Code differentiation

Homework #1

Topical research themes 2 – Differentiable Programming

1 Introduction

In this assignment, you will delve into the practical applications of differentiation. You will implement and evaluate numerical differentiation for vector-valued and explore forward mode automatic differentiation for multivariate functions.

2 Numerical differentiation (2 %)

Implement numerical differentiation for vector valued functions

1. Use the programming language of your choice which supports such implementation.
2. The function called `numericalDerivative` takes 2 – 3 parameters:
   a. vector valued function `f` to differentiate,
   b. a vector `vec` at which to differentiate `f`, and
   c. a small number `h` representing the finite difference interval which is optional. If it is not passed, consider some default value.
3. The function `f` should be defined such that it takes a vector as input and returns a vector as output.
4. Implement the function to compute the Jacobian matrix as described, ensuring that each element of the matrix is calculated correctly.
5. Consider edge cases, such as small values of `h` leading to numerical instability.

Test your implementation

1. Create a test function, for example, `f(vec) = [sin(x) + cos(y), cos(x) - sin(y)]` where `vec = [x, y]`.
2. Test your `numericalDerivative` function at a specific point like `vec = [π/4, π/3]` with a small `h` such as `0.0001`.
3. Print the output Jacobian matrix.

Verify correctness

- Calculate the expected results manually or using a mathematical tool and compare them with the results produced by your function.
- Include at least five verification case in your homework where you detail both the expected and actual outputs.

3 Forward Mode Automatic Differentiation (3 %)

Implement forward mode automatic differentiation for multivariate functions

1. Use the programming language of your choice which supports such implementation.
2. Implement differentiable variable data structure (or class) with supporting functions which handle evaluation and differentiation of multivariate functions.
3. Implement functions for addition `add`, multiplication `mul`, power `pow`, sinus `sin` and cosine `cos` derivation.

Test your implementation

1. Test with few examples such as for specific variable values:
   `f(x, y) = 3x + 4y + 5`
   `f(x, y) = 3xy + 5`
   `f(x, y, z) = 5x + 3y + 4xyz`
   `f(x, y) = 2sin(x) + 3cos(y)`
2. Print the function values and derivative function value.

Verify correctness

- Calculate the expected results manually or using a mathematical tool and compare them with the results produced by your function.
- Include at least five verification case in your homework where you detail both the expected and actual outputs.