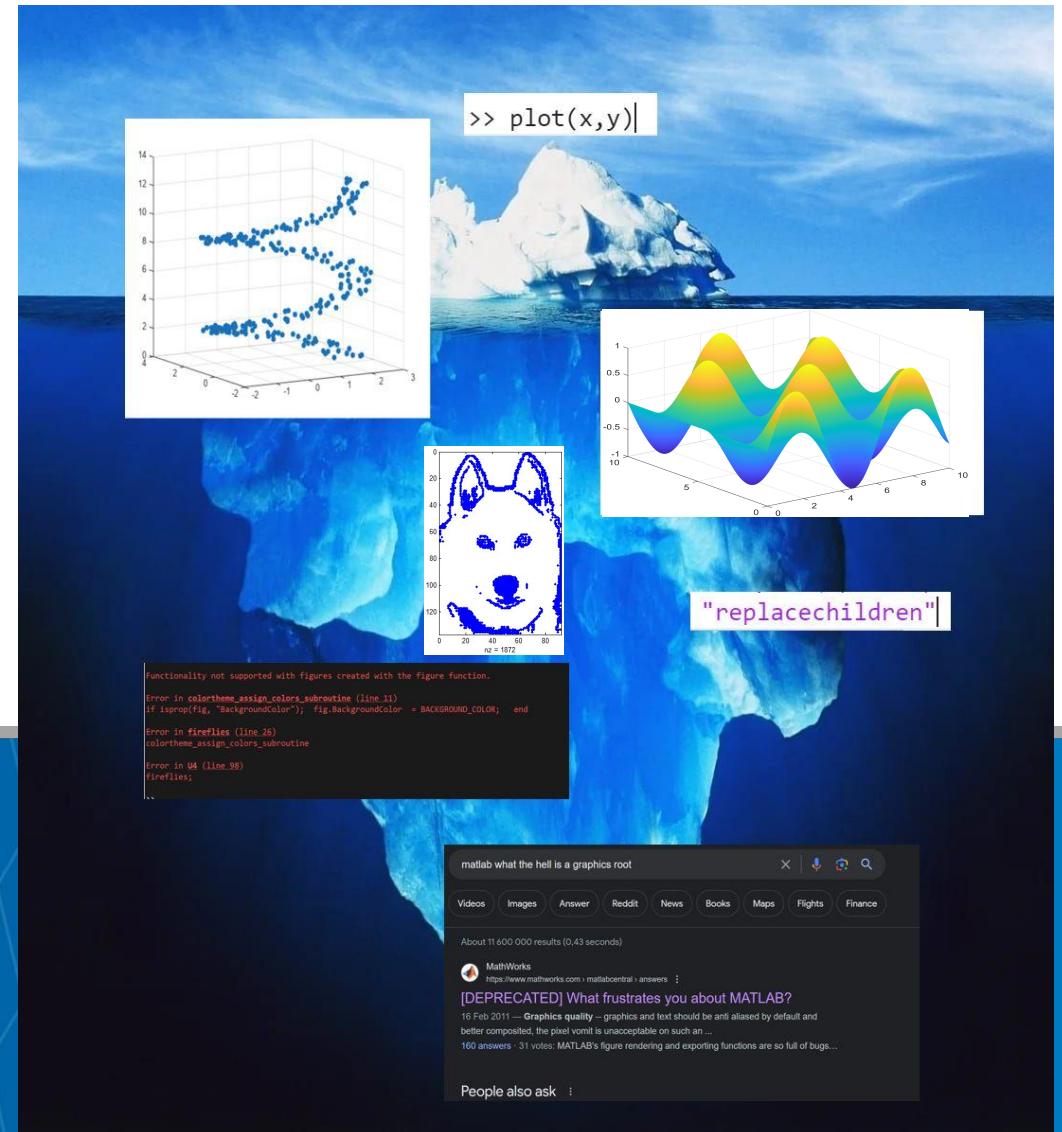
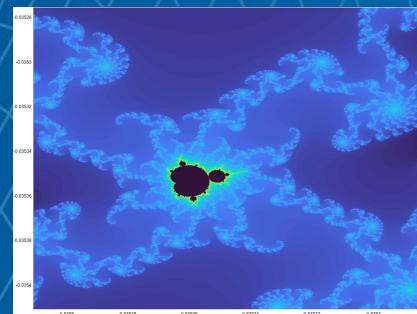
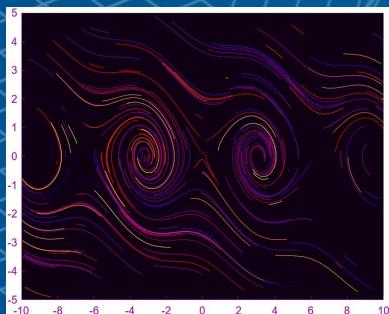
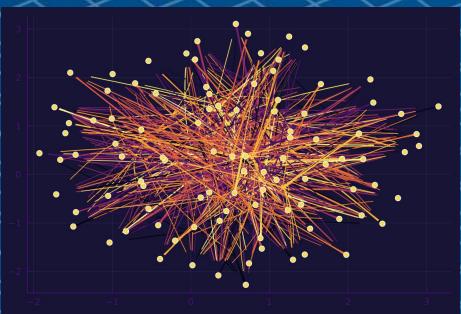
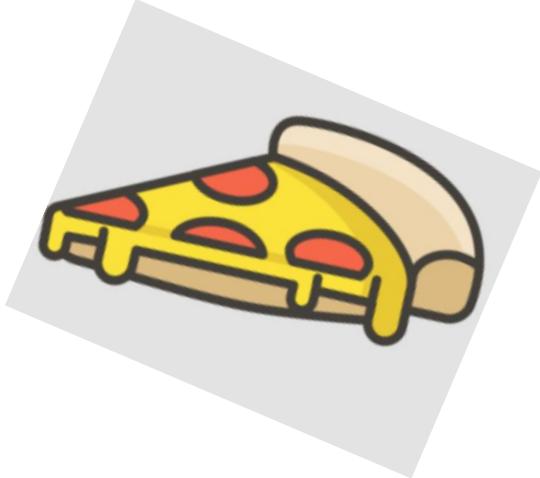
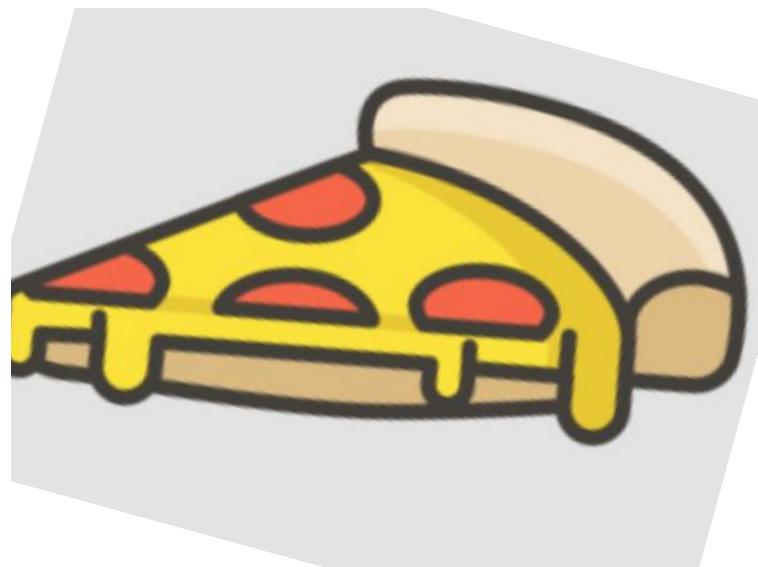


Data visualization in MATLAB

Vilgot Lötberg
18-12-2023





First off, logistics

What kind of Pizza do you want? *

- Vesuvio (ham)
- Margherita (cheese)
- Kebab-pizza
- Pepperoni
- Pompei (bacon, onion)
- Hawaii (pineapple, ham)
- Altonno (tuna, onion)

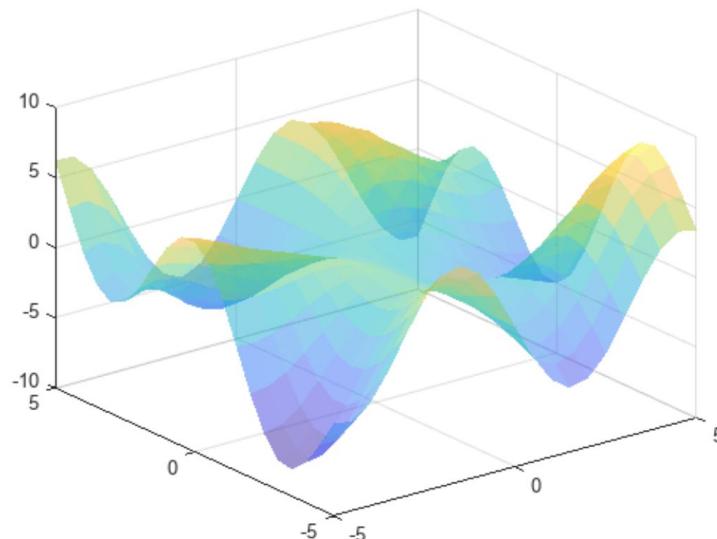
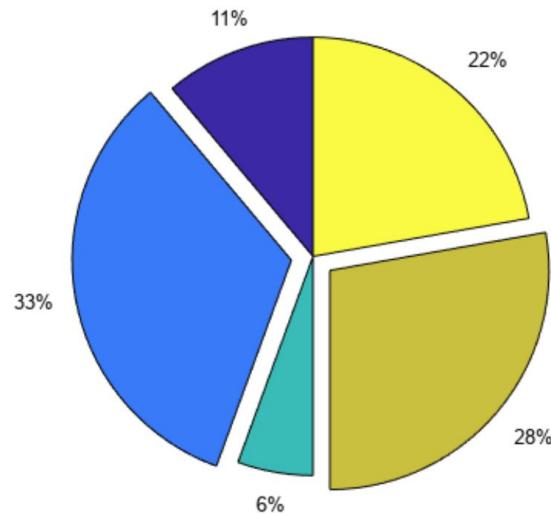
Agenda for tonight:

- General plot types
- Line plots
- Surface plots
- Quiver plots
- Demos
- Using surface to make 3D-models
- Importing 3D-models
- Colorthemes- library

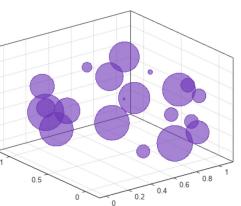
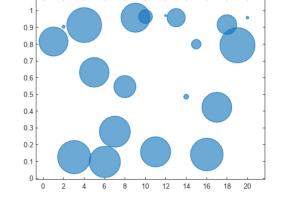
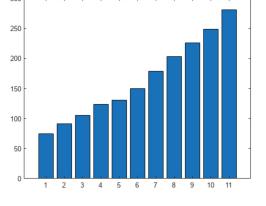
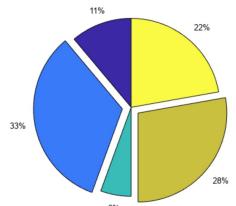
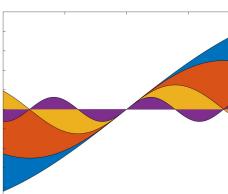
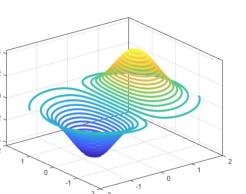
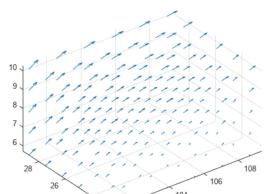
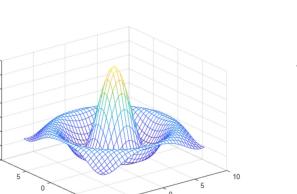
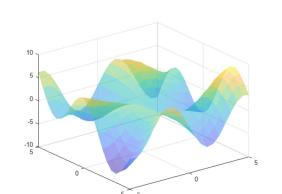
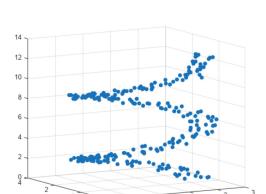
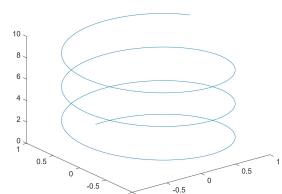
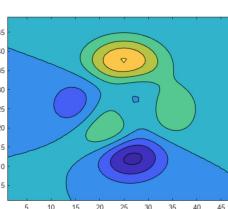
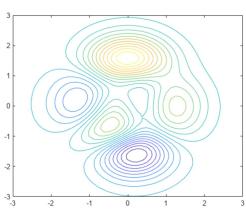
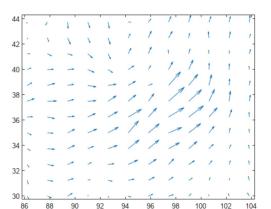
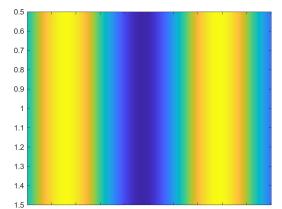
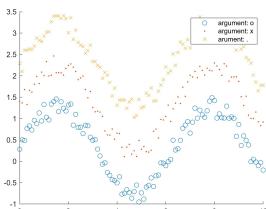
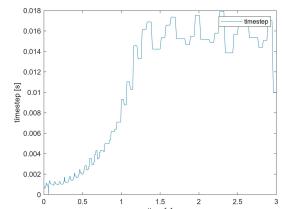
- Food!!
- Problem-solving
- Merch giveaway!
- Kahoot if we have time

So, you want to plot some data!

- Continuous function?
- Discrete number of data-points?
- Dimensions? 2D or 3D or even 4D?
- Images?
- Model of a real world system?

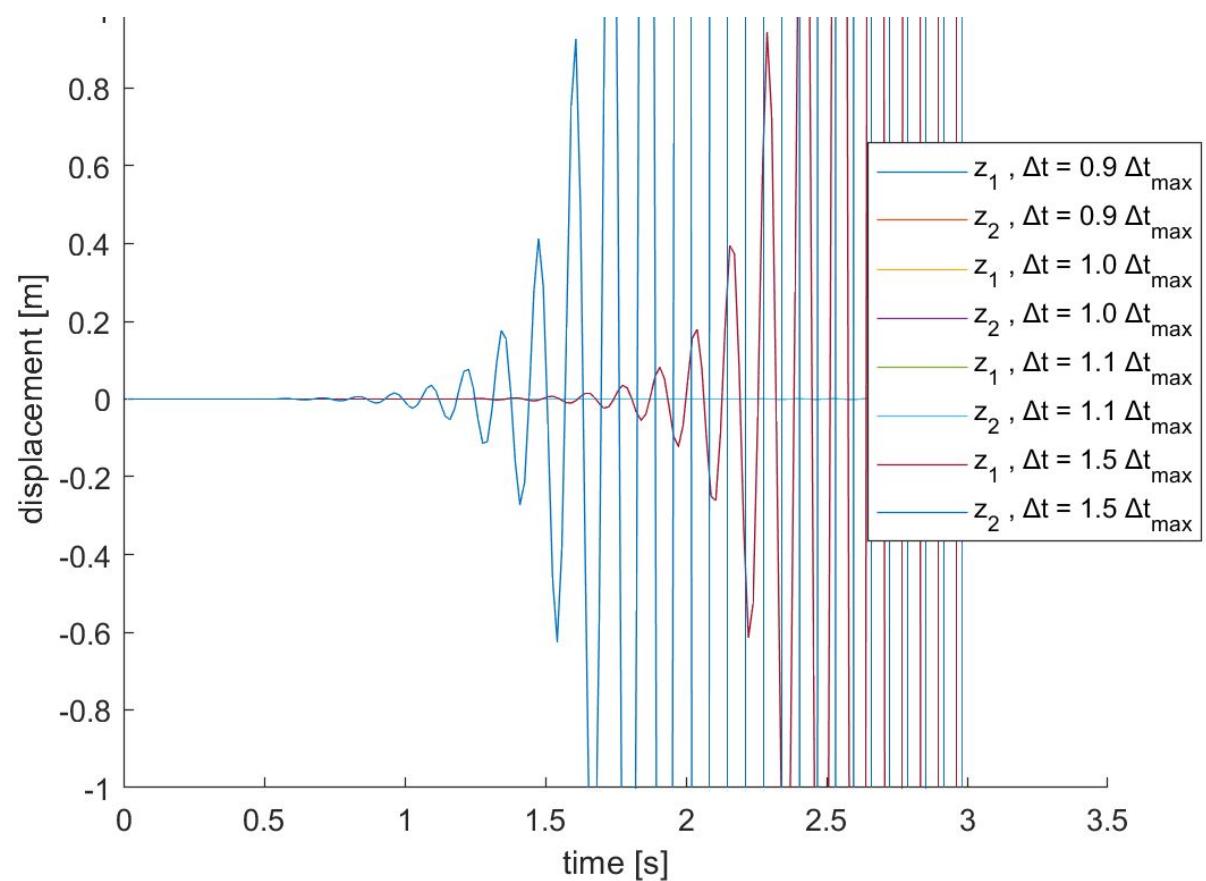
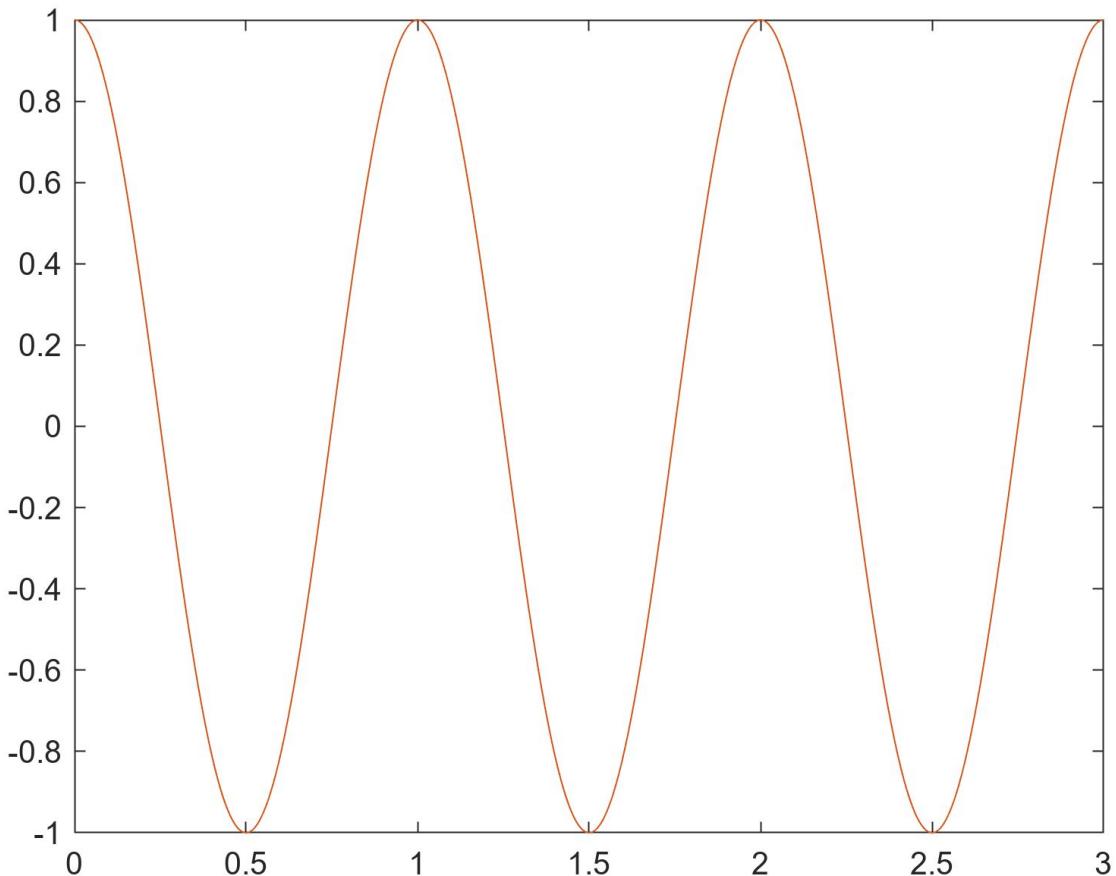


Choose wisely...



