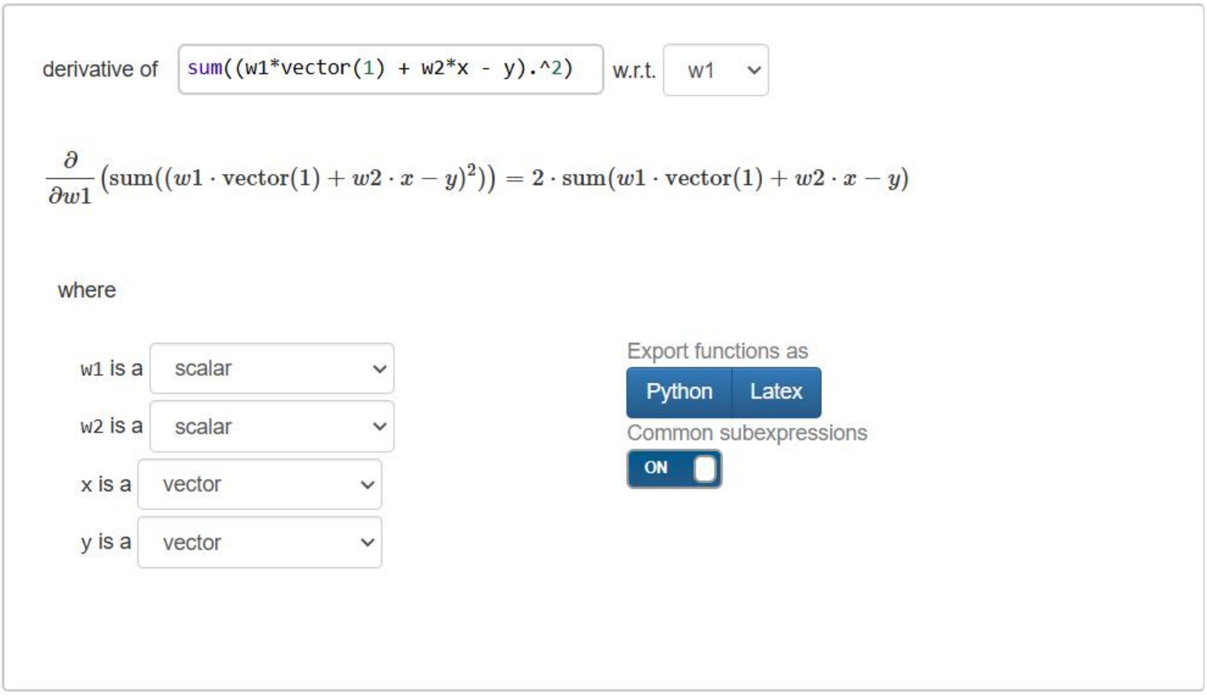
Exercise 1

a)

