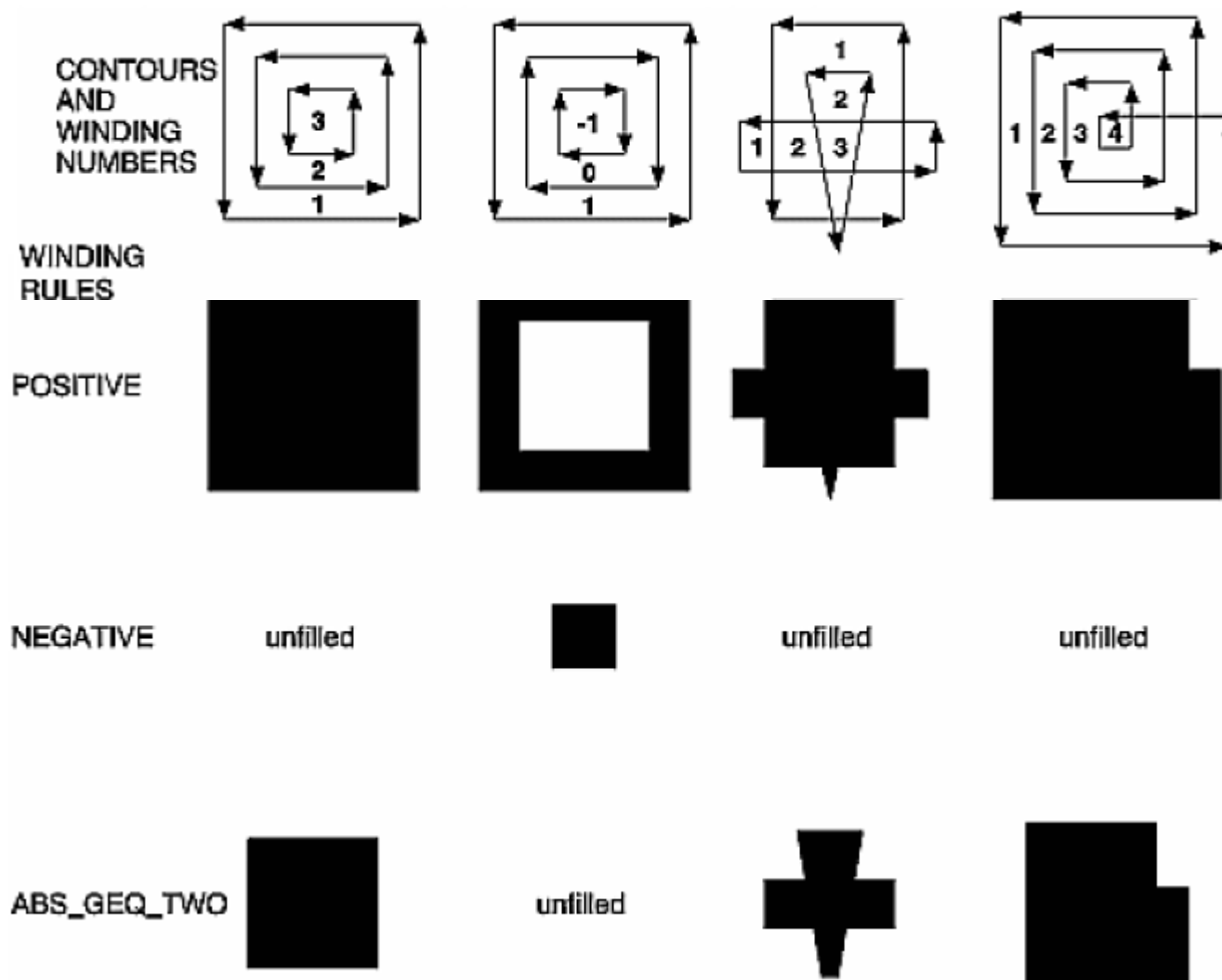
- OpenGL поддерживает задание и отображение простых выпуклых многоугольников (задаваемых единственным несамопересекающимся контуром), для остальных случаев необходимо применять процедуру разбиения (tessellation)
- процедура разбиения
 1. создание объекта разбиения: **gluNewTess()**;
 2. определение функций обратного вызова для процедуры разбиения: **gluTessCallback()**
 - определение наборов возвращаемых данных (GL_TRIANGLE_FAN, GL_TRIANGLE_STRIP, GL_TRIANGLES, GL_LINE_LOOP);
 - результат разбиения в виде последовательности вершин;
 - вычисление атрибутов дополнительных вершин в случае самопересечений;
 - информация об ошибках;
 3. определение режима разбиения: **gluTessProperty()**
 - *GLU_TESS_BOUNDARY_ONLY* – получение границ результирующих областей;
 - *GLU_TESS_TOLERANCE* – точность операций с вершинами (анализа на совпадение);
 - *GLU_TESS_WINDING_RULE* – определение правила задания внутренних и внешних областей (*GLU_TESS_WINDING_ODD* – по умолчанию, *GLU_TESS_WINDING_NONZERO*, *GLU_TESS_WINDING_POSITIVE*, *GLU_TESS_WINDING_NEGATIVE*, *GLU_TESS_WINDING_ABS_GEQ_TWO*);
 4. задание набора многоугольников через набор контуров
 - **gluTessBeginPolygon () / gluTessEndPolygon();**
 - **gluTessBeginContour () / gluTessEndContour();**
 - **gluTessVertex ();**
 5. удаление объекта разбиения: **gluDeleteTess()**.