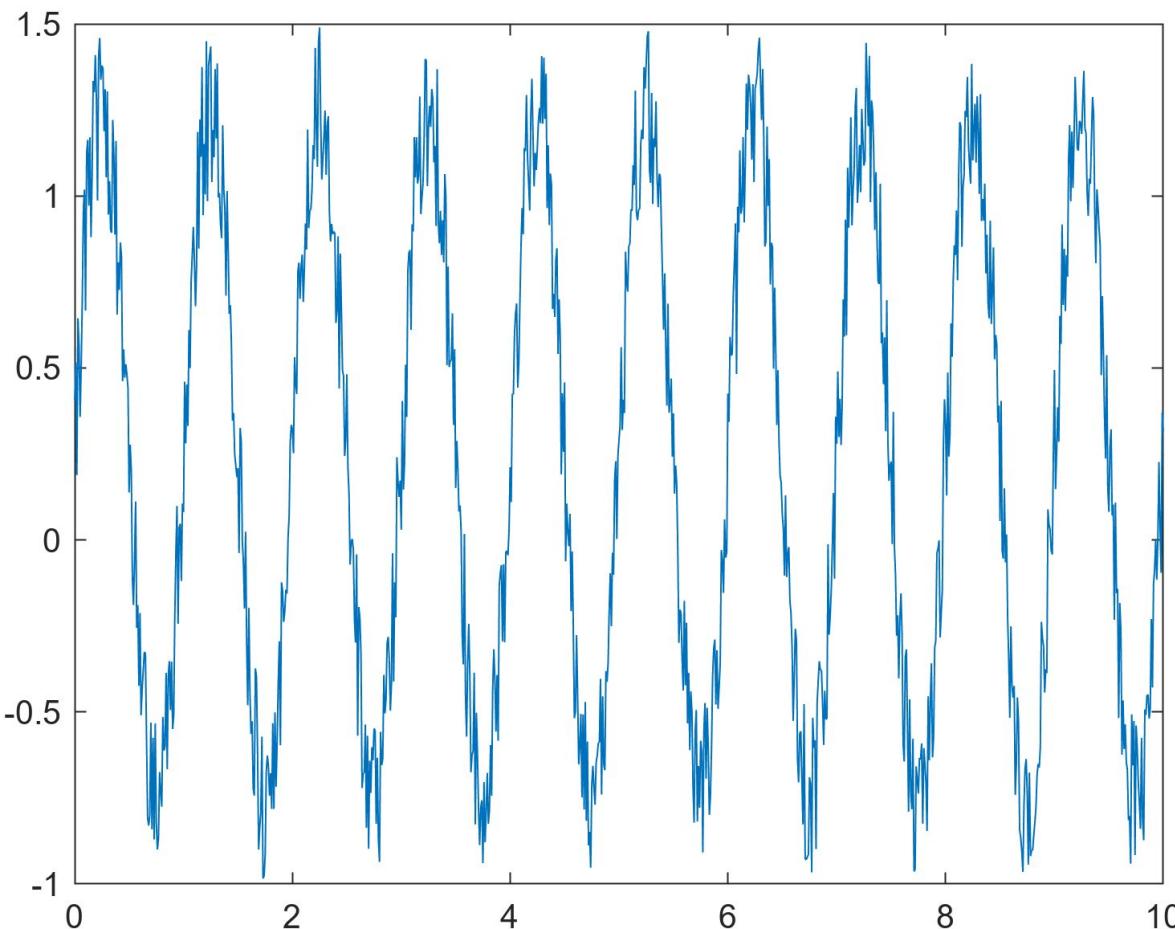
Standard line-plots:

```
x = 0:0.001:3;  
y = cos(2*pi*x);  
  
plot(x,y);
```



Standard line-plots:

```
x = 0:0.01:10;  
y = sin(2*pi*x) + 0.5*rand(size(x));  
  
plot(x,y);
```



Customization:

```
x = 0:0.01:10;  
y = sin(pi*x);  
  
plot(x,y);  
hold on  
  
plot(x,y+1, "-.");  
plot(x,y+2, ":" );  
plot(x,y+3, "--");  
  
hold off
```

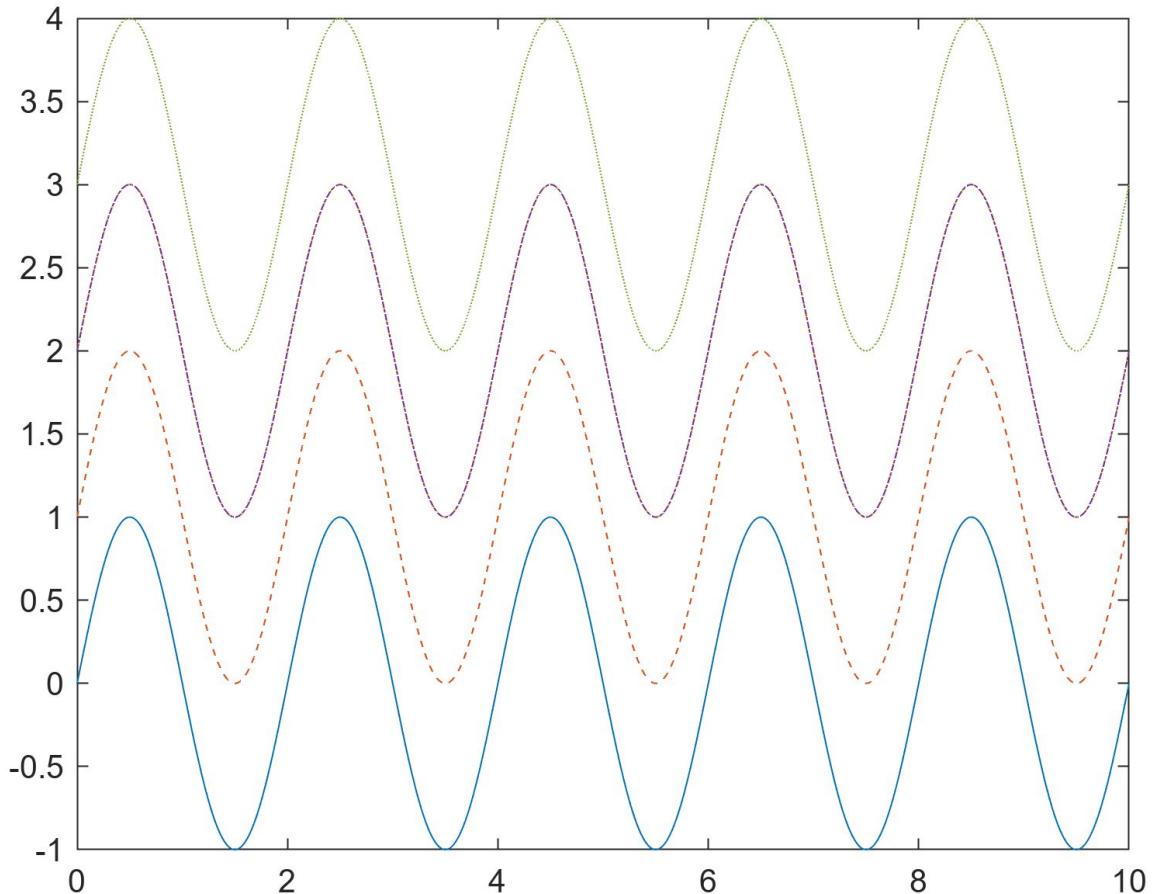
Line Style

"_"

"--"

": "

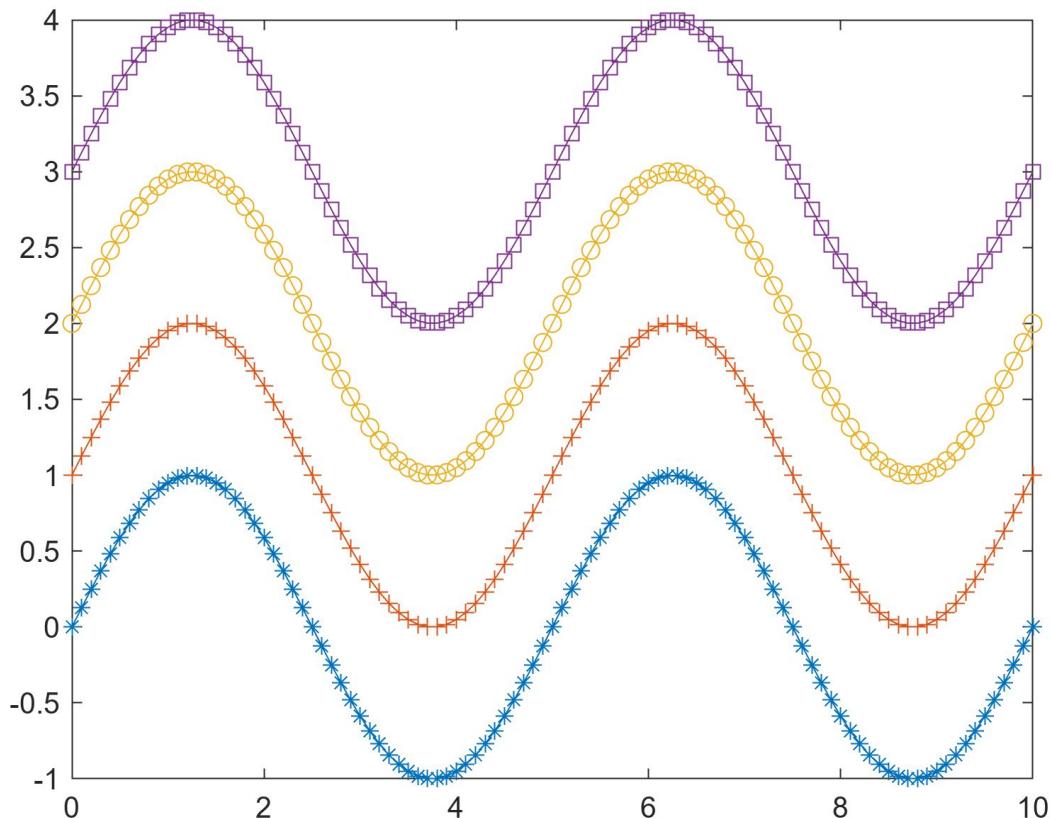
"- ."



Markers:

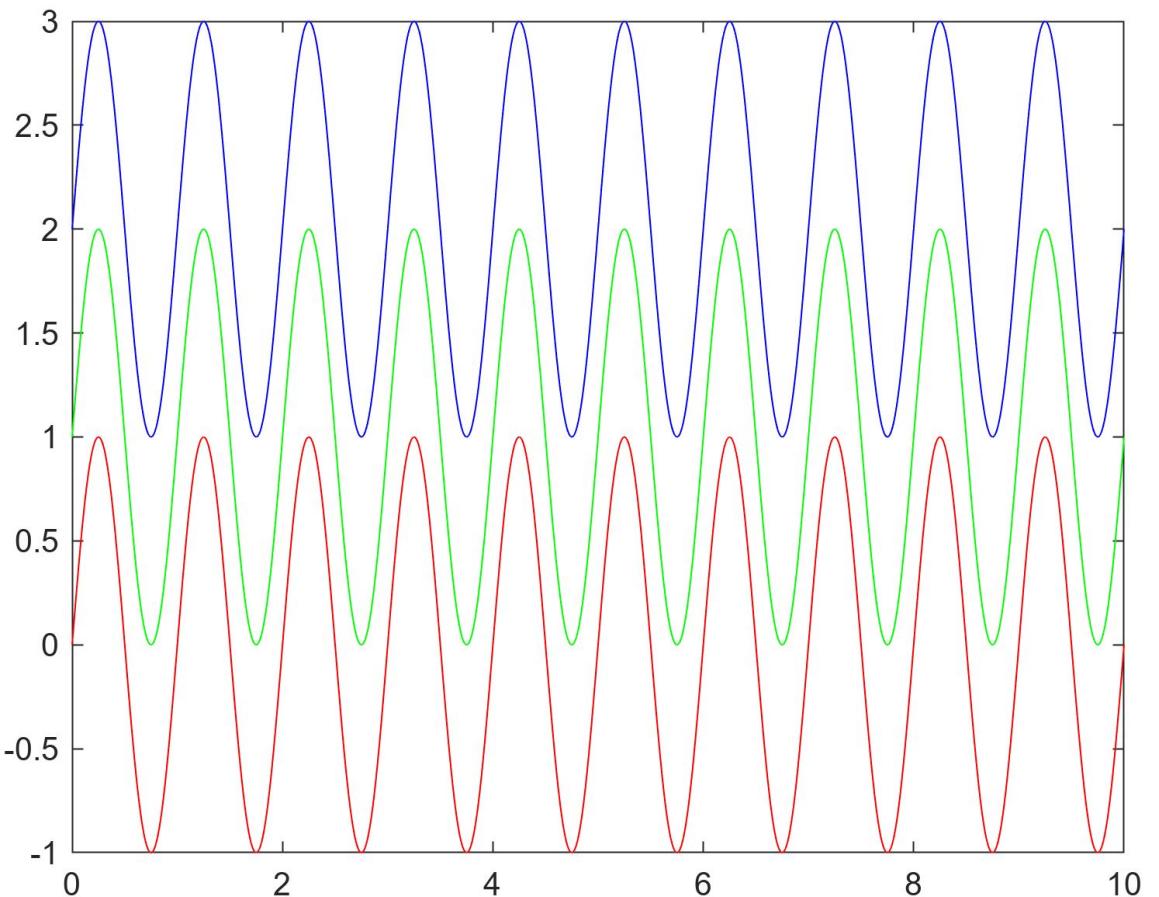
Marker
"o"
"+"
"*"
". "
"x"
"_ "
" "
"square"
"diamond"
"^"
"v"
<"
"pentagram"
"hexagram"

```
myax = axes();  
  
x = 0:0.1:10;  
y = sin(x);  
  
plot(x,y, "Marker", "*")  
hold on  
plot(x,y+1, "Marker", "+")  
plot(x,y+2, "Marker", "o")  
plot(x,y+3, "Marker", "square")  
hold off
```



Change colors:

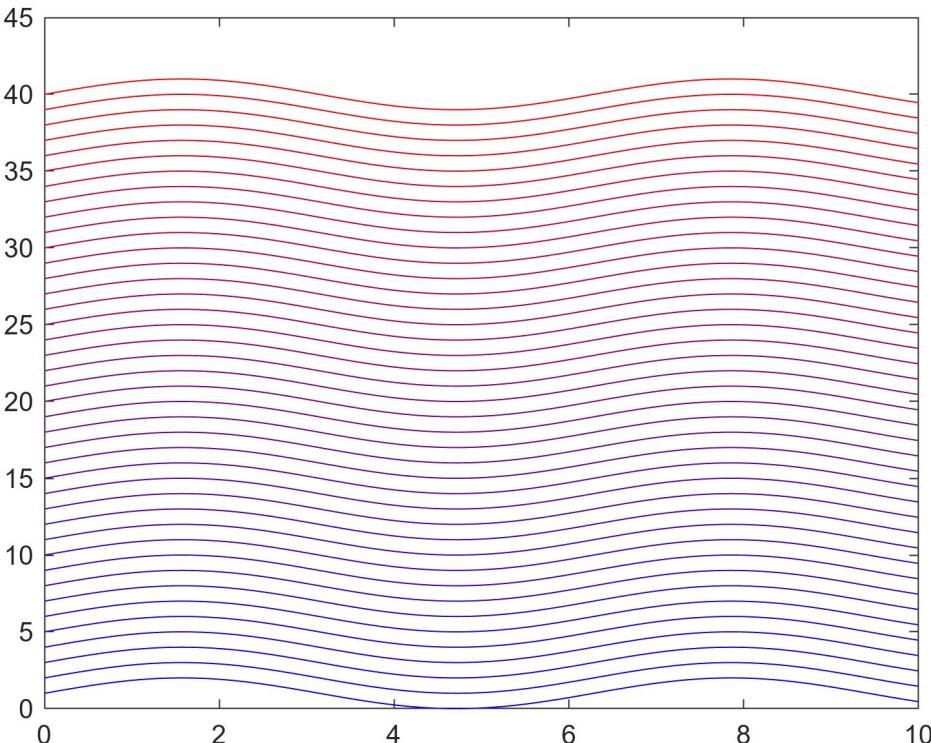
```
x = 0:0.01:10;  
y = sin(2*pi*x);  
  
plot(x,y, "Color",[1 0 0]);  
hold on  
plot(x,y+1, "Color",[0 1 0]);  
plot(x,y+2, "Color",[0 0 1]);  
hold off
```



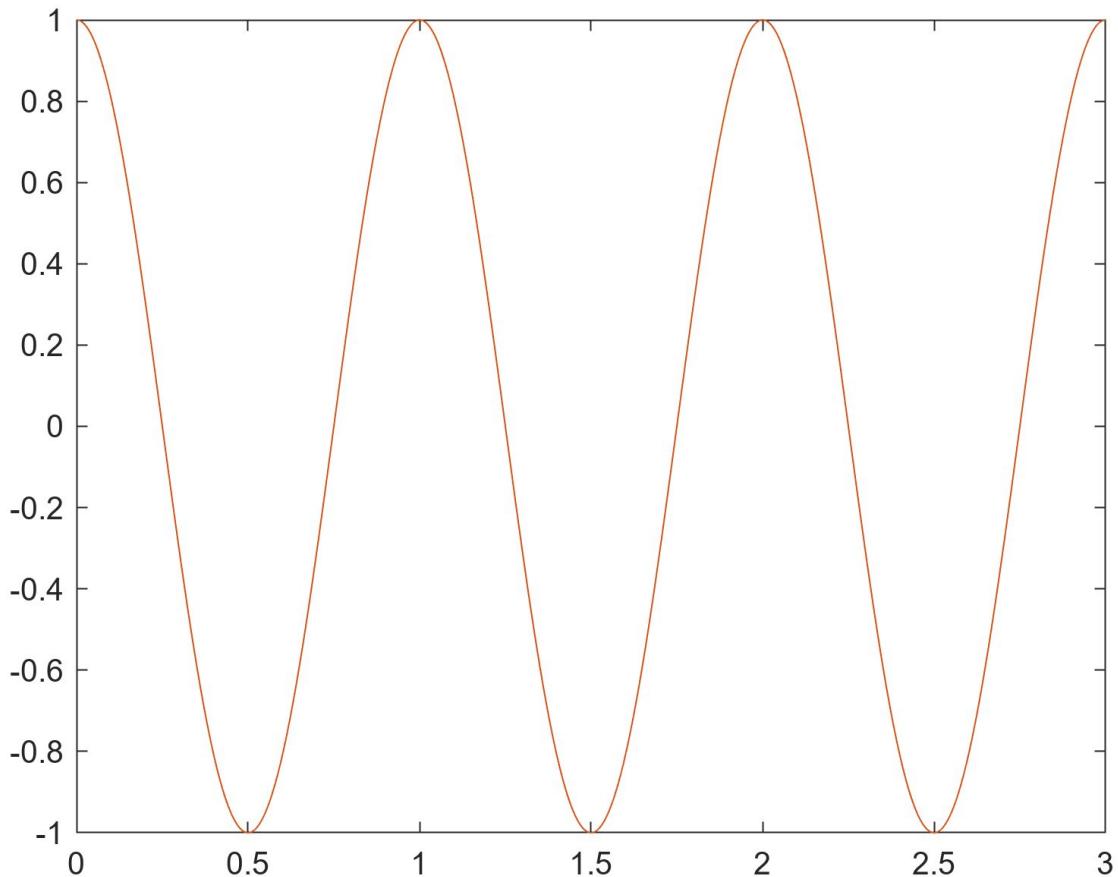
Change colors:

```
myax =  
  
Axes with properties:  
  
    XLim: [0 1]  
    YLim: [0 1]  
    XScale: 'linear'  
    YScale: 'linear'  
    GridLineStyle: '-'  
    Position: [0.1300 0.1100 0.7750 0.8150]  
    Units: 'normalized'  
  
Show all properties  
  
>> myax.ColorOrder  
  
ans =  
  
    0    0.4470    0.7410  
0.8500    0.3250    0.0980  
0.9290    0.6940    0.1250  
0.4940    0.1840    0.5560  
0.4660    0.6740    0.1880  
0.3010    0.7450    0.9330  
0.6350    0.0780    0.1840
```

```
myax = axes();  
  
x = 0:0.01:10;  
y = sin(x) + (1:40)';  
  
plot(x,y)  
  
myax.ColorOrder = [linspace(0,1,40);  
                   linspace(0,0,40);  
                   linspace(1,0,40)]';
```

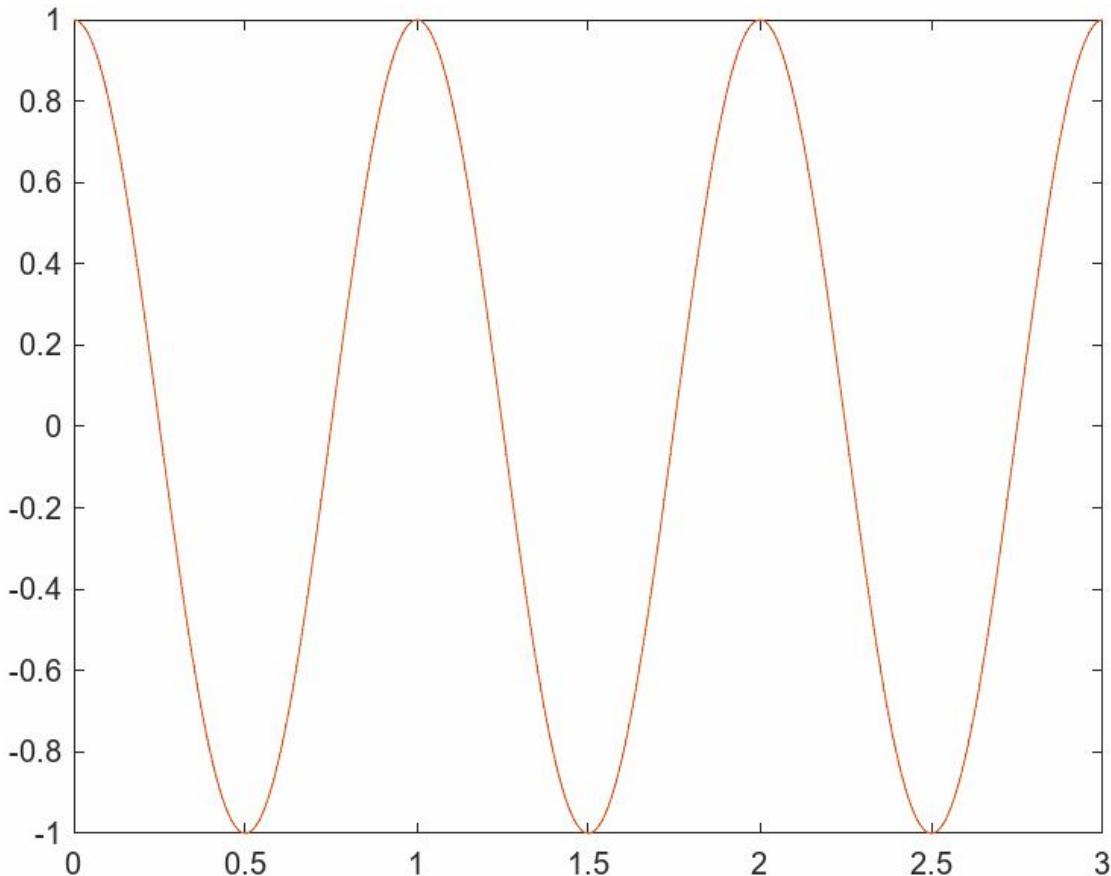


In 3D?



```
x = 0:0.001:3;  
y = cos(2*pi*x);  
  
plot(x,y);
```

In 3D?