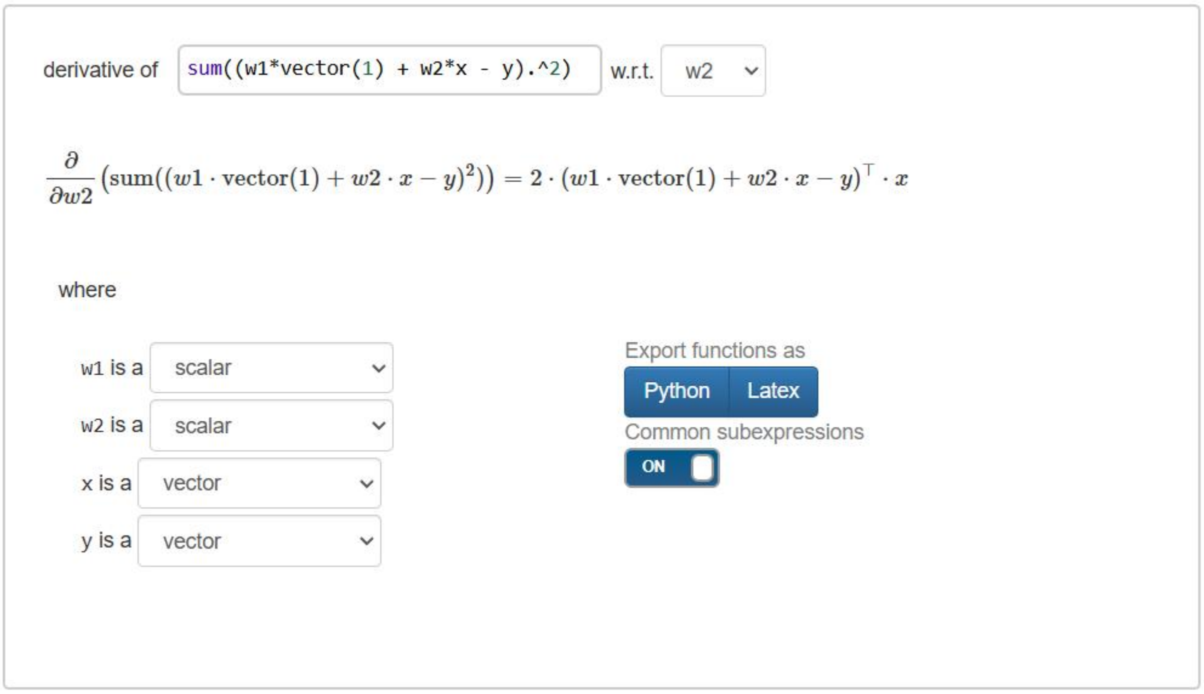


derivative of `sum((w1*vector(1) + w2*x - y).^2)` w.r.t. w1

$$\frac{\partial}{\partial w1}(\mathrm{sum}((w1 \cdot \mathrm{vector}(1) + w2 \cdot x - y)^2)) = 2 \cdot \mathrm{sum}(w1 \cdot \mathrm{vector}(1) + w2 \cdot x - y)$$

where

w1 is a	scalar
w2 is a	scalar
x is a	vector
y is a	vector



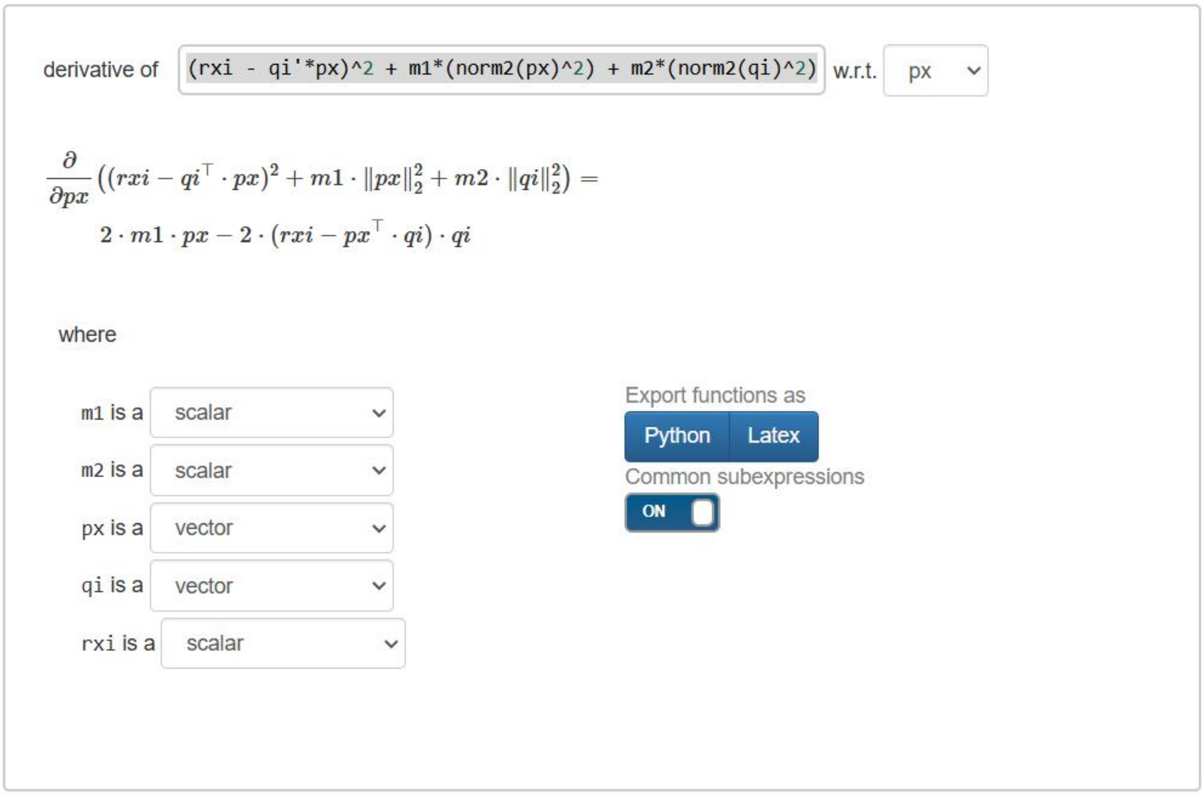
derivative of `sum((w1*vector(1) + w2*x - y).^2)` w.r.t. w2