Задание внутренних областей в OpenGL

- подсчитывается и суммируется число витков (winding numbers) контуров вокруг точки, лежащей в рассматриваемой области;
- задается правило, по которому в соответствии с вычисленным количеством витков области классифицируются на внутренние (interior) и внешние (exterior).

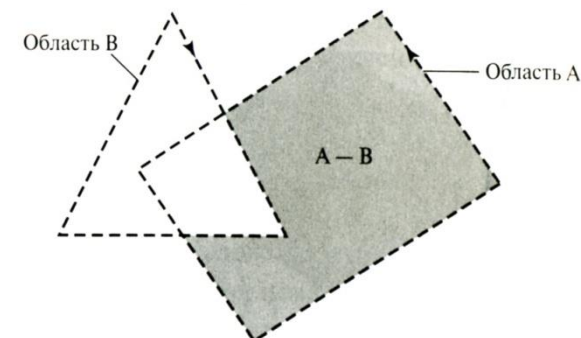
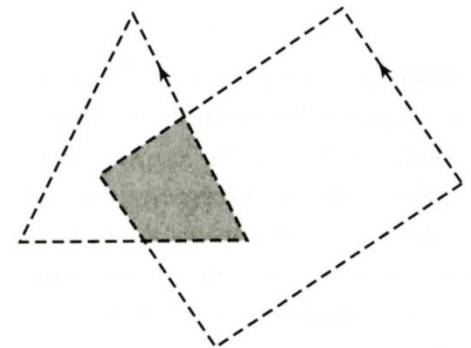
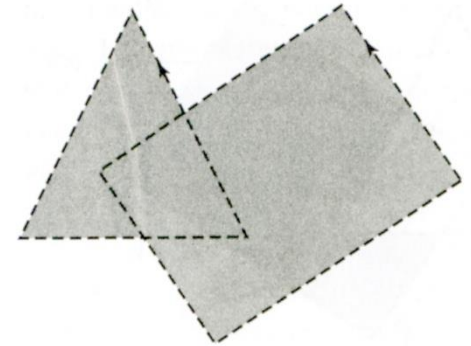


Задание внутренних областей в OpenGL (продолжение)



Реализация булевых операций через задание правил подсчета витков

- рассматриваются простые, непересекающиеся контуры. Контуры с направлением обхода против часовой стрелки задают внешние границы многоугольника, с направлением обхода по часовой стрелке – внутренние границы (отверстия). Таким образом, для точек внутренней области количество витков будет равно 1, а для точек внешней – 0. Для того, чтобы выполнить данное требование, можно предварительно применить операцию выделения границ (GLU_TESS_BOUNDARY_ONLY) для произвольно заданных контуров;
- реализация булевых операций:
 - ОБЪЕДИНЕНИЕ (UNION):
 - все рассматриваемые контуры вводятся как единственный многоугольник;
 - в качестве правила подсчета задается GLU_TESS_WINDING_NONZERO (нечувствительно к направлению обхода) или GLU_TESS_WINDING_POSITIVE;
 - ПЕРЕСЕЧЕНИЕ (INTERSECTION):
 - контуры рассматриваются попарно;
 - в качестве правила подсчета задается GLU_TESS_WINDING_ABS_GEQ_TWO;
 - РАЗНОСТЬ (DIFFERENCE):
 - вершины вычитаемых контуров вводятся в обратном порядке;
 - в качестве правила подсчета задается GLU_TESS_WINDING_POSITIVE.