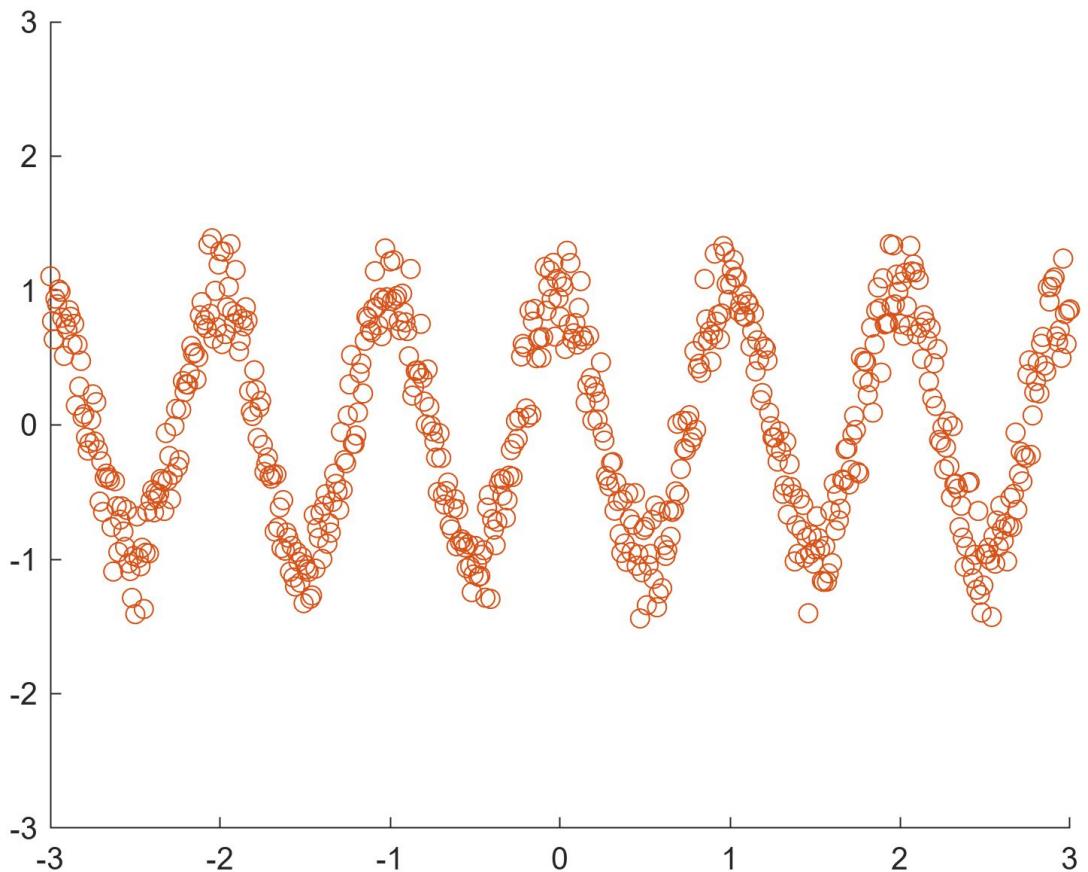


```
x = 0:0.01:10;  
y = sin(2*pi*x);  
z = cos(2*pi*x);
```

```
plot3(x,y,z);|
```

scatter()

```
x = 0:0.1:10;  
y = sin(x) + 0.1*rand(size(x));  
  
scatter(x, y, "o")
```

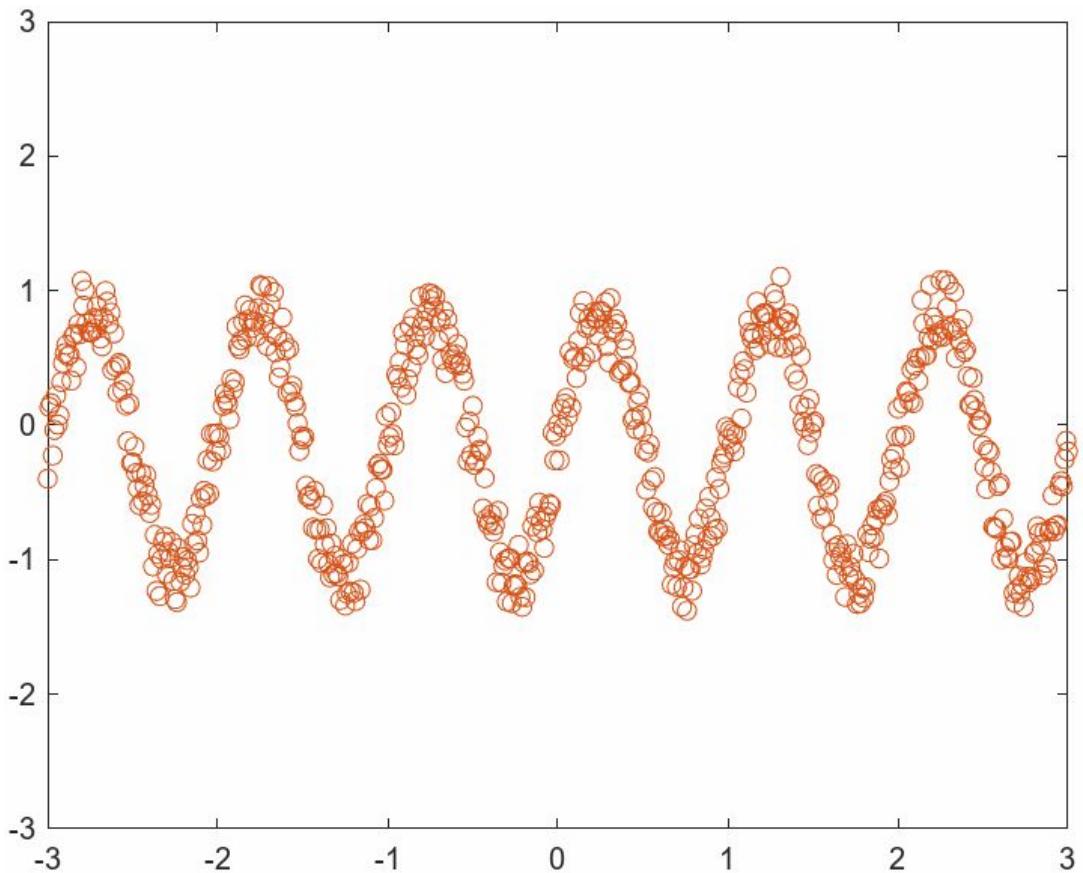


Good for:

- Visualizing discrete data points
- Clear visual representation of the amount of data

scatter3()

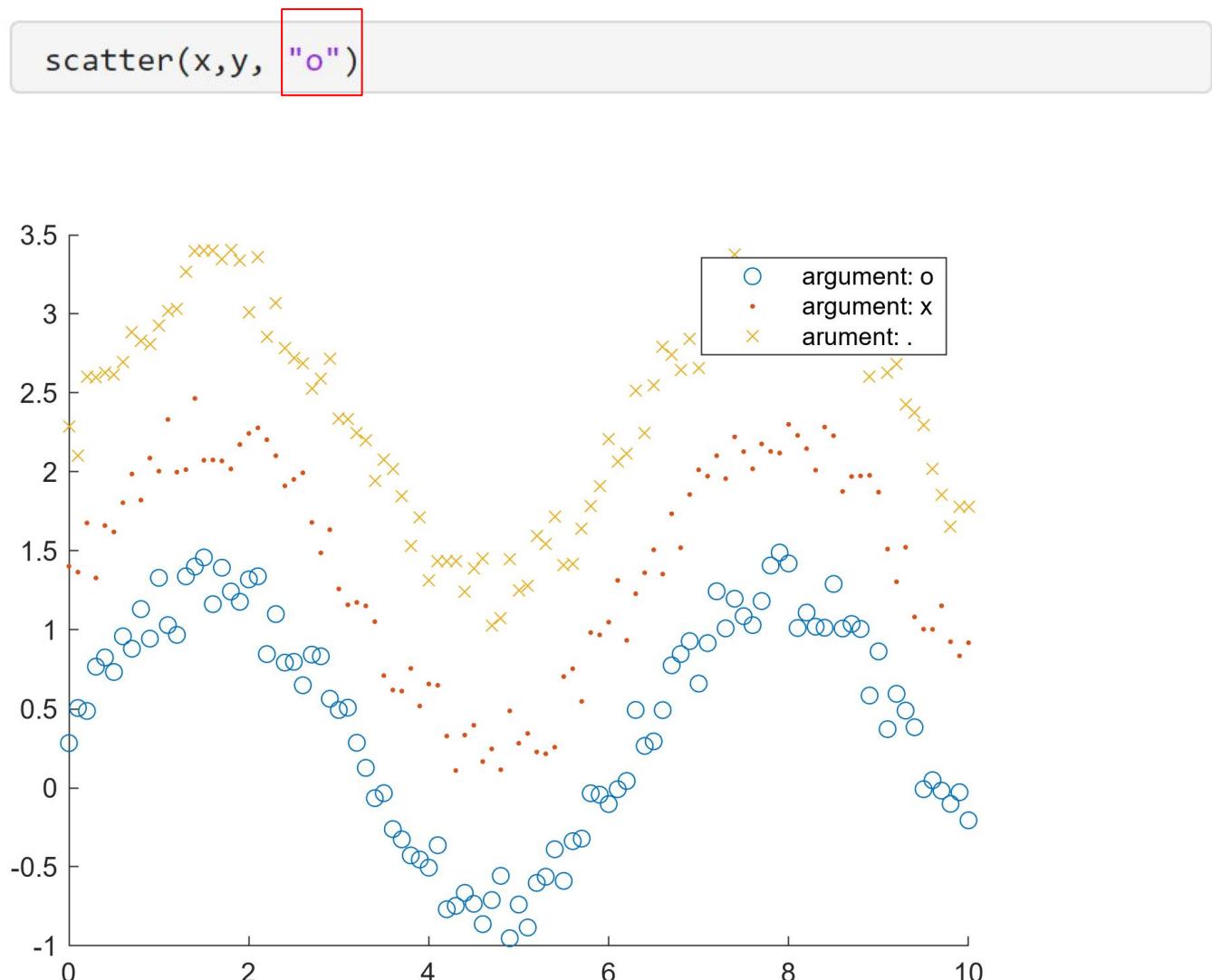
```
x = 0:0.1:10;  
y = sin(x) + 0.1*rand(size(x));  
z = cos(x) + 0.1*rand(size(x));  
  
scatter(x,y,z, "o")
```



Good for:

- Visualizing discrete data points
- Clear visual representation of the amount of data

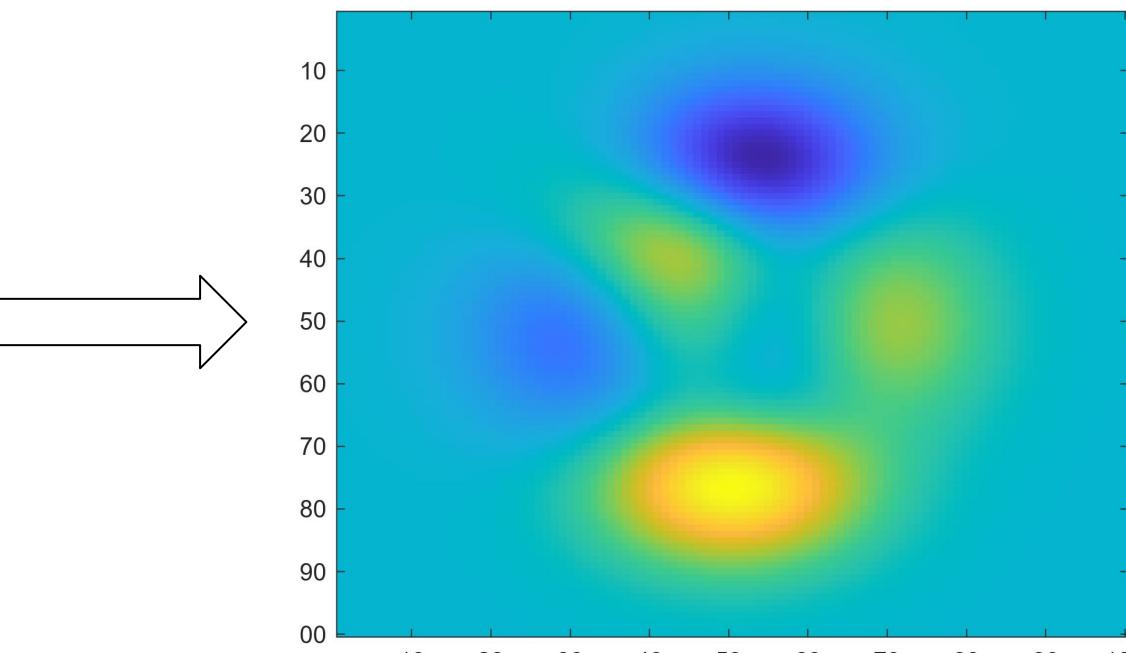
Markers:



Marker	Resulting Marker
"o"	○
"+"	+
"*"	*
".."	•
"x"	×
"_"	—
" "	
"square"	□
"diamond"	◇
"^"	△
"v"	▽
▷	
<"	◁
"pentagram"	☆
"hexagram"	★

Visualizing matrix-data: imagesc()

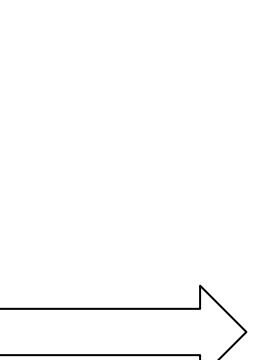
C																			
0.7282	0.6822	0.4987	0.4990	0.3322	0.2673	0.2132	0.1684	0.1318	0.1023	0.0786	0.0599	0.0453	0.0339	0.0252					
0.6756	0.5648	0.4676	0.3834	0.3114	0.2585	0.1997	0.1578	0.1235	0.0958	0.0737	0.0561	0.0424	0.0317	0.0236					
0.6381	0.5265	0.4358	0.3572	0.2901	0.2333	0.1868	0.1469	0.1149	0.0892	0.0685	0.0522	0.0394	0.0295	0.0219					
0.5845	0.4882	0.4839	0.3309	0.2686	0.2160	0.1721	0.1359	0.1063	0.0824	0.0634	0.0483	0.0365	0.0273	0.0203					
0.5396	0.4586	0.3724	0.3049	0.2474	0.1980	0.1584	0.1250	0.0978	0.0755	0.0582	0.0443	0.0335	0.0251	0.0186					
0.4961	0.4137	0.3417	0.2796	0.2267	0.1821	0.1458	0.1143	0.0894	0.0693	0.0532	0.0405	0.0306	0.0229	0.0178					
0.4546	0.3948	0.3124	0.2554	0.2068	0.1666	0.1328	0.1041	0.0813	0.0630	0.0473	0.0356	0.0278	0.0208	0.0154					
0.4154	0.3454	0.2946	0.2324	0.1868	0.1508	0.1198	0.0943	0.0736	0.0570	0.0437	0.0322	0.0259	0.0181	0.0139					
0.3799	0.3145	0.2346	0.1908	0.1539	0.1230	0.1075	0.0766	0.0597	0.0461	0.0353	0.0268	0.0202	0.0150	0.0111					
0.3452	0.2858	0.2246	0.1808	0.1539	0.1230	0.1075	0.0766	0.0597	0.0461	0.0353	0.0268	0.0202	0.0150	0.0111					
0.3143	0.2595	0.2124	0.1724	0.1388	0.1107	0.0876	0.0687	0.0534	0.0412	0.0315	0.0239	0.0180	0.0134	0.0099					
0.2862	0.2356	0.1923	0.1556	0.1249	0.0995	0.0785	0.0615	0.0477	0.0437	0.0281	0.0212	0.0160	0.0119	0.0088					
0.2697	0.2138	0.1748	0.1404	0.1124	0.0892	0.0703	0.0549	0.0425	0.0327	0.0249	0.0188	0.0141	0.0105	0.0077					
0.2377	0.1942	0.1574	0.1266	0.1010	0.0880	0.0628	0.0490	0.0378	0.0299	0.0221	0.0167	0.0125	0.0093	0.0068					
0.2168	0.1764	0.1424	0.1142	0.0908	0.0717	0.0561	0.0436	0.0338	0.0257	0.0195	0.0147	0.0110	0.0081	0.0068					
0.1978	0.1682	0.1289	0.1029	0.0816	0.0642	0.0581	0.0388	0.0298	0.0228	0.0172	0.0130	0.0097	0.0071	0.0052					
0.1804	0.1455	0.1186	0.0927	0.0732	0.0574	0.0447	0.0345	0.0264	0.0288	0.0152	0.0114	0.0085	0.0063	0.0046					
0.1645	0.1321	0.1054	0.0835	0.0657	0.0513	0.0398	0.0306	0.0234	0.0178	0.0134	0.0100	0.0074	0.0055	0.0048					
0.1497	0.1193	0.0952	0.0751	0.0589	0.0458	0.0354	0.0272	0.0207	0.0157	0.0118	0.0088	0.0065	0.0048	0.0035					
0.1360	0.1083	0.0858	0.0674	0.0527	0.0408	0.0315	0.0241	0.0183	0.0130	0.0093	0.0061	0.0041	0.0030	0.0020					
0.1231	0.0979	0.0771	0.0590	0.0460	0.0356	0.0270	0.0211	0.0141	0.0104	0.0070	0.0057	0.0040	0.0030	0.0026					
0.1110	0.0873	0.0691	0.0539	0.0418	0.0347	0.0287	0.0214	0.0166	0.0097	0.0058	0.0043	0.0031	0.0023	0.0019					
0.0997	0.0786	0.0616	0.0479	0.0371	0.0285	0.0217	0.0165	0.0124	0.0093	0.0069	0.0051	0.0037	0.0027	0.0019					
0.0899	0.0700	0.0547	0.0424	0.0327	0.0251	0.0191	0.0144	0.0108	0.0081	0.0060	0.0044	0.0032	0.0023	0.0017					
0.0799	0.0628	0.0483	0.0374	0.0287	0.0220	0.0167	0.0125	0.0094	0.0070	0.0052	0.0038	0.0028	0.0020	0.0014					
0.0697	0.0545	0.0424	0.0327	0.0251	0.0191	0.0145	0.0109	0.0081	0.0060	0.0044	0.0032	0.0024	0.0017	0.0012					
0.0610	0.0477	0.0370	0.0285	0.0218	0.0166	0.0125	0.0094	0.0070	0.0052	0.0038	0.0028	0.0020	0.0014	0.0010					
0.0530	0.0413	0.0328	0.0246	0.0188	0.0143	0.0107	0.0088	0.0060	0.0044	0.0032	0.0024	0.0017	0.0012	0.0009					
0.0457	0.0356	0.0275	0.0211	0.0161	0.0122	0.0092	0.0069	0.0051	0.0037	0.0027	0.0020	0.0014	0.0010	0.0007					
0.0391	0.0304	0.0235	0.0188	0.0137	0.0103	0.0078	0.0058	0.0043	0.0032	0.0023	0.0017	0.0012	0.0009	0.0006					
0.0332	0.0258	0.0199	0.0152	0.0115	0.0087	0.0065	0.0049	0.0036	0.0026	0.0019	0.0014	0.0010	0.0007	0.0005					
0.0279	0.0216	0.0167	0.0127	0.0097	0.0073	0.0055	0.0041	0.0030	0.0022	0.0016	0.0012	0.0008	0.0004						
0.0233	0.0186	0.0139	0.0106	0.0080	0.0066	0.0045	0.0034	0.0025	0.0018	0.0013	0.0010	0.0007	0.0005	0.0003					
0.0192	0.0149	0.0114	0.0087	0.0066	0.0050	0.0037	0.0028	0.0020	0.0015	0.0011	0.0008	0.0006	0.0004	0.0003					
0.0162	0.0109	0.0084	0.0064	0.0046	0.0033	0.0024	0.0018	0.0013	0.0010	0.0007	0.0005	0.0004	0.0003	0.0002					
0.0128	0.0099	0.0076	0.0058	0.0044	0.0033	0.0024	0.0018	0.0013	0.0010	0.0007	0.0005	0.0004	0.0003	0.0002					
0.0103	0.0080	0.0061	0.0046	0.0035	0.0026	0.0020	0.0015	0.0011	0.0008	0.0006	0.0004	0.0003	0.0002	0.0001					
0.0082	0.0063	0.0049	0.0037	0.0028	0.0021	0.0016	0.0012	0.0008	0.0006	0.0004	0.0003	0.0002	0.0001	0.0001					
0.0065	0.0050	0.0038	0.0029	0.0022	0.0017	0.0012	0.0009	0.0007	0.0005	0.0004	0.0003	0.0002	0.0001	0.0001					
0.0051	0.0039	0.0030	0.0023	0.0017	0.0013	0.0010	0.0007	0.0005	0.0003	0.0002	0.0001	0.0001	0.0001	0.0001					
0.0040	0.0030	0.0023	0.0018	0.0013	0.0010	0.0007	0.0005	0.0004	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001					
0.0030	0.0023	0.0018	0.0014	0.0010	0.0008	0.0006	0.0004	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001	0.0000					



```
z = peaks(20);
imagesc(z)
```

Visualizing matrix-data: surf()

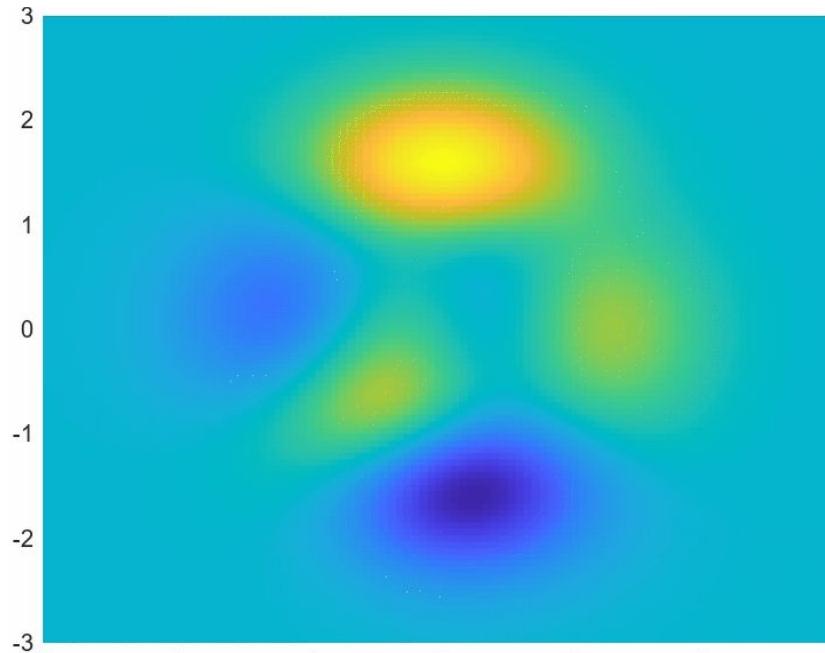
Z																			
0.7282	0.6822	0.4987	0.4999	0.3322	0.2673	0.2132	0.1684	0.1318	0.1023	0.0786	0.0599	0.0453	0.0339	0.0252					
0.6756	0.5648	0.4676	0.3834	0.3114	0.2595	0.1997	0.1578	0.1235	0.0958	0.0737	0.0561	0.0424	0.0317	0.0236					
0.6381	0.5265	0.4358	0.3572	0.2901	0.2333	0.1868	0.1469	0.1149	0.0892	0.0685	0.0522	0.0394	0.0295	0.0219					
0.5845	0.4882	0.4839	0.3309	0.2686	0.2160	0.1721	0.1359	0.1063	0.0824	0.0634	0.0483	0.0365	0.0273	0.0203					
0.5396	0.4586	0.3724	0.3049	0.2474	0.1988	0.1584	0.1258	0.0978	0.0755	0.0582	0.0443	0.0335	0.0251	0.0186					
0.4961	0.4137	0.3417	0.2796	0.2267	0.1821	0.1458	0.1143	0.0894	0.0693	0.0532	0.0405	0.0306	0.0229	0.0179					
0.4546	0.3948	0.3124	0.2554	0.2068	0.1666	0.1328	0.1041	0.0813	0.0630	0.0473	0.0356	0.0278	0.0208	0.0154					
0.4154	0.3454	0.2946	0.2324	0.1898	0.1509	0.1198	0.0943	0.0736	0.0570	0.0437	0.0322	0.0259	0.0187	0.0139					
0.3799	0.2945	0.2346	0.1908	0.1539	0.1239	0.0975	0.0766	0.0597	0.0461	0.0353	0.0268	0.0202	0.0159	0.0111					
0.3452	0.2858	0.2346	0.1908	0.1539	0.1239	0.0975	0.0766	0.0597	0.0461	0.0353	0.0268	0.0202	0.0159	0.0111					
0.3143	0.2595	0.2124	0.1724	0.1388	0.1107	0.0876	0.0687	0.0534	0.0412	0.0315	0.0239	0.0180	0.0134	0.0099					
0.2862	0.2356	0.1923	0.1556	0.1249	0.0995	0.0785	0.0615	0.0477	0.0367	0.0281	0.0212	0.0160	0.0119	0.0088					
0.2697	0.2138	0.1748	0.1404	0.1124	0.0892	0.0703	0.0549	0.0425	0.0327	0.0249	0.0188	0.0141	0.0105	0.0077					
0.2377	0.1942	0.1574	0.1266	0.1010	0.0880	0.0628	0.0490	0.0378	0.0299	0.0211	0.0167	0.0125	0.0093	0.0068					
0.2168	0.1764	0.1424	0.1142	0.0908	0.0717	0.0561	0.0436	0.0338	0.0257	0.0195	0.0147	0.0110	0.0081	0.0060					
0.1978	0.1602	0.1289	0.1029	0.0816	0.0642	0.0501	0.0388	0.0298	0.0228	0.0172	0.0130	0.0097	0.0071	0.0052					
0.1804	0.1455	0.1168	0.0927	0.0732	0.0574	0.0447	0.0345	0.0264	0.0208	0.0152	0.0114	0.0085	0.0063	0.0046					
0.1645	0.1321	0.1054	0.0835	0.0657	0.0513	0.0398	0.0306	0.0234	0.0178	0.0134	0.0100	0.0074	0.0055	0.0048					
0.1497	0.1193	0.0952	0.0751	0.0589	0.0458	0.0354	0.0272	0.0207	0.0157	0.0118	0.0088	0.0065	0.0048	0.0035					
0.1360	0.1083	0.0858	0.0674	0.0527	0.0408	0.0315	0.0241	0.0183	0.0134	0.0103	0.0074	0.0057	0.0041	0.0030					
0.1231	0.0960	0.0771	0.0604	0.0470	0.0356	0.0270	0.0214	0.0164	0.0124	0.0096	0.0070	0.0057	0.0041	0.0030					
0.1140	0.0873	0.0691	0.0539	0.0418	0.0342	0.0247	0.0187	0.0144	0.0106	0.0079	0.0058	0.0043	0.0031	0.0026					
0.0997	0.0786	0.0616	0.0479	0.0371	0.0285	0.0217	0.0165	0.0124	0.0093	0.0069	0.0051	0.0037	0.0027	0.0019					
0.0899	0.0700	0.0547	0.0424	0.0327	0.0251	0.0191	0.0144	0.0108	0.0081	0.0060	0.0044	0.0032	0.0023	0.0017					
0.0799	0.0628	0.0483	0.0374	0.0287	0.0220	0.0167	0.0125	0.0094	0.0070	0.0052	0.0038	0.0028	0.0020	0.0014					
0.0697	0.0545	0.0424	0.0327	0.0251	0.0191	0.0145	0.0109	0.0081	0.0060	0.0044	0.0032	0.0024	0.0017	0.0012					
0.0610	0.0477	0.0370	0.0285	0.0218	0.0166	0.0125	0.0094	0.0070	0.0052	0.0038	0.0028	0.0020	0.0014	0.0010					
0.0530	0.0413	0.0328	0.0246	0.0188	0.0143	0.0107	0.0088	0.0060	0.0044	0.0032	0.0024	0.0017	0.0012	0.0009					
0.0457	0.0356	0.0275	0.0211	0.0161	0.0122	0.0092	0.0069	0.0051	0.0037	0.0027	0.0020	0.0014	0.0010	0.0007					
0.0391	0.0304	0.0235	0.0180	0.0137	0.0103	0.0078	0.0058	0.0043	0.0032	0.0023	0.0017	0.0012	0.0009	0.0006					
0.0332	0.0258	0.0199	0.0152	0.0115	0.0087	0.0065	0.0049	0.0036	0.0026	0.0019	0.0014	0.0010	0.0007	0.0005					
0.0279	0.0216	0.0167	0.0127	0.0097	0.0073	0.0055	0.0041	0.0030	0.0022	0.0016	0.0012	0.0008	0.0004						
0.0233	0.0186	0.0139	0.0106	0.0080	0.0066	0.0045	0.0034	0.0025	0.0018	0.0013	0.0010	0.0007	0.0005	0.0003					
0.0192	0.0149	0.0114	0.0087	0.0066	0.0050	0.0037	0.0028	0.0020	0.0015	0.0011	0.0008	0.0006	0.0003						
0.0155	0.0108	0.0074	0.0054	0.0036	0.0024	0.0016	0.0010	0.0007	0.0004	0.0002	0.0001	0.0001	0.0001	0.0002					
0.0128	0.0099	0.0076	0.0058	0.0044	0.0033	0.0024	0.0018	0.0013	0.0010	0.0007	0.0005	0.0004	0.0003	0.0002					
0.0103	0.0080	0.0061	0.0046	0.0035	0.0026	0.0019	0.0015	0.0011	0.0008	0.0006	0.0004	0.0003	0.0002	0.0001					
0.0082	0.0063	0.0049	0.0037	0.0028	0.0021	0.0016	0.0012	0.0008	0.0006	0.0004	0.0003	0.0002	0.0002	0.0001					
0.0065	0.0050	0.0038	0.0029	0.0022	0.0017	0.0011	0.0009	0.0007	0.0005	0.0003	0.0002	0.0001	0.0001	0.0001					
0.0051	0.0039	0.0030	0.0023	0.0017	0.0013	0.0010	0.0007	0.0005	0.0003	0.0002	0.0001	0.0001	0.0001	0.0001					
0.0040	0.0030	0.0023	0.0018	0.0013	0.0010	0.0007	0.0005	0.0004	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001					
0.0030	0.0023	0.0018	0.0014	0.0010	0.0008	0.0006	0.0004	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001	0.0000					



```
[X,Y] = meshgrid(linspace(-3, 3, 100),...
    linspace(-3, 3, 100));

Z = peaks(100);

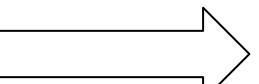
surf(X,Y,Z, "EdgeColor", "none")
```



