1 idw.m

```
function [zgrid, npts] = idw(varargin)
% IDW Performs Inverse Distance Weighting to Grid Data
%   Grid sparse data onto a grid using inverse distance weighting algorithm
%   The weighted average of points within a specified radius(r) are computed
%   using the equation:
%
%                     W = distance^power
%
%   The most common implementation uses a power value of −2
%
%   ∗ NOTE that the equation requires a negative power to be inverse
%
%   [zgrid, npts] = idw(x,y,xgrid,r,p)                         % 1D Case
%   [zgrid, npts] = idw(x,y,z,xgrid,ygrid,r,p)                 % 2D Case
%   [zgrid, npts] = idw(x,y,z,I,xgrid,ygrid,zgrid,r,p)         % 3D Case
```

1.1 Motivation/Concept

Inverse Distance Weighting(IDW) is a very common algorithm for interpolating sparse data. IDW performs a weighted average on points within a certain radius based the equation:

$$W = distance^{power}$$

Or more formally

$$\begin{aligned} d_i &= \text{distance to point i} \\ p &= \text{idw power constant} \\ r &= \text{radius constant} \\ n &= \text{number of points within radius (r)} \\ Y &= \frac{\sum\limits_{i=1}^{n} y_i d_i^p}{\sum\limits_{i=1}^{n} d_i^p} \end{aligned}$$

1.2 Inputs/Outputs

```
% Inputs:
%   − x     : vector of x data
%   − y     : vector of y data
%   − z     : vector of z data
%   − xgrid : x grid nodes to query data at
%   − ygrid : y grid nodes to query data at
%   − r     : IDW search radius
%   − p     : IDW power (default=−2)
%
% Outputs:
%   − zgrid : z data at (xgrid,ygrid) nodes (nan if no data)
%   − npts  : number of points at (xgrid,ygrid) nodes
```

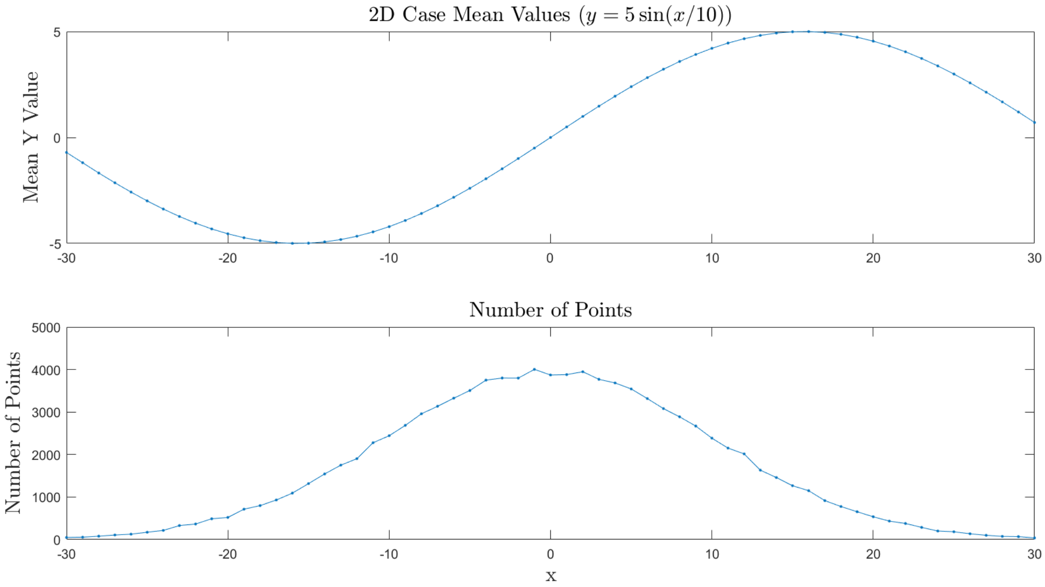
1.3 Examples *exampleIdw.m*

Example IDW Line

This example demonstrates generation of a line using IDW with the default power of -2.

```matlab
%% EXAMPLE 1D
npts1 = 100000;
x = randn(npts1,1)*10;
y = sin(.1*x)*5;
xgi = -30:1:30;

[val,numpts]=idw(x,y,xgi,0.5);
```

