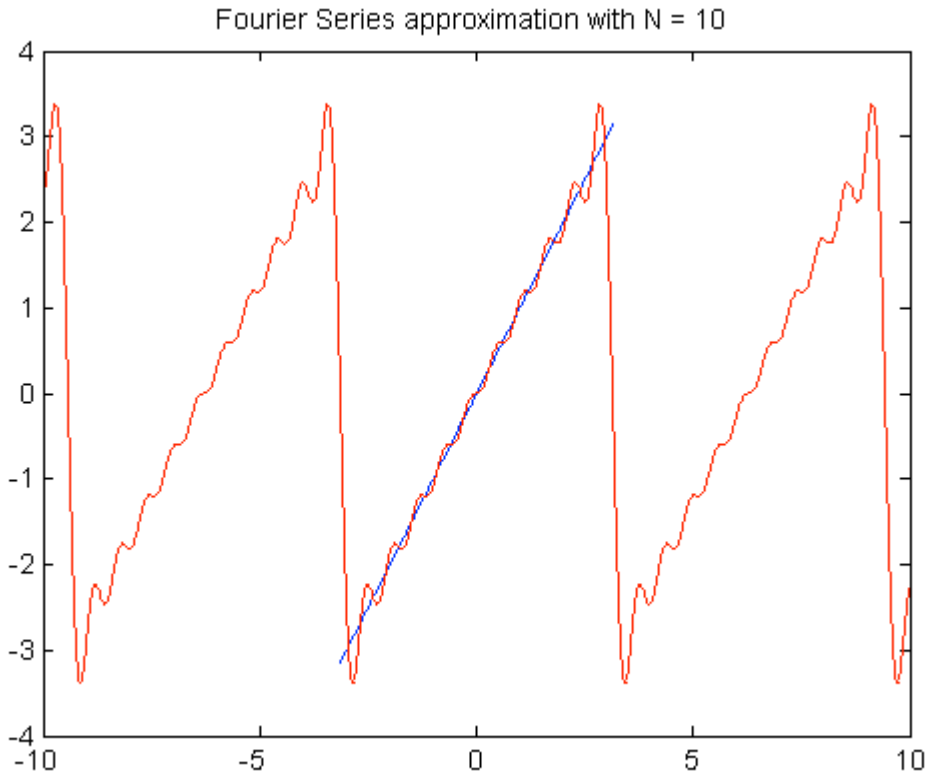
Lincoln Maths, Fourier Series and PDEs

Hilary 2013, Week 2

The below code will allow you to effectively compute the Fourier coefficients of any 2pi-periodically extended function on [-pi, pi]

```
clear           % clear all variables
close all       % close all figures

% **** How many modes do you want? ****
N = 10;

% **** Choose your function ****
%myfunc = @(x) (x>0).*cos(x);
myfunc = @(x)x;
%myfunc = @(x) (x>0).*sin(x);
%myfunc = @(x) exp(x);


nx = 100;                   % number of points in [-pi, pi]
x = linspace(-pi, pi, nx);  % create vector of x coordinates
f = myfunc(x);              % create vector of f values

[a0 a b] = fmodes(x, f, N); % compute Fourier modes


xnew = linspace(-10, 10, nx*100);   % Define a new grid for the whole line

plot(x, f, 'b');            % plot the original f
hold on

% Below computes and plots the sums from k = 1 to N
fapprox = a0/2;
p = plot(xnew, fapprox, 'r');
for k = 1:N
    fapprox = fapprox + a(k)*cos(k*xnew) + b(k)*sin(k*xnew);
    delete(p);
    p = plot(xnew, fapprox, 'r');
    title(['Fourier Series approximation with N = ', num2str(k)]);
    drawnow

    pause(0.2)
end
hold off
```



Published with MATLAB® 7.12

Compute Fourier coefficients

This function file computes the Fourier coefficients given a vector for x, a vector for f, and the number of desired modes.

It does so by numerically integrating the requisite function (using Matlab's trapz function)

```
function [a0 a b] = fmodes(x, f, N)

a = zeros(N, 1);
b = zeros(N, 1);

a0 = 1/pi*trapz(x, f);

for k = 1:N
    a(k) = 1/pi*trapz(x, f.*cos(k*x));
    b(k) = 1/pi*trapz(x, f.*sin(k*x));
end
```

Published with MATLAB® 7.12