Adding a colormap: surf()

C

0.7282	0.6822	0.4987	0.4090	0.3322	0.2673	0.2132	0.1684	0.1318	0.1023	0.0786	0.0599	0.0453	0.0339	0.0252
0.6756	0.5648	0.4676	0.3834	0.3114	0.2595	0.1997	0.1578	0.1235	0.0958	0.0737	0.0561	0.0424	0.0317	0.0236
0.6381	0.5265	0.4358	0.3572	0.2901	0.2333	0.1868	0.1469	0.1149	0.0892	0.0685	0.0522	0.0394	0.0295	0.0219
0.5845	0.4882	0.4039	0.3309	0.2686	0.2160	0.1721	0.1359	0.1063	0.0824	0.0634	0.0483	0.0365	0.0273	0.0203
0.5396	0.4508	0.3724	0.3049	0.2474	0.1980	0.1584	0.1250	0.0978	0.0755	0.0582	0.0443	0.0335	0.0251	0.0186
0.4961	0.4137	0.3417	0.2796	0.2267	0.1821	0.1458	0.1143	0.0894	0.0693	0.0532	0.0405	0.0306	0.0229	0.0178
0.4546	0.3778	0.3124	0.2554	0.2068	0.1666	0.1328	0.1041	0.0813	0.0630	0.0473	0.0356	0.0278	0.0208	0.0154
0.4154	0.3454	0.2946	0.2324	0.1868	0.1509	0.1198	0.0943	0.0736	0.0570	0.0437	0.0322	0.0258	0.0187	0.0139
0.3799	0.3145	0.2526	0.1908	0.1539	0.1230	0.0975	0.0766	0.0597	0.0461	0.0353	0.0268	0.0202	0.0150	0.0111
0.3452	0.2858	0.2346	0.1908	0.1539	0.1230	0.0975	0.0766	0.0597	0.0461	0.0353	0.0268	0.0202	0.0150	0.0111
0.3143	0.2595	0.2124	0.1724	0.1388	0.1107	0.0876	0.0687	0.0534	0.0412	0.0315	0.0239	0.0180	0.0134	0.0099
0.2862	0.2356	0.1923	0.1556	0.1249	0.0995	0.0785	0.0615	0.0477	0.0367	0.0281	0.0212	0.0160	0.0119	0.0088
0.2697	0.2138	0.1748	0.1404	0.1124	0.0892	0.0703	0.0549	0.0425	0.0327	0.0249	0.0188	0.0141	0.0105	0.0077
0.2377	0.1942	0.1574	0.1266	0.1010	0.0880	0.0628	0.0490	0.0378	0.0299	0.0221	0.0167	0.0125	0.0093	0.0068
0.2168	0.1764	0.1424	0.1142	0.0908	0.0717	0.0561	0.0436	0.0338	0.0257	0.0195	0.0147	0.0110	0.0081	0.0060
0.1978	0.1602	0.1289	0.1029	0.0816	0.0642	0.0501	0.0388	0.0298	0.0228	0.0172	0.0130	0.0097	0.0071	0.0052
0.1804	0.1455	0.1168	0.0927	0.0732	0.0574	0.0447	0.0345	0.0264	0.0281	0.0152	0.0114	0.0085	0.0063	0.0046
0.1645	0.1321	0.1054	0.0835	0.0657	0.0513	0.0398	0.0306	0.0234	0.0178	0.0134	0.0100	0.0074	0.0055	0.0048
0.1497	0.1193	0.0952	0.0751	0.0589	0.0458	0.0354	0.0272	0.0207	0.0157	0.0118	0.0088	0.0065	0.0048	0.0035
0.1360	0.1083	0.0858	0.0674	0.0527	0.0408	0.0315	0.0241	0.0183	0.0141	0.0103	0.0074	0.0051	0.0038	0.0030
0.1231	0.0971	0.0751	0.0570	0.0440	0.0336	0.0270	0.0211	0.0154	0.0124	0.0097	0.0071	0.0050	0.0036	0.0026
0.1110	0.0879	0.0669	0.0539	0.0418	0.0347	0.0287	0.0214	0.0166	0.0109	0.0058	0.0043	0.0031	0.0023	0.0019
0.0997	0.0786	0.0616	0.0479	0.0371	0.0285	0.0217	0.0165	0.0124	0.0093	0.0059	0.0051	0.0037	0.0021	0.0019
0.0899	0.0700	0.0547	0.0424	0.0327	0.0251	0.0191	0.0144	0.0108	0.0081	0.0060	0.0044	0.0032	0.0023	0.0017
0.0799	0.0628	0.0483	0.0374	0.0287	0.0220	0.0167	0.0125	0.0094	0.0070	0.0052	0.0038	0.0028	0.0020	0.0014
0.0697	0.0545	0.0424	0.0327	0.0251	0.0191	0.0145	0.0109	0.0081	0.0060	0.0044	0.0032	0.0024	0.0017	0.0012
0.0610	0.0477	0.0370	0.0285	0.0218	0.0166	0.0125	0.0094	0.0070	0.0052	0.0038	0.0028	0.0020	0.0014	0.0010
0.0530	0.0413	0.0328	0.0246	0.0188	0.0143	0.0107	0.0088	0.0060	0.0044	0.0032	0.0024	0.0017	0.0012	0.0009
0.0457	0.0356	0.0275	0.0211	0.0161	0.0122	0.0092	0.0069	0.0051	0.0037	0.0027	0.0020	0.0014	0.0010	0.0007
0.0391	0.0304	0.0235	0.0188	0.0137	0.0103	0.0078	0.0058	0.0043	0.0032	0.0023	0.0017	0.0012	0.0009	0.0006
0.0332	0.0258	0.0199	0.0152	0.0115	0.0087	0.0065	0.0049	0.0036	0.0026	0.0019	0.0014	0.0010	0.0007	0.0005
0.0279	0.0216	0.0167	0.0127	0.0097	0.0073	0.0055	0.0041	0.0030	0.0022	0.0016	0.0012	0.0008	0.0004	0.0003
0.0233	0.0186	0.0139	0.0106	0.0080	0.0066	0.0045	0.0034	0.0025	0.0018	0.0013	0.0010	0.0007	0.0005	0.0003
0.0192	0.0149	0.0114	0.0087	0.0066	0.0050	0.0037	0.0028	0.0020	0.0015	0.0011	0.0008	0.0006	0.0004	0.0003
0.0152	0.0109	0.0074	0.0054	0.0039	0.0027	0.0020	0.0015	0.0012	0.0009	0.0006	0.0004	0.0003	0.0002	0.0002
0.0128	0.0099	0.0076	0.0058	0.0044	0.0033	0.0024	0.0018	0.0013	0.0010	0.0007	0.0005	0.0004	0.0003	0.0002
0.0103	0.0080	0.0061	0.0046	0.0035	0.0026	0.0020	0.0015	0.0011	0.0008	0.0006	0.0004	0.0003	0.0002	0.0001
0.0082	0.0063	0.0049	0.0037	0.0028	0.0021	0.0016	0.0012	0.0008	0.0006	0.0004	0.0003	0.0002	0.0002	0.0001
0.0065	0.0050	0.0038	0.0029	0.0022	0.0017	0.0012	0.0009	0.0007	0.0005	0.0004	0.0003	0.0002	0.0001	0.0001
0.0051	0.0039	0.0030	0.0023	0.0017	0.0013	0.0010	0.0007	0.0005	0.0003	0.0002	0.0001	0.0001	0.0001	0.0001
0.0040	0.0030	0.0023	0.0018	0.0013	0.0010	0.0007	0.0005	0.0004	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001
0.0030	0.0023	0.0018	0.0014	0.0010	0.0008	0.0006	0.0004	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001	0.0000

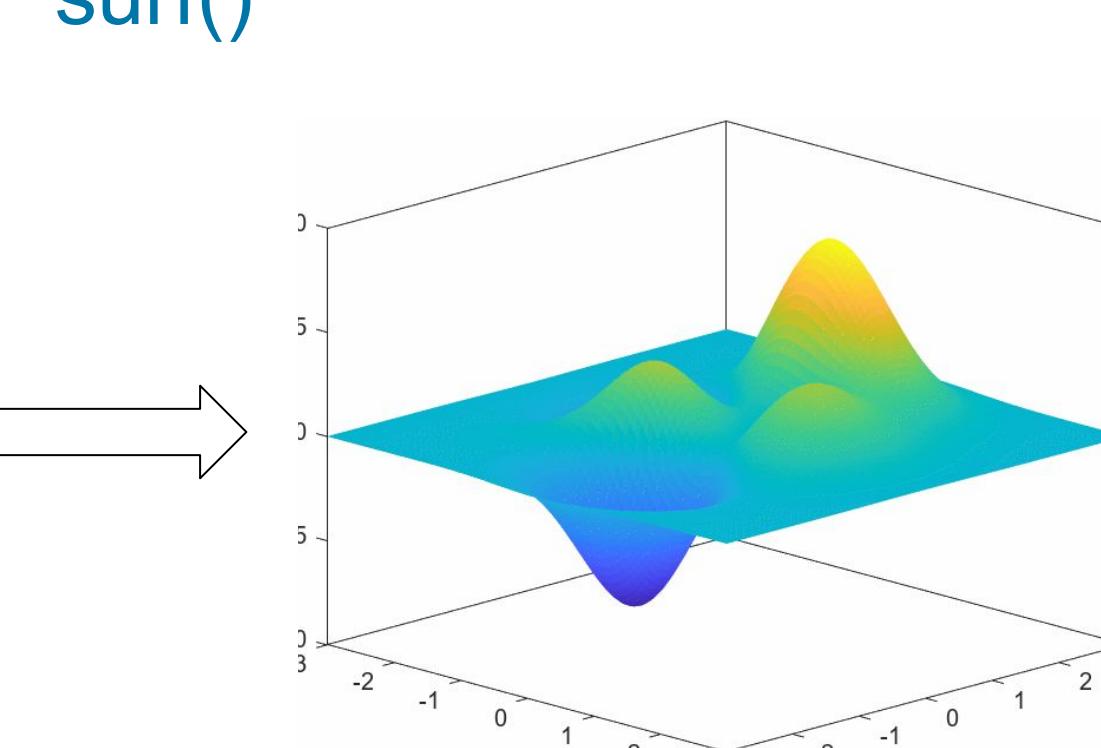


```
[X,Y] = meshgrid(linspace(-3, 3, 100),...
    linspace(-3, 3, 100));

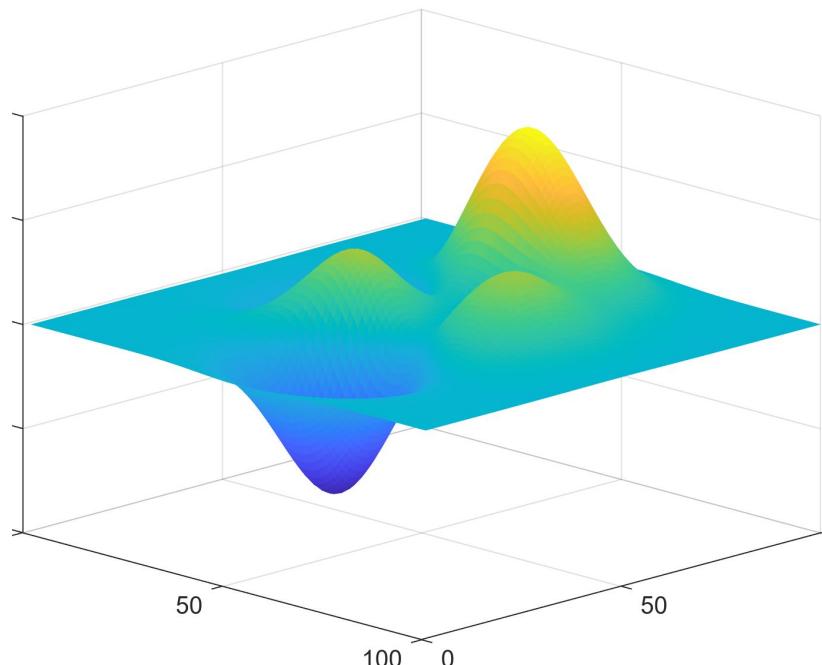
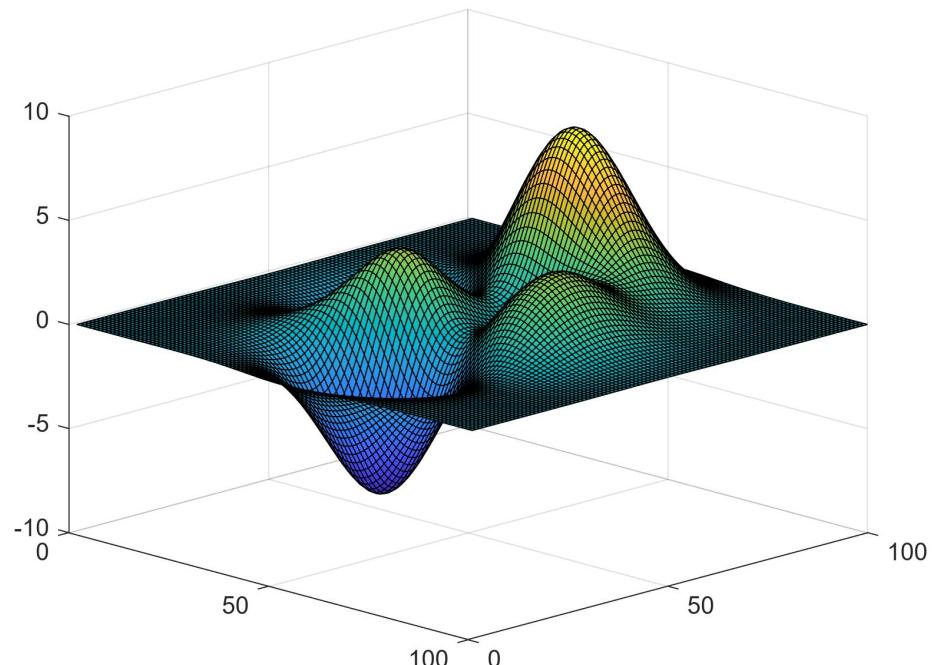
Z = peaks(100);

C = sin(2*pi*X);

surf(X,Y,Z, C, "EdgeColor", "none")
```



Customization: surf()



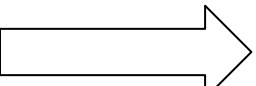
Line Style
"_"
"--"
"-.."
"none"

```
[X,Y] = meshgrid(linspace(-3, 3, 100),...  
linspace(-3, 3, 100));  
  
Z = peaks(100);  
  
surf(X,Y,Z, "LineStyle","none")
```

Visualizing matrix-data: contour3()

Z

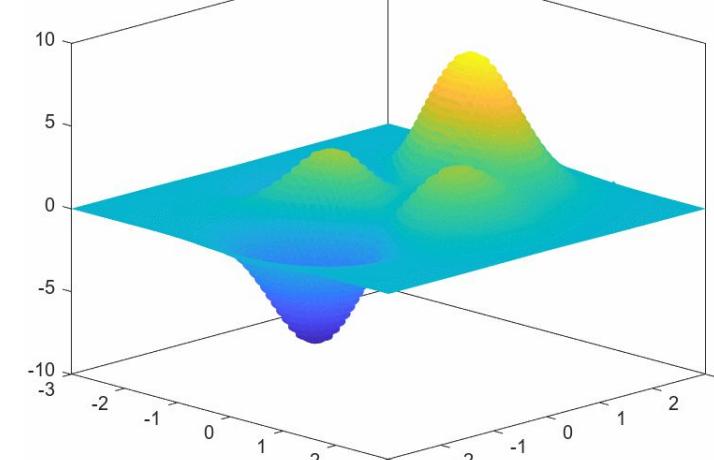
	0.7282	0.6822	0.4987	0.4090	0.3322	0.2673	0.2132	0.1684	0.1318	0.1023	0.0786	0.0599	0.0453	0.0339	0.0252
0.6756	0.5648	0.4676	0.3834	0.3114	0.2595	0.1997	0.1578	0.1235	0.0958	0.0737	0.0561	0.0424	0.0317	0.0236	
0.6381	0.5265	0.4358	0.3572	0.2901	0.2333	0.1868	0.1469	0.1149	0.0892	0.0685	0.0522	0.0394	0.0295	0.0219	
0.5845	0.4882	0.4039	0.3309	0.2686	0.2160	0.1721	0.1359	0.1063	0.0824	0.0634	0.0483	0.0365	0.0273	0.0203	
0.5396	0.4508	0.3724	0.3049	0.2474	0.1988	0.1584	0.1258	0.0978	0.0755	0.0582	0.0443	0.0335	0.0251	0.0186	
0.4961	0.4137	0.3417	0.2796	0.2267	0.1821	0.1458	0.1143	0.0894	0.0693	0.0532	0.0405	0.0306	0.0229	0.0178	
0.4546	0.3748	0.3124	0.2554	0.2068	0.1666	0.1328	0.1041	0.0813	0.0630	0.0473	0.0356	0.0278	0.0208	0.0154	
0.4154	0.3454	0.2946	0.2324	0.1868	0.1509	0.1198	0.0943	0.0736	0.0570	0.0437	0.0322	0.0258	0.0187	0.0139	
0.3770	0.3145	0.2526	0.1908	0.1539	0.1230	0.0975	0.0766	0.0597	0.0461	0.0353	0.0268	0.0202	0.0150	0.0111	
0.3452	0.2858	0.2346	0.1808	0.1539	0.1230	0.0975	0.0766	0.0597	0.0461	0.0353	0.0268	0.0202	0.0150	0.0111	
0.3143	0.2595	0.2124	0.1724	0.1388	0.1107	0.0876	0.0687	0.0534	0.0412	0.0315	0.0239	0.0180	0.0134	0.0099	
0.2862	0.2356	0.1923	0.1556	0.1249	0.0995	0.0785	0.0615	0.0477	0.0437	0.0281	0.0212	0.0160	0.0119	0.0088	
0.2697	0.2138	0.1748	0.1404	0.1124	0.0892	0.0703	0.0549	0.0425	0.0327	0.0249	0.0188	0.0141	0.0105	0.0077	
0.2377	0.1942	0.1574	0.1266	0.1010	0.0880	0.0628	0.0490	0.0378	0.0299	0.0221	0.0167	0.0125	0.0093	0.0068	
0.2168	0.1764	0.1424	0.1142	0.0908	0.0717	0.0561	0.0436	0.0338	0.0257	0.0195	0.0147	0.0110	0.0081	0.0068	
0.1978	0.1682	0.1289	0.1029	0.0816	0.0642	0.0501	0.0388	0.0298	0.0228	0.0172	0.0130	0.0097	0.0071	0.0052	
0.1804	0.1455	0.1186	0.0927	0.0732	0.0574	0.0447	0.0345	0.0264	0.0208	0.0152	0.0114	0.0085	0.0063	0.0046	
0.1645	0.1321	0.1054	0.0835	0.0657	0.0513	0.0398	0.0306	0.0234	0.0178	0.0134	0.0100	0.0074	0.0055	0.0048	
0.1497	0.1193	0.0952	0.0751	0.0589	0.0458	0.0354	0.0272	0.0207	0.0157	0.0118	0.0088	0.0065	0.0048	0.0035	
0.1360	0.1083	0.0858	0.0674	0.0527	0.0408	0.0315	0.0241	0.0183	0.0130	0.0083	0.0057	0.0041	0.0030	0.0026	
0.1231	0.0961	0.0771	0.0604	0.0476	0.0356	0.0270	0.0214	0.0160	0.0114	0.0070	0.0057	0.0041	0.0030	0.0026	
0.1104	0.0873	0.0691	0.0539	0.0418	0.0322	0.0247	0.0187	0.0144	0.0106	0.0079	0.0058	0.0043	0.0031	0.0023	
0.0997	0.0786	0.0616	0.0479	0.0371	0.0285	0.0217	0.0165	0.0124	0.0093	0.0069	0.0051	0.0037	0.0021	0.0019	
0.0899	0.0700	0.0547	0.0424	0.0327	0.0251	0.0191	0.0144	0.0108	0.0081	0.0060	0.0044	0.0032	0.0023	0.0017	
0.0799	0.0628	0.0483	0.0374	0.0287	0.0220	0.0167	0.0125	0.0094	0.0070	0.0052	0.0038	0.0028	0.0020	0.0014	
0.0697	0.0545	0.0424	0.0327	0.0251	0.0191	0.0145	0.0109	0.0081	0.0060	0.0044	0.0032	0.0024	0.0017	0.0012	
0.0610	0.0477	0.0370	0.0285	0.0218	0.0166	0.0125	0.0094	0.0070	0.0052	0.0038	0.0028	0.0020	0.0014	0.0010	
0.0530	0.0413	0.0328	0.0246	0.0188	0.0143	0.0107	0.0088	0.0060	0.0044	0.0032	0.0024	0.0017	0.0012	0.0009	
0.0457	0.0356	0.0275	0.0211	0.0161	0.0122	0.0092	0.0069	0.0051	0.0037	0.0027	0.0020	0.0014	0.0010	0.0007	
0.0391	0.0304	0.0235	0.0188	0.0137	0.0103	0.0078	0.0058	0.0043	0.0032	0.0023	0.0017	0.0012	0.0009	0.0006	
0.0332	0.0258	0.0199	0.0152	0.0115	0.0087	0.0065	0.0049	0.0036	0.0026	0.0019	0.0014	0.0010	0.0007	0.0005	
0.0279	0.0216	0.0167	0.0127	0.0097	0.0073	0.0055	0.0041	0.0030	0.0022	0.0016	0.0012	0.0008	0.0004	0.0003	
0.0233	0.0186	0.0139	0.0106	0.0080	0.0066	0.0045	0.0034	0.0025	0.0018	0.0013	0.0010	0.0007	0.0005	0.0003	
0.0192	0.0149	0.0114	0.0087	0.0066	0.0050	0.0037	0.0028	0.0020	0.0015	0.0011	0.0008	0.0006	0.0004	0.0003	
0.0155	0.0109	0.0074	0.0052	0.0037	0.0026	0.0017	0.0010	0.0007	0.0005	0.0003	0.0002	0.0001	0.0001	0.0001	
0.0128	0.0099	0.0076	0.0058	0.0044	0.0033	0.0024	0.0018	0.0013	0.0010	0.0007	0.0005	0.0004	0.0003	0.0002	
0.0103	0.0080	0.0061	0.0046	0.0035	0.0026	0.0019	0.0015	0.0011	0.0008	0.0006	0.0004	0.0003	0.0002	0.0001	
0.0082	0.0063	0.0049	0.0037	0.0028	0.0021	0.0016	0.0012	0.0008	0.0006	0.0004	0.0003	0.0002	0.0002	0.0001	
0.0065	0.0050	0.0038	0.0029	0.0022	0.0017	0.0010	0.0009	0.0007	0.0005	0.0004	0.0003	0.0002	0.0001	0.0001	
0.0051	0.0039	0.0030	0.0023	0.0017	0.0013	0.0010	0.0007	0.0005	0.0003	0.0002	0.0001	0.0001	0.0001	0.0001	
0.0040	0.0030	0.0023	0.0018	0.0013	0.0010	0.0007	0.0005	0.0004	0.0003	0.0002	0.0002	0.0001	0.0001	0.0000	
0.0030	0.0023	0.0018	0.0014	0.0010	0.0008	0.0006	0.0004	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001	0.0000	



