

# Matlab Plotting

Hendrik Speleers

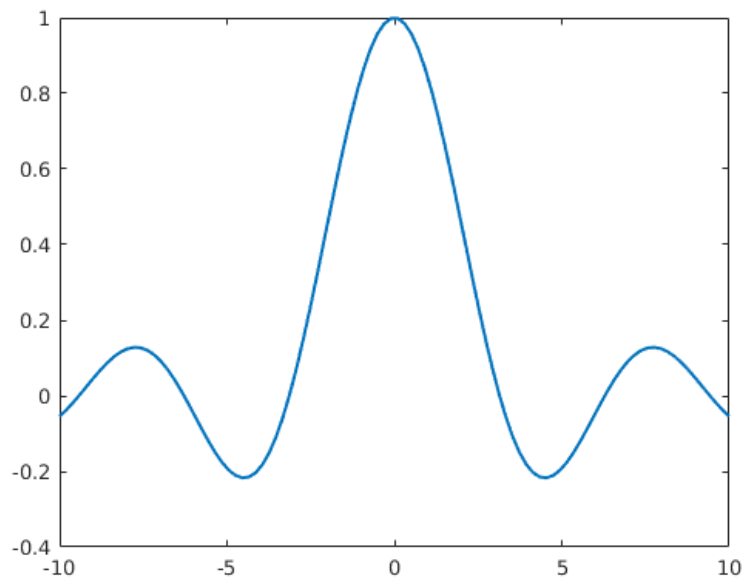
# Matlab Plotting

- Overview
  - 2D plotting
    - Standard line plotting
    - Other plotting
  - 3D plotting
    - 3D line/surface plotting
    - Contour plotting
  - Input and output
    - Save and load figures

# Matlab Plotting

- 2D plotting
  - Syntax is vector-based

```
>> x = linspace(-10, 10, 100);  
>> y = sin(x)./x;  
>> plot(x, y);
```



# Matlab Plotting

- 2D plotting
  - Standard line plotting: basic syntax

```
plot(y)
plot(x, y)
plot(x, y, 'clm')
plot(x1, y1, 'clm', x2, y2, 'clm')
```

- Connect data points (x, y) with optional format string
- Color (c): b, g, r, c, m, y, k, w
- Linestyle (l): -, --, -., :
- Marker (m): o, \*, ., +, x, s, d, ^, <, >, p, h, ...

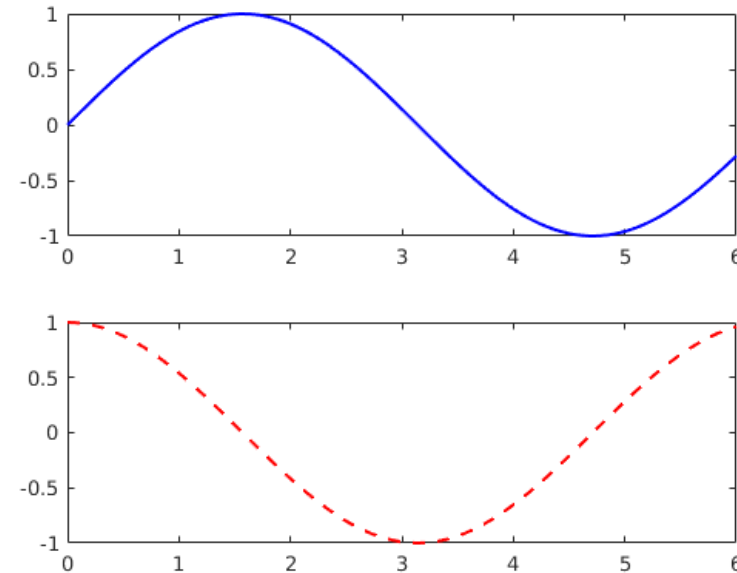
# Matlab Plotting

- 2D plotting

- Multiple plots

- New figure: `figure`
    - Same figure: `hold on/off`
    - Subplot: `subplot(nrows, ncols, index)`

```
>> x = linspace(0, 6, 100);  
>> subplot(2, 1, 1);  
>> plot(x, sin(x), 'b');  
>> subplot(2, 1, 2);  
>> plot(x, cos(x), 'r--');
```



