Armstrong State University
Engineering Studies
MATLAB Marina –Differentiation Primer

Prerequisites

The Differentiation Primer assumes knowledge of the MATLAB IDE, MATLAB help, arithmetic operations, built in functions, scripts, variables, arrays, logic expressions, conditional structures, iteration, functions, debugging, characters and strings, cell arrays, structures, file input and output, 2D plotting, 3D plotting, interpolation, and curve fitting. Material on these topics is covered in the MATLAB Marina Introduction to MATLAB module, MATLAB Marina Variables module, MATLAB Marina Arrays module, MATLAB Marina Logic Expressions module, MATLAB Marina Conditional Structures module, MATLAB Marina Iteration module, MATLAB Marina Functions module, MATLAB Marina debugging module, MATLAB Marina Character and Strings module, MATLAB Marina Cell Arrays module, MATLAB Marina Structures module, MATLAB Marina File Input and Output module, MATLAB Marina Plotting module, MATLAB Marina Interpolation Module, and MATLAB Marina Curve Fitting Module.

Learning Objectives

1. Be able to use MATLAB to compute approximate derivatives (numerical derivatives).
2. Understand the limitations of numerical differentiation.

Terms

approximate derivative, numerical derivative, forward difference, backwards difference, central difference

MATLAB Functions, Keywords, and Operators

diff

Differentiation

The derivative of a function $f(x)$ is the rate of change of the functions with respect to x, written $\frac{df(x)}{dx}$. Numerical differentiation is typically used when a formula for the function to differentiate is not available or the derivative would be difficult to determine analytically. Numerical differentiation can yield accurate results if the function is piecewise continuous, relatively smooth, the increment between function points is relatively small, and the data is accurate. Numerical differentiation is very sensitive to inaccuracies in the data (noisy data).

There are three main methods for computing approximate (numerical) derivatives: backward difference approximation, forward difference approximation, and central or midpoint difference approximation. Given vectors x and f of the same length:

- The backwards difference approximation of the derivative of $f(x)$ is the slope of the line between the current and the previous point computed at each point x_k in x via

$$f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

- The forward difference approximation of the derivative of $f(x)$ is the slope of the line between the current and the next point computed at each point x_k in x via

$$f'(x_k) = \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}$$

- The central difference approximation of the derivative of $f(x)$ is the slope of the line between the previous and the next point computed at each point x_k in x via

$$f'(x_k) = \frac{f(x_{k+1}) - f(x_{k-1})}{x_{k+1} - x_{k-1}}$$

Approximate Derivatives using MATLAB

MATLAB has a built in function `diff` that can used to compute approximate derivatives. The function `diff` takes one argument and returns a vector of the differences of length one less than the original. For example, `diff(x)` returns `[x(2)-x(1), x(3)-x(2), … x(N)-x(N-1)]`, where N is the length of the vector. Thus given two vectors x and f, `diff(f)./diff(x)` computes either the forward (if values are interpreted as being for `x(1:end-1)`) or backward difference approximation (if values are interpreted as being for `x(2:end)`).

The MATLAB program of Figure 1a computes the forward and central difference approximations of $\frac{dy(x)}{dx}$ for the polynomial function $y(x) = x^2 + 2x + 1$. The central difference has one less point than the forward difference. Figure 1b shows the polynomial function and the approximate derivative (forward difference) of the function.

For a backwards difference approximation, the difference `dydxNum` values are interpreted for the x value range of 2 to the end. The plot statement for the forward difference in Figure 1a would be replaced with

```
plot(x,y,'b-',x(2:end),dydxNum,'g-')
```

for a backwards difference.