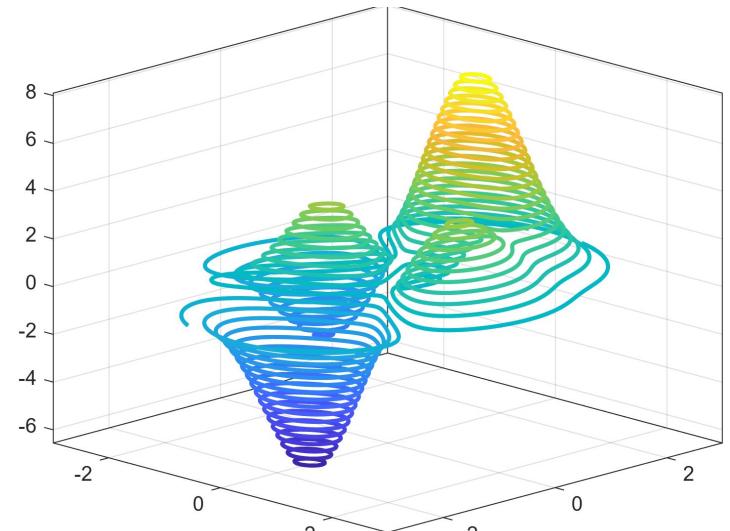
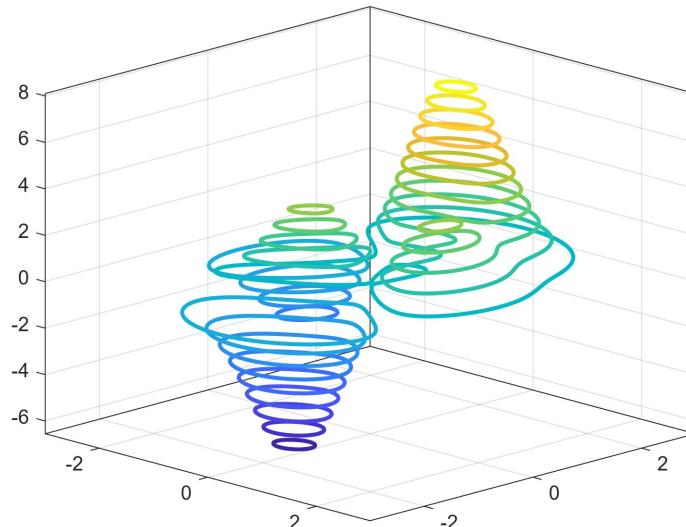
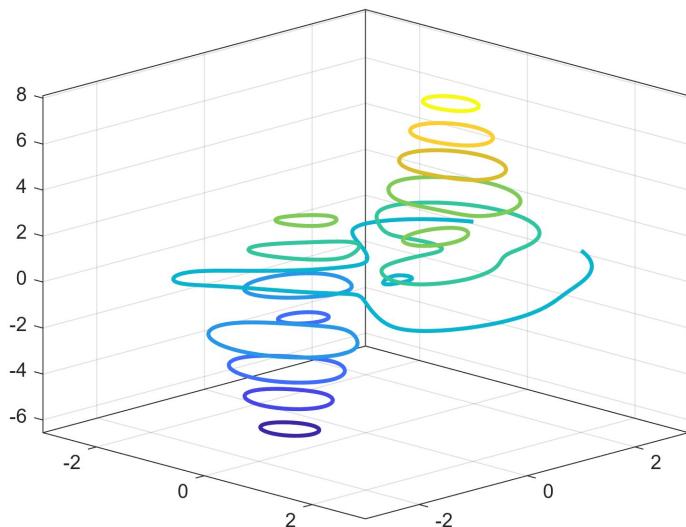
```
[X,Y] = meshgrid(linspace(-3, 3, 100),...
    linspace(-3, 3, 100));

Z = peaks(100);

contour3(X,Y,Z)
```



Customization: contour3()



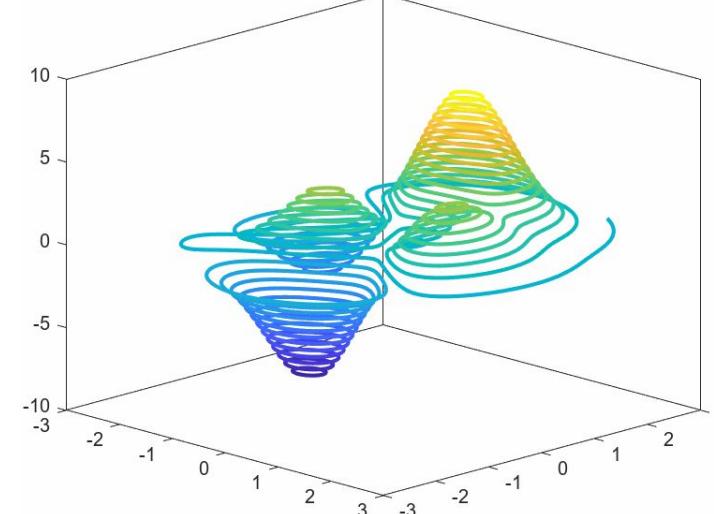
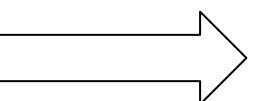
```
[X,Y] = meshgrid(linspace(-3, 3, 100),...  
    linspace(-3, 3, 100));  
  
Z = peaks(100);  
  
lines = 40;  
  
contour3(X,Y,Z, lines)
```

Visualizing matrix-data: contour()

```

0.7282 0.6822 0.4987 0.4099 0.3322 0.2673 0.2132 0.1684 0.1318 0.1023 0.0786 0.0599 0.0453 0.0339 0.0252
0.6756 0.5648 0.4676 0.3834 0.3114 0.2595 0.1997 0.1578 0.1235 0.0958 0.0737 0.0561 0.0424 0.0317 0.0236
0.6381 0.5265 0.4358 0.3572 0.2981 0.2333 0.1868 0.1469 0.1149 0.0892 0.0685 0.0522 0.0394 0.0295 0.0219
0.5845 0.4882 0.4039 0.3309 0.2686 0.2160 0.1721 0.1359 0.1063 0.0824 0.0634 0.0483 0.0365 0.0273 0.0203
0.5396 0.4586 0.3724 0.3049 0.2474 0.1988 0.1584 0.1258 0.0978 0.0753 0.0582 0.0443 0.0335 0.0251 0.0186
0.4961 0.4137 0.3417 0.2796 0.2267 0.1821 0.1458 0.1143 0.0894 0.0693 0.0532 0.0405 0.0306 0.0229 0.0178
0.4546 0.3981 0.3124 0.2554 0.2068 0.1666 0.1328 0.1041 0.0813 0.0630 0.0473 0.0356 0.0278 0.0208 0.0154
0.4154 0.3524 0.2946 0.2324 0.1868 0.1508 0.1198 0.0943 0.0736 0.0571 0.0437 0.0322 0.0232 0.0187 0.0139
0.3799 0.3145 0.2526 0.1908 0.1539 0.1239 0.1095 0.0766 0.0597 0.0461 0.0353 0.0268 0.0202 0.0159 0.0111
0.3452 0.2858 0.2246 0.1724 0.1388 0.1107 0.0876 0.0687 0.0534 0.0412 0.0315 0.0239 0.0180 0.0134 0.0099
0.3143 0.2595 0.2124 0.1724 0.1388 0.1107 0.0876 0.0687 0.0534 0.0412 0.0315 0.0239 0.0180 0.0134 0.0099
0.2862 0.2356 0.1923 0.1556 0.1249 0.0995 0.0785 0.0615 0.0477 0.0367 0.0281 0.0212 0.0160 0.0119 0.0088
0.2697 0.2138 0.1748 0.1404 0.1124 0.0892 0.0703 0.0549 0.0425 0.0327 0.0249 0.0188 0.0141 0.0105 0.0077
0.2377 0.1942 0.1574 0.1266 0.1010 0.0880 0.0628 0.0490 0.0378 0.0299 0.0211 0.0167 0.0125 0.0093 0.0068
0.2168 0.1764 0.1424 0.1142 0.0908 0.0717 0.0561 0.0436 0.0336 0.0257 0.0195 0.0147 0.0110 0.0081 0.0068
0.1978 0.1682 0.1289 0.1029 0.0816 0.0642 0.0581 0.0388 0.0298 0.0228 0.0172 0.0130 0.0097 0.0071 0.0052
0.1804 0.1455 0.1116 0.0897 0.0732 0.0574 0.0447 0.0345 0.0264 0.0208 0.0152 0.0114 0.0085 0.0063 0.0046
0.1645 0.1321 0.1054 0.0835 0.0657 0.0513 0.0398 0.0306 0.0234 0.0178 0.0134 0.0100 0.0074 0.0055 0.0048
0.1497 0.1193 0.0952 0.0751 0.0589 0.0458 0.0354 0.0272 0.0207 0.0157 0.0118 0.0088 0.0065 0.0048 0.0035
0.1360 0.1083 0.0858 0.0674 0.0527 0.0408 0.0315 0.0246 0.0183 0.0130 0.0093 0.0067 0.0041 0.0030 0.0030
0.1231 0.0971 0.0771 0.0604 0.0460 0.0356 0.0270 0.0214 0.0154 0.0114 0.0076 0.0057 0.0037 0.0026 0.0026
0.1110 0.0873 0.0691 0.0539 0.0418 0.0342 0.0247 0.0187 0.0141 0.0106 0.0079 0.0058 0.0043 0.0021 0.0023
0.0997 0.0766 0.0616 0.0479 0.0371 0.0385 0.0217 0.0165 0.0124 0.0093 0.0069 0.0051 0.0037 0.0021 0.0019
0.0899 0.0700 0.0547 0.0424 0.0327 0.0251 0.0191 0.0144 0.0108 0.0081 0.0060 0.0044 0.0032 0.0023 0.0017
0.0799 0.0628 0.0483 0.0374 0.0287 0.0220 0.0167 0.0125 0.0094 0.0070 0.0052 0.0038 0.0028 0.0020 0.0014
0.0697 0.0545 0.0424 0.0327 0.0251 0.0191 0.0145 0.0109 0.0081 0.0060 0.0044 0.0032 0.0024 0.0017 0.0012
0.0610 0.0477 0.0370 0.0285 0.0218 0.0166 0.0125 0.0094 0.0070 0.0052 0.0038 0.0028 0.0020 0.0014 0.0018
0.0530 0.0413 0.0328 0.0246 0.0188 0.0143 0.0107 0.0088 0.0060 0.0044 0.0032 0.0024 0.0017 0.0012 0.0009
0.0457 0.0356 0.0275 0.0211 0.0161 0.0122 0.0092 0.0069 0.0051 0.0037 0.0027 0.0020 0.0014 0.0010 0.0007
0.0391 0.0304 0.0235 0.0188 0.0137 0.0103 0.0078 0.0058 0.0043 0.0032 0.0023 0.0017 0.0012 0.0009 0.0006
0.0332 0.0258 0.0199 0.0152 0.0115 0.0087 0.0065 0.0049 0.0036 0.0026 0.0019 0.0014 0.0010 0.0007 0.0005
0.0279 0.0216 0.0167 0.0127 0.0097 0.0073 0.0055 0.0041 0.0030 0.0022 0.0016 0.0012 0.0008 0.0006 0.0004
0.0233 0.0186 0.0139 0.0106 0.0080 0.0066 0.0045 0.0034 0.0025 0.0018 0.0013 0.0010 0.0007 0.0005 0.0003
0.0192 0.0149 0.0114 0.0087 0.0066 0.0050 0.0037 0.0028 0.0020 0.0015 0.0011 0.0008 0.0006 0.0004 0.0003
0.0159 0.0116 0.0084 0.0060 0.0042 0.0030 0.0022 0.0016 0.0011 0.0007 0.0004 0.0003 0.0002 0.0001 0.0001
0.0128 0.0099 0.0076 0.0058 0.0044 0.0033 0.0024 0.0018 0.0013 0.0010 0.0007 0.0005 0.0004 0.0003 0.0002
0.0103 0.0080 0.0061 0.0046 0.0035 0.0026 0.0020 0.0015 0.0011 0.0008 0.0006 0.0004 0.0003 0.0002 0.0001
0.0082 0.0063 0.0049 0.0037 0.0028 0.0021 0.0016 0.0012 0.0008 0.0006 0.0004 0.0003 0.0002 0.0002 0.0001
0.0065 0.0050 0.0038 0.0029 0.0022 0.0017 0.0012 0.0009 0.0007 0.0005 0.0004 0.0003 0.0002 0.0001 0.0001
0.0051 0.0039 0.0030 0.0023 0.0017 0.0013 0.0010 0.0007 0.0005 0.0003 0.0002 0.0001 0.0001 0.0001 0.0001
0.0040 0.0030 0.0023 0.0018 0.0013 0.0010 0.0007 0.0005 0.0004 0.0003 0.0002 0.0001 0.0001 0.0001 0.0001
0.0030 0.0023 0.0018 0.0014 0.0010 0.0008 0.0006 0.0004 0.0003 0.0002 0.0002 0.0001 0.0001 0.0001 0.0000

```



```

[X,Y] = meshgrid(linspace(-3, 3, 100),...
                  linspace(-3, 3, 100));

Z = peaks(100);

lines = 40;

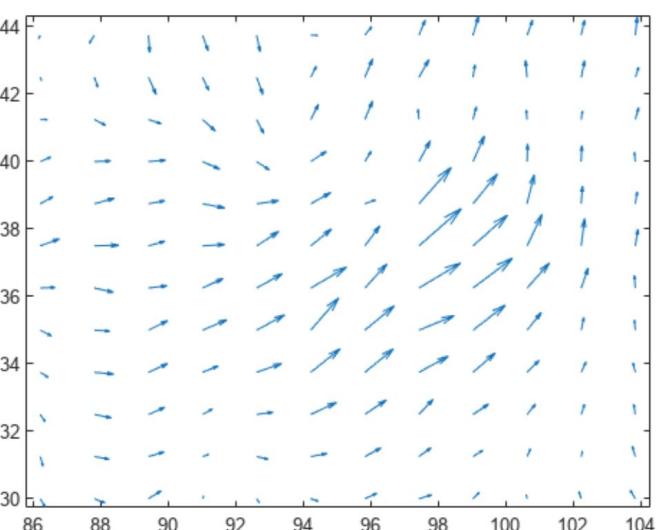
contour(X,Y,Z, lines)

```

Visualizing matrix-data: quiver()

Arrow-position

Arrow-direction



```
[X,Y] = meshgrid(linspace(-3, 3, 100),  
                  linspace(-3, 3, 100));
```

~~[U,V] = magnetic field(X,Y)~~

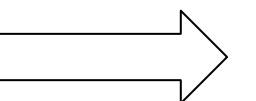
```
quiver(X,Y,U,V)
```

```
function [dxdt, dydt] = magnetic_field(x,y)
%%...
end
```

Visualizing matrix-data: quiver3()

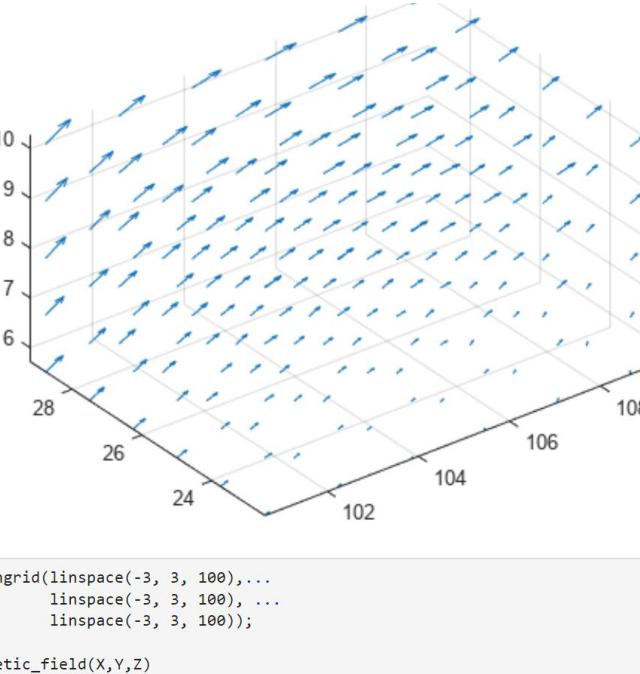
Arrow-position

0.7282	0.4902	0.4987	0.4909	0.3322	0.2673	0.2332	0.1884	0.1138	0.1023	0.0785	0.0599	0.0443	0.0339	0.0252
0.5396	0.3555	0.4588	0.4572	0.2981	0.2333	0.1868	0.1469	0.1149	0.0992	0.0685	0.0522	0.0394	0.0295	0.0193
0.4546	0.3452	0.4097	0.4099	0.3322	0.2773	0.2332	0.1694	0.1138	0.1023	0.0786	0.0599	0.0443	0.0339	0.0252
0.3785	0.3454	0.4082	0.4099	0.3322	0.2773	0.2332	0.1694	0.1138	0.1023	0.0786	0.0599	0.0443	0.0339	0.0252
0.3412	0.4046	0.4576	0.4581	0.3314	0.2686	0.1997	0.1578	0.1255	0.1023	0.0786	0.0599	0.0443	0.0339	0.0252
0.2862	0.4154	0.4842	0.4839	0.3389	0.2658	0.2158	0.1709	0.1372	0.1255	0.1023	0.0786	0.0599	0.0443	0.0339
0.2377	0.3452	0.4093	0.4137	0.3417	0.2796	0.2287	0.1821	0.1459	0.1243	0.0984	0.0693	0.0532	0.0408	0.0278
0.1979	0.2862	0.4154	0.3451	0.2646	0.2323	0.1848	0.1567	0.1159	0.0943	0.0776	0.0578	0.0437	0.0312	0.0251
0.1645	0.2377	0.3789	0.3145	0.2288	0.1793	0.1354	0.1082	0.0852	0.0664	0.0513	0.0439	0.0323	0.0215	0.0151
0.1304	0.1978	0.3143	0.2591	0.2124	0.1738	0.1368	0.1107	0.0879	0.0667	0.0518	0.0412	0.0315	0.0219	0.0109
0.1113	0.1645	0.2697	0.2118	0.1740	0.1484	0.1124	0.0892	0.0703	0.0649	0.0445	0.0327	0.0249	0.0186	0.0077
0.0899	0.1258	0.2156	0.1764	0.1424	0.1142	0.0896	0.0717	0.0551	0.0436	0.0336	0.0227	0.0159	0.0147	0.0068
0.0897	0.1118	0.2394	0.1455	0.1256	0.0927	0.0712	0.0574	0.0443	0.0345	0.0254	0.0152	0.0145	0.0085	0.0066
0.0510	0.0698	0.1467	0.1159	0.0952	0.0751	0.0589	0.0458	0.0354	0.0272	0.0207	0.0157	0.0118	0.0089	0.0015
0.0445	0.0709	0.1211	0.0977	0.0771	0.0648	0.0467	0.0343	0.0279	0.0213	0.0161	0.0122	0.0089	0.0016	0.0026
0.0332	0.0618	0.0907	0.0771	0.0648	0.0467	0.0343	0.0279	0.0213	0.0161	0.0122	0.0089	0.0016	0.0023	0.0013
0.0213	0.0457	0.0798	0.0625	0.0483	0.0374	0.0287	0.0219	0.0157	0.0125	0.0094	0.0071	0.0052	0.0028	0.0014
0.0158	0.0322	0.0616	0.0477	0.0379	0.0218	0.0166	0.0094	0.0070	0.0065	0.0050	0.0032	0.0020	0.0007	0.0009
0.0103	0.0233	0.0463	0.0354	0.0275	0.0211	0.0161	0.0122	0.0092	0.0065	0.0051	0.0037	0.0020	0.0007	0.0009
0.0084	0.0203	0.0479	0.0329	0.0217	0.0097	0.0073	0.0041	0.0031	0.0022	0.0016	0.0012	0.0005	0.0003	0.0004
0.0055	0.0156	0.0310	0.0214	0.0167	0.0065	0.0049	0.0017	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0002
0.0051	0.0126	0.0209	0.0176	0.0125	0.0064	0.0045	0.0015	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0002
0.0048	0.01218	0.0189	0.0095	0.0055	0.0025	0.0018	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0001	0.0002
0.0045	0.0108	0.0189	0.0083	0.0053	0.0025	0.0018	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0001	0.0002
0.0043	0.0103	0.0189	0.0083	0.0053	0.0025	0.0018	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0001	0.0002
0.0040	0.0083	0.0189	0.0083	0.0053	0.0025	0.0018	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0001	0.0002
0.0038	0.0081	0.0189	0.0083	0.0053	0.0025	0.0018	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0001	0.0002



