TF502: Numerical Analysis
Homework 3

Prof. Boris Houska

Deadline: Oct 31, 2017

This homework is about algorithmic differentiation of factorable functions over the library

$$\mathcal{L} = \{+, *, \sin, \cos\} \ .$$

Here, a function $f : \mathbb{R} \to \mathbb{R}$ is called factorable over $\mathcal{L}$, if it can be composed from a finite number of atom operations from the library $\mathcal{L}$. For example, the JULIA code

```
function f(x)
a = x*x;
b = cos(a);
c = a + b;
return c;
```

is a factorable function over the library $\mathcal{L}$, which could also be written in the form

$$\forall x \in \mathbb{R}, \qquad f(x) = x^2 + \cos(x^2) \ ,$$

if you prefer mathematical syntax.

1. Prove that all factorable functions $f : \mathbb{R} \to \mathbb{R}$ over the library $\mathcal{L}$ are infinitely often differentiable. Also prove that the library $\mathcal{L}$ is closed under differentiation, i.e., all derivatives, $f'(x), f''(x), \ldots$ are factorable functions over the library $\mathcal{L}$, too. (*Hint: write down a generic algorithmic differentiation rule and use induction.*) **(10 points)**

2. Write a Julia code for algorithmic differentiation in forward mode by using the syntax

   ```
   function ADforward(f)
   ...
   df = ...
   return c;
   ```

   Your code should return the derivative of any factorable input functions $f : \mathbb{R} \to \mathbb{R}$ over the library $\mathcal{L}$. You can store the function `ADforward` in a file named `ADforward.jl`. Test your code with simple functions first, e.g., run the code

   ```
   include("ADforward.jl")
   f(x) = sin(x)
   df = ADforward(f)
   plot(df)
   ```

   to check whether your code is working for the atom functions. If you succeed debugging, test your code with more complicated functions. Please also check whether your code works recursively, e.g., the code

```
include("ADforward.jl")
f(x) = sin(x)
df = ADforward(f)
ddf = ADforward(df)
plot(ddf)
```

should return the second derivative of your input function. Your code should

- be bug-free and compute derivatives correctly (we will automatically check your code with several test functions) **(60 points)**
- be readable, well-structured, and brief (you should include comments in your code, and break your overall code into a reasonable amount of sub-functions) **(10 points)**