Многогранники

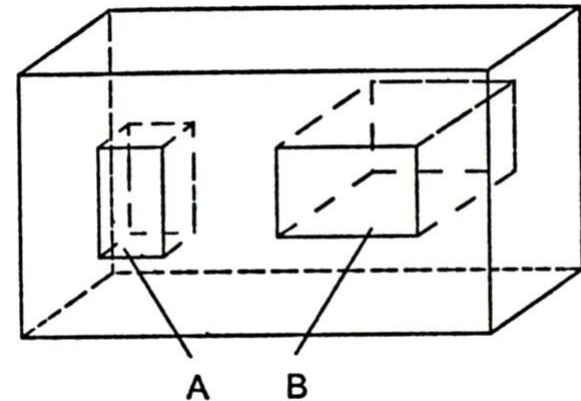
- *многогранник (полиэдр)* — связная сетка из простых плоских многоугольников, ограничивающих конечный объем пространства;
- многоугольники, формирующие многогранник, называются *гранями*, их стороны — *рёбрами*, а их вершины — *вершинами* многогранника;
- грани многогранника называются *смежными*, если они имеют общее ребро (для любой грани существует единственная грань, инцидентная по данному ребру);
- вершины многогранника называются *соседними (смежными)*, если они являются концами одного из его ребер;
- грани, содержащие некоторую вершину, называются *смежными* с данной вершиной.

Проверка правильности задания многогранника (формула Эйлера)

- свойства многогранников:
 - ребро принадлежит ровно двум граням;
 - в вершине встречается не менее трех ребер;
 - грани не являются взаимопроникающими;
- формула Эйлера: соотношение, связывающее количество вершин, ребер и граней:

$$V - E + F - H = 2 * (C - G)$$

- V – количество вершин;
- E – количество ребер;
- F – количество граней;
- H – количество отверстий в гранях;
- C – количество компонент;
- G – количество сквозных отверстий;

