Plotting in 3D and animation

Dr. Mihail

October 2, 2018

3D Plots

Plots of 1D functions (e.g., $f(x) = x^2$) are trivially extended to 2D by using a second input y: $f(x, y)$. When the outputs of these functions is a scalar, we can visualize it in several different ways.

MATLAB function

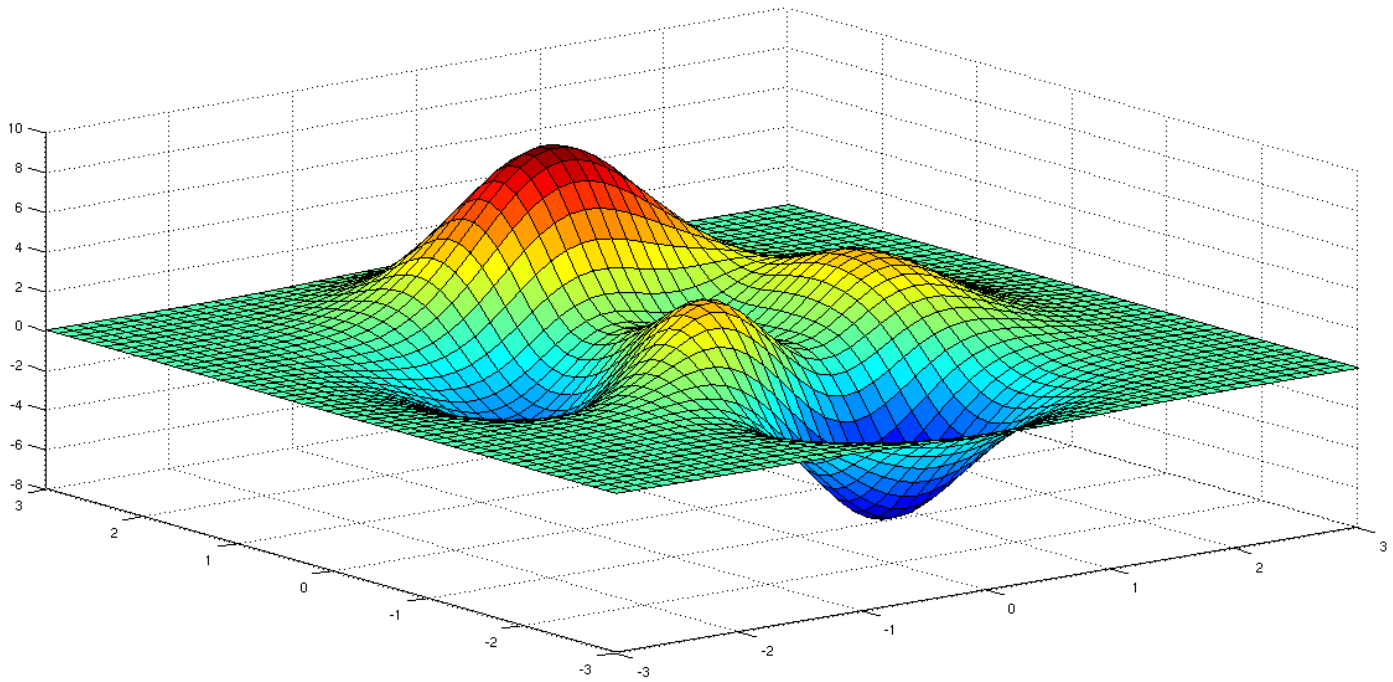
peaks

We will use use a built-in MATLAB function useful for demonstrating 3D plots called `peaks`. In particular, the version of `peaks` with three outputs: `[x, y, z] = peaks(n);` will generate an output (z) for every pair of x and y, on a grid of size n.

```
[x, y, z] = peaks(50);
```