# Matlab Plotting

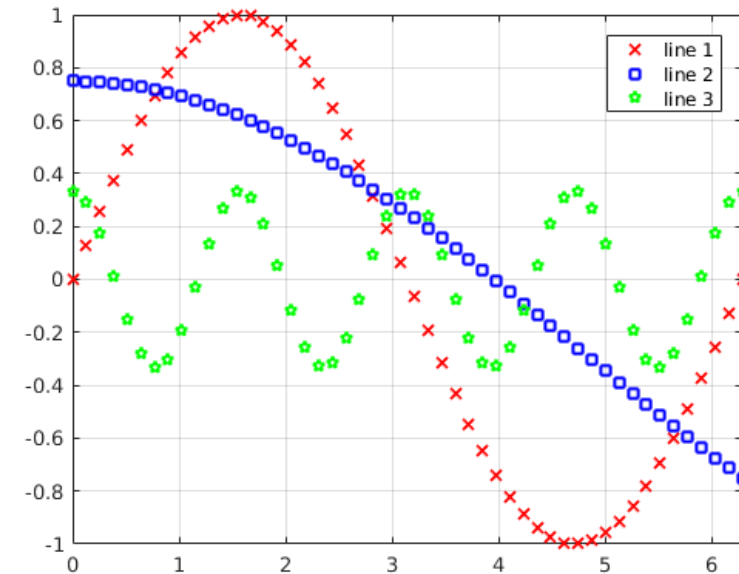
- 2D plotting
  - Figure options
    - Get or set limits: `xlim`, `ylim`, `axis`
      - `xlim([left, right])`, `ylim([bottom, top])`
      - `axis([left, right, bottom, top])`
      - `axis equal`, `axis on/off`, ...
      - `ax = axis`
    - Set labels: `xlabel(txt)`, `ylabel(txt)`
    - Set title: `title(txt)`
    - Set legend: `legend(label1, label2, ...)`
    - Others: `box on/off`, `grid on/off`, ...

# Matlab Plotting

- 2D plotting

- Example

```
>> x = linspace(0, 2*pi, 50);  
>> plot(x, sin(x), 'rx');  
>> hold on;  
>> plot(x, cos(x/3)-0.25, 'bs');  
>> plot(x, cos(4*x)/3, 'gp');  
>> hold off;  
>> xlim([0, 2*pi]);  
>> grid on;  
>> legend('line 1', 'line 2', 'line 3');
```



# Matlab Plotting

- 2D plotting

- Standard line plotting: advanced syntax

```
plot(__, 'name', value)
```

- Each plot option is a name-value pair (name is a string)
    - **Name:** Color, LineStyle, LineWidth, Marker, MarkerFaceColor, MarkerEdgeColor, MarkerEdgeWidth, MarkerSize, DisplayName, ...

- Legend: advanced syntax

```
legend(__, 'name', value)
```

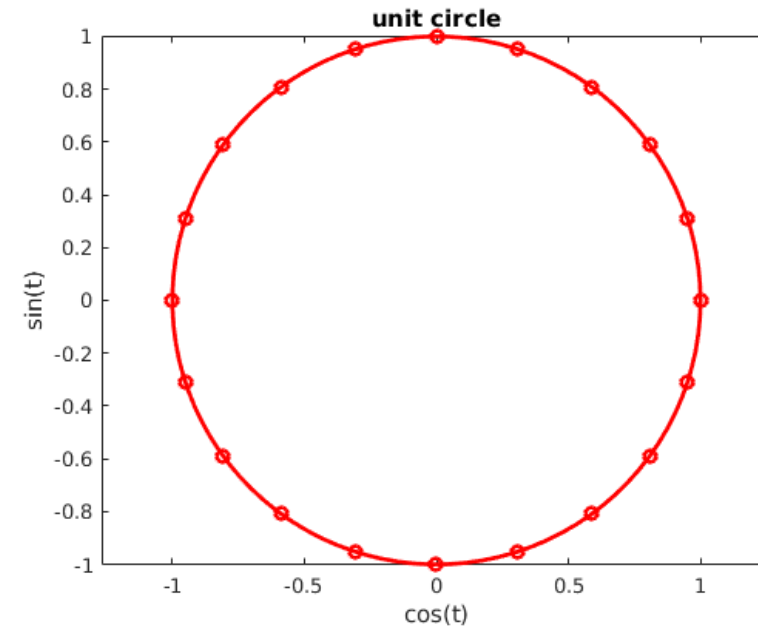
- **Name:** Location, Orientation, ...