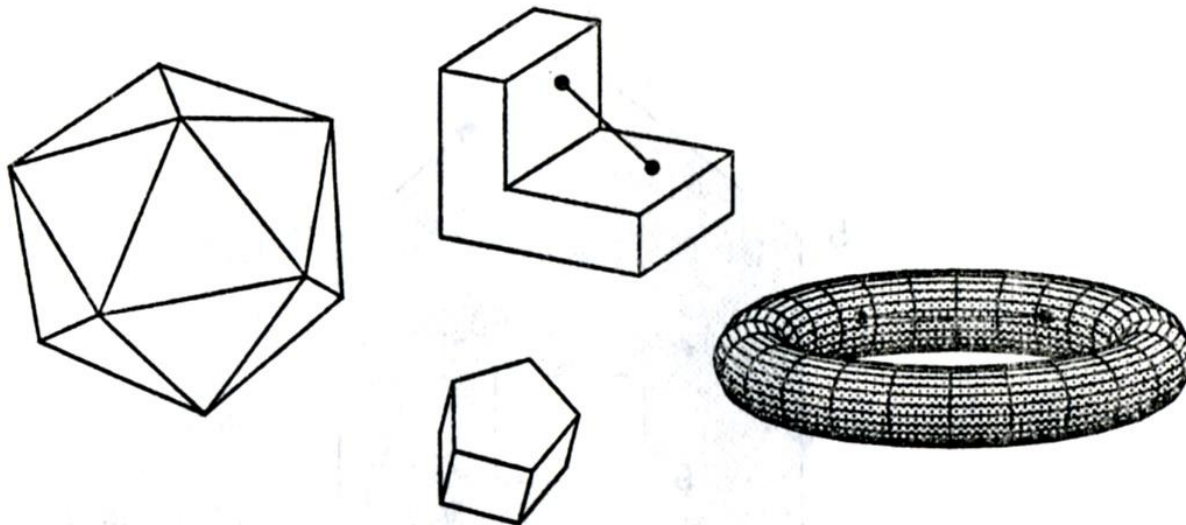


*A - углубление
B – отверстие*

$$24 - 36 + 15 - 3 = 2 * (1 - 1)$$

Критерии выпуклости многогранников

- многогранник называют *выпуклым*, если выполнено одно из следующих (эквивалентных) условий:
 - многогранник лежит по одну сторону относительно любой плоскости, содержащей его грань;
 - любой отрезок с концами в точках, принадлежащих многограннику, целиком ему принадлежит.

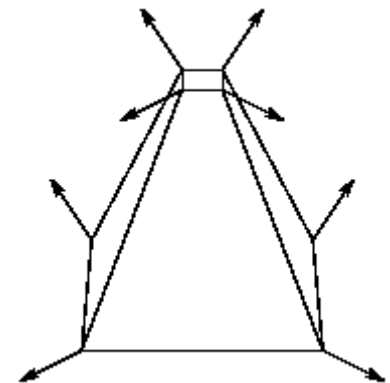
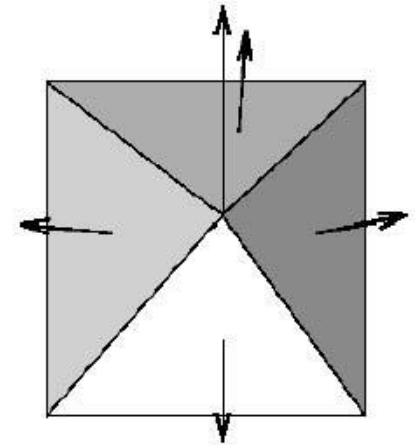


Определение (тест) выпуклости и разбиение многогранников

- тест выпуклости:
 1. расположение вершин многогранника относительно любой плоскости, содержащей его грань (анализ знаков скалярного произведения нормали к грани и векторов, соединяющих одну из вершин ребра грани и все остальные вершины многогранника):
 - по одну сторону (скалярные произведения одного знака) – многогранник выпуклый;
 - иначе – вогнутый;
 2. метод вращения: последовательно каждая грань многогранника совмещается с плоскостью XOY , и анализируются знаки координаты z для всех вершин, не принадлежащих рассматриваемой грани (если знаки разные, то нарушается условие выпуклости для данной грани);
- разбиение многогранников: метод вращения – отсечение гранью, для которой нарушается условие выпуклости.

Формирование полигональной сетки

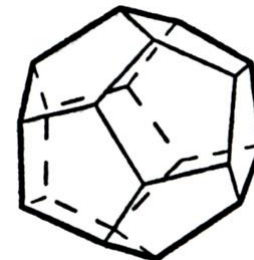
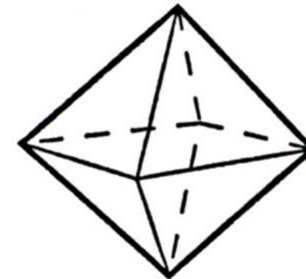
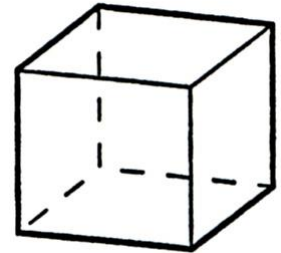
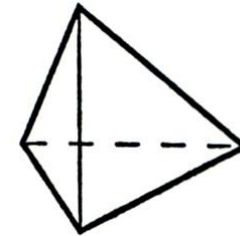
- структура данных (как правило, используется представление, индексированное по ребрам):
 - вершины: геометрия (координаты);
 - грани: топология (связность);
 - нормали: ориентация;
- формирование полигональной сетки состоит из следующих этапов:
 - формирование вершин (принадлежат поверхности);
 - формирование ребер и граней;
 - определение нормалей (в вершинах);
- нормали:
 - нормаль к граням (внешняя нормаль - векторное произведение любых двух соседних ребер при обходе грани против часовой стрелки);
 - нормаль в вершине (вычисляется усреднением нормалей смежных граней).



