Euler Method with FORTRAN 90

Computer Languages for Engineers

Statik und Dynamik der Tragwerke

University of Duisburg-Essen

Presenting to:

- Dr.-Ing. R. Ostwald

Presented by:

- Bhargav Sanchania (3106382)
- Vashishth Shukla (3098673)

UNIVERSITÄT DUISBURG ESSEN

Open-Minded

Contents:

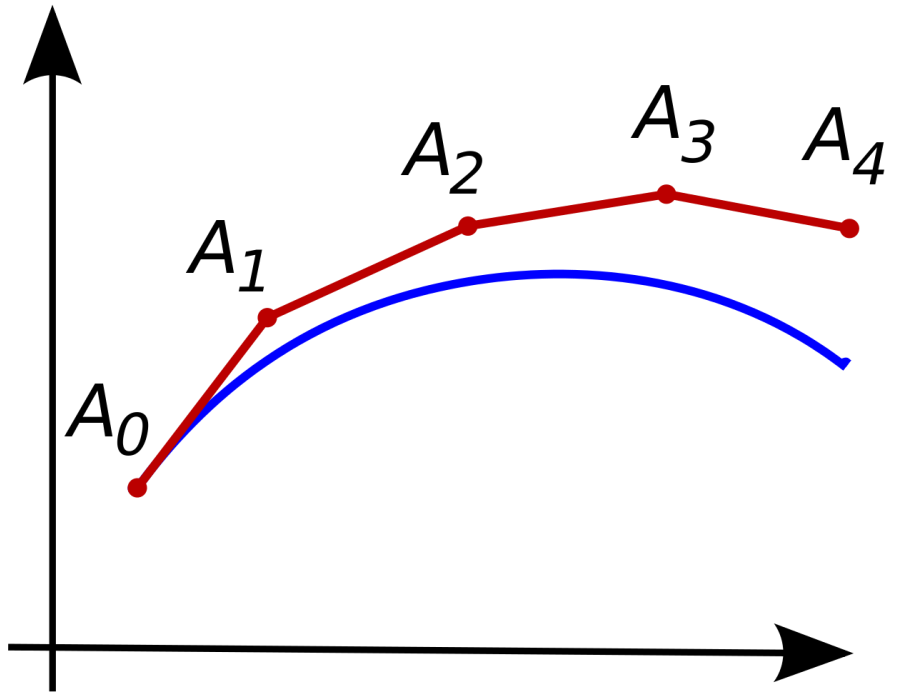
UNIVERSITÄT DUISBURG ESSEN
Open-Minded

- Euler methods
- User guide for the program
- FORTRAN 90
- Outputs
- Conclusion
- References

Euler Methods:

- Motivation:
  - Its not easy / possible to find antiderivatives of all functions
  - Computers aren't good with Calculus
  - We have some function

    $f'(y, t) = y^2 + 1 - t^2$



Euler Methods:

- Forward Euler method:

- Approximation of next value

$$y_{n+1} = y_n + \Delta t\, y'_n$$

- Advantages
  - Easy to calculate
- Limitations
  - Error explodes

Euler Methods:

- Backward Euler method:

- Approximation of next value

$$y_{n+1} = y_n + \Delta t\, y'_{n+1}$$

- Advantages
  - More accurate than forward Euler
- Limitations
  - Not easy to compute

Euler Methods:

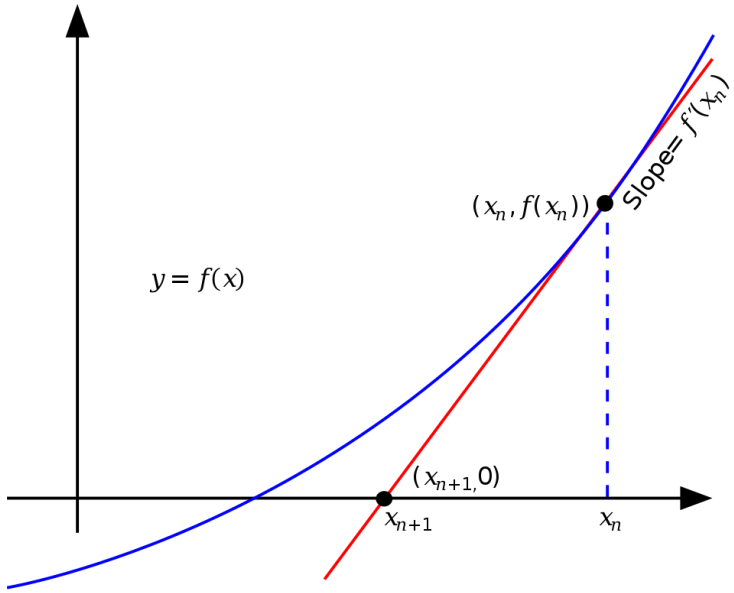
- Newton method:
  - To calculate such an equation,