Arrow-direction

0.7282	0.4902	0.4987	0.4909	0.3322	0.2673	0.2332	0.1884	0.1138	0.1023	0.0785	0.0599	0.0443	0.0339	0.0252
0.5396	0.3555	0.4588	0.4572	0.2981	0.2333	0.1868	0.1469	0.1149	0.0992	0.0685	0.0522	0.0394	0.0295	0.0193
0.4546	0.3452	0.4097	0.4099	0.3322	0.2773	0.2332	0.1694	0.1138	0.1023	0.0786	0.0599	0.0443	0.0339	0.0252
0.3785	0.3454	0.4082	0.4099	0.3322	0.2773	0.2332	0.1694	0.1138	0.1023	0.0786	0.0599	0.0443	0.0339	0.0252
0.3412	0.4046	0.4576	0.4581	0.3314	0.2686	0.1997	0.1578	0.1255	0.1023	0.0786	0.0599	0.0443	0.0339	0.0252
0.2862	0.4154	0.4842	0.4839	0.3389	0.2658	0.2158	0.1709	0.1372	0.1255	0.1023	0.0786	0.0599	0.0443	0.0339
0.2377	0.3453	0.4564	0.3789	0.3124	0.2558	0.2088	0.1646	0.1341	0.1038	0.0813	0.0630	0.0443	0.0328	0.0154
0.1979	0.2862	0.4154	0.3451	0.2646	0.2323	0.1848	0.1567	0.1159	0.0943	0.0776	0.0578	0.0437	0.0312	0.0159
0.1645	0.2377	0.3789	0.3145	0.2288	0.1793	0.1354	0.1082	0.0852	0.0664	0.0513	0.0439	0.0323	0.0215	0.0151
0.1304	0.1978	0.3143	0.2591	0.2124	0.1738	0.1368	0.1107	0.0879	0.0667	0.0518	0.0412	0.0315	0.0219	0.0109
0.1113	0.1645	0.2697	0.2118	0.1740	0.1484	0.1124	0.0892	0.0703	0.0649	0.0445	0.0327	0.0227	0.0159	0.0087
0.0899	0.1258	0.2156	0.1764	0.1424	0.1142	0.0896	0.0717	0.0551	0.0436	0.0336	0.0227	0.0159	0.0147	0.0068
0.0897	0.1118	0.2394	0.1455	0.1256	0.0927	0.0712	0.0574	0.0443	0.0345	0.0254	0.0152	0.0145	0.0085	0.0066
0.0510	0.0698	0.1467	0.1159	0.0952	0.0751	0.0589	0.0458	0.0354	0.0272	0.0207	0.0157	0.0118	0.0089	0.0015
0.0445	0.0709	0.1211	0.0977	0.0771	0.0648	0.0467	0.0343	0.0279	0.0213	0.0161	0.0122	0.0089	0.0016	0.0026
0.0332	0.0618	0.0907	0.0771	0.0648	0.0467	0.0343	0.0279	0.0213	0.0161	0.0122	0.0089	0.0016	0.0023	0.0013
0.0213	0.0457	0.0798	0.0625	0.0483	0.0374	0.0287	0.0219	0.0157	0.0125	0.0094	0.0071	0.0052	0.0028	0.0014
0.0158	0.0322	0.0616	0.0477	0.0379	0.0218	0.0166	0.0094	0.0070	0.0065	0.0050	0.0032	0.0020	0.0007	0.0009
0.0103	0.0233	0.0463	0.0354	0.0275	0.0211	0.0161	0.0122	0.0092	0.0065	0.0051	0.0037	0.0020	0.0007	0.0009
0.0084	0.0203	0.0479	0.0329	0.0217	0.0161	0.0096	0.0073	0.0051	0.0038	0.0022	0.0016	0.0012	0.0005	0.0004
0.0055	0.0156	0.0314	0.0214	0.0167	0.0065	0.0049	0.0017	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0002
0.0051	0.0126	0.0209	0.0176	0.0125	0.0064	0.0045	0.0015	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0002
0.0048	0.01218	0.0189	0.0095	0.0055	0.0025	0.0018	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0001	0.0002
0.0045	0.0108	0.0189	0.0083	0.0053	0.0025	0.0018	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0001	0.0002
0.0043	0.0103	0.0189	0.0083	0.0053	0.0025	0.0018	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0001	0.0002
0.0040	0.0083	0.0189	0.0083	0.0053	0.0025	0.0018	0.0011	0.0008	0.0006	0.0004	0.0002	0.0001	0.0001	0.0002



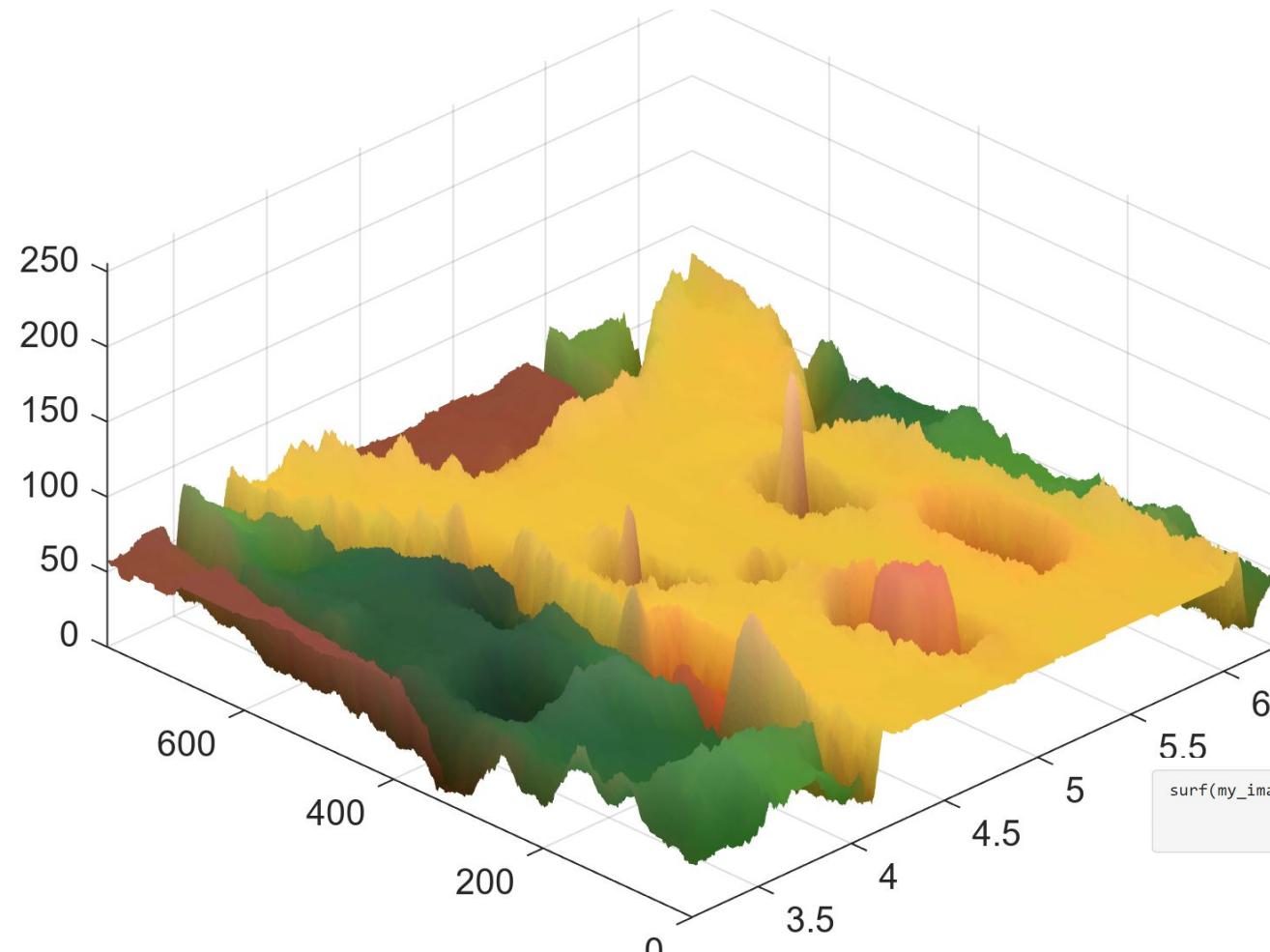
Images are matrices too

```
my_image = imread("pikachu.jpg");  
imshow(my_image)
```



```
red_matrix = my_image(:, :, 1);  
blue_matrix = my_image(:, :, 2);  
green_matrix = my_image(:, :, 3);
```

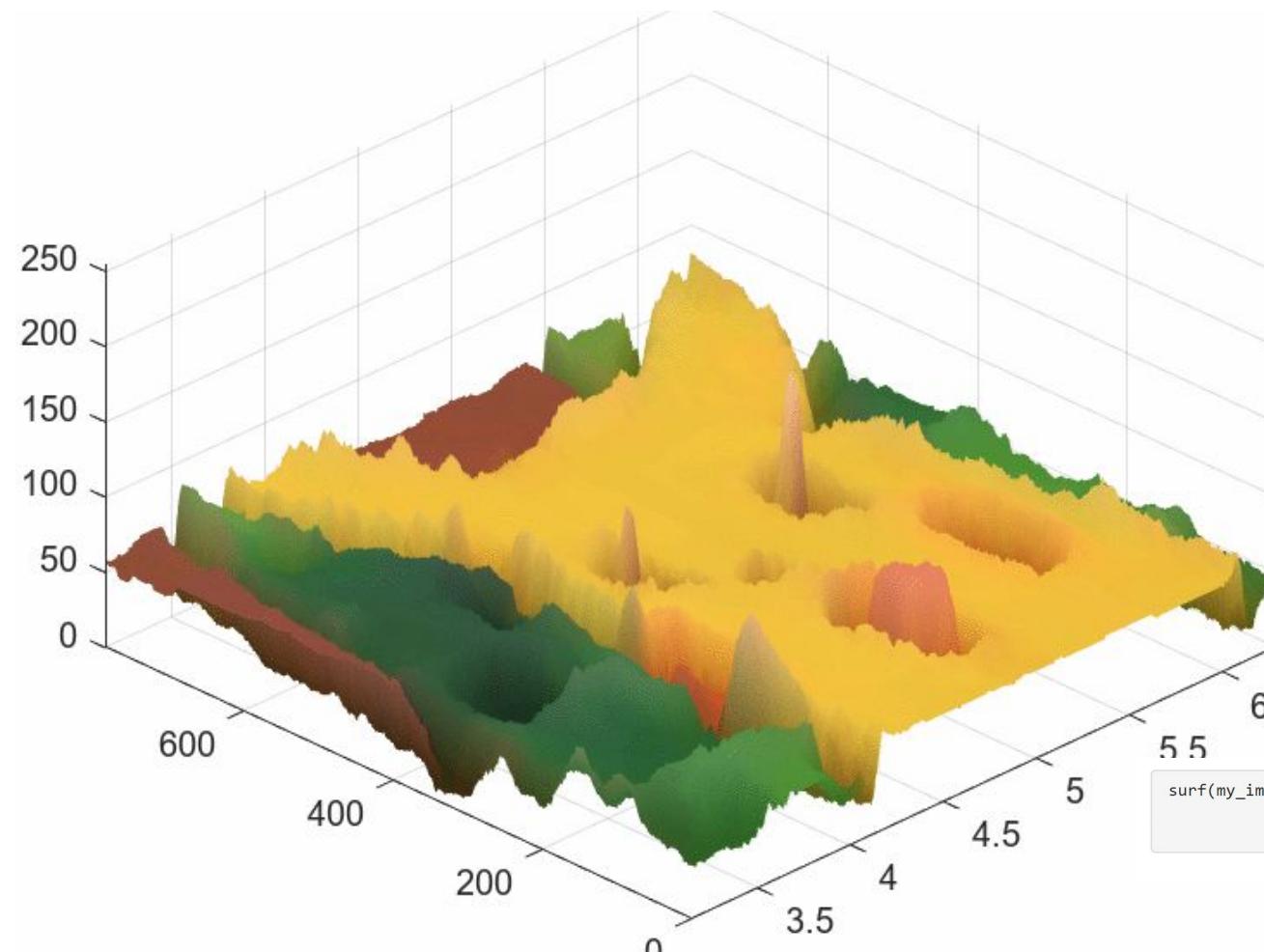
So... 3D pikachu??



```
surf(my_image(:,:,3), 'FaceColor','texturemap',...
    'EdgeColor','none',...
    'Cdata',my_image)
```

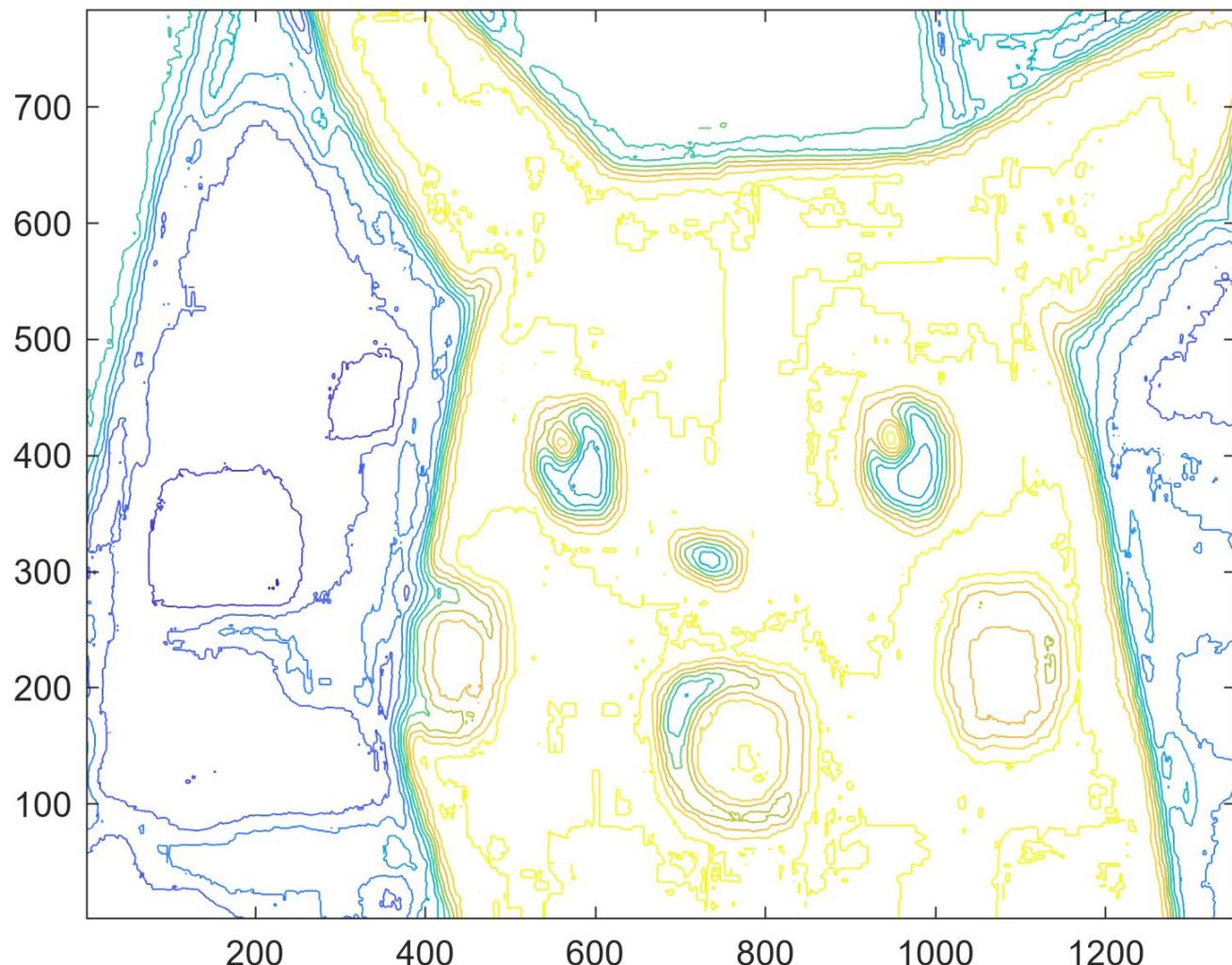
(if you have to...)

So... 3D pikachu??



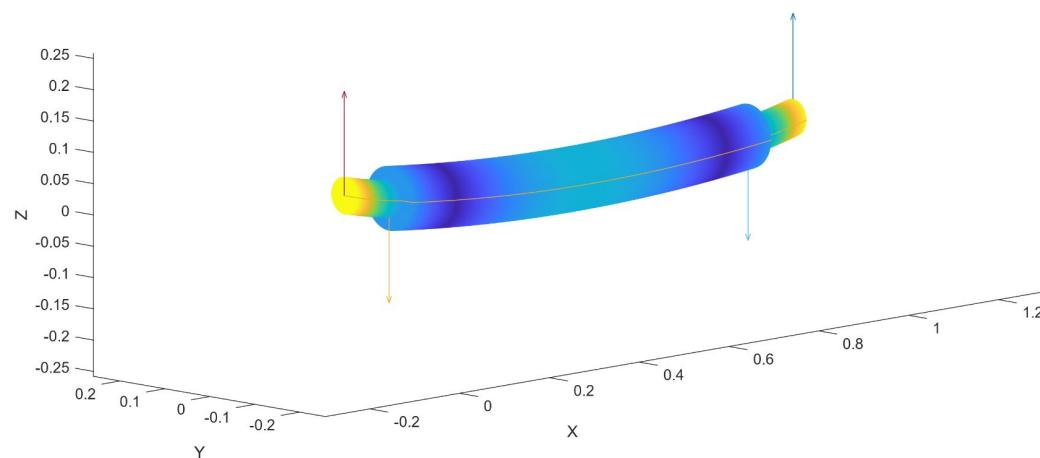
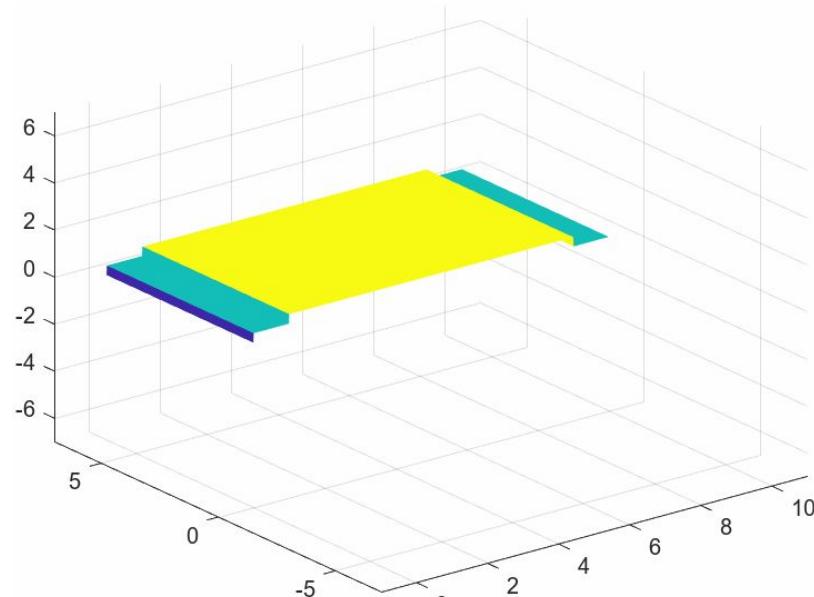
```
surf(my_image(:,:,3), 'FaceColor','texturemap',...
    'EdgeColor','none',...
    'Cdata',my_image)
```

(if you have to...)



```
my_image = imread("pikachu.jpg");
contour(my_image(:,:,1), 40);
```

Plots work best when used together:



```
radius = @(l) ... some radius-function for our axel;
length_vec = 0:0.01:10;
theta_vec = linspace(0, 2*pi, 100);

[length_mat, theta_mat] = meshgrid(length_vec, theta_vec);
radius_mat = radius(length_mat);

[y,z] = pol2cart(theta_mat, radius_mat);
x = length_mat;
```

Force-vectors:

```
my_force = [1;0;0];
my_force_position = 1;
```

rendering:

```
myax = axes();
myax.NextPlot = "add";
surf(myax, x,y,z);
quiver3(myax, my_force_position, 0, 0, ...
my_force(1), my_force(2), my_force(3));
% Etc...
% All the other stuff that you might want to layer on top
myax.NextPlot = "replacechildren";
```

For more complex geometries, use the STLread library!

Installed



STL File Reader
Version 1.2.0.0 (1.6 MB) by Eric Johnson
STLREAD imports geometry from a binary stereolithography (STL) file into MATLAB.
 Function

★★★★★ (75)
37.6K Downloads 
Updated 20 Jul 2011
[View License](#)

[Open Folder](#) [Manage](#)

[Overview](#) [Functions](#) [Examples](#) [Version History](#) [Reviews \(75\)](#) [Discussions \(21\)](#)

In addition to the STLREAD import function, this submission also includes a small demo that loads an STL model of a human femur bone.

`FV = STLREAD(FILENAME)` imports triangular faces from the binary STL file indicated by `FILENAME`, and returns the patch struct `FV`, with fields '`faces`' and '`vertices`'.

`[F,V] = STLREAD(FILENAME)` returns the faces `F` and vertices `V` separately.

`[F,V,N] = STLREAD(FILENAME)` also returns the face normal vectors.

The faces and vertices are arranged in the format used by the `PATCH` plot object.

Cite As

Eric Johnson (2023). STL File Reader (<https://www.mathworks.com/matlabcentral/fileexchange/22409-stl-file-reader>), MATLAB Central File Exchange. Retrieved December 17, 2023.

