Algorithmic Differentiation
Exercise 2

Consider the vector-valued multivariate function $F : x \in \mathbb{R}^3 \mapsto y \in \mathbb{R}^3$:

$$y = F(x) = \begin{bmatrix} x_1^2 * x_2 + \sin(x_1 * x_2 * x_3) + 0.415145069 \\ \exp(|x_1 * x_2 * x_3|) + x_1 * x_2 * x_3 - 1.066476035 \\ x_1 * x_2 + x_1^2 * x_2^2 * x_3^2 + 0.194335938 \end{bmatrix}.$$

Single-Assignment-Code (SAC)

a) Implement the above function as a single-assignment-code in C. Name the intermediate variables in your code as `v1,v2,v3,...`

Tangent expressions and Jacobian

b) Derive tangent expressions for all the intermediate variables and finally for the vector of dependent variables y.

c) Using the tangent expressions, implement a C function, which computes the Jacobian matrix of F at a point x.

d) Validate your Jacobian implementation using finite difference approximations at the point $x = (\frac{1}{4}, -\frac{5}{4}, \frac{11}{10})^\top$.

Solving nonlinear equations using the damped-Newton method

e) Implement a C-program to find the roots of $F(x) = 0$ using the damped-Newton method. As the starting point take $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)})^\top = (\frac{1}{2}, -\frac{1}{2}, \frac{1}{2})^\top$.

Damped-Newton method:

A system of nonlinear equations $F(x) = 0$, can be solved using the damped-Newton iterations:

$$x^{(k+1)} = x^{(k)} - \alpha * J^{-1}(x^{(k)}) * F(x^{(k)}),\ k = 0, 1, \dots,$$

where $\alpha \in (0, 1]$ is a positive scalar. The matrix J in the above equation is known as the **Jacobian** matrix and it is defined as:

$$J(x^{(k)}) = \begin{bmatrix} \frac{\partial y_1(x^{(k)})}{\partial x_1} & \frac{\partial y_1(x^{(k)})}{\partial x_2} & \frac{\partial y_1(x^{(k)})}{\partial x_3} \\ \frac{\partial y_2(x^{(k)})}{\partial x_1} & \frac{\partial y_2(x^{(k)})}{\partial x_2} & \frac{\partial y_2(x^{(k)})}{\partial x_3} \\ \frac{\partial y_3(x^{(k)})}{\partial x_1} & \frac{\partial y_3(x^{(k)})}{\partial x_2} & \frac{\partial y_3(x^{(k)})}{\partial x_3} \end{bmatrix}.$$

Either, submit your implementation with a result file and a running code example to max.sagebaum@scicomp.uni-kl.de or visit the exercise class on 18.11.2019 and do/present it there.