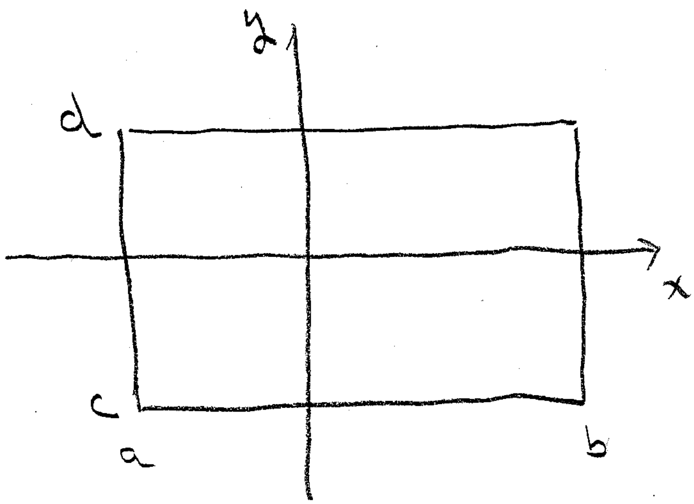
Viewing 2D functions $z = f(x,y)$ on irregular domains

We have examined plotting a surface function of $z = f(x,y)$ before. In this case, the domains were rectangles of the form

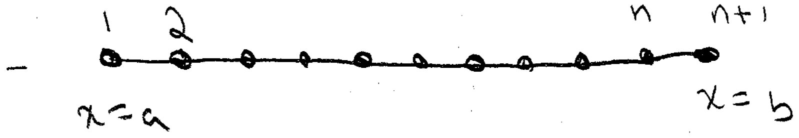
$x \in [a,b]$, $y \in [c,d]$



To do this, we first <u>discretized</u> the continuous domain.

<u>discretize</u> - create a discrete approximation to a continuous region.

- we have done this before when we take the interval $x \in [a,b]$ and divide into $n+1$ equally spaced points



the set of points $x_1, x_2, \cdots, x_n, x_{n+1}$
is called the discretization of $x \in [a,b]$

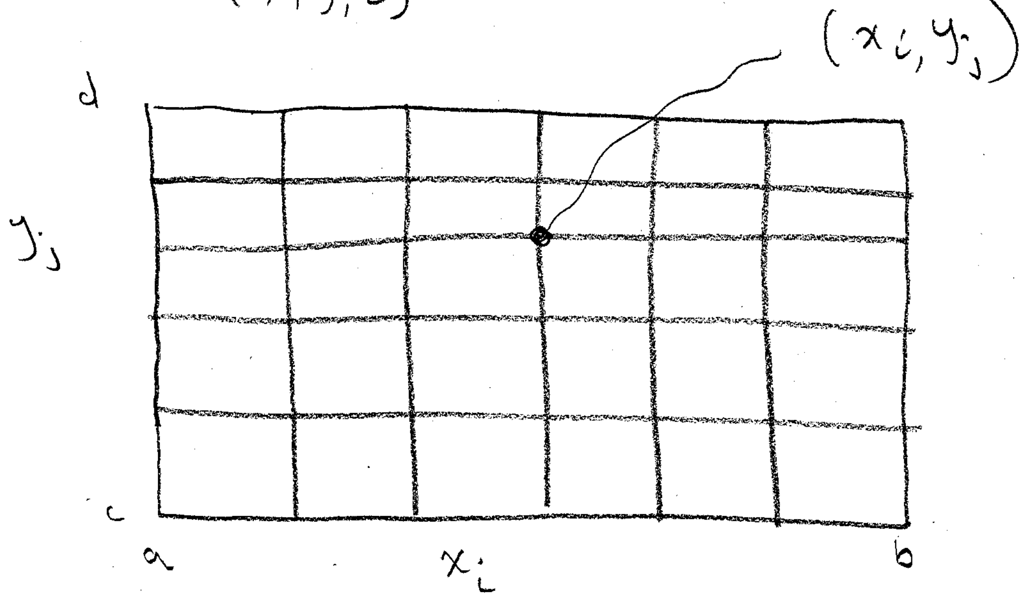
Previously we used meshgrid to discretize the 2D domain

x = linspace(a,b, n+1)'; $\quad h = \frac{b-a}{n}$
y = linspace (c,d, m+1)'; $\quad k = \frac{d-c}{m}$