Правильные многогранники: платоновы тела

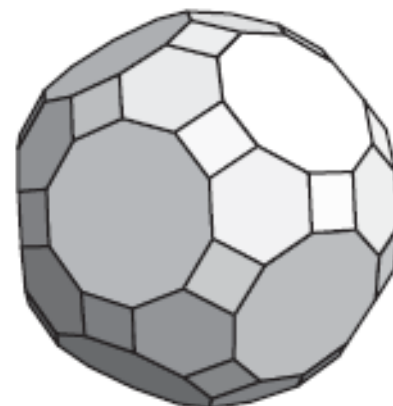
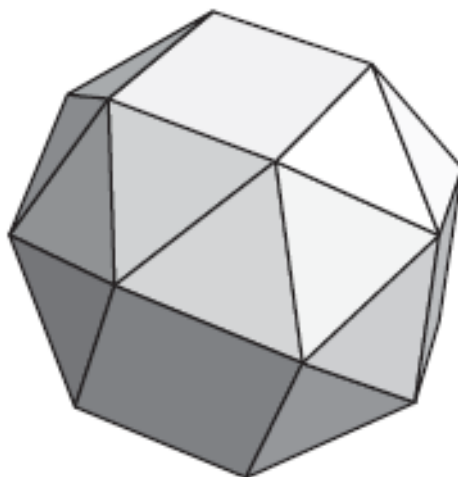
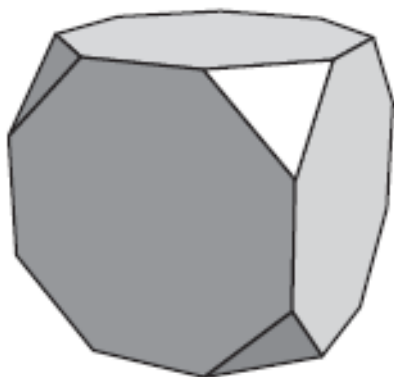
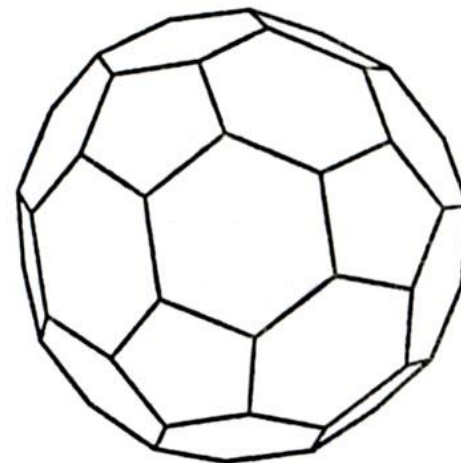
- все грани – идентичные правильные многоугольники:
 - все стороны многогранника равны;
 - все плоские углы при вершинах равны;
 - все углы между гранями равны;
- 5 правильных многогранников:
 - тетраэдр (треугольная пирамида, 4 грани);
 - гексаэдр (куб, 6 граней);
 - октаэдр (8 граней);
 - додекаэдр (12 граней);
 - икосаэдр (20 граней)
- нормаль к грани вычисляется как вектор из центра многогранника (среднее всех вершин) к центру грани (среднее вершин грани).

Тело	V	F	E
Тетраэдр	4	4	6
Гексаэдр	8	6	12
Октаэдр	6	8	12
Икосаэдр	12	20	30
Додекаэдр	20	12	30