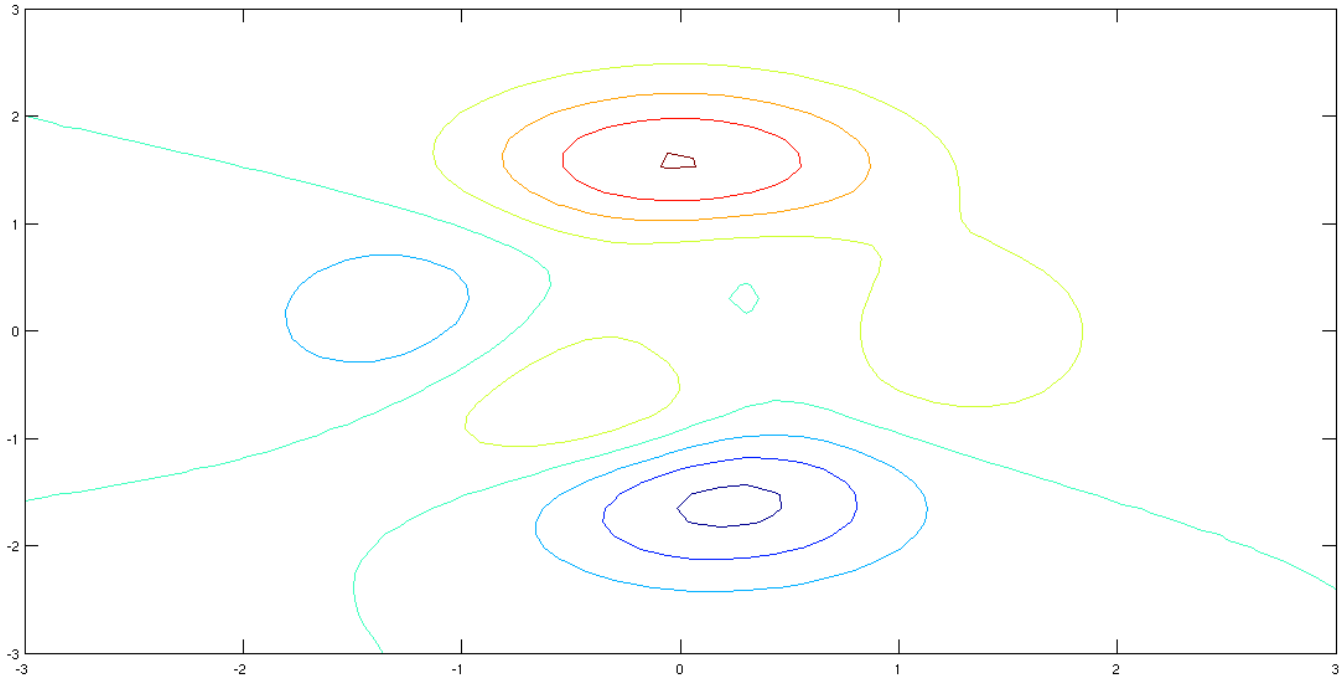
Surface plot

```
[x, y, z] = peaks(50);surf(x, y, z);
```