# Matlab Plotting

- 2D plotting

- Example

```
>> t = linspace(0, 2*pi, 100);  
>> x = cos(t); y = sin(t);  
>> tm = linspace(0, 2*pi, 21);  
>> xm = cos(tm); ym = sin(tm);  
>> plot(x, y, 'r', xm, ym, 'ro', ...  
        'LineWidth', 2);  
>> xlabel('cos(t)'); ylabel('sin(t)');  
>> title('unit circle');  
>> axis equal;
```



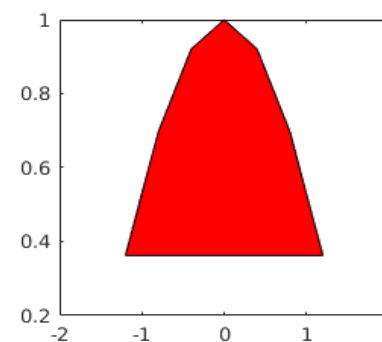
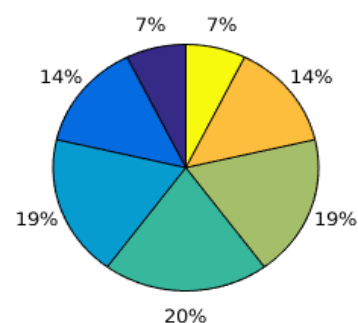
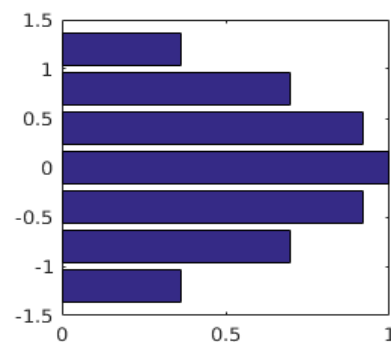
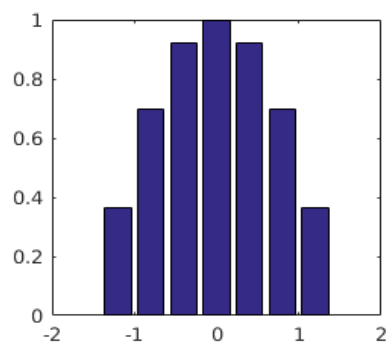
# Matlab Plotting

- **2D plotting**
  - Other plotting
    - Log plots: `loglog(x, y)`, `semilogx(x, y)`, `semilogy(x, y)`
    - Polar plots: `polar(theta, r)`
    - Scatter plots: `scatter(x, y)`
    - Bar graphs: `bar(x, height)`, `barh(y, width)`
    - Pie charts: `pie(x)`
    - Histogram: `histogram(x, bins)`
    - Filled curves: `fill(x, y, color)`
  - For full method details, check out `help method`

