EOSC 213 Quiz 4 March $28^{\text{th}}$, 2019
Name: ID:

# EOSC 213 - Quiz

---

**Instructions (30 points in total)**

- Read the examination before beginning.
- Calculators are allowed (if you don't have one, just give the expression to type in a calculator).
- You have exactly 45 minutes for the examination.
- Be as precise and clear as possible.
- This is a closed book examination.
- If you get stuck, make an assumption, state what it is and try to carry on.

---

**Question 0: English is a funny language test**

**Q0** Complete the sentence with your favourite word. *If vegetarians consume vegetables, then humanitarians consume .....* [**1 point**]

**Question 1: Taylor's approximation**
Let us consider the exponential function described in equation 1

$$f(x) = \exp(x). \tag{1}$$

**Q1a** Compute a first order approximation to the first derivative of the exponential function at $x = 0$ ($f'(0)$) using $\Delta x = 0.2$. [**2 points**]

**Q1b** Compute a second order approximation to the first derivative of the exponential function at $x = 0$ ($f'(0)$) using $\Delta x = 0.2$. [**2 points**]

**Question 2: Diffusion**
The 1D transient diffusion equation with homogeneous diffusion coefficient can be written as:

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}. \tag{2}$$

Let us consider the physical domain $x \in [-1; 1]\,\mathrm{m}$, with the specified boundary conditions $x(1) = x(-1) = 0$. Consider the following function:

$$c(x,t) = c_0 \cos\left(\frac{\pi x}{2}\right) \exp\left(-\alpha t\right) \tag{3}$$

**Q2a** Show that the function described in equation 3 satisfies the boundary conditions at all times [**2 points**]

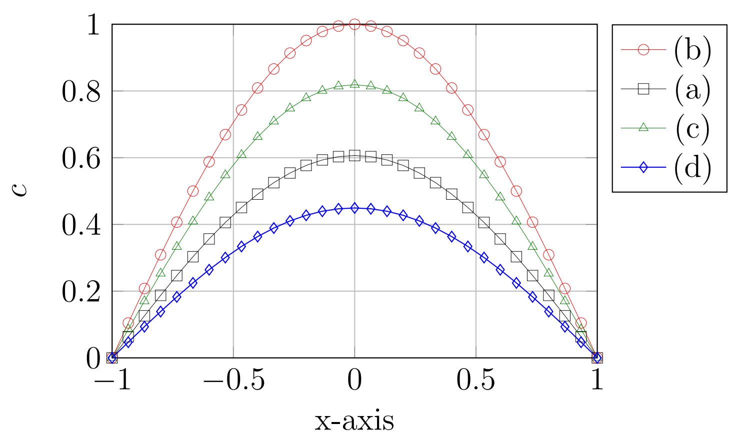
**Q2b** Function described in equation 3 is ... a solution to the diffusion problem. **Justify** your answer. [**2 points**]

**(a)** Always

**(b)** Sometimes

**(c)** Never

**Q2c** Concentrations were measured at four different times and are represented in the graph below. The messy person who did the measurements does not remember at which time these measurements were taken. Can you help him, using your physical intuition? Give the temporal sequence of the 4 curves. [**2 points**]




**Q2d** What can you say about the total mass evolution in the system with time? Is that consistent with the boundary conditions? [**2 points**]

**Q2e** Use your physical reasoning (and equation 3) to describe the asymptotic/final solution. [**2 points**]

**Question 3: Conservation equation** The continuity equation says:

$$\frac{\partial \,\text{Stuff}}{\partial t} = -\overrightarrow{\nabla} \cdot \overrightarrow{\jmath}\,, \tag{4}$$

or, in cartesian coordinates

$$\frac{\partial \, \text{Stuff}}{\partial t} = -\frac{\partial j_x}{\partial x} - \frac{\partial j_y}{\partial y} - \frac{\partial j_z}{\partial z}. \tag{5}$$

$\overrightarrow{j}$ is a flux vector that describes the rate at which "stuff", the conserved quantity, fluxes (moves).

**Q3a** Provide two examples of what "Stuff", the **conserved quantity**, could represent. [**2 points**]

**Q3b** Write the PDE conservation law (either in nabla notation or in cartesian coordinates) for the case of the diffusion of a solute (not in a porous media) where the flux is given by Fick's law $\overrightarrow{j_{\text{diff}}} = -D\overrightarrow{\nabla}c$ [**3 points**]

**Q3c** What is the physical meaning of the term $\frac{\partial \, \text{Stuff}}{\partial t}$ in the case of a diffusing solute? Provide a one or two sentence(s) explanation. [**2 points**]

**Q3d** Write the PDE conservation law (either in nabla notation or in cartesian coordinates) for the case where the flux is given by advection $\overrightarrow{j_{\text{adv}}} = \overrightarrow{v}c$ [**3 points**]

## Question 4: Python

This is a question about the following code:

---

```
class Problem_Def:
    """
    this class holds the specifcation for the domain,
    including the value of the porosity
    """

    def __init__(self, nx, ny, poro, wx, wy):
        self.nx = nx
        self.ny = ny
        self.poro = poro
        self.wx = wx
        self.wy = wy


def get_spacing(nx=4, ny=3, poro=0.4, wx=10, wy=20):
    the_prob = Problem_Def(nx, ny, poro, wx, wy)
```

```
    delx = the_prob.wx / the_prob.nx
    dely = the_prob.wy / the_prob.ny
    return delx, dely
```

---

**Q4a** Given the code above, what does the following python statement print? [**1 point**]

```
print(f"{get_spacing(nx=6)}")
```

**Q4b** modify the `Problem_Def` class to incorporate `get_spacing` as an instance method [**2 points**]

That is, create a version of `Problem_Def` for which the following will work::

```
the_instance = Problem_Def()
delx, dely = the_instance.get_spacing()
```

where the new constructor has the signature::

```
def __init__(self,nx=4,ny=3,poro=0.4,wx=10,wy=20):
    ...
```

**Q4c** something [**2 points**]