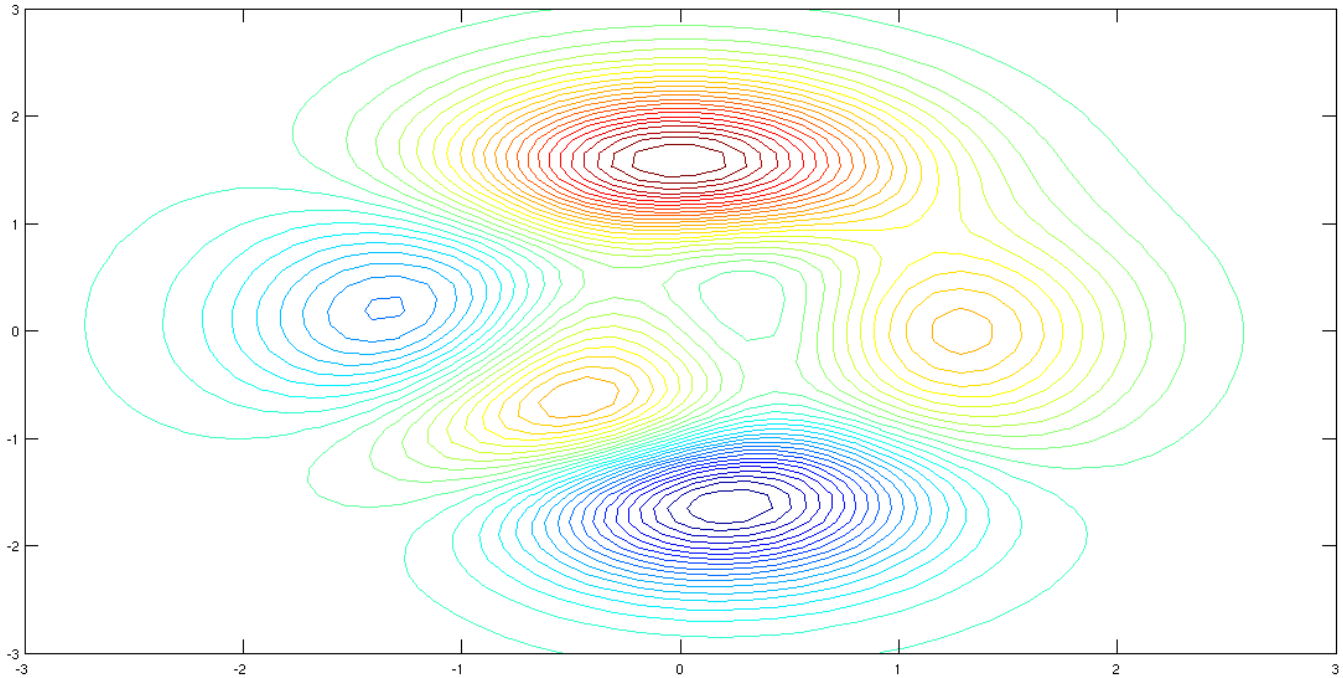
Contour plot

```
[x, y, z] = peaks(50);contour(x, y, z);
```



Contour plot with more contour levels

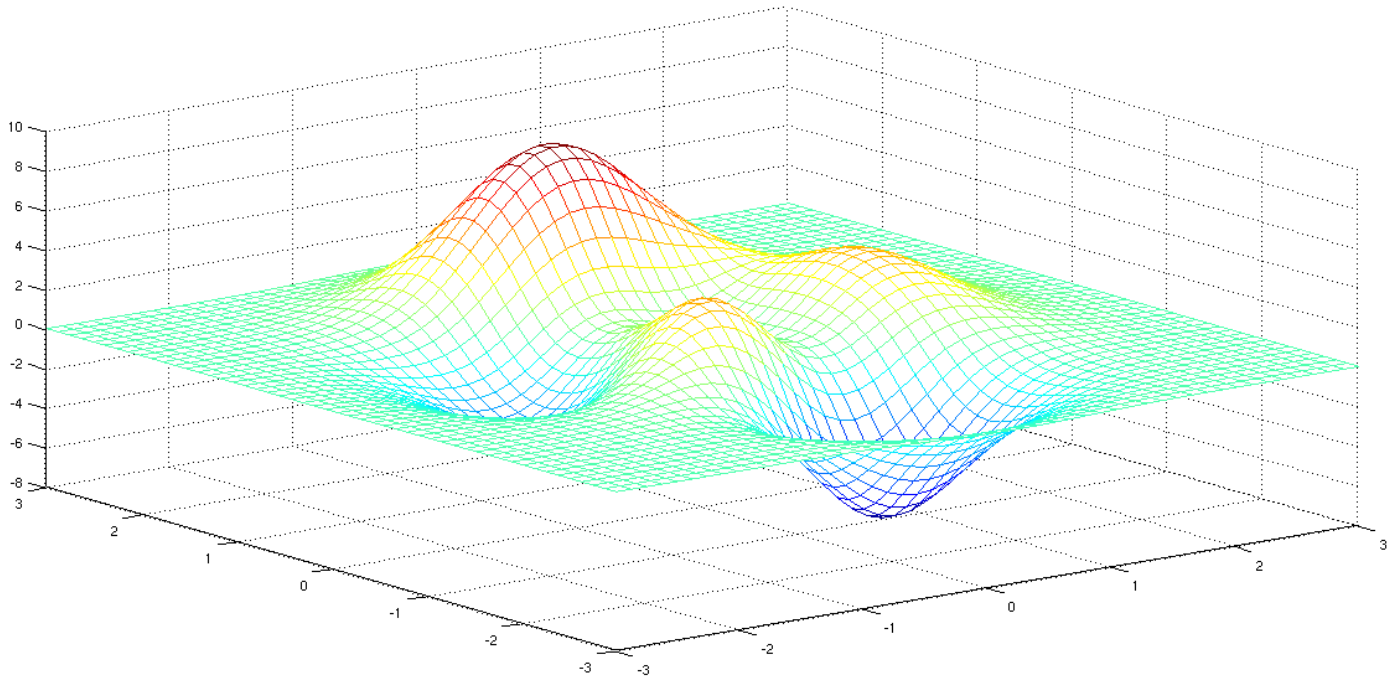
```
[x, y, z] = peaks(50);contour(x, y, z, 40);
```