# Matlab Plotting

- 2D plotting
  - Example

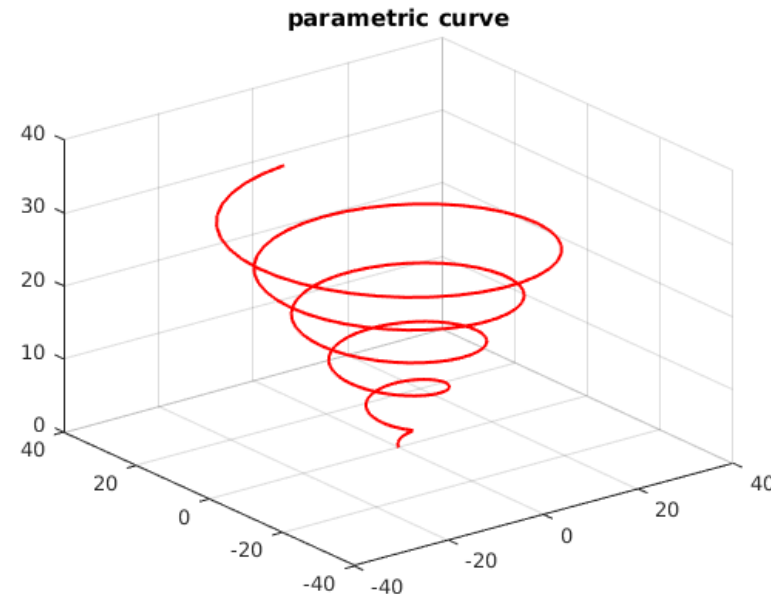
```
>> x = linspace(-1.2, 1.2, 8); y = cos(x);  
>> subplot(1, 4, 1); bar(x, y);  
>> subplot(1, 4, 2); barh(x, y);  
>> subplot(1, 4, 3); pie(y);  
>> subplot(1, 4, 4); fill(x, y, 'r');
```



# Matlab Plotting

- 3D plotting
  - Natural plot extensions
    - Line plots: `plot3(x, y, z)`
    - Scatter plots: `scatter3(x, y, z)`

```
>> z = linspace(0, 10*pi, 200);  
>> x = z.*sin(z);  
>> y = z.*cos(z);  
>> plot3(x, y, z, 'r');  
>> title('parametric curve');  
>> grid on;
```



# Matlab Plotting

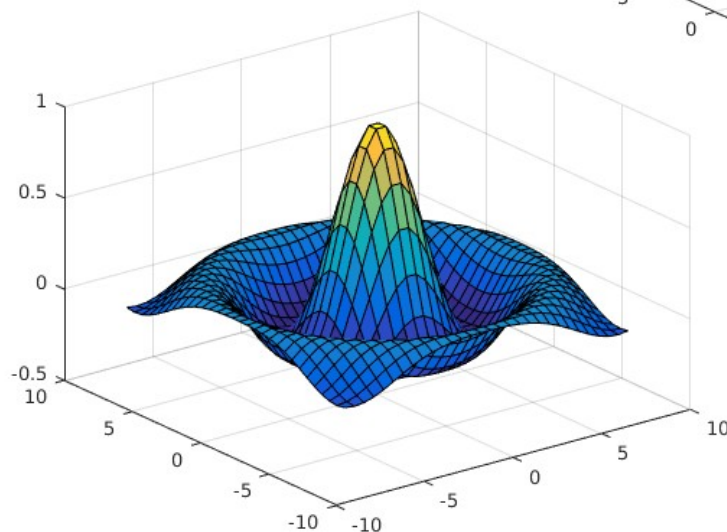
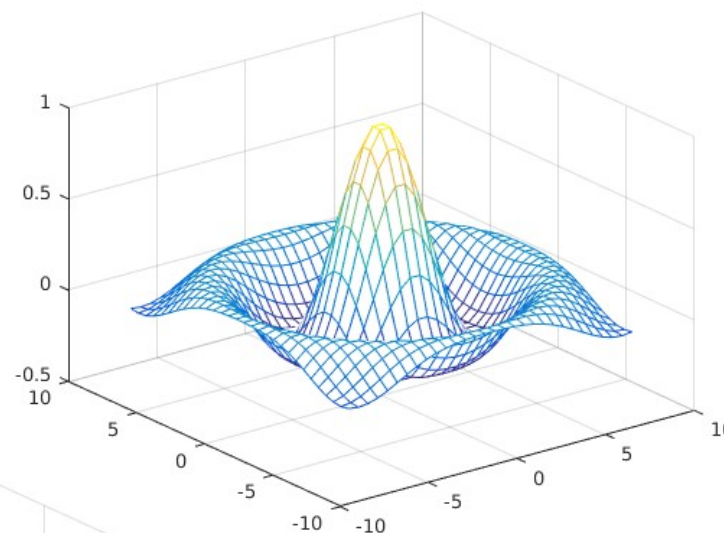
- 3D plotting
  - Surface plotting: syntax is matrix-based
    - Wireframe plot: `mesh(X, Y, Z)`
    - Surface plot: `surf(X, Y, Z)`
  - Grid creation: create coordinate matrices from coordinate vectors
    - `[X, Y] = meshgrid(x, y)`
      - Matrix `X` contains in each row values of vector `x`
      - Matrix `Y` contains in each column values of vector `y`
  - Figure options extend to 3D
    - `axis, xlabel, ylabel, zlabel, title, ...`