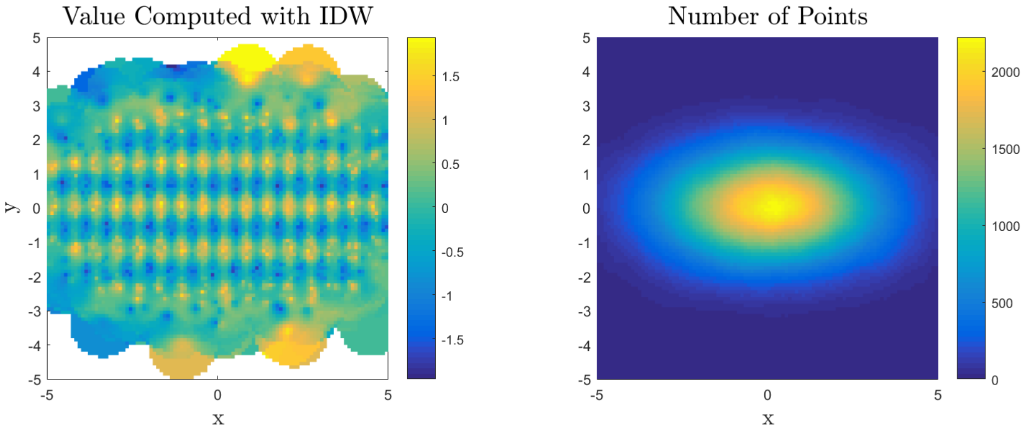


Example IDW 2D

This example demonstrates generation of a plane grid using IDW with the default power of -2.

```
%% EXAMPLE 2D
RADIUS = 1;
POWER = -2;
x = randn(10000,1)*2;
y = randn(10000,1)*1;
z = sin(10*x) + cos(5*y);
[xg,yg]=meshgrid(-5:.1:5,-5:.1:5);

[zgrid, npts] = idw(x,y,z,xg,yg,RADIUS,POWER);
```



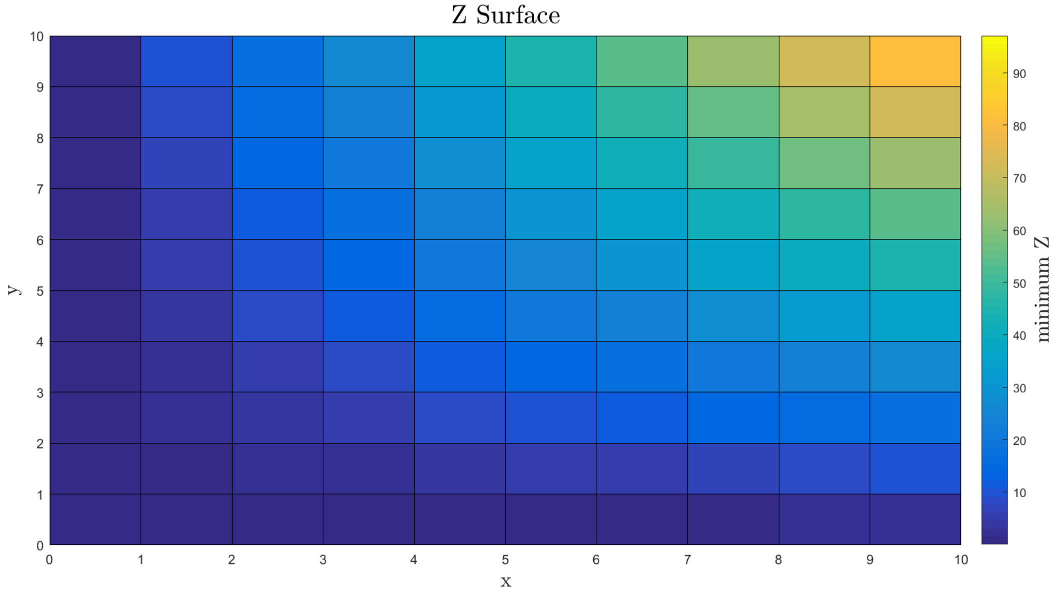
Value Computed with IDW

Number of Points

Example IDW Plane (Power = -1)

This example demonstrates generation of a plane grid using IDW with a power of -1.

