MATH1072

Practical 2: Exploring Sources and Bounds of Error: Linear and Quadratic Approximations

Motivation: As alluded to in your last practical, numerical approximations can give rise to error in the approximate solution when compared to the true, analytic solution. In reality, we often cannot analytically solve complex problems, so we rely on the established *theory in numerical analysis* to help us find upper bounds on the error of our numerical solution. Taylor expansion is the foundation of this rich theory.

Although this course will not delve into much depth in numerical analysis, (we'll leave that for your third-year courses), we have introduced a key concept that will be used to understand error: **The Tangent Plane** (or 1st order approximation, or linear approximation).

1 Linear and Quadratic Approximations in 1D

(a) Plot the function $z(x) = \frac{x^3}{2} + x^2$ over the interval $[-1, 1]$ by defining it implicitly and using the `linspace` and `plot` functions. We will use this figure for all of part 1. Type the command `hold on` to use the same figure for plots.

```
z = @(x) %define the function z(x) implicitly
x = linspace() %complete the argument
plot() %complete the argument
hold on %command to start using this figure
```

(b) Perform, by hand, a linear (1st order) approximation $f(x)$ of the function $z(x)$ around the point $x = 0$. Type your solution into MATLAB as an implicit function. Recall from lecture the linear approximation around the point $x = a$ is

$$f(x) = z(a) + \frac{dz(a)}{dx}(x - a).$$

Plot the approximation on the same figure.

```
dz = @(x) %implicit function representing the derivative
   of z(x)
f = @(x) %implicit function representing the linear
   approximation around x = 0.
plot() %complete the argument
```

(c) Calculate the error of your linear approximation given by,

$$E(x) = |f(x) - z(x)|$$

at the points $x = 0.5$ and $x = 0.8$. What is happening to the error as you evaluate further away from the point of expansion, $x = 0$? This is the result of *numerical error*: error coming from the (linear) approximation of the solution.

```
E1 = %calculate error at x = 0.5
E2 = %calculate error at x = 0.8
```

(d) Perform, by hand, a quadratic (2nd order) approximation $f_2(x)$ of the function $z(x)$ around the point $x = 0$. Type your solution into MATLAB as an implicit function. Recall from lecture the quadratic approximation around the point $x = a$ is

$$f_2(x) = z(a) + \frac{dz(a)}{dx}(x - a) + \frac{1}{2}\frac{d^2z(a)}{dz^2}(x - a)^2.$$

Plot the approximation on the same figure.

```
ddz = @(x) %implicit function representing the second
   derivative of z(x)
f2 = @(x) %implicit function representing the quadratic
   approximation around x = 0.
plot() %complete the argument
xlabel() %complete the argument
ylabel() %complete the argument
hold off %command to stop using this figure
```

(e) Calculate the error of your quadratic approximation given by,

$$E(x) = |f_2(x) - z(x)|$$

at the points $x = 0.5$ and $x = 0.8$. How does this *numerical error* compare to your result in 1(c)?

```
E21 = %calculate error at x = 0.5
E22 = %calculate error at x = 0.8
```

2 Linear Approximations in 2D: The Tangent Plane

(a) Plot the function $z(x,y) = x^2 + y^2(1-x)^3$ over the interval $[-1,1] \times [-1,1]$ by defining it implicitly and using the `meshgrid` and `surf` functions. We will use this figure for all of part 2. Type the command `hold on` to use the same figure for plots.

```
[X, Y] = meshgrid() %complete the argument
Z = @(x,y) %implicit function representing the function z(
   x,y)
surf() %complete the argument
hold on %command to start using this figure
```

(b) Perform, by hand, a linear (1st order) approximation $f(x,y)$ of the function $z(x,y)$ around the point $x = (-1,0)$. Type your solution into MATLAB as an implicit function. Recall from lecture the linear approximation around the point $(x,y) = (a,b)$ is

$$f(x,y) = z(a,b) + \frac{dz(a,b)}{dx}(x-a) + \frac{dz(a,b)}{dy}(y-b).$$

Plot the approximation on the same figure.

```
dZx = @(x,y) %implicit function representing the
   derivative of z(x,y) with respect to x
dZy = @(x,y) %implicit function representing the
   derivative of z(x,y) with respect to y
F = @(x,y) %implicit function representing the linear
   approximation around (x,y) = (-1,0)
surf() %complete the argument
hold off %command to stop using figure
xlabel() %complete the argument
ylabel() %complete the argument
zlabel() %complete the argument
```

(c) Calculate the error at the point of expansion $(a,b) = (-1,0)$ and at the point $(-1.1, 0.2)$. Is this expected? Why do you think they are different? This is the result of *numerical error*.

```
EE1 = %calculate error at (x,y) = (-1,0)
EE2 = %calculate error at (x,y) = (-1.1,0.2)
```

(d) Suppose we have some small error(s), $\varepsilon_1 = 0.1$, $\varepsilon_2 = 1$, in our measurement of the point of expansion $(x,y) = (a+\varepsilon_1, b+\varepsilon_2) = (-1+0.1, 0+1)$. Estimate the error on our approximation of the solution at $(x, y) = (a, b)$ using the expression given in lecture,

$$|E| \approx |\frac{dz}{dx}\varepsilon_1| + |\frac{dz}{dy}\varepsilon_2|.$$

This is the result of *experimental error*. Does the size of the error ε_2 matter?

```
EE = %calculate error using
```