# Matlab Plotting

- 3D plotting

- Example

```
>> x = linspace(-8, 8, 30); y = x;  
>> [X, Y] = meshgrid(x, y);  
>> R = sqrt(X.^2 + Y.^2);  
>> Z = sin(R)./R;  
>> figure(1);  
>> mesh(X, Y, Z);  
>> figure(2);  
>> surf(X, Y, Z);
```



# Matlab Plotting

- 3D plotting

- Surface options

- Viewpoint specification

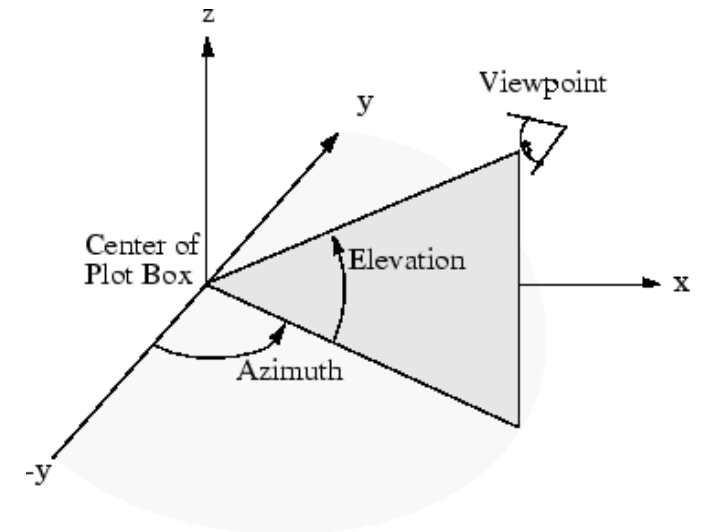
- Set viewpoint: `view(az, el), view([x, y, z])`
      - Change viewpoint: `rotate3d on/off`

- Color maps: mapping between numeric values and colors

- Keyword: `colormap type`
      - Type: `hsv, flag, jet, hot, cool, parula, winter, ...`

- Shading: how to color surfaces

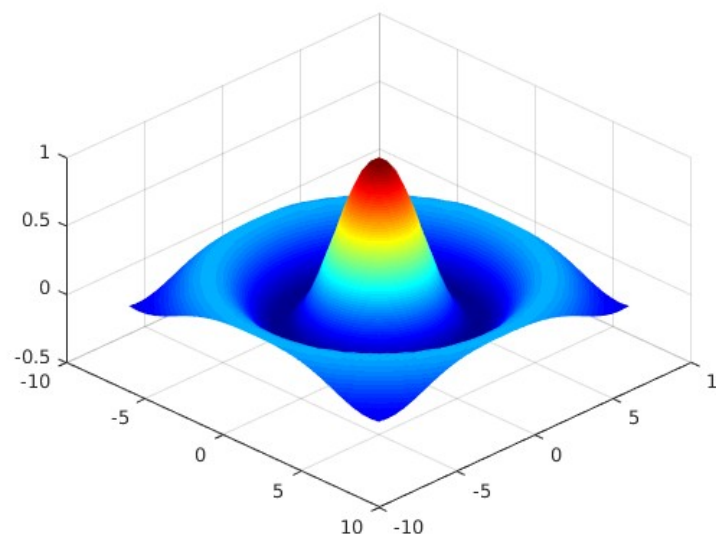
- Keyword: `shading type`
      - Type: `flat, faceted, interp`