Mesh

Mesh plot

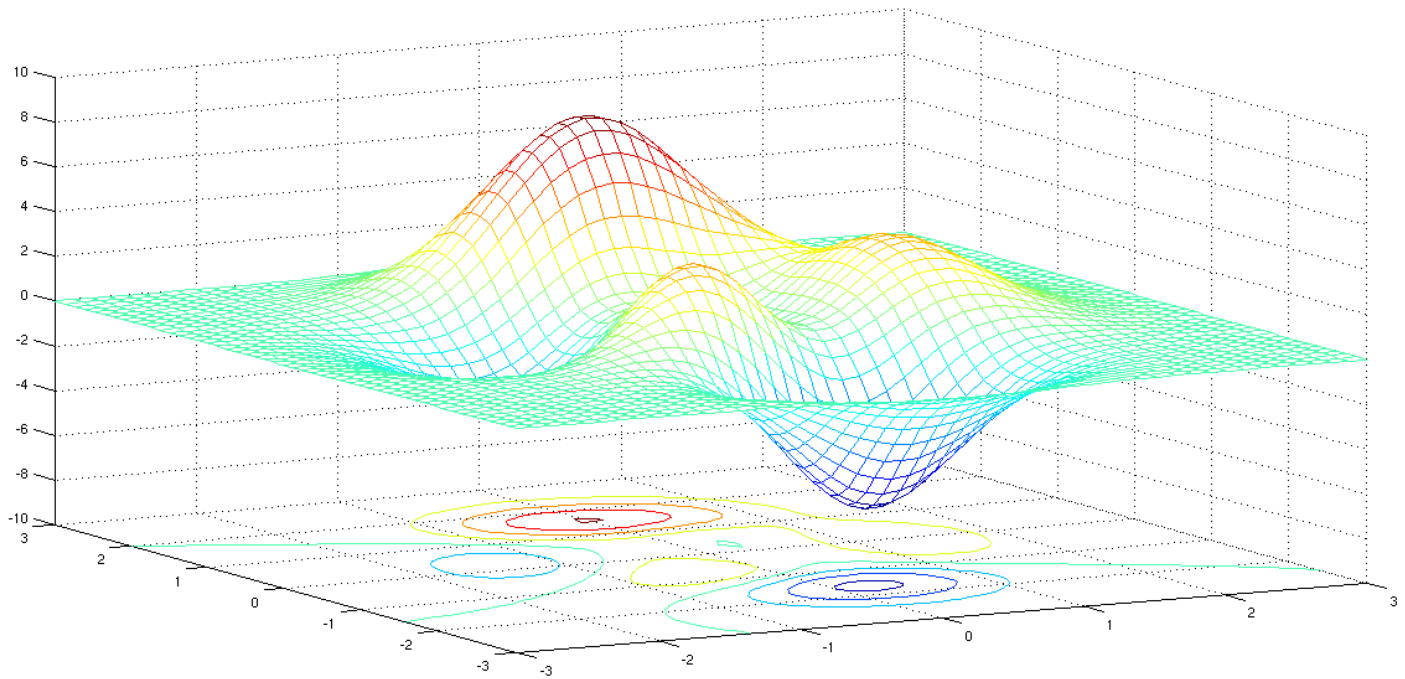
```
[x, y, z] = peaks(50);mesh(x, y, z);
```



