Lab Work - Week 3

Math 105B Lab

Summer Session II - 2022

1 Numerical Differentiation

CODING ASSIGNMENT 6 - Forward Difference Numerical Differentiation

Produce a function with the following specifications:

NAME:	forwardNumericalDeriv_########
INPUT:	f, X
OUTPUT:	YPrime
DESCRIPTION:	YPrime= $(f'(x(1), \ldots, f'(x(n-1))$ is the vector consisting of the derivative of f evaluated at the points $x_1, \ldots, x_{n-1}$ using the forward difference method. Note that $X = (x_1, \ldots, x_n)$.
PSEUDOCODE:	See Pg. 172, equation 4.1 for a description of the computation.

CODING ASSIGNMENT 7 - Backward Difference Numerical Differentiation

Produce a function with the following specifications:

NAME:	backwardNumericalDeriv_########
INPUT:	f, X
OUTPUT:	YPrime
DESCRIPTION:	YPrime= $(f'(x(2), \ldots, f'(x(n))$ is the vector consisting of the derivative of f evaluated at the points $x_2, \ldots, x_n$ using the backward difference method. Note that $X = (x_1, \ldots, x_n)$.
PSEUDOCODE:	See Pg. 172, equation 4.1 for a description of the computation.

CODING ASSIGNMENT 8 - Midpoint Numerical Differentiation

Produce a function with the following specifications:

NAME:	centralNumericalDeriv_########
INPUT:	f, X
OUTPUT:	YPrime
DESCRIPTION:	YPrime= $(f'(x(2), \ldots, f'(x(n-1))$ is the vector consisting of the derivative of f evaluated at the points $x_2, \ldots, x_{n-1}$ using the midpoint method. Note that $X = (x_1, \ldots, x_n)$.
PSEUDOCODE:	See Pg. 175, equation 4.5 for a description of the computation.

Useful MATLAB concepts/functions

- In the functions above, we need to pass a function to our functions as an input. To do so, we can use what is called *anonymous functions*. Say we wanted to pass the function $f(x) = \frac{1}{x^4 - 5x}$ to our central numerical derivative function and evaluate the central numerical derivative at 101 linearly spaced points between -5 and 5. We would use the following code:

```
X = linspace(-5,5,101);
YPrime = centralNumericalDeriv_74964706(@(x)(1/(x^4+5*x)),X);
```

  Note that when we pass anonymous functions defined by algebraic operations, we use:

  @(x)(arithmetic expression goes here)

  If we wanted to pass an already defined function, such as e^x, we simply pass the function handle @exp to the function, i.e.

```
X = linspace(-5,5,101);
YPrime = centralNumericalDeriv_74964706(@exp,X);
```

EXERCISES

Consider the function $f(x) = \frac{1}{1+x^2}$ defined on [-5, 5] . Choose equally-spaced grid points $x_i = -5 + ih$, where $i = 0, \ldots, n$ and $h = 10/n$ with $n = 11, 21, 51, 101$.

1. Numerically approximate $f'(x_i)$ for $i = 1, \ldots, n-1$ using forward difference, backward difference, central difference approximations. Plot the results on a graph.

2. **(Participation Assessment)** What is the error between the numerical solution from part (1) and the exact solution? Plot the results versus n. How does the error scale with n? Compare the result to the error estimate. Is the actual error about the same size as the error estimate? Explain your answer.

2 Numerical Integration

CODING ASSIGNMENT 9 - Composite Simpson's Rule

Produce a function with the following specifications:

NAME:	simpsonInt_########
INPUT:	f,a,b,n
OUTPUT:	XI
DESCRIPTION:	XI is the approximate value of $\int_a^b f(x)\,dx$ found using composite Simpson's Rule.
PSEUDOCODE:	Pg. 205

CODING ASSIGNMENT 10 - Composite Trapezoidal Rule

Produce a function with the following specifications:

NAME:	trapezoidInt_########
INPUT:	f,a,b,n
OUTPUT:	XI
DESCRIPTION:	XI is the approximate value of $\int_a^b f(x)\,dx$ found using the composite Trapezoidal Rule.
PSEUDOCODE:	Modify coding assignment 9 using the rule on pg. 193.

Useful MATLAB concepts/functions

- To check if an integer is even or odd we can use the remainder mod m function. For example, to compute the remainder of the integer i modulo 2, we would use the MATLAB command:

  mod(i,2)

  This function returns 1 if i is odd, and 0 if i is even. Since MATLAB (and most other computing languages) interpret 1 as a true value, to check if i is even, we would use the command:

  ∼mod(i,2)

  which performs the logical negation of mod(i,2). That is, it outputs 1 if i is even, and 0 if i is odd.

EXERCISES

Consider the integral $\int_1^2 x \ln x \, dx$.

1. Calculate the exact value of the integral.
2. Use the trapezoid rule and Simpson's rule over the whole interval to evaluate the integral. What is the error? Compare the result to the error estimate.
3. For $n = 10, 20, 50, 100$, use trapezoidInt_######## and simpsonInt_######## to approximate the given integral. Produce a single plot showing the output of these algorithms as you vary n (i.e. put n on the x-axis). There will be two curves on this plot: one for the composite Simpson's rule and one for the composite trapezoid rule.
4. Compare the error to the exact error estimates given in the book. Produce one plot showing the error of the numerical quadrature as a function of n. There will be two curves on this plot: one for the composite Simpson's rule and one for the composite trapezoid rule.