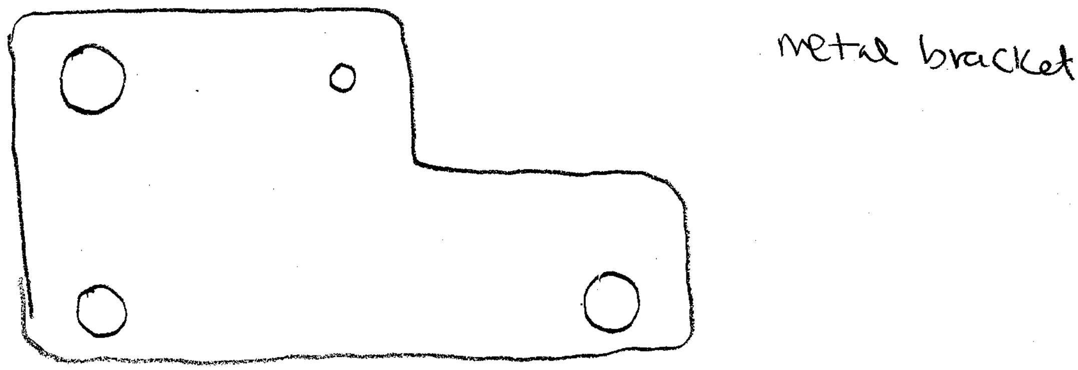
[X,Y] = meshgrid (x,y)

z = f (X, Y)

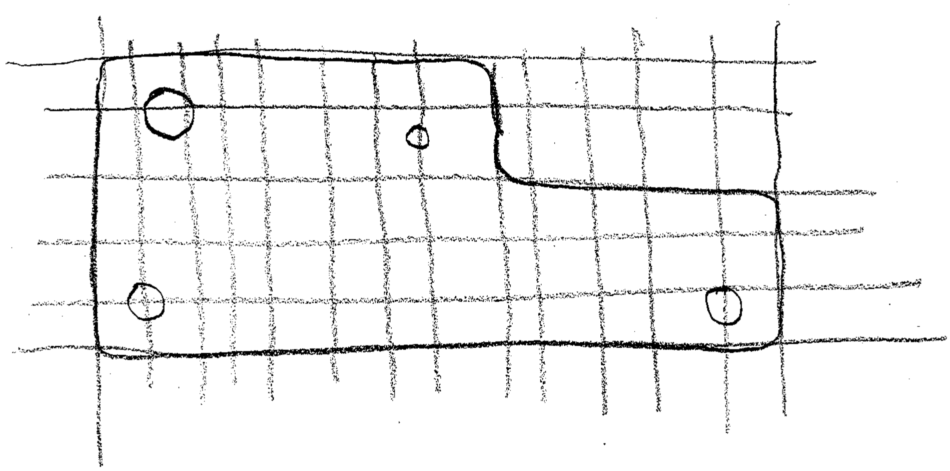
surf (x,y,z)



How do view a function on a more irregular type of domain?