$$\frac{\partial}{\partial w2}(\mathrm{sum}((w1 \cdot \mathrm{vector}(1) + w2 \cdot x - y)^2)) = 2 \cdot (w1 \cdot \mathrm{vector}(1) + w2 \cdot x - y)^\top \cdot x$$

where

w1 is a	scalar
w2 is a	scalar
x is a	vector
y is a	vector

b)

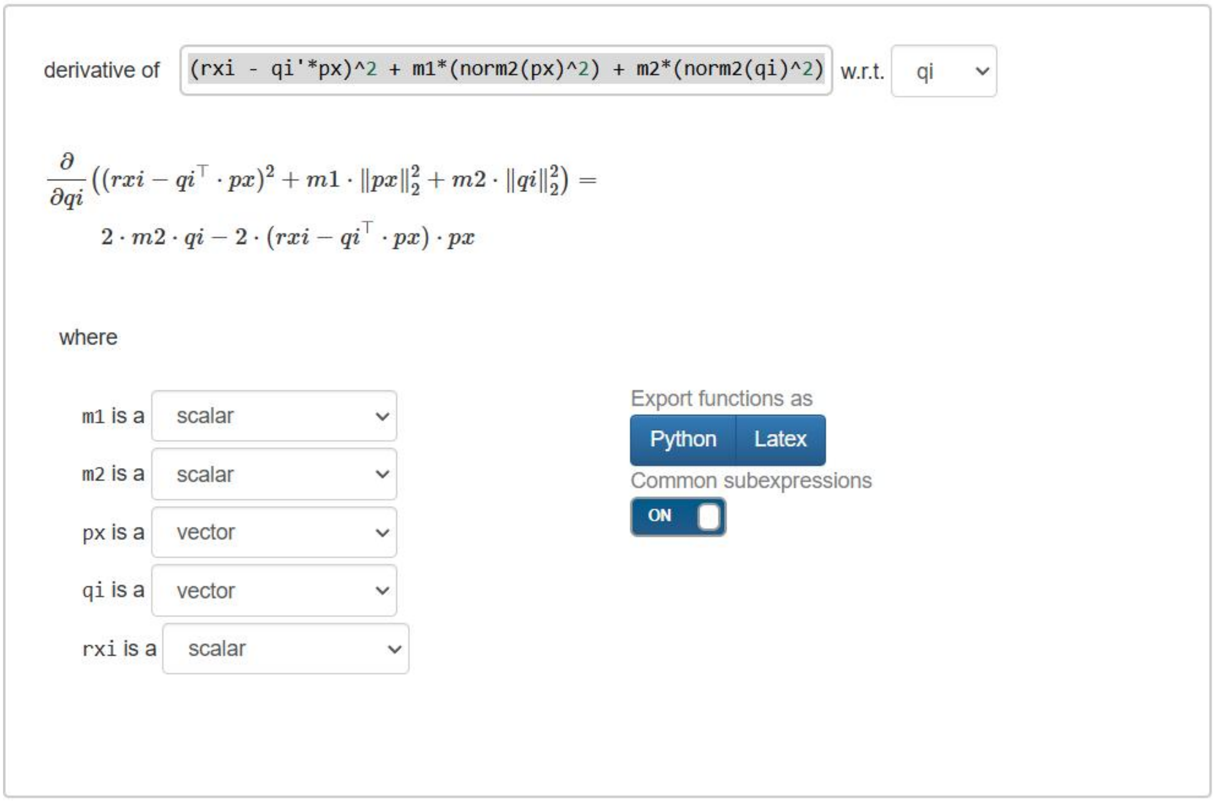


derivative of `(rxi - qi'*px)^2 + m1*(norm2(px)^2) + m2*(norm2(qi)^2)` w.r.t. px

$$\frac{\partial}{\partial px}\left((rxi - qi^\top \cdot px)^2 + m1 \cdot \|px\|_2^2 + m2 \cdot \|qi\|_2^2\right) =$$
$$2 \cdot m1 \cdot px - 2 \cdot (rxi - px^\top \cdot qi) \cdot qi$$

where

m1 is a	scalar
m2 is a	scalar
px is a	vector
qi is a	vector
rxi is a	scalar



derivative of `(rxi - qi'*px)^2 + m1*(norm2(px)^2) + m2*(norm2(qi)^2)` w.r.t. qi

$$\frac{\partial}{\partial qi}\left((rxi - qi^\top \cdot px)^2 + m1 \cdot \|px\|_2^2 + m2 \cdot \|qi\|_2^2\right) =$$
$$2 \cdot m2 \cdot qi - 2 \cdot (rxi - qi^\top \cdot px) \cdot px$$

where

m1 is a	scalar
m2 is a	scalar
px is a	vector
qi is a	vector
rxi is a	scalar

Exercise 2

a) The purpose of automatic differentiation (AD) is to compute differential values of functions efficiently. While computing derivatives, it assigns intermediate variables and uses these intermediate variables to reach the final function.
   The main difference between numerical differentiation is that AD operates on symbols, but numerical differentiation operates on computing finite differences between two nearby points. Numerical differentiation uses a very small “h” value even though h theoretically should be infinitely close to 0. In result, AD results are closer to the true value compared to numerical differentiation.
   Symbolic differentiation operates on symbols just like AD. However, it is not as efficient as AD. Symbolic differentiation calculates derivative result with symbols which cost extra computations for the algorithm. AD does not calculate final derivative in symbols; however, it requires fewer computation to reach numerical derivative value.

b) Forward mode of AD starts with the input variables and creates intermediate variables via combining input variables with operations. It continues combining until it reaches all output variables. Then it calculates derivatives of the intermediates for a chosen input value and reach the output value.
   Reverse mode of AD also starts with creating intermediate variables and reaching output variables. However, it calculates derivatives of the intermediates according to output variables and reach input variables.
   In forward mode, while calculating derivatives of intermediates, some output variables may contain mutual intermediates which could speed up computations. Therefore, in forward mode having many output variables increases efficiency. In reverse mode, output variables can reach multiple input variables so having multiple input variables increases efficiency.

c) Time complexity of forward mode is dependent on input variables and time complexity of reverse mode is dependent on output variables. So forward mode is better at bigger input variable count and reverse mode is better at bigger output variable count.

d) An example of when checkpointing would be essential, could be a neural network. If the network is too large, it might be hard to store all the data. Thanks to checkpointing strategies, we could store only a few layers of the network and reduce memory consumption significantly.