MATLAB Release Compatibility

Created with R2008b
Compatible with any release

Platform Compatibility

Windows macOS Linux

Categories

[MATLAB > Data Import and Analysis](#)

[Physical Modeling > Simscape](#)
[Multibody > Model Import](#)

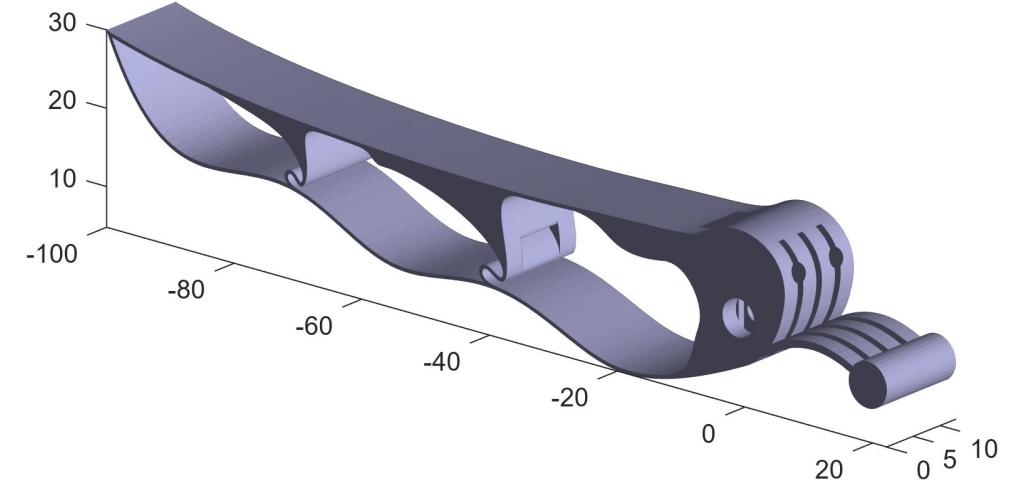
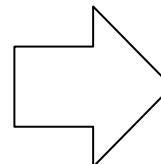
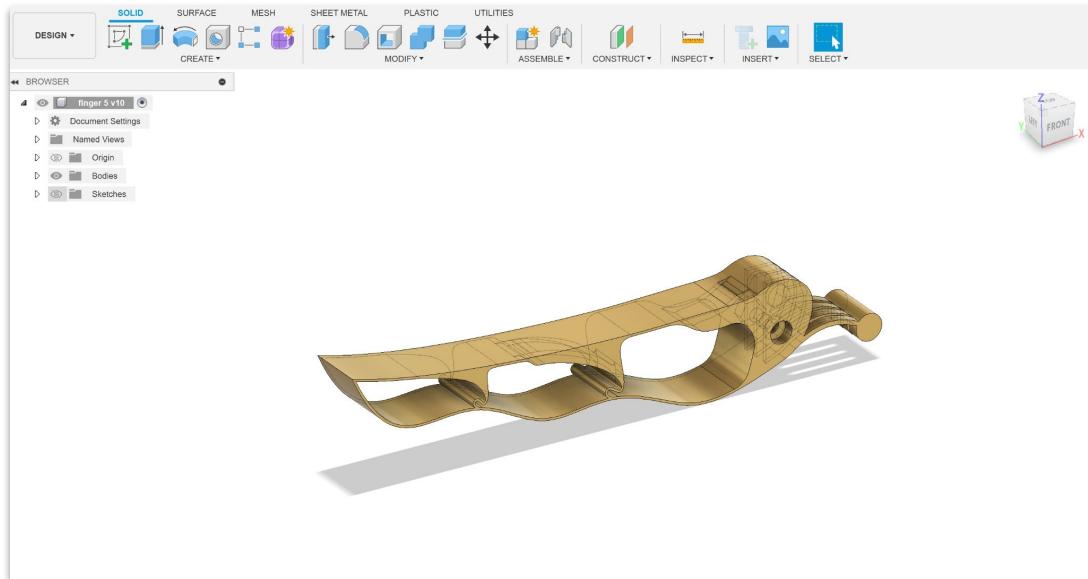
[MATLAB > Data Import and Analysis >](#)
[Data Import and Export > Standard File Formats > STL \(Stereolithography\)](#)

Tags

Add Tags

[cad](#) [data export](#)

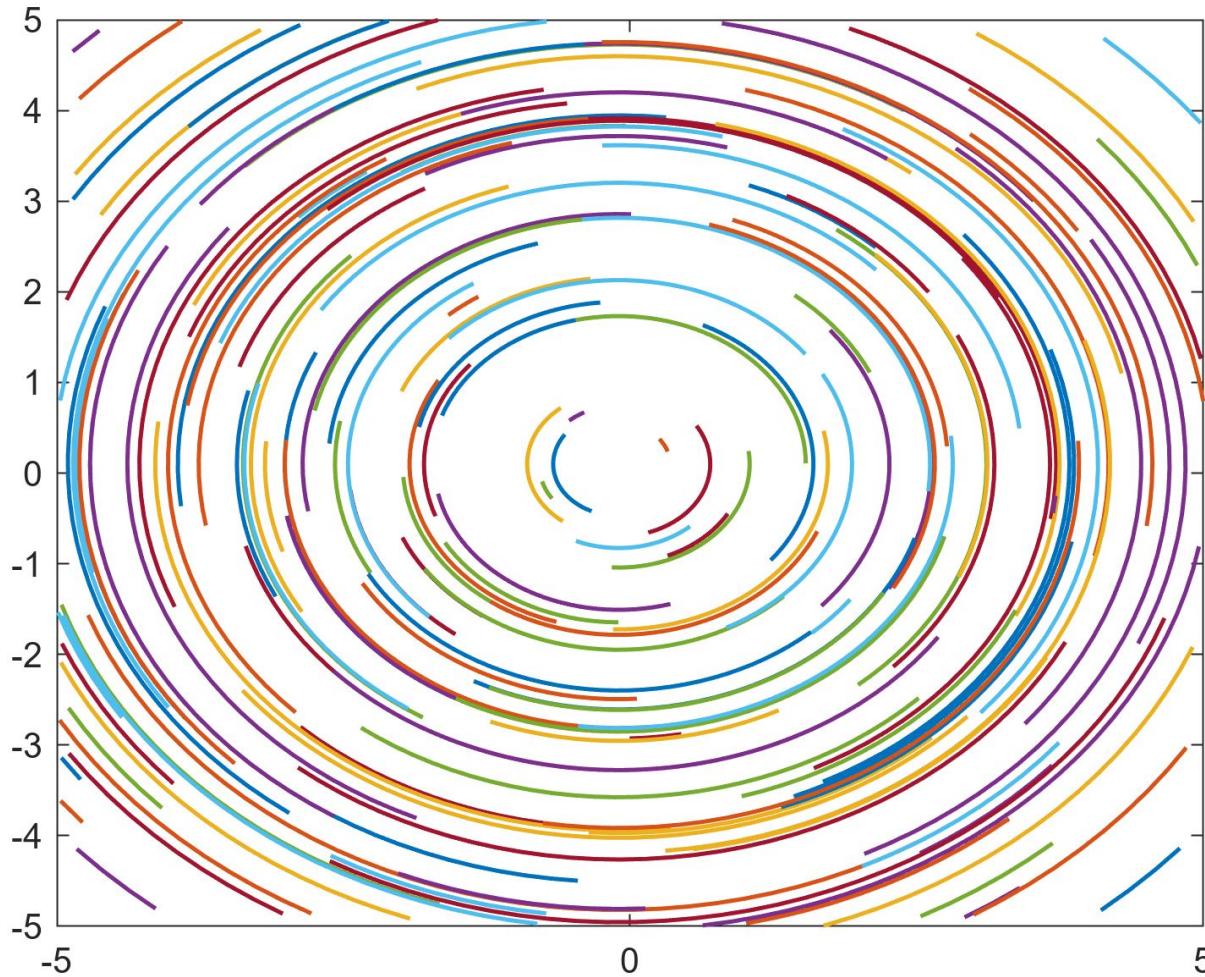
Reading STL-files into MATLAB:



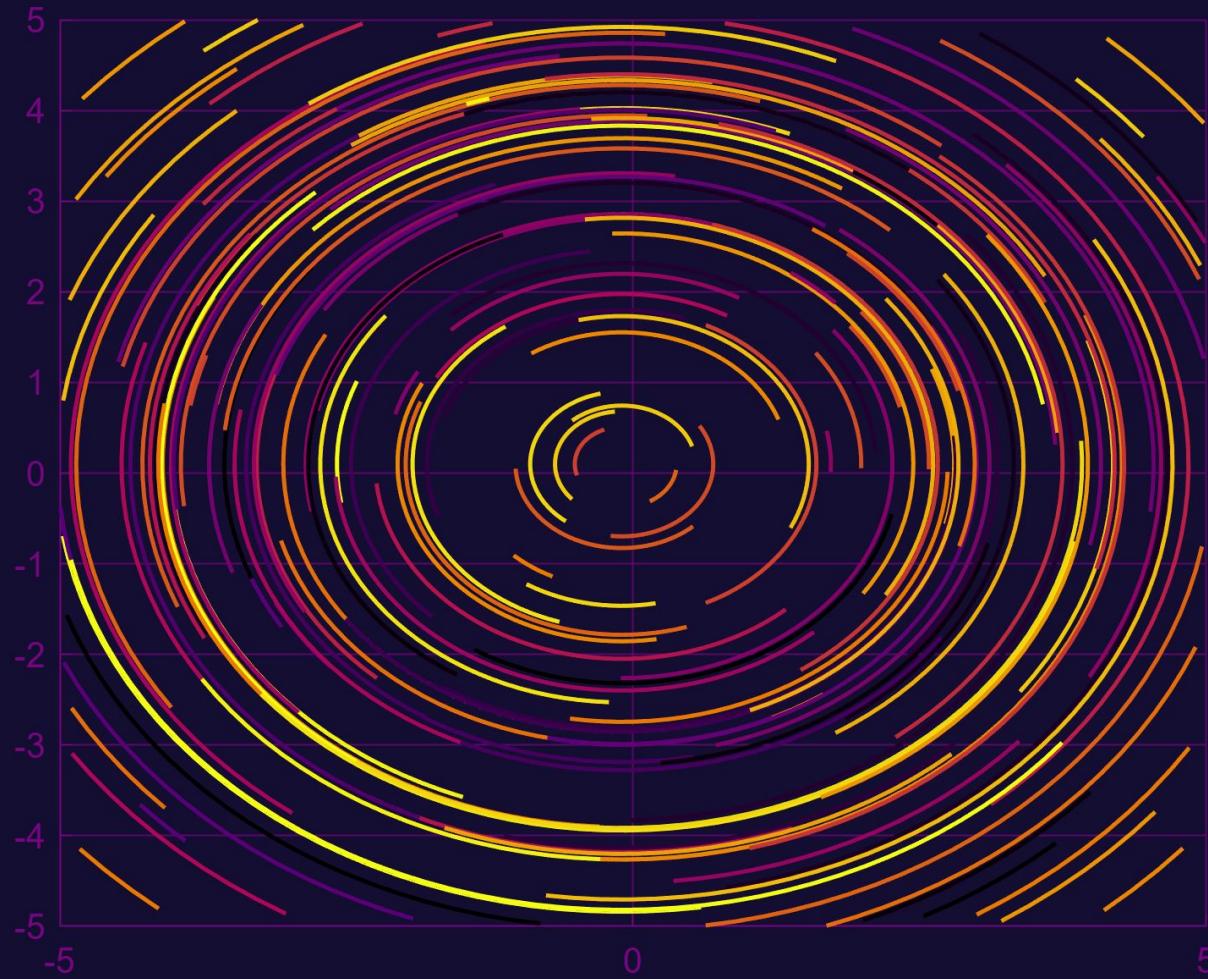
```
my_patch = stlread("finger.stl");
myax = axes();
axis("equal")

patch(myax, my_patch, 'FaceColor', [0.8 0.8 1.0], ...
      'EdgeColor', 'none', ...
      'FaceLighting', 'gouraud', ...
      'AmbientStrength', 0.15);
camlight(70,45)
material('dull');
view(45, 45)
axis("tight")
drawnow
```

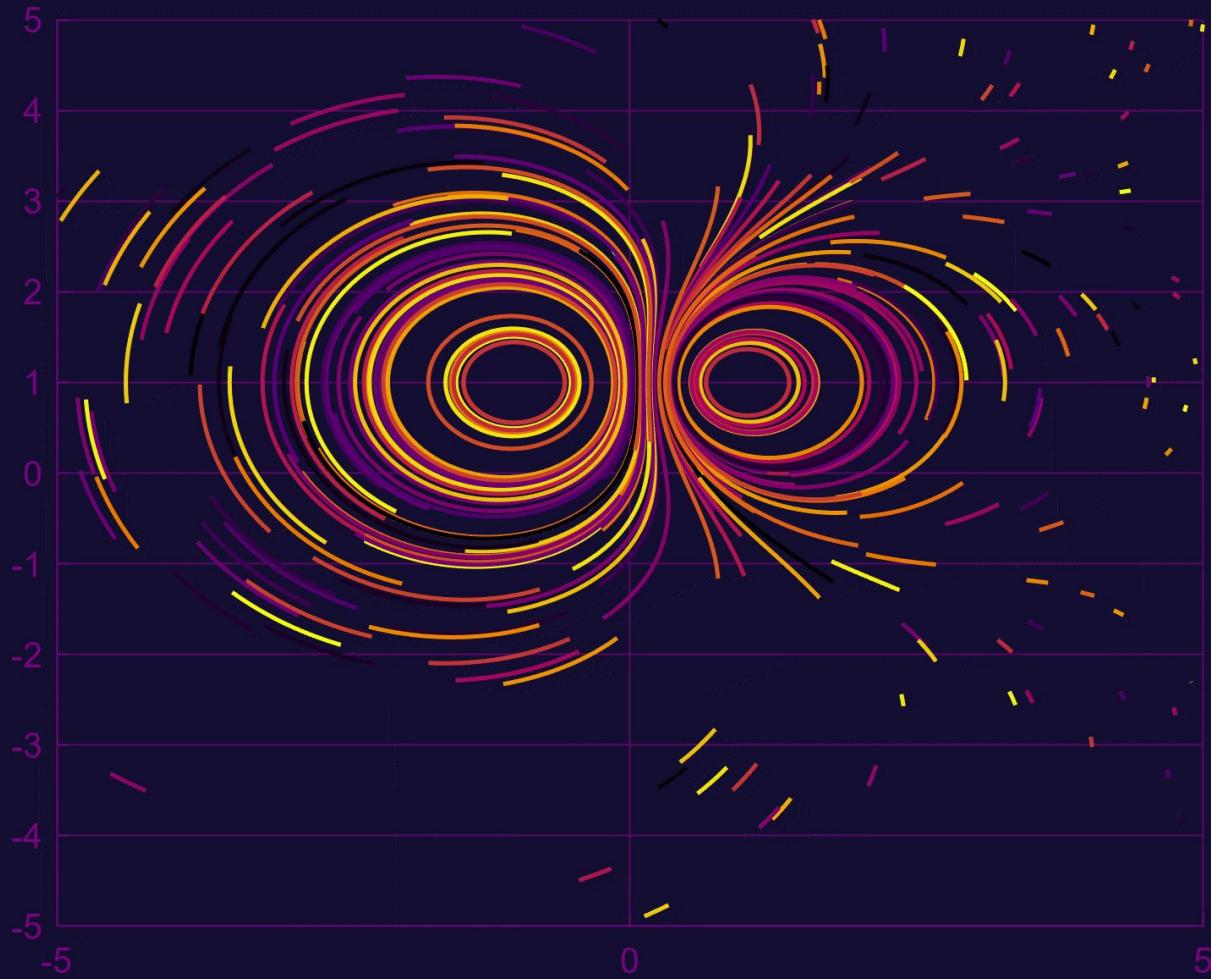
These plots are cool and all, but my eyes hurt...



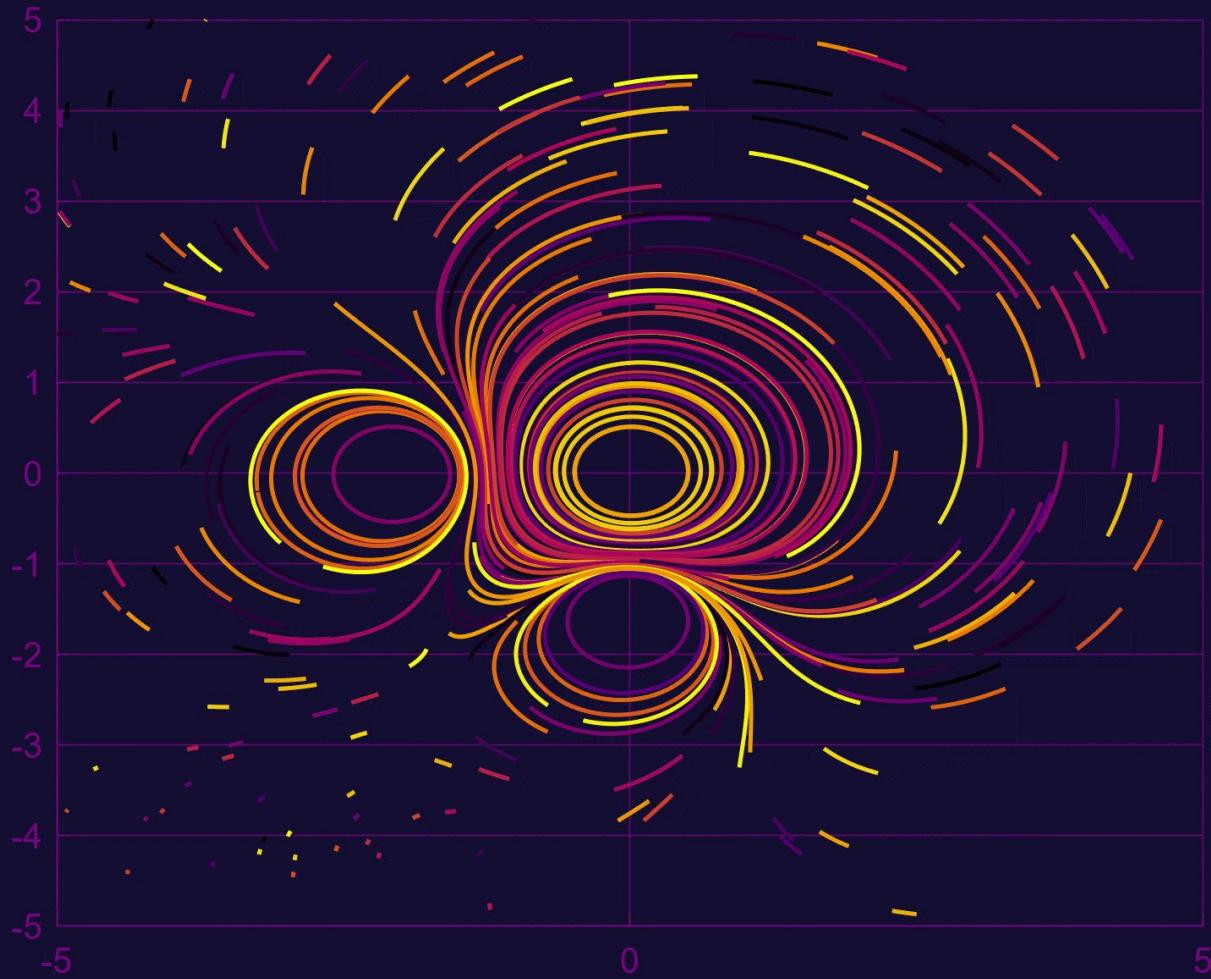
Join the dark side.



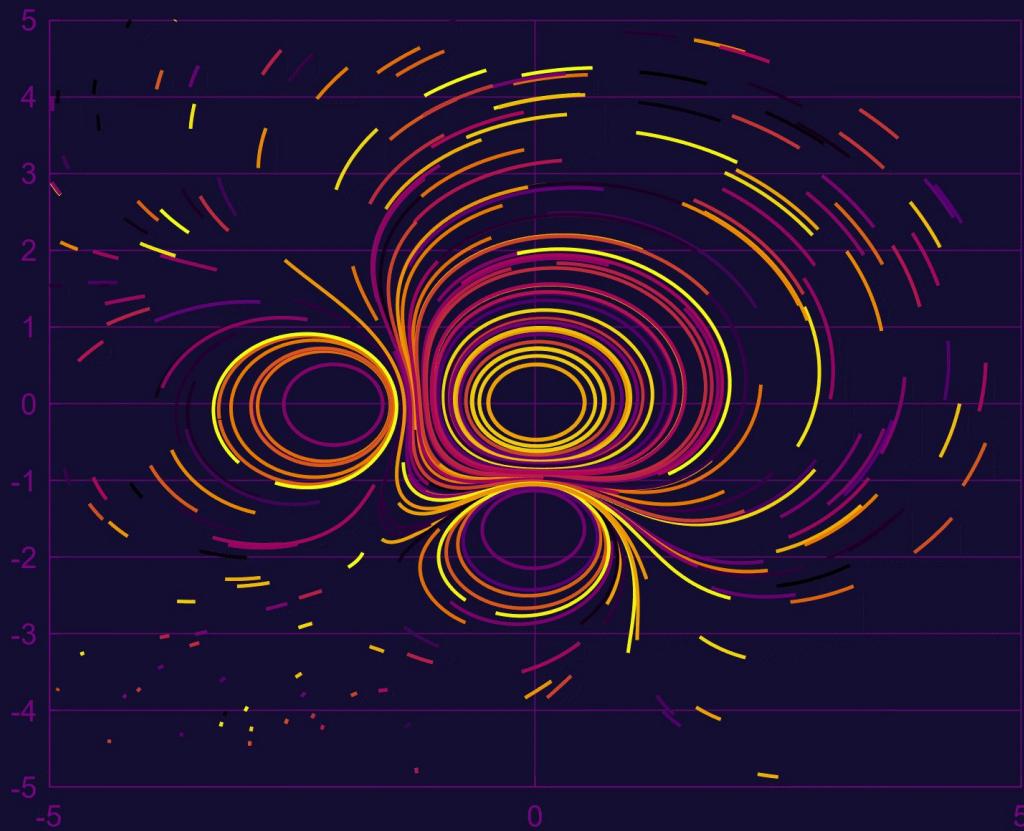
fireflies()



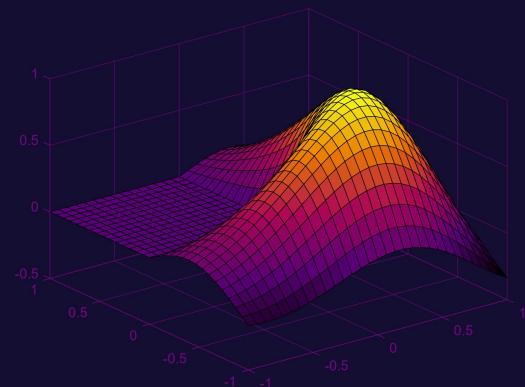
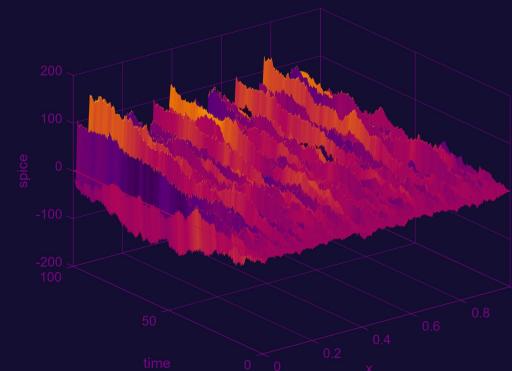
fireflies()



