Lipschitz continuity

Which of the following functions are globally Lipschitz continuous? ($f_i\colon \mathbb{R} \to \mathbb{R}, x \mapsto f_i(x)$, $i = 1, \ldots, 4$)
Choose all that apply.

(a) $f_1(x) = \max(0, x)$

(b) $f_2(x) = \mathrm{sign}(x)$

(c) $f_3(x) = \sqrt{x^2}$

(d) $f_4(x) = \sqrt{|x|}$

Convexity of functions

Which of the following functions are convex?
($f_i\colon \mathbb{R} \to \mathbb{R}, x \mapsto f_i(x)$, $i = 1, \ldots, 4$)
Choose all that apply.

(a) $f_1(x) = \max(0, x)$

(b) $f_2(x) = \exp(x^2)$

(c) $f_3(x) = \sqrt{x^2}\sin(x)$

(d) $f_4(x) = \sqrt{|x|}$

Optimal Control Problems - Sequential approach

We consider an optimal control problem (OCP) in discrete time. The state and control vectors at each time instance have dimension $n_x = 4$ resp. $n_u = 2$, and the problem has time horizon $N = 10$. The initial value is eliminated as $x_0 = \bar{x}_0$. We choose the **sequential** approach for the formulation of the OCP, and collect all decision variables in the vector $w \in \mathbb{R}^{n_w}$. As answer, please enter the dimension n_w of this vector.

$n_w = \ldots?$

Optimal Control Problem - Simultaneous approach

We consider an optimal control problem (OCP) in discrete time. The state and control vectors at each time instance have dimension $n_x = 4$ resp. $n_u = 2$, and the problem has time horizon $N = 10$. The initial value x_0 is kept as a variable. We choose the **simultaneous** approach for the formulation of the OCP, and collect all decision variables in the vector $w \in \mathbb{R}^{n_w}$. As answer, please enter the dimension n_w of this vector.

$n_w = \ldots ?$

Newton's method

Regard the following equation system:

$$\frac{1}{x} - y = 0,$$
$$x^4 + y^4 - 1 = 0.$$

We summarize it as $F(w) = 0$, where $w = (x, y)$ and $F : \mathbb{R}^2 \to \mathbb{R}^2$. We want to solve this root finding problem using (exact) Newton's method. Our current iterate is $w_k = (\frac{1}{2}, 0)$ (i.e., $x_k = \frac{1}{2}$, $y_k = 0$.) Use Newton's method to find the next iterate $w_{k+1} = w_k + \Delta w_k$, where $\Delta w_k = (\Delta x_k, \Delta y_k)$.
As answer, please enter the value of Δy_k:

$\Delta y_k = \ldots ?$

Note: You can solve this task by pen&paper or on the computer.

If necessary, round the value to two decimal digits after the decimal separator by simply dropping the superfluous digits (e.g. "3.149" becomes "3.14").

Computing Derivatives

Considered the following function, as constructed in Matlab resp. Python and CasADi:

```
% MATLAB
x = MX.sym('x');
y = 1 + exp(x);
for k = 1:5
    y = y * (sin(k*x) + cos(x));
end
f = Function('f', {x}, {y});
```

```
# python
import casadi as ca
x = ca.MX.sym('x')
y = 1 + ca.exp(x)
for k in range(1,6):
    y = y * (ca.sin(k*x) + ca.cos(x))
end
f = ca.Function('f', [x], [y])
```

Use CasADi to compute its derivative $f'(x)$ and evaluate it at $\bar{x} = 1.7$.
As answer, please enter the value of $f'(\bar{x})$:

$$f'(\bar{x}) = \ldots?$$

If necessary, round the value to two decimal digits after the decimal separator by simply dropping the superfluous digits (e.g. "3.149" becomes "3.14").

Numerical Integration

Consider the following ordinary differential equation,

$$\dot{u} = u - uv,$$
$$\dot{v} = uv - v,$$

describing the interaction of a predator population v with a prey population u, where $u,\ v \in \mathbb{R}$ are the size of the respective population (for simplicity we allow non-integer values)[a]. We collect them in state $x = (u, v)$.

The initial state is given as $x_0 = (0.3, 0.4)$. Use the Runge-Kutta method of fourth order (RK4) to integrate this differential equation, with a step length of $h = 0.1$. Compute the state of the system after $N = 150$ integration steps. As answer, please enter the corresponding prey population size u_N after N steps.

$u_N = \ldots ?$

If necessary, round the value to two decimal digits after the decimal separator by simply dropping the superfluous digits (e.g. "3.149" becomes "3.14").

---

[a] Also known as the Lotka-Volterra equations, cf. `https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations`

Optimization using CasADi

Regard the following optimization problem:

$$\begin{aligned} \min_{w \in \mathbb{R}^2} \quad & (w_1 - 1)^2 + w_2^2 \\ \text{s.t.} \quad & 2w_1^2 + w_2^2 \geq 1, \\ & w_2 - w_1^2 = 0, \end{aligned}$$

where $w = (w_1, w_2)$. Use CasADi and the solver IPOPT to find the minimizer $w^* = (w_1^*, w_2^*)$ of this problem.
As answer, please enter the value of w_2^*:

$w_2^* = \ldots ?$

If necessary, round the value to two decimal digits after the decimal separator by simply dropping the superfluous digits (e.g. "3.149" becomes "3.14").