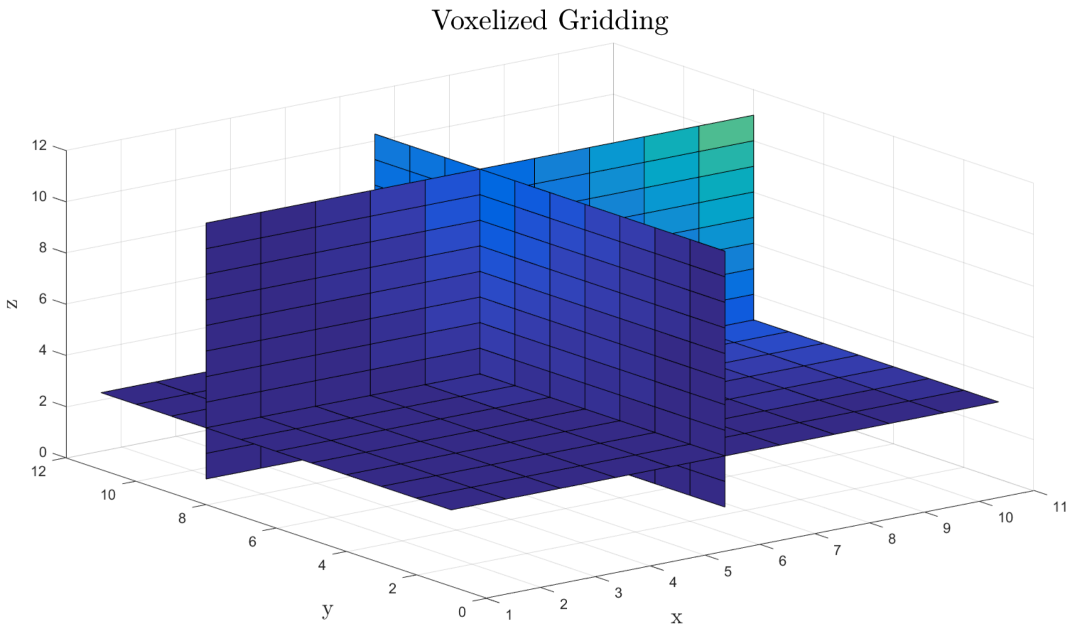
```
%% EXAMPLE 2D
npts2 = 10000;
x = rand(npts2,1)*10;
y = rand(npts2,1)*10;
z = x.*y;
xgi = 0:1:10;
ygi = 0:1:10;
[xg,yg]=meshgrid(xgi,ygi);
val=idw(x,y,z,xg,yg,0.5,−1);
```



Example IDW Voxels

This example demonstrates generation of voxels.

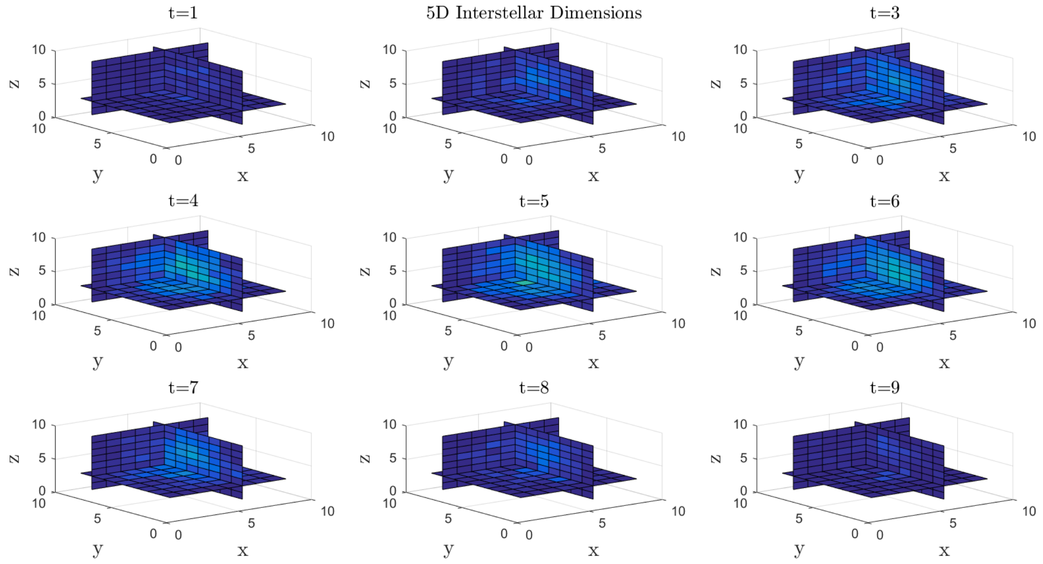
```
%% EXAMPLE 3D *uses meshgrid array
x = rand(100000,1)*10;
y = rand(100000,1)*10;
z = rand(100000,1)*10;
I=x.^2.*y.*z;
xgi = 0:1:10;
ygi = 0:1:10;
zgi = 0:1:10;
[xg,yg,zg]=meshgrid(xgi,ygi,zgi);
val=idw(x,y,z,I,xg,yg,zg,0.5); % Mean Intensity voxels
```



Example Temporal Voxels

This example demonstrates generation a 4d grid using x,y,z, and time.

```matlab
%% Example 4D
npts4 = 100000;
xn = randn(npts4,5)*2+5;
xni = 1:1:9;
[xg1,xg2,xg3,xg4]=ndgrid(xni,xni,xni,xni);
[val,numpts]=idw(xn(:,1),xn(:,2),xn(:,3),xn(:,4),xn(:,5),...
    xg1,xg2,xg3,xg4,0.5);% interstellar dimensions
```


t=1
5D Interstellar Dimensions
t=3
t=4
t=5
t=6
t=7
t=8
t=9
z
y
x