The approximate derivative is very close to the analytic derivative for the polynomial function and interval used. The maximum variation between `dydx` and `dydxNum` is 0.101 over this x range. For 1000 x points, the maximum variation between `dydx` and `dydxNum` is 0.010 over this x range. The approximate derivative is accurate in this example because: the polynomial function is smooth, the function values are accurate, and the independent variable interval used is small relative to how rapidly the function changes.

```
% second order polynomial function
x = linspace(-5,5,100);
y = x.^2 + 2*x + 1;
% analytical derivative
dydx = 2*x + 2;
% approximate derivative of polynomial function
diffy = diff(y);
diffx = diff(x);
% forward difference
dydxNum = diffy./diffx;
% central difference
centralDifference = (diffy(2:end)+diffy(1:end-1))./ ...
    (diffx(2:end) + diffx(1:end-1));

figure(1)
plot(x,y,'b-',x(1:end-1),dydxNum,'g-')
% plot(x,y,'b-',x(2:end-1),centralDifference,'g-')
xlabel('x'), legend('y','dy/dx (numerical)')
```

Figure 1a, MATLAB Program for Determining Approximate Derivative of Polynomial Function

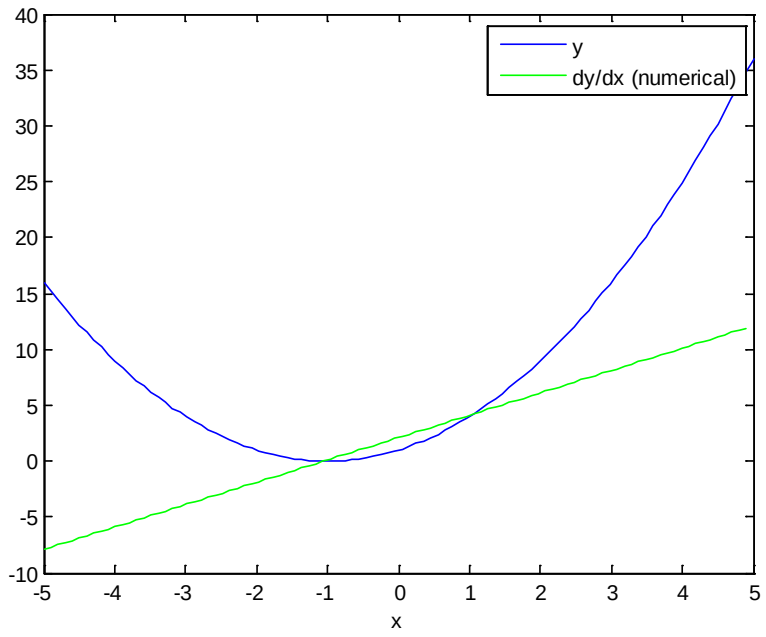

Figure 1b, Plot of Polynomial Function and Approximate Derivative of Polynomial Function

Typically, one does not have access to the function describing the data like in the example of Figure 1a, but rather one has measured data values for the function. The approximate

derivative will be less accurate for functions whose data values are less accurate and for which the interval between measurements is larger relative to how rapidly the function changes.

Additional Notes on the diff Function

The `diff` function can also be used for approximations of higher order derivatives and can be used to take differences along a dimension of a 2D array. For example:

- `diff(x, 2);` is the 2nd order difference of x and is equivalent to `diff(diff(x))`
- `diff(x, N);` is the Nth order difference of x
- `diff(X,1,1);` is the 1st order difference of x along dimension 1 (difference along columns)
- `diff(X,1,2);` is the 1st order difference of x along dimension 2 (difference along rows)

An approximate second order derivative of the polynomial function $y(x) = x^2 + 2x + 1$ could be computed using the MATLAB code of Figure 2.

```
% second order polynomial function
x = linspace(-5,5,100);
y = x.^2 + 2*x + 1;
% approximate second order derivative of polynomial function
diffx = diff(x);
d2ydx2Num = diff(y,2)./(diffx(1:end-1).^2);
```

Figure 2, MATLAB Code for Approximate 2nd Order Derivative

The second difference of y will have one less point than the first difference of x. Also note that the approximate second order derivative is not computed using

`diff(y,2)./ diff(x,2)`

which would not only be incorrect and but if the interval between the x values was constant like in the example of Figure 2, `diff(x,2)` will be a vector of all 0s.

Last modified Thursday, November 13, 2014