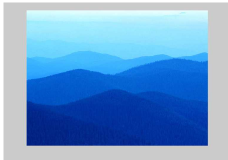
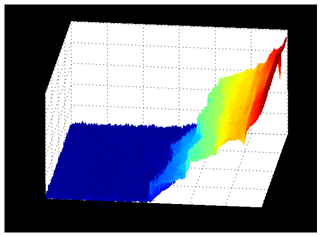
DEFINING OBJECTS - 3D REPRESENTATIONS

3D surface representation - continued
- Height maps
- Parametric surfaces

Heightmaps from bitmaps



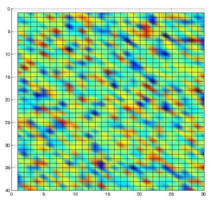


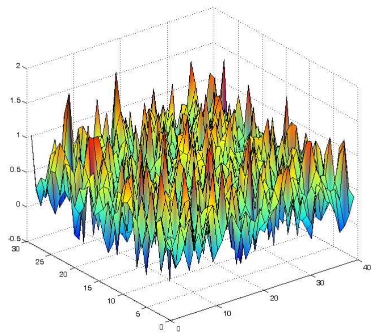
Heightmaps from bitmaps

- Bitmap – image data
  - Rectangular grid X × Y of values
  - Value in the bitmap corresponds to height (Z) in the heightmap
  - Image rendered using normally the Painter algorithm (see later lectures)

230	230	230	229	225
212	223	222	219	226
215	218	220	222	220
203	201	210	225	215
124	146	169	187	196
72	83	100	108	114
58	68	69	63	70
43	50	56	63	63
40	39	42	50	44
49	45	49	47	46
40	46	40	37	41

Heightmaps using random number generation





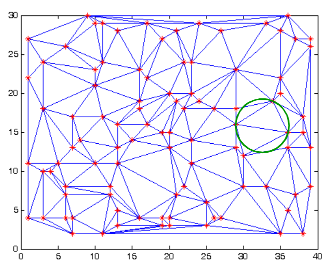
Heightmaps using random number generation

- Bitmap – contents created using random number generation
  - Rectangular grid X × Y of values
  - Value in the bitmap corresponds to height (Z) in the heightmap
  - Image rendered using normally the Painter algorithm (see later lectures)

```
grid=zeros(Nx,Ny);
for xi=1:Nx
   for yi=1:Ny
      grid(xi,yi)=gmin+(gmax-gmin)*rand(); % Uniform distribution
      grid(xi,yi)=0.5+sqrt(0.1)*randn();    % Normal distribution
   end
end
```

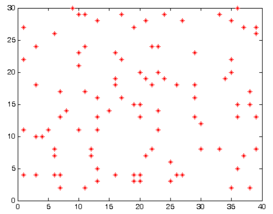
Delaunay triangulation

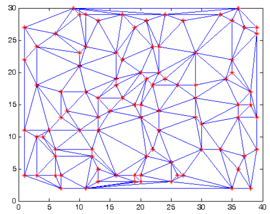
- For the data points defined by vertices x and y, Delaunay triangulation returns a set of triangles such that no data points are contained in any triangle's circumscribed circle.



For overview of Delunay triangulation see http://www.cs.berkeley.edu/~jrs/meshpapers/BernEppstein.pdf

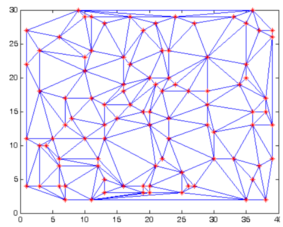
Delaunay triangulation

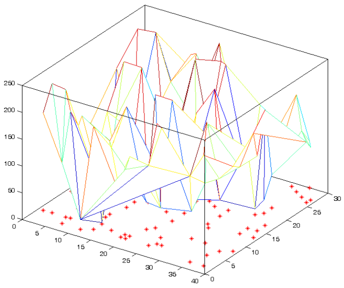




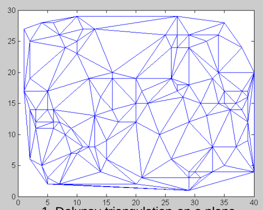
- Generate points (vertices) at random locations on a rectangular grid
- At each point generate random height
- Delunay triangulation creates triangles by connecting the nearest three vertices. The lines between the vertices must not intersect.
- This generates triangular faces.

Delaunay triangulation

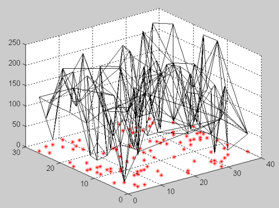




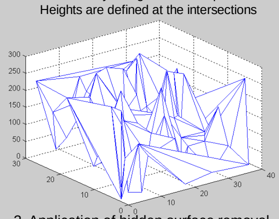
- Display the surfaces



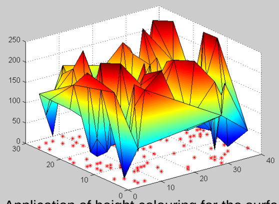
1. Delunay triangulation on a plane. Heights are defined at the intersections



2. Wireframe representation



3. Application of hidden surface removal



4. Application of height colouring for the surfaces

Curved surfaces (and lines)

- **The generation of 3D curved lines and surfaces**
  - An input set of mathematical functions
  - A set of user-specified data points (splines, discussed later)
- **Curve and surface equations**
  - Nonparametric
  - **Parametric**
- **Quadric Surfaces**
  - Sphere
  - Ellipsoid
  - Torus

Parametric equations

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

- u is the parameter
- x(u), y(u) and z(u) are functions of the parameter u which generate x, y and z coordinates of the curve or surface P.