Mesh with contour plot

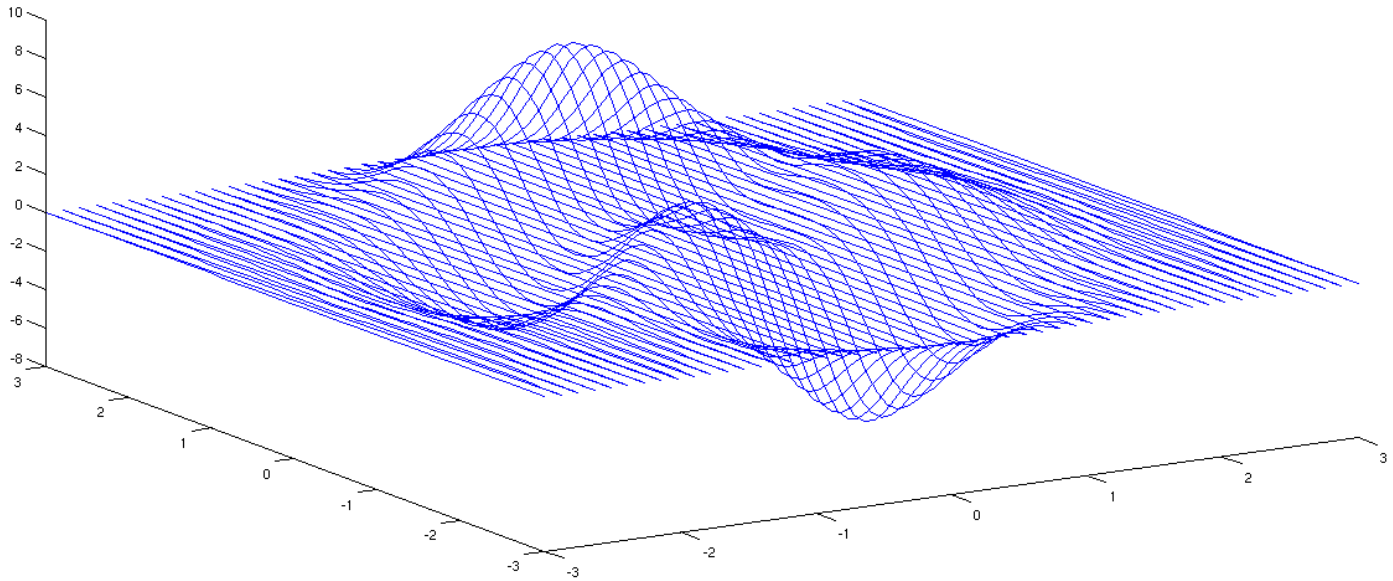
Mesh with contour

```
[x, y, z] = peaks(50);meshc(x, y, z);
```



3D line plot

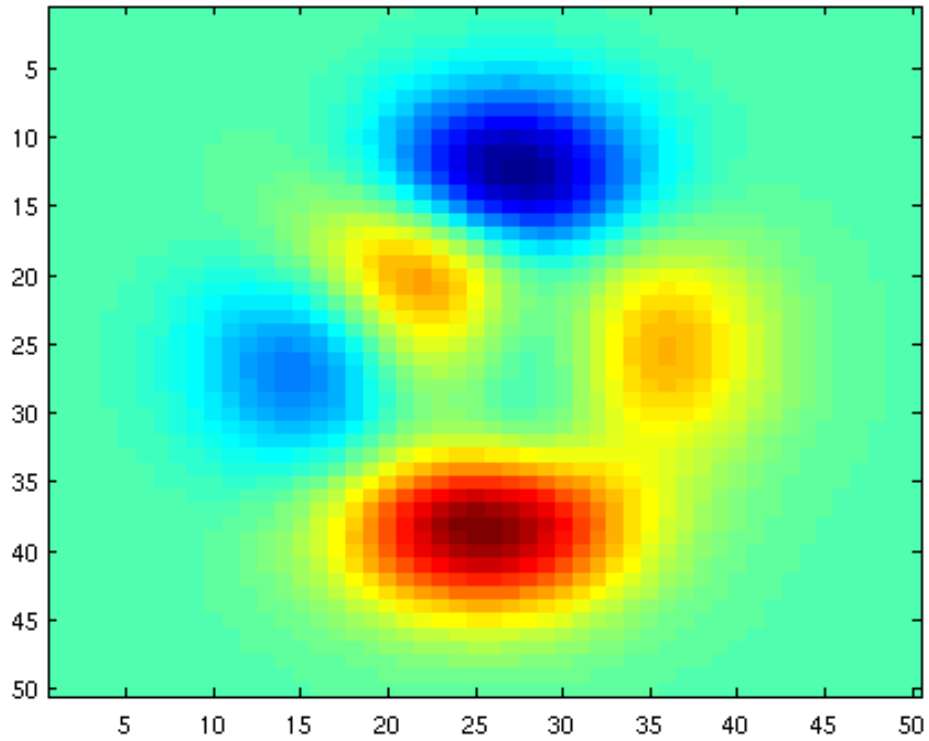
```
[x, y, z] = peaks(50);plot3(x(:), y(:), z(:));
```



