Introduction to computational thinking and programming for CFD (13251)

Dr. rer. nat. Marten Klein

Chair of Numerical Fluid and Gas Dynamics, BTU Cottbus - Senftenberg

Written Module Examination – Winter Semester 2024/2025

---

1. Explain with your own words and give an example.

   (a) What does mesh generation in the context of CFD mean? (3 pts.)

   (b) What does the term "numerical error" mean? (3 pts.)

   (c) What is a 2-D array? (3 pts.)

2. The following sequence of random numbers has been obtained with a Python program:

   `seq = [0, 4, 3, 4, 7, 8, 2, 6, 0, 8, 6, 9, 5, 6, 5, 2, 1, 7, 3, 9, 1, 3]`

   (a) Sketch the histogram on paper. Label the axes and indicate the data ranges. (5 pts.)

   (b) Are the random numbers uniformly distributed or not? Explain your choice. (2 pts.)

   (c) Write a Python script that plots the histogram. You can use library functions. Add axis labels and a title. Change the plot color to "green". Save the image as "`histogram.png`". (8 pts.)

3. The following Python code generates 4-digit random numbers.

```python
seed_number = int(input("Please enter a 4-digit number: "))
number = seed_number
already_seen = []
counter = 0

while number not in already_seen:
    counter += 1
    already_seen.append(number)
    number = int(str(number * number).zfill(8)[2:6])  # zfill adds zero padding
    print(f"#{counter}: {number}")

print(f"Starting with {seed_number}," +
      f"the sequence repeats after {counter} steps with number {number}.")
```

   (a) Explain briefly, line by line, what is happening in the code. (12 pts. – *Note: It is not one point per line of code!*)

   (b) Modify the code such that not more than 1000 iterations are performed and the generated sequence is saved to a file. (4 pts.)

4. A derived quantity $Q(y)$ is defined as

$$Q(y) = y^2 \cdot \frac{\mathrm{d}u}{\mathrm{d}y} \,, \tag{1}$$

where $u(y)$ denotes the value of a known physical variable u that resides at location y. Gridded data $u_i = u(y_i)$ has been obtained and is collected in the following arrays:

```
y = np.array([-3., 0., 1., 2., 3.])             # locations
u = np.array([-129., -17., 1., 3., 55.])        # values
```

(a) Discretize the right hand side of equation (1) with the central difference approximation. (6 pts.)

(b) Evaluate the discretized expression $Q(y)$ for location `y[2]`. Sketch the problem on paper and mark the points that are used in the differencing operation. (6 pts.)

(c) Write a Python function that implements the discretized expression and give the corresponding function call for the gridded data that is given above. (6 pts.)