

## Exercise 3

### Files, Allocation

(14.11.2025)

#### Review

- |                                              |                                                     |
|----------------------------------------------|-----------------------------------------------------|
| 1. What is not allowed in an OPEN statement? | <input type="checkbox"/> ELSEIF (BTEST) THEN ; STOP |
| <input type="checkbox"/> ACTION='READWRITE'  | <input type="checkbox"/> IF ( .TRUE. ) THEN STOP    |
| <input type="checkbox"/> FILE=FILE           | 3. Which statement terminates the current loop?     |
| <input type="checkbox"/> IER=102             | <input type="checkbox"/> EXIT                       |
| 2. Which IF statement is correct?            | <input type="checkbox"/> END                        |
| <input type="checkbox"/> IF ( K != 3 )       | <input type="checkbox"/> CONTINUE                   |
- 

#### 1. Task *Files* (4 P)

Use the OPEN, WRITE, and CLOSE statements to create an ASCII text-file, that contains an  $xy$  table, where the  $x$ - and  $y$ - values are computed with help of this recursive formula:

$$y_{n+1} = x \cdot y_n - x \cdot y_n^2 \quad (1)$$

Where the  $x$ -values should start from 0.0 and run till 4.0 in steps of 0.001. For each  $x$  value calculate the sequence  $y_n$  with  $y_0 = 0.5$  up to  $n = 600$  and write out only  $y_n$  for  $n > 401$  pairwise with its corresponding  $x$ -value to a file `bifi.txt`. Finally, plot this file, e.g., with `gnuplot`, plot style is `dots`.

The first entries of this file should look like (depending on formatting):

```
0.0 0.0
0.0 0.0
0.0 0.0
```

#### 2. Task *More!* (2 P)

Write a little Fortran program that uses the ALLOCATE statement to determine the maximum memory you can allocate for your own program.

- Use a loop starting from `I=10000` up to the *largest possible* integer in steps of 1000, where in each iteration an array is allocated and then deallocated.
- The array should consist of 64-bit REALs and have dimension (rank) 2.
- Before each allocation attempt, print how much memory (in MB) is being attempted for allocation. The memory required for the array is calculated from the size of the array multiplied by the size of its individual elements.