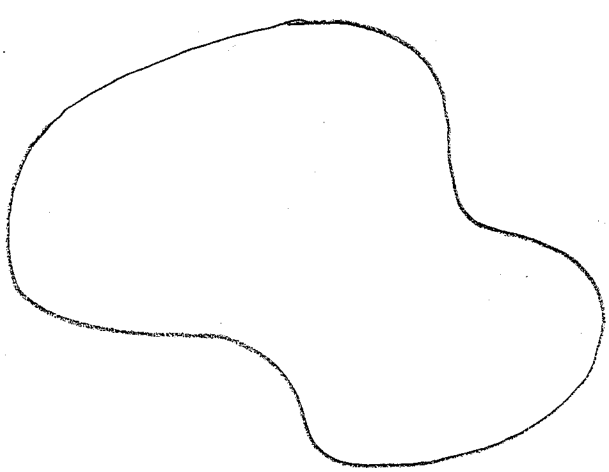


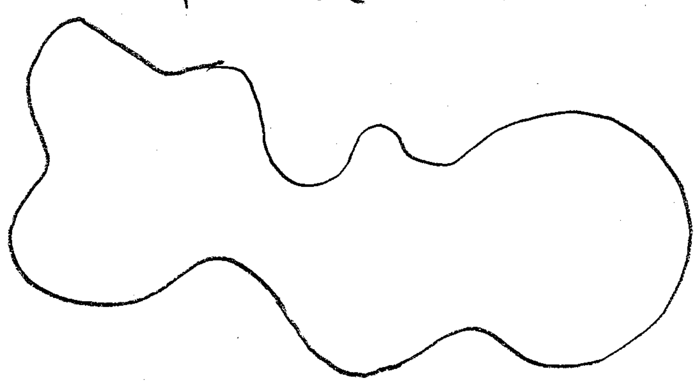
The process from before doesn't work here



The circles and the upper right area cause a problem.

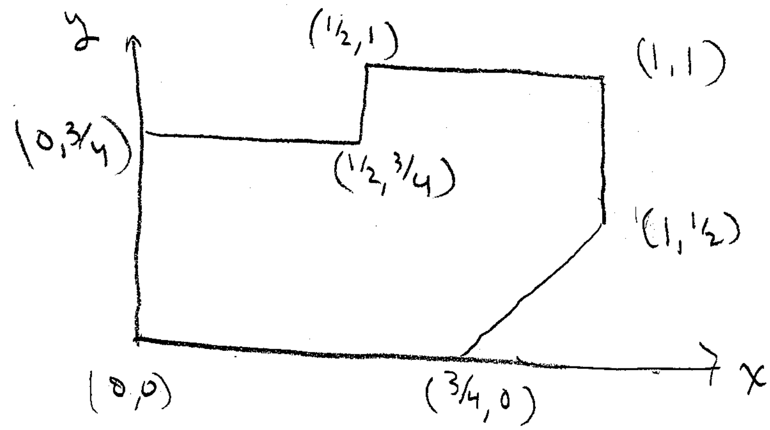
Even more problematic shapes are possible



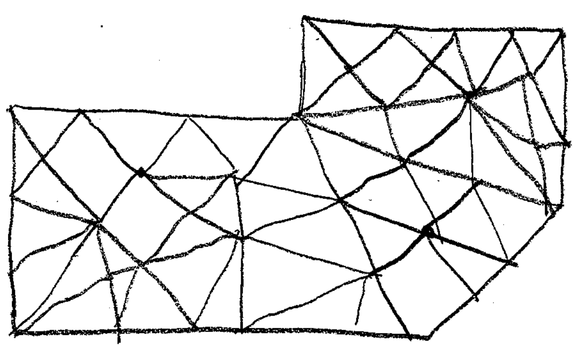


We need a way to discretize these shapes. This will require a different way to store the domain information

For now we will assume the domain consists of line segments



We will perform a triangularization of the domain (you can also use quadrilaterals.)



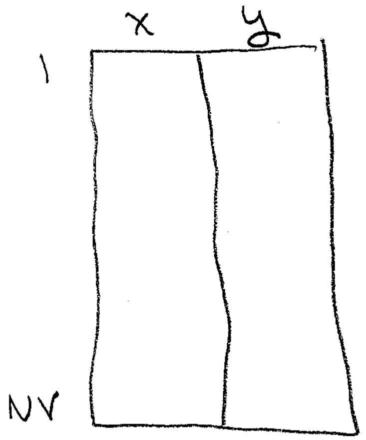
Divide the region into triangles. This creates a set of vertices (x,y). $z = f(x,y)$ can be evaluated at each of those vertices. We can then use 3D techniques to visualize the function z.