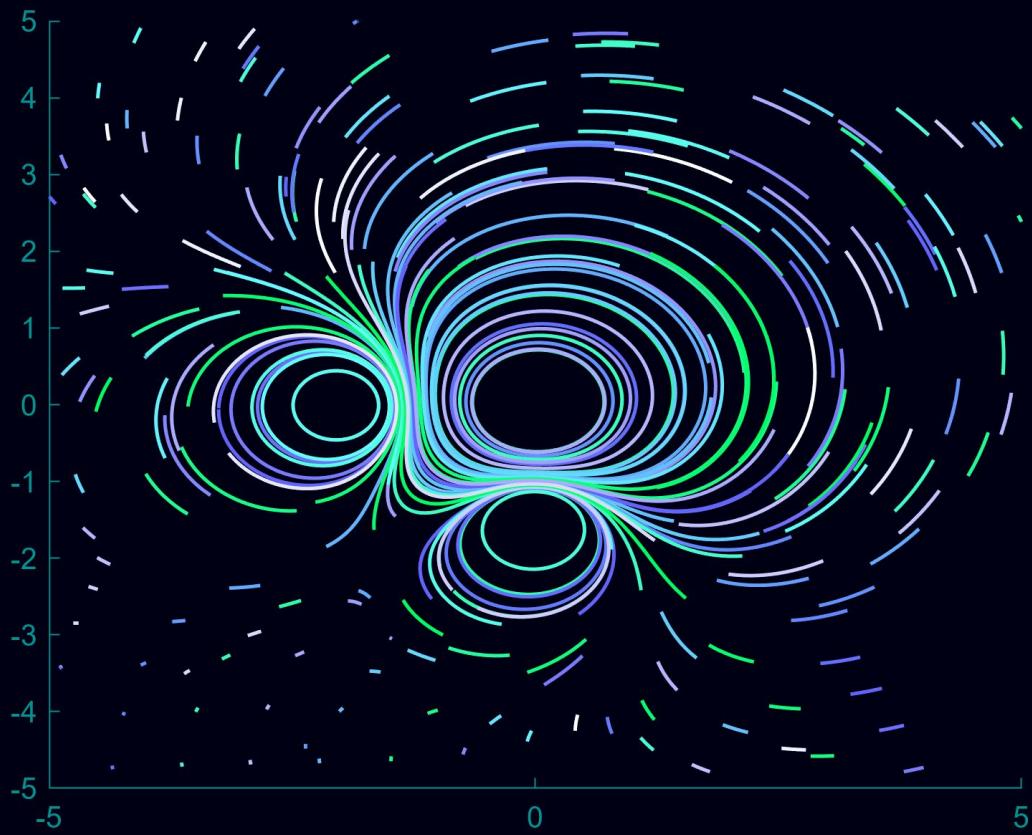
fireflies()



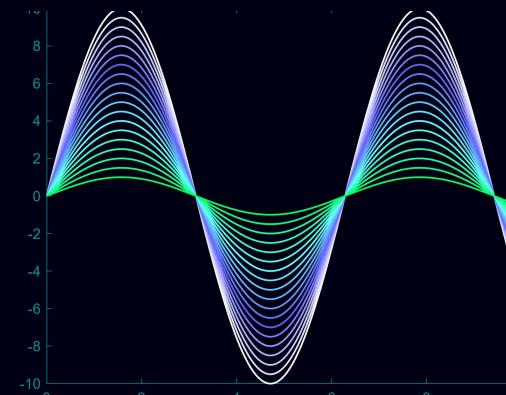
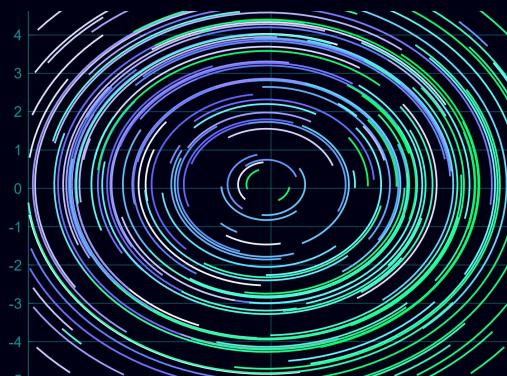
```
my_field = particle_field(); % Not part of standard library  
  
myaxes = axes();  
fireflies();  
  
animate_field("ParticleField",  
             "Function",  
             "Axes",  
             my_field, ...  
             @magnetic_field, ...  
             myaxes); % Not part of standard library
```



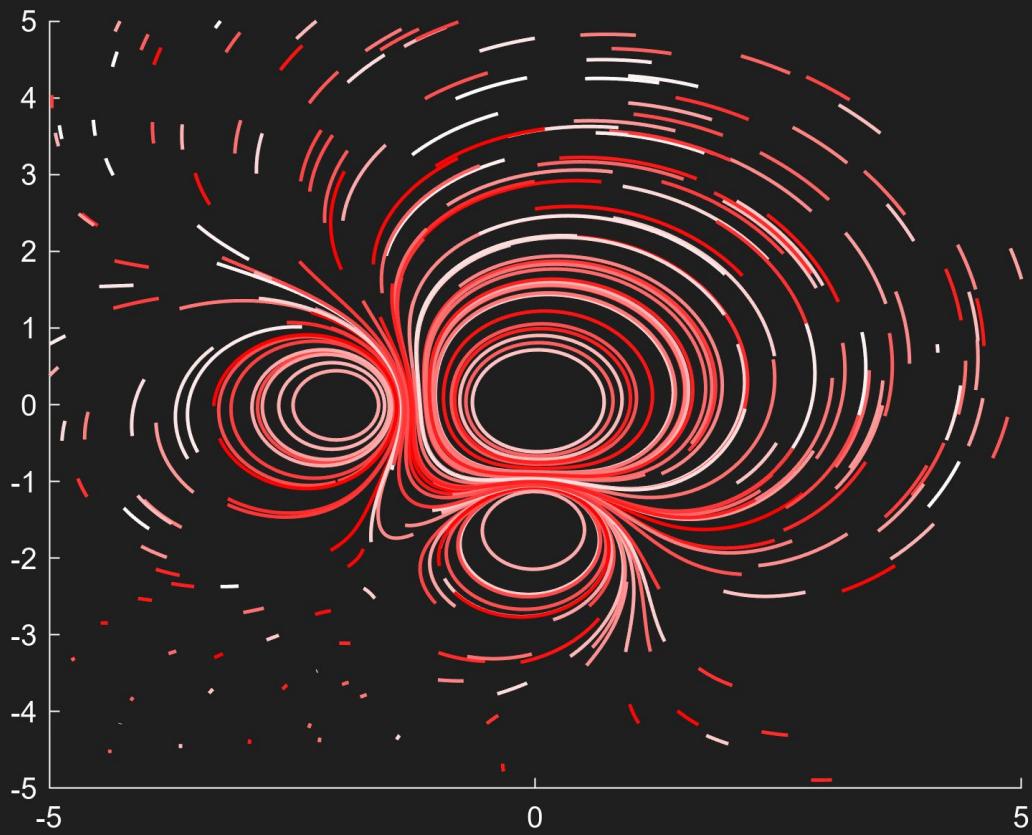
borealis()



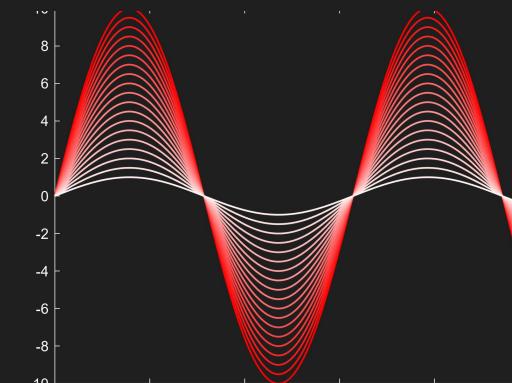
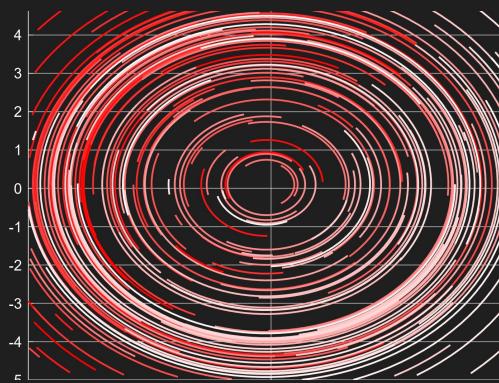
```
my_field = particle_field(); % Not part of standard library  
  
myaxes = axes();  
fireflies();  
  
animate_field("ParticleField",  
             "Function",  
             "Axes",  
             my_field, ...  
             @magnetic_field, ...  
             myaxes); % Not part of standard library
```



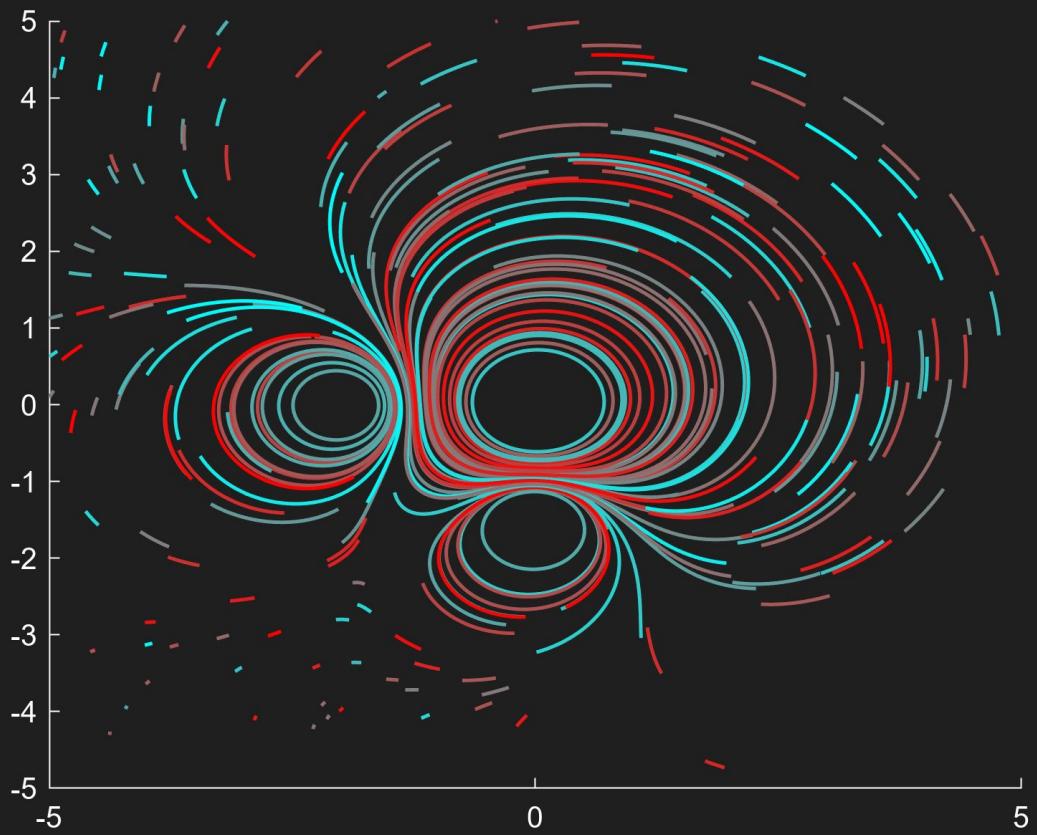
dark_mode()



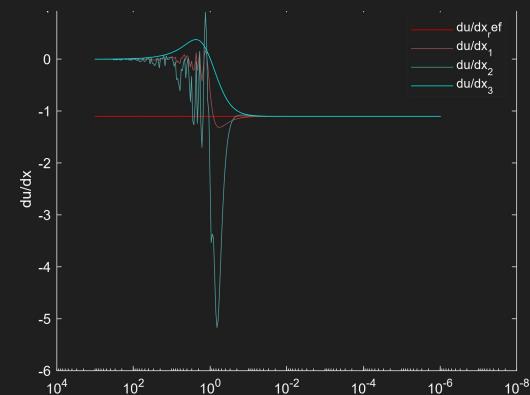
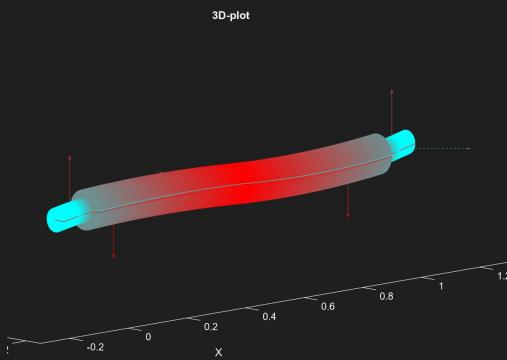
```
my_field = particle_field(); % Not part of standard library  
  
myaxes = axes();  
fireflies();  
  
animate_field("ParticleField",  
             "Function",  
             "Axes",  
             my_field, ...  
             @magnetic_field, ...  
             myaxes); % Not part of standard library
```



dark_mode2()



```
my_field = particle_field(); % Not part of standard library  
  
myaxes = axes();  
fireflies();  
  
animate_field("ParticleField",  
             "Function",  
             "Axes",  
             my_field, ...  
             @magnetic_field, ...  
             myaxes); % Not part of standard library
```



Integration into VS-code

File Edit Selection View Go Run Terminal Help ← → ⌘ diffprojekt_fjadring (Workspace) □ □ □ 0° - + ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋

EXPLORER ... preview newtons_animated.m MAIN_DOCUMENT.md ...

OPEN EDITORS

GROUP 1

- Preview newton...
MAIN_DOCUMENT....

GROUP 2

- Preview MAIN_...

DIFFPROJEKT_FJADRING (...)

diffprojekt_fjadrin...

- sym_math
- .MATLABDriveTag
- Betrakta en linjär O...
- c_s & c_k.png
- convergence_order.m
- convergence.png
- dependent_paramet...
- diffprojekt_fjadrin...
- error_fun.m
- Euler_fjadrin.... M
- Euler_fjadrin.... M
- Euler_fwd.m
- euler_solver.m
- euler_stiff.png M
- Euler_timestep.... M
- Euler_timestep.... M
- imp_trapetz.csv
- imp_trapetz.m
- impli_trapetz.m
- impli_trapetzus.m
- implicit_solver.m
- implicit_stiff.png M
- implicit_trapetsmo...
- Jacobian_transfer_fu...
- k_ref_comparis... M
- MAIN_DOCUMENT....
- matlab_proj_1b_u5....
- MATLAB_proj1B_u5....

> OUTLINE

> TIMELINE

diffprojekt_fjadrin > MAIN DOCUMENT.md > # U1

213 Where $\boldsymbol{J}_{\Delta \vec{T}}$ is the Jacobian matrix for the transformation $\Delta \vec{T} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$.

214 The Jacobian $\boldsymbol{J}_{\Delta \vec{T}}$ is given by:

215

```
216 $ \boldsymbol{J}_{\Delta \vec{T}} = \begin{pmatrix} \frac{\partial \Delta T_1}{\partial k_1} & \frac{\partial \Delta T_1}{\partial k_2} \\ \frac{\partial \Delta T_2}{\partial k_1} & \frac{\partial \Delta T_2}{\partial k_2} \end{pmatrix}
```

217 The Jacobian $\boldsymbol{J}_{\Delta \vec{T}}$ is given by:

218

```
219 $ \boldsymbol{J}_{\Delta \vec{T}} = \begin{pmatrix} \frac{\partial \Delta T_1}{\partial k_1} & \frac{\partial \Delta T_1}{\partial k_2} \\ \frac{\partial \Delta T_2}{\partial k_1} & \frac{\partial \Delta T_2}{\partial k_2} \end{pmatrix}
```

220

```
221 $ \boldsymbol{J}_{\Delta \vec{T}} = \begin{pmatrix} \frac{\partial \Delta T_1}{\partial k_1} & \frac{\partial \Delta T_1}{\partial k_2} \\ \frac{\partial \Delta T_2}{\partial k_1} & \frac{\partial \Delta T_2}{\partial k_2} \end{pmatrix}
```

222

```
223 $ \boldsymbol{J}_{\Delta \vec{T}} = \begin{pmatrix} \frac{\partial \Delta T_1}{\partial k_1} & \frac{\partial \Delta T_1}{\partial k_2} \\ \frac{\partial \Delta T_2}{\partial k_1} & \frac{\partial \Delta T_2}{\partial k_2} \end{pmatrix}
```

224

```
225 $ \boldsymbol{J}_{\Delta \vec{T}} = \begin{pmatrix} \frac{\partial \Delta T_1}{\partial k_1} & \frac{\partial \Delta T_1}{\partial k_2} \\ \frac{\partial \Delta T_2}{\partial k_1} & \frac{\partial \Delta T_2}{\partial k_2} \end{pmatrix}
```

226

```
227 $ \boldsymbol{J}_{\Delta \vec{T}} = \begin{pmatrix} \frac{\partial \Delta T_1}{\partial k_1} & \frac{\partial \Delta T_1}{\partial k_2} \\ \frac{\partial \Delta T_2}{\partial k_1} & \frac{\partial \Delta T_2}{\partial k_2} \end{pmatrix}
```

228

229 Thus we can use the Jacobian given to us without modification to do the Newton-iteration.

230

231

232 The convergence we got for different starting-values was the following:

Where $\boldsymbol{J}_{\Delta \vec{T}}$ is the Jacobian matrix for the transformation $\Delta \vec{T} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$.

The Jacobian $\boldsymbol{J}_{\Delta \vec{T}}$ is given by:

$$\boldsymbol{J}_{\Delta \vec{T}} = \begin{pmatrix} \frac{\partial \Delta T_1}{\partial k_1} & \frac{\partial \Delta T_1}{\partial k_2} \\ \frac{\partial \Delta T_2}{\partial k_1} & \frac{\partial \Delta T_2}{\partial k_2} \end{pmatrix}$$
$$\frac{\partial \Delta T_i}{\partial k_j} = \frac{\partial T_i(k_1, k_2) - T_{i,final}}{\partial k_j} = \frac{\partial T_i(k_1, k_2)}{\partial k_j} = \frac{\partial F_i(k_1, k_2) + \phi_i}{\partial k_j} = \frac{\partial F_i(k_1, k_2)}{\partial k_j}$$
$$\Leftrightarrow \boldsymbol{J}_{\Delta \vec{T}} = \boldsymbol{J}_F$$

Thus we can use the Jacobian given to us without modification to do the Newton-iteration.

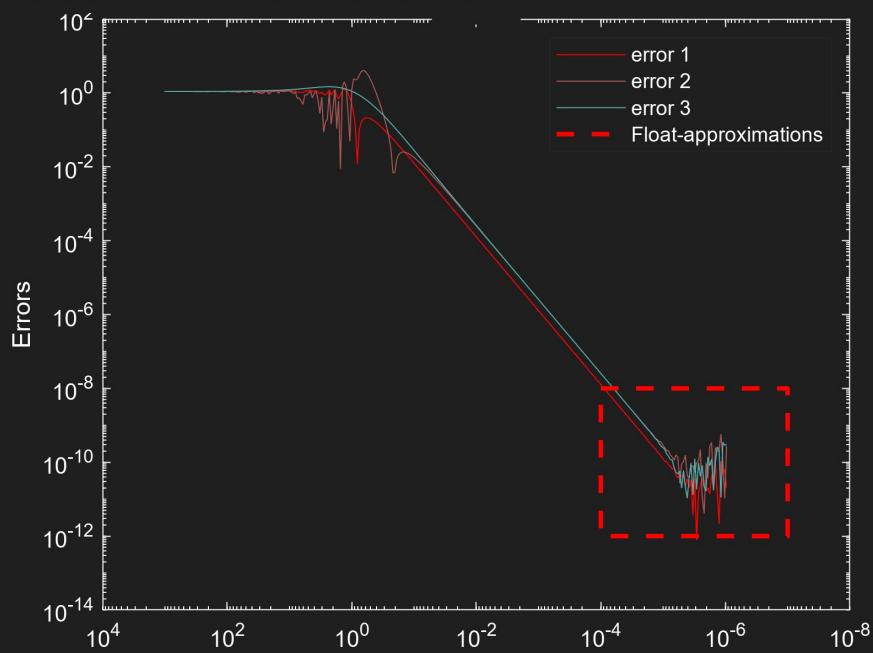
The convergence we got for different starting-values was the following:

Errors

error 1
error 2
error 3
Float-approximations

10⁻¹⁴ 10⁻¹² 10⁻¹⁰ 10⁻⁸ 10⁻⁶ 10⁻⁴ 10⁻² 10⁰ 10² 10⁴

10⁻¹⁴ 10⁻¹² 10⁻¹⁰ 10⁻⁸ 10⁻⁶ 10⁻⁴ 10⁻² 10⁰ 10² 10⁴



Includes color theme template!

```
function my_colortHEME(varargin)

colortHEME_process_inputs_subroutine

color1 = [255 255 100]/255;
color2 = [255 0 0]/255;
color3 = [70 0 220]/255;
color4 = [0 0 0]/255;

BACKGROUND_COLOR = [15 0 20]/255;
AXES_COLOR = [250 0 150]/255;

NEW_COLORORDER = create_colormap(color1, color2, color3, color4, number_of_hues);
NEW_COLORMAP = create_colormap(color4, color3, color2, color1, 256);

colortHEME_assign_colors_subroutine

end
```

Make your own color themes for your plots



Find them on github!

github.com/spiggen/MATLAB_color_themes

Links can also be found on matlab_kth
instagram and facebook-group!

And if you do use it, please show me!!! I'd be
happy to add more contributors if you're someone
who likes this sort of stuff! :)

Questions?



SCAN ME

Problem to work on after eating:

<https://se.mathworks.com/matlabcentral/cody/problems/55770-jack-o-lantern/solutions/new>

Like (3)

Difficulty: ●●○ (57) Rate

Solve Later Add To Group

If visualized correctly, the data contained in the matrix A will look like a jack-o'-lantern.

Create a function that will visualize A as an indexed color image. Make sure the orientation of the face is correct! Remove the x-ticks and y-ticks. Change the colormap so that the background (A has the value 1) is white, the facial features (A has the value 0) are black, and the face (A has the value 0.5) is orange.

Your function should return the figure handle as output.



Prize!! MATLAB frisbee and sunglasses 😎😎

Merch giveaway requirements:

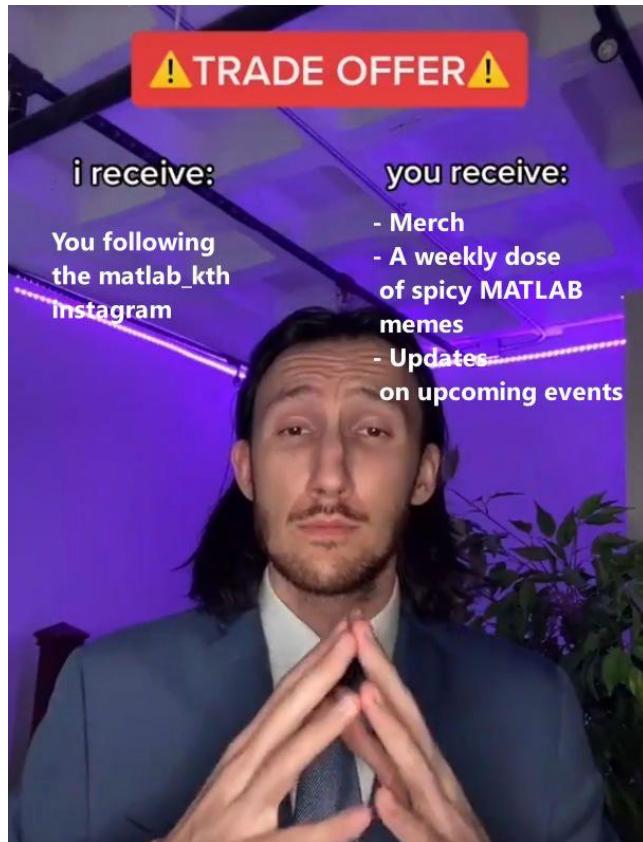
- Follow **matlab_kth** on instagram **Or** join the facebook-group!
- Follow **farmenflyg** on instagram **Or** Flygsektionens Arbetsmarknadsenhet on facebook if you're from the flight-chapter! (You're welcome to do so if you're not from the flight-chapter too but you don't have too :))



matlab_kth
instagram



matlab_kth
facebook
group



farmenflyg
instagram



Flygsektionens
Arbetsmarknadsenhet
facebook