Curve equations - circle

Canonical form $$\left(\frac{x}{r}\right)^2 + \left(\frac{y}{r}\right)^2 = 1$$

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

Parametric form

Parameter

Parameter

$$x = r\cos\theta$$
$$y = r\ \sin\theta$$
$$-\pi \le \theta \le \pi$$

Generating a circle in Matlab

```
Nphi=30;
Dphi=pi/Nphi;
R=2;
phi=(-pi : Dphi : pi);

for i=1:length( phi )
   X(i)=R*cos( phi(i) );
   Y(i)=R*sin( phi(i) );
end

plot( X, Y);
```

Examples of parametric equations: curves

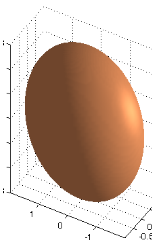
Circle
$$x = r\cos\theta \qquad y = r\ \sin\theta \qquad -\pi \le \theta \le \pi$$

Ellipse
$$x = r_x\cos\theta \qquad y = r_y\sin\theta \qquad -\pi \le \theta \le \pi$$

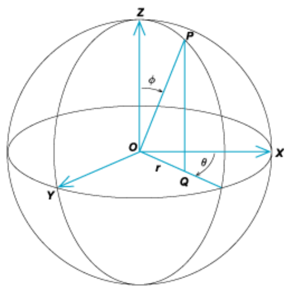
Quadric Surfaces: canonical form

Sphere
$$\left(\frac{x}{r}\right)^2 + \left(\frac{y}{r}\right)^2 + \left(\frac{z}{r}\right)^2 = 1$$

Ellipsoid
$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1$$



Spherical coordinates



$$x = r\cos\phi\cos\theta$$
$$y = r\cos\phi\sin\theta$$
$$z = r\sin\phi$$

$$-\pi/2 \le \phi \le \pi/2$$
$$-\pi \le \theta \le \pi$$

Quadric Surfaces

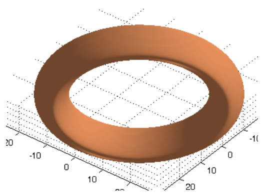
Parameters

Sphere
$$x = r\cos\phi\cos\theta \qquad y = r\cos\phi\sin\theta \qquad z = r\sin\phi$$
$$-\pi/2 \le \phi \le \pi/2 \qquad -\pi \le \theta \le \pi$$

Ellipsoid
$$x = r_x\cos\phi\cos\theta \qquad y = r_y\cos\phi\sin\theta \qquad z = r_z\sin\phi$$
$$-\pi/2 \le \phi \le \pi/2 \qquad -\pi \le \theta \le \pi$$

Quadric Surfaces

Torus
$$x = r_x(r + \cos\phi)\cos\theta \qquad y = r_y(r + \cos\phi)\sin\theta \qquad z = r_z\sin\phi$$
$$-\pi \le \phi \le \pi \qquad -\pi \le \theta \le \pi$$



Superquadrics

- **A generalization of the quadric representations**
  - Incorporate additional parameters into the quadric equations
  - The number of additional parameters used is equal to the dimension of the object.

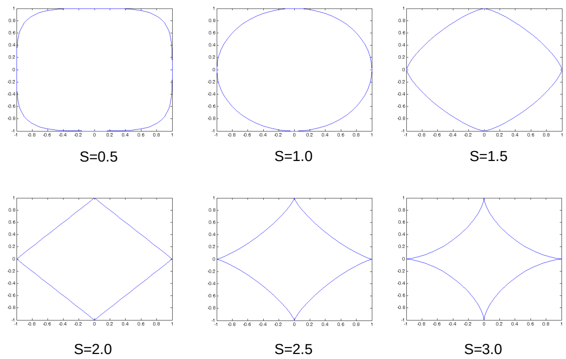
Ellipse

$$x = r_x \cos\theta \qquad -\pi \le \theta \le \pi$$
$$y = r_y \sin\theta$$

Superellipse

$$x = r_x \cos^s\theta \qquad -\pi \le \theta \le \pi$$
$$y = r_y \sin^s\theta$$

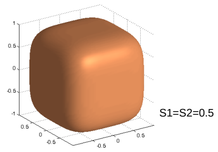
Superellipses

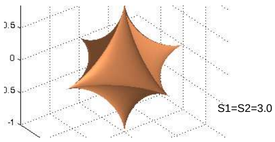


Superquadrics

Superellipsoid

$$x = r_x \cos^{s1}\phi \cos^{s2}\theta \qquad -\pi/2 \le \phi \le \pi/2$$
$$y = r_y \cos^{s1}\phi \sin^{s2}\theta \qquad -\pi \le \theta \le \pi$$
$$z = r_z \sin^{s1}\phi$$





Homework

- Explore the range of superellipses by further modifying parameter *s*
- Explore the range of superellipsoids by modifying parameters *s1* and *s2*
- Relevant Matlab code is at ex3_surffaces.zip

Next lecture

Surface rendering - overview