To represent the triangularization, we will use 2 special matrices. Historically these are called the vertex table and the triangle table.

Vertex Table

Size = NV x 2   Where NV = # of Vertices

This is just a listing of the coordinates of the vertices in the triangulation

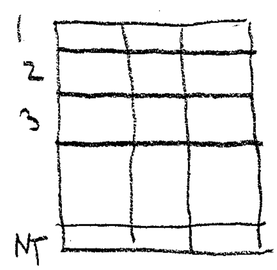
V = 

column 1 is x coordinates
column 2 is y coordinates

Triangle Table

Size = NT x 3   NT = number of triangles

This is a listing of the verticies that make up each triangle (usually ordered counter clockwise

T = 

← Verticies for triangle 1
← ——— 2
← ——— 3

How to use this to visualize $z = f(x,y)$?

we need to create the connectivity matrix. This is easy to do in matlab.

The connectivity matrix is a matrix of size $NV \times NV$

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} & & j & & \\ 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ i \;\; 3 \\ 4 \\ 5 \end{array} &
\left[\begin{array}{ccccc} & & & 1 & 1 \\ & & & & \\ & & & & \\ 1 & & & & 1 \\ 1 & & & 1 & \end{array}\right]
\end{array}
$$

$(i,j) = 1 \Rightarrow$ i and j are connected

<u>Ex</u> Triangle 1 in example is 1 5 4

This means there are connections between

$1 \to 5,\ 5 \to 1,\ 1 \to 4,\ 4 \to 1,\ 5 \to 4,\ 4 \to 5$

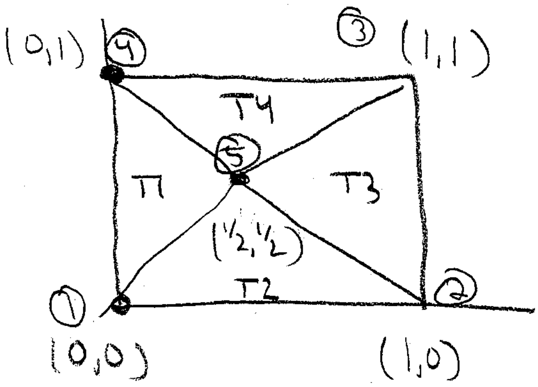
So set

$(1,5), (5,1), (1,4), (4,1), (5,4), (4,5) = 1$

loop over all the triangles. The connectivity matrix would look like

$$
\begin{array}{cc}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\left[\begin{array}{ccccc} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{array}\right]
\end{array}
$$

Example



○ - vertex numbers

T - triangle numbers

The vertices can be numbered in any way you want (same with the triangles)

$$V = \begin{matrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1/2 & 1/2 \end{matrix} \qquad T = \begin{matrix} 1 & 5 & 4 \\ 1 & 2 & 5 \\ 2 & 3 & 5 \\ 3 & 4 & 5 \end{matrix}$$

note: T1 could be

1 5 4 or

5 4 1 or

4 1 5

Note creating a triangulation is difficult. People have written Matlab routines for this. The examples we will use are written using a C program called triangle (mainly because I have an existing set of routines that work with this program).

Viewing Mesh

$A = \text{sparse}(NV, NV)$ create $NV \times NV$ matrix of 0's

for $k = 1 : NT$

$v = T(k, :)$ → get row k of T. This is the 3 vertices that make up triangle k

$A(v, v) = \text{ones}(3)$ → 3x3 matrix of ones.

End

gplot (A, V) → matlab function for plotting undirected graphs

Now it is easy to view z

Suppose $z = (x+y)\,e^{-(x+y)}$

given V, T

```
x = V(:,1);
y = V(:,2);
z = (x+y).*exp(-(x+y));
trisurf(T, x, y, z)
```

← built in Matlab command for viewing functions on triangularizations.