Color codes

Color coded image

```
[x, y, z] = peaks(50);imagesc(z);
```

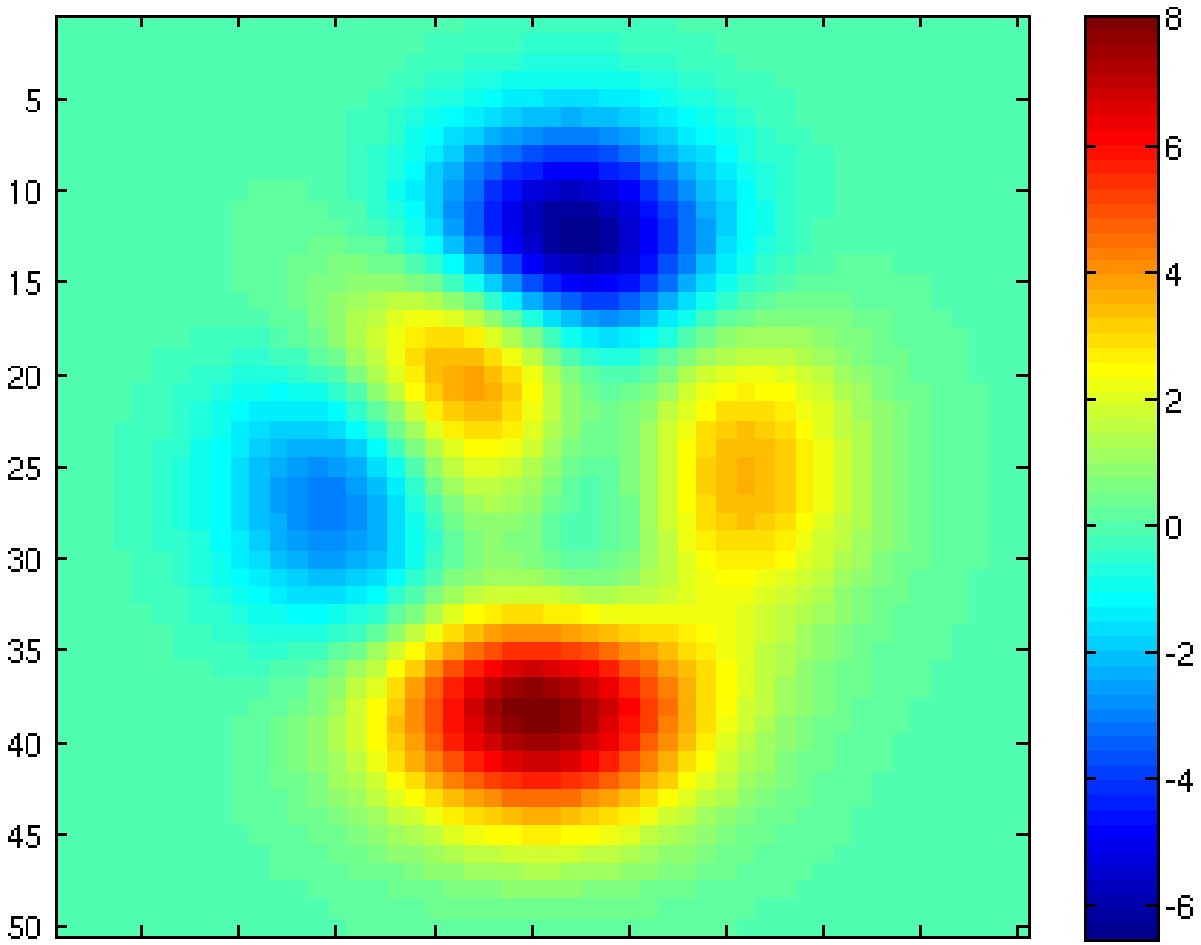


We know relative shape, but each color represents a number. We need to add the `colorbar`.

Color codes

Color coded image

```
[x, y, z] = peaks(50);imagesc(z);colorbar;
```



Animations

Basic idea

Plot several times a second with slightly different parameters (the ones you want to animate), cleaning the figure each frame. This naturally leads to the use of loops. The quadratic family of functions is:

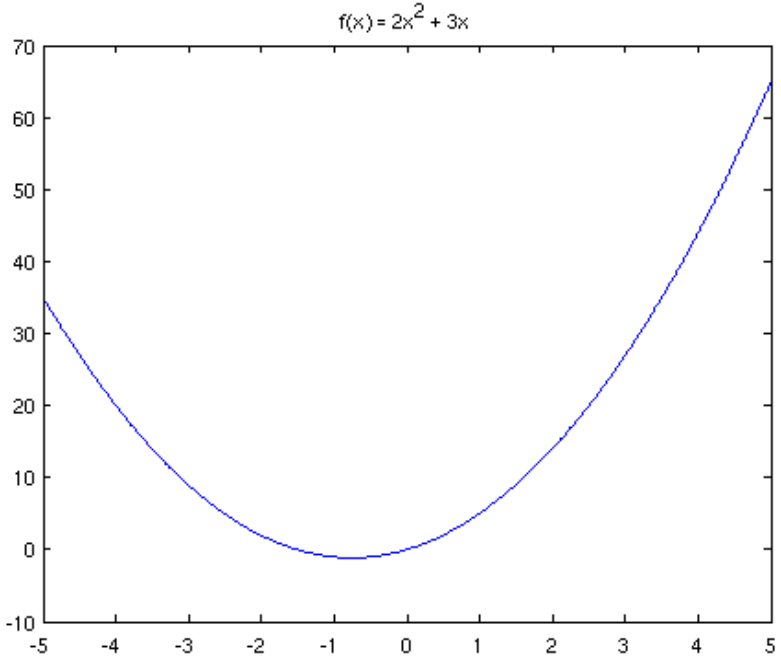
$$f(x) = ax^2 + bx + c$$

Let's pick the values 2, 3 and 0 for a, b and c:

$$f(x) = 2x^2 + 3x$$

Quadratic

```
x = linspace(-5, 5, 100); y = 2*x.^2 + 3*x; plot(x, y);
```



Let's animate $a = 2$ above, from 1 to 3.

Animating *a*

```
a = linspace(1, 3, 100); % 100 choices for a, between 1 and 3
x = linspace(-5, 5, 100); % x never changes
for one_a = a
    figure(1);clf; % create and clear figure
    y = one_a*x.^2 + 3*x; % new function for a specific a
    plot(x, y); % plot
    xlim([-5, 5]); % set x-limits
    ylim([-10, 100]); % set y-limits
    title(['a = ' num2str(one_a)]); % set title
    pause(0.1); % pause one 10th of second each frame
end
```

Animating c in $f(x) = 2x^2 + 3x + c$

```
c = linspace(-4, 15, 100);
x = linspace(-5, 5, 100); % x never changes
for one_c = c
    figure(1);clf; % create and clear figure
    y = 2*x.^2 + 3*x + one_c; % new function for a specific b
    plot(x, y); % plot
    xlim([-5, 5]); % set x-limits
    ylim([-10, 100]); % set y-limits
    title(['c = ' num2str(one_b) '; in f(x) = 2*x^2 + 3*x + c']
    pause(0.1); % pause each frame
end
```