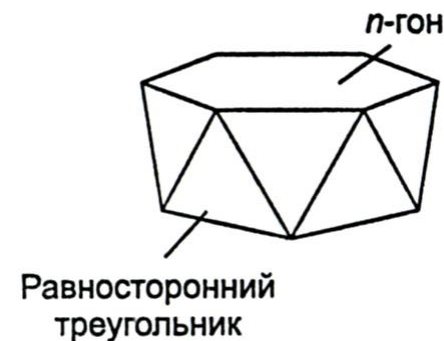
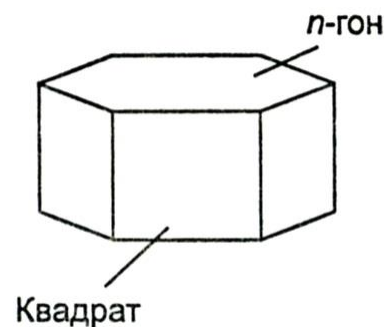
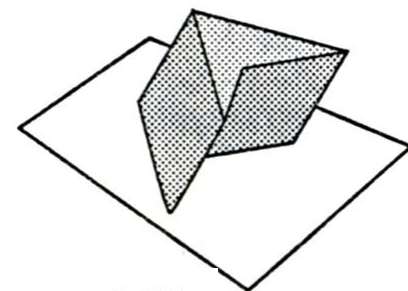
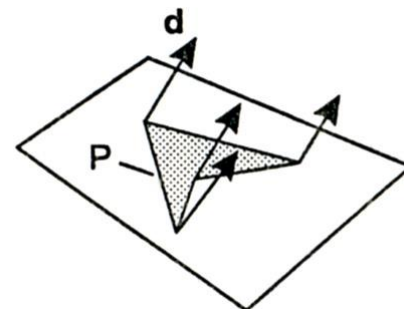
Полуправильные многогранники: архимедовы тела

- в качестве граней имеют несколько различных типов правильных многоугольников;
- каждая вершина окружена одной и той же совокупностью многоугольников в одном и том же порядке;
- частные случаи:
 - усеченный куб;
 - усеченный икосаэдр.



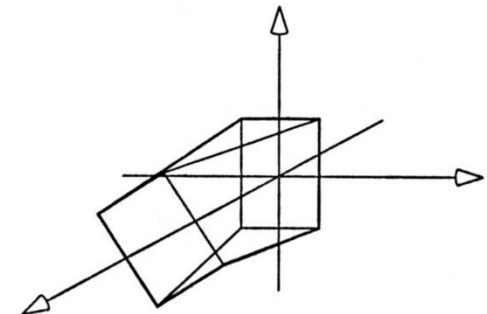
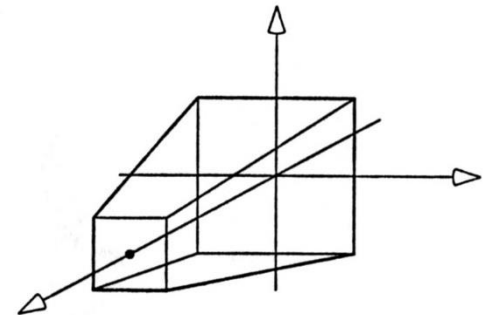
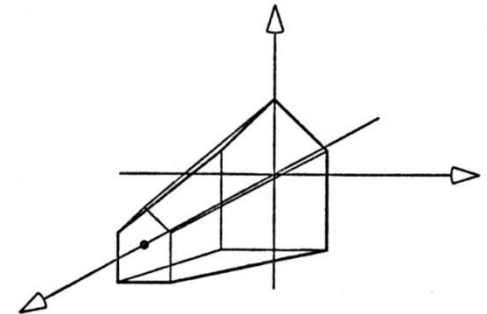
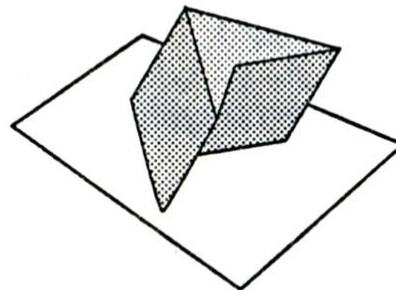
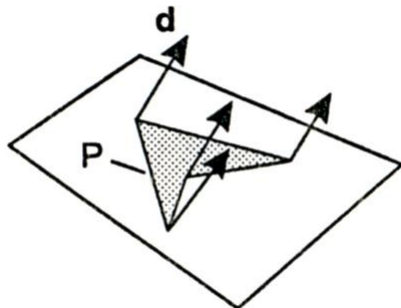
Призмы и антипризмы

- *призма* получается в результате экструзии (extruding, выдавливания) многоугольника вдоль направляющего вектора \mathbf{d} :
 - [нижнее] основание (base);
 - боковые грани;
 - крышка (cap, верхнее основание);
- *прямая призма (right prism)*: направляющий вектор перпендикулярен плоскости основания (иначе – *наклонная призма*);
- *правильная призма (regular)* имеет в основании правильный многоугольник, а в качестве боковых граней – квадраты (прямоугольники);
- *антипризма (antiprism)*: верхний n -угольник повернут на $180/n$ градусов, каждая вершина верхнего основания соединена с парой ближайших вершин нижнего основания (таким образом, что формируются боковые грани в виде равносторонних треугольников).