# Matlab Plotting

- 3D plotting
  - Example

```
>> x = linspace(-8, 8, 30); y = x;  
>> [X, Y] = meshgrid(x, y);  
>> R = sqrt(X.^2 + Y.^2);  
>> Z = sin(R)./R;  
>> surf(X, Y, Z);  
>> colormap jet;  
>> shading interp;  
>> view(45, 45);
```



# Matlab Plotting

- Contour plotting

- Contour lines: basic syntax

```
contour(Z)  
contour(X, Y, Z)  
contour(X, Y, Z, levels)
```

- Other related functions:

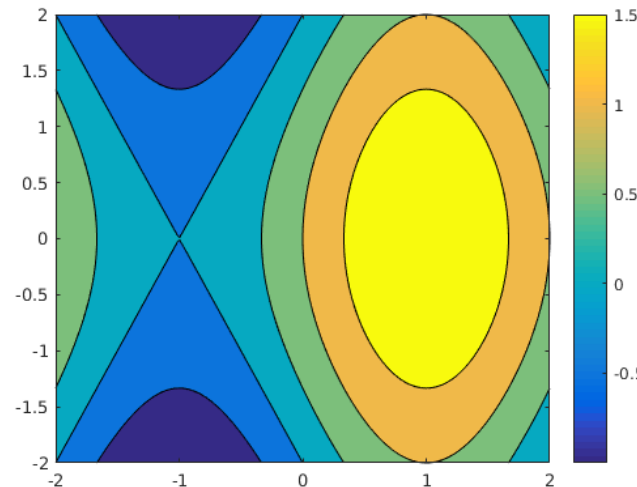
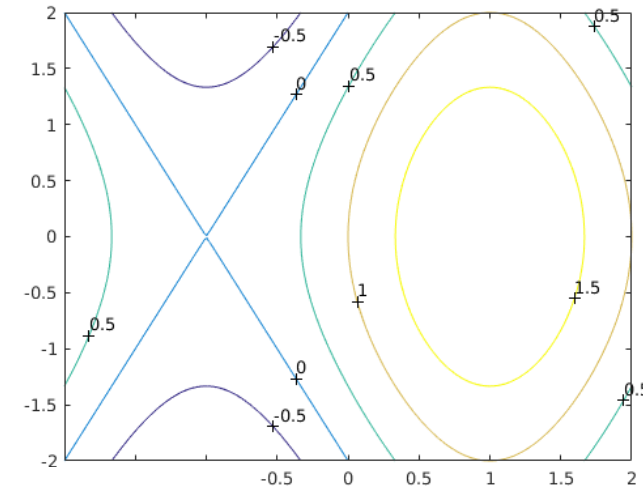
- Filled contours: `contourf(X, Y, Z, levels)`
    - Contour identification: `clabel(cs), colorbar`
    - 3D contour lines: `contour3(X, Y, Z, levels)`

# Matlab Plotting

- Contour plotting

- Example

```
>> t = linspace(-2, 2, 201);  
>> [X, Y] = meshgrid(t, t);  
>> Z = sin(X*pi/2) + cos(Y*pi/4);  
>> figure(1);  
>> cs = contour(X, Y, Z);  
>> clabel(cs);  
>> figure(2);  
>> cs = contourf(X, Y, Z);  
>> colorbar;
```



# Matlab Plotting

- **Input and output**
  - Save figures
    - Internal format: `savefig('filename.fig')`
    - All formats: `saveas(figid, 'filename.ext')`
      - Matlab supports bmp, jpg, png, pdf, eps/epsc, svg
  - Load figures
    - Internal format: `openfig(filename)`

```
>> plot([1, 2, 4, 2]);  
>> savefig('plot.fig');  
>> saveas(gcf, 'plot.png');
```