Mini-Matlab Lesson 3: Numerical differentiation

How do you take the derivative of a function?

Contents

Two powerful commands

The 'clear' command clears all the variables from the workspace. Be careful with mistakingly writing code that reuses the old values of certain variables. The 'close all' command closes all the open figure windows

```
clear
close all
```

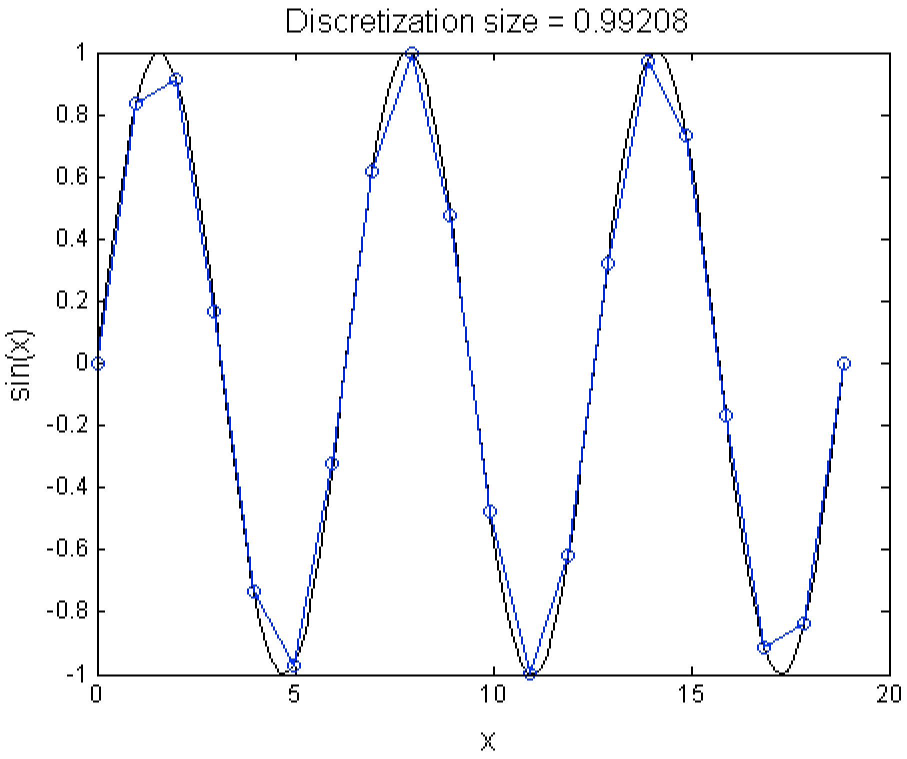
Anonymous functions

```
myfunc = @(x) sin(x);
myfuncp = @(x) cos(x);
```

Discretization of functions

We would like to discretize a continuous function, f(x), by representing it as a finite set of points. Consider the function f(x) = sin(x) on the domain [0, 6*pi]

```
N = 20;

x = linspace(0, 6*pi, N);    % Rough mesh
f = myfunc(x);

xs = linspace(0, 6*pi, 200); % Fine mesh for comparison

figure(1)
plot(xs, myfunc(xs), 'k');   % 'k' creates a black line
hold on                      % Hold the plot
plot(x, f, 'bo-');           % 'bo-' creates a blue line with circles
hold off
xlabel('x', 'FontSize', 16);
ylabel('sin(x)', 'FontSize', 16);
title(['Discretization size = ', num2str(x(2)-x(1))], 'FontSize', 16);
```



Numerical differentiation

Recall the definition of a derivative:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Let x_i be the i'th mesh point corresponding to f_i, and i = 1, ..., N, then the *forward finite difference* is defined by

$$f_i' = \frac{f_{i+1} - f_i}{x_{i+1} - x_i}$$

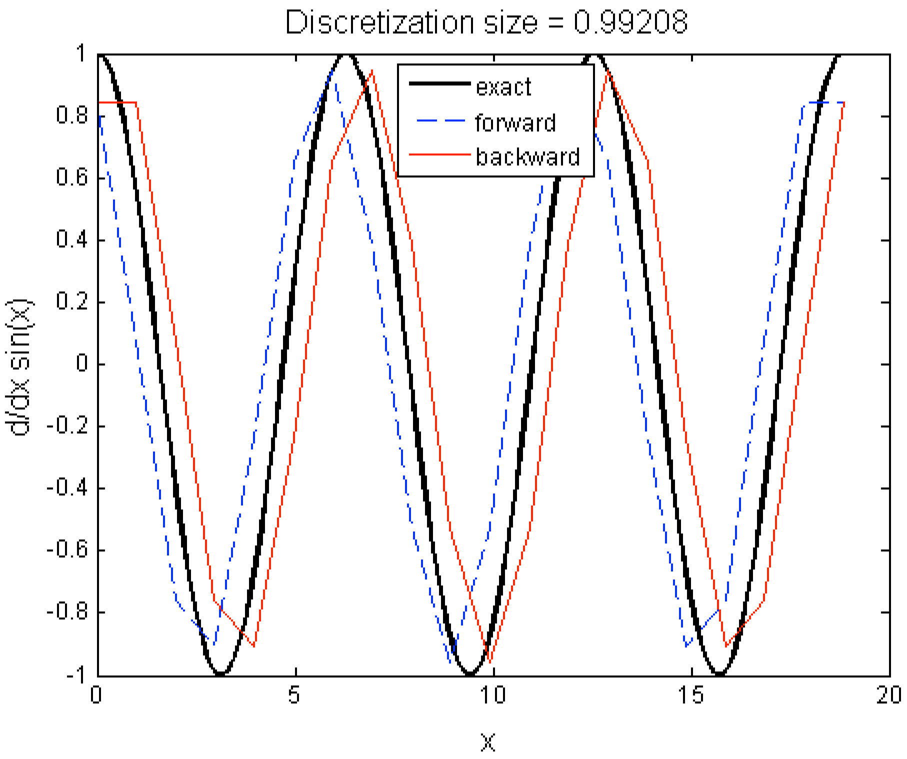
Similarly, the *backwards finite difference* is defined by