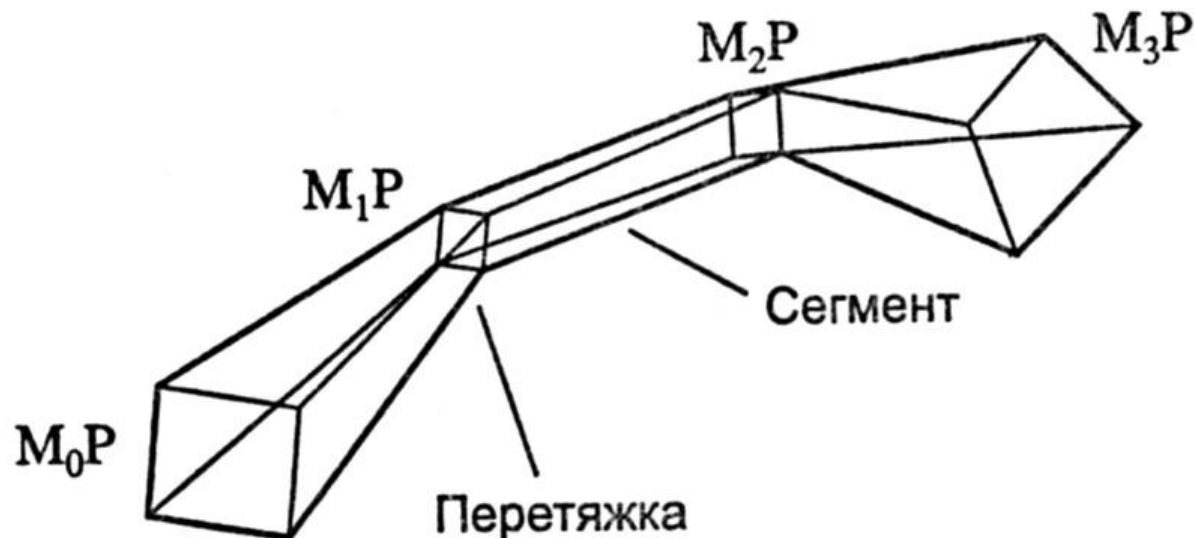
Экструзивные формы

- экструзия – *выдавливание* (extruding), или *заметание* (sweeping) двумерной формы в пространстве (заметаящие представления могут использоваться при создании объектов с трансляционной, вращательной и другими типами симметрии):
 - линейная (вдоль некоторого вектора):
 - призма (прямая, наклонная);
 - нелинейная (с использованием аффинных преобразований основания):
 - пирамида;
 - повернутые призмы;
 - поверхности вращения (surface of revolution).



Сегментированная экструзия

- применение последовательности экструзий, каждая со своим собственным преобразованием;
- *сегменты* экструзии формируют *трубку*;
- преобразованные основания (двумерные формы, на основании которых строится каждый из сегментов экструзии) называются *перетяжками*.



Конструирование трубок на базе трехмерных кривых

- сегментированная экструзия задается трехмерной кривой (хребтом, spine) $C(t)$;
- ориентация основания каждого из сегментов определяется путем вычисления *базиса Френе*:
 - единичный касательный вектор к кривой $T(ti) = C'(ti) / |C'(ti)|$;
 - *единичный вектор бинормали* $B(ti) = (C'(ti) \times C''(ti)) / |C'(ti) \times C''(ti)|$;
 - $N(t) = B(t) \times T(t)$;
- матрица преобразований основания задается через полученные векторы:
$$M_i = (N(ti) \mid B(ti) \mid T(ti) \mid C(ti))$$
- возможна численная аппроксимация производных.