$$f_i' = \frac{f_i - f_{i-1}}{x_i - x_{i-1}}$$

Note that the forward differences can only be defined for i = 1 to N−1 and the backward differences from i = 2 to N. We will have to use backwards and forward differences for the endpoints (respectively).

```
% Define for i = 1 ... N-1
fp_forward = (f(2:N)-f(1:N-1))./(x(2:N)-x(1:N-1));

% Define backwards difference for i = N
fp_forward(end+1) = (f(N)-f(N-1))/(x(N)-x(N-1));

% Note that I could have used 'N' as the vector size measure instead of
% 'end'.

% Define fowards differences for i = 1
```

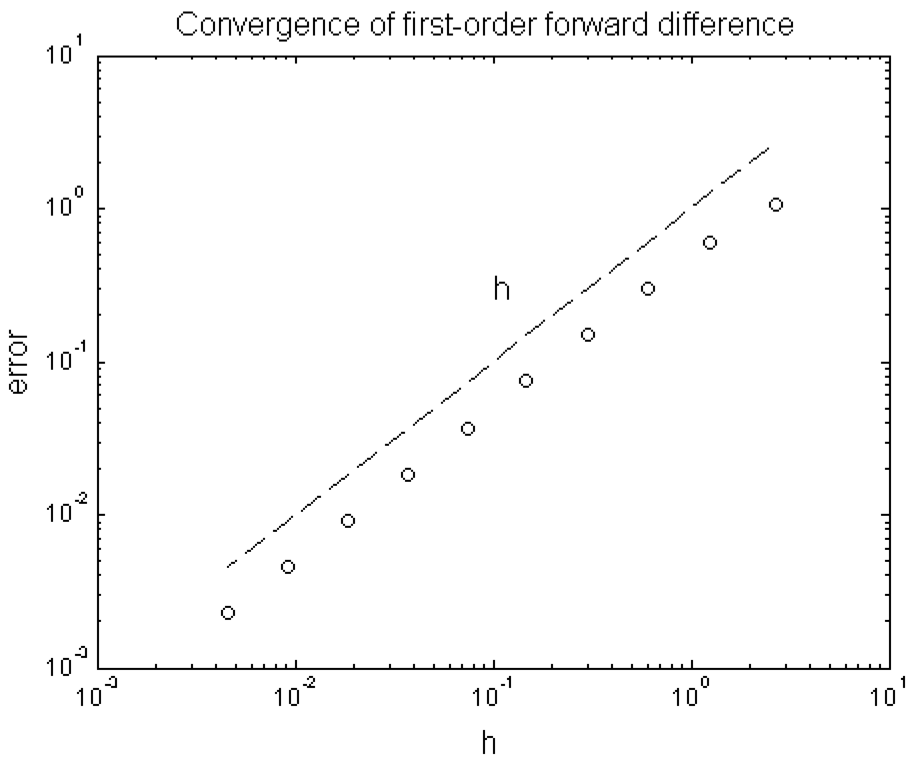
```
fp_backward = (f(2)-f(1))/(x(2)-x(1));

% Define backwards differences from i = 2 to N
fp_backward(2:N) = [f(2:N)-f(1:N-1)]./(x(2:N)-x(1:N-1));

figure(2)
plot(xs, myfuncp(xs), 'k', 'LineWidth', 2);
hold on
plot(x, fp_forward, 'b--');
plot(x, fp_backward, 'r');
hold off
xlabel('x', 'FontSize', 16);
ylabel('d/dx sin(x)', 'FontSize', 16);
title(['Discretization size = ', num2str(x(2)-x(1))], 'FontSize', 16);
legend('exact', 'forward', 'backward', 'Location', 'North');
```



Error analysis

We would expect that as more points are included, the accuracy gets better.

```
Nvec = 2.^(3:12);

for j = 1:length(Nvec)

    M = Nvec(j);
    x = linspace(0, 6*pi, M);
    f = myfunc(x);

    h(j) = x(2) - x(1);
    fp = [(f(2:M)-f(1:M-1)) (f(M)-f(M-1))]/h(j);
    error(j) = norm(fp - myfuncp(x), inf);
end

figure(3)
loglog(h, error, 'o');
hold on
loglog(h, h, 'k--');
hold off
xlabel('h', 'FontSize', 16);
ylabel('error', 'FontSize', 16);
title('Convergence of first-order forward difference', 'FontSize', 16);
text(1/10, 0.3,'h', 'FontSize', 16)
```



Taylor series

Can we explain the error? Note that

$$f(x) = f(x) + f'(x)h + f''(c)h^2$$

where c is in [0, 6*pi]. So we have

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h),$$

where O(h^2) means 'on the order of h^2'

Published with MATLAB® 7.12