$$R(y_{n+1}) := y_{n+1} - y_n - \Delta t * y'(y_{n+1}, t_{n+1}) \approx 0$$

$$y^{(k+1)}_{n+1} = y^{(k)}_{n+1} - \frac{R^{(k)}}{R'^{(k)}}$$

with

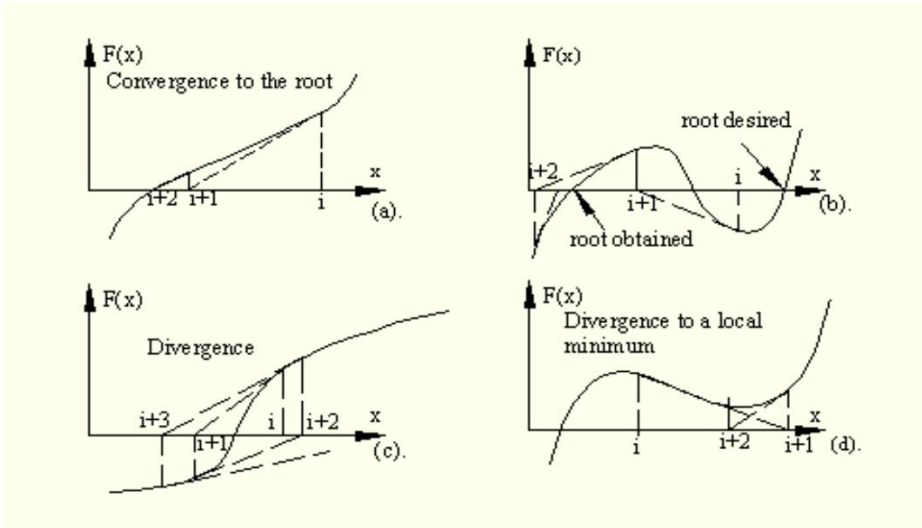
$$R'^{(k)} = \frac{\partial R(y_{n+1})}{\partial y_{n+1}} | y^{(k)}_{n+1}$$



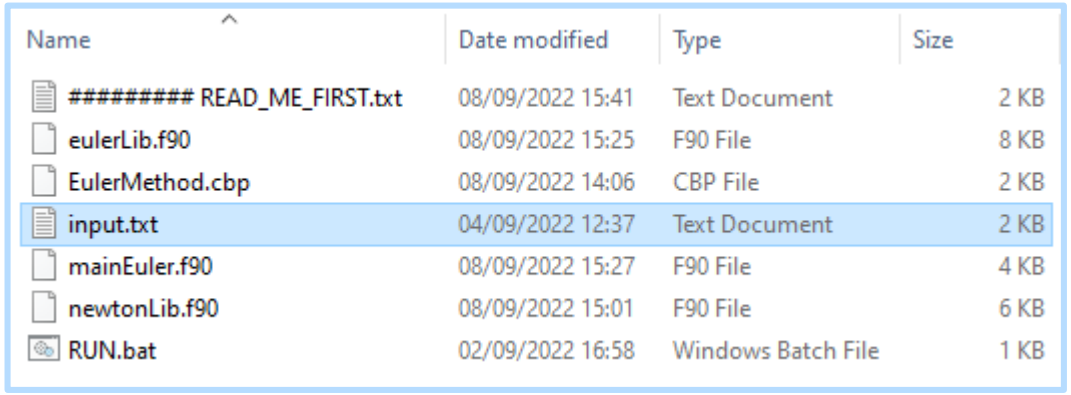
  - Advantages
    - Can slove any equation having solution

Euler Methods:

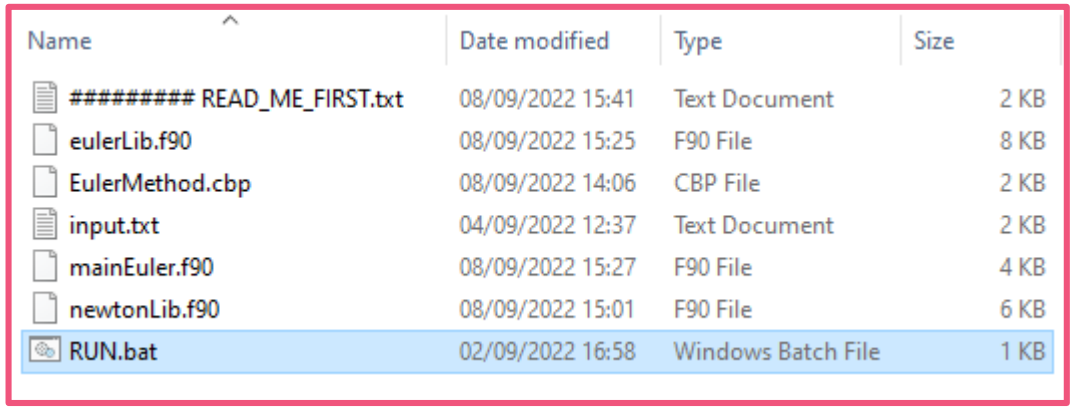
- Newton method:
  - Limitations
    - It's convergence is not guaranteed. So, sometimes, for given equation and for given guess we may not get solution.
    - Division by zero problem can occur.
    - Root jumping might take place thereby not getting intended solution.
    - Inflection point issue might occur.
    - Symbolic derivative is required.
    - In case of multiple roots, this method converges slowly.
    - Near local maxima and local minima, due to oscillation, its convergence is slow.



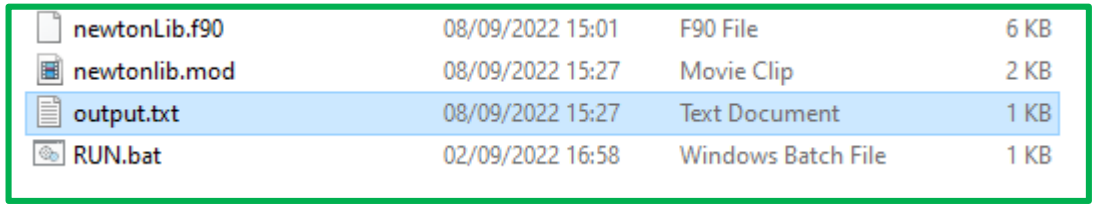
User guide for the program



Name	Date modified	Type	Size
######### READ_ME_FIRST.txt	08/09/2022 15:41	Text Document	2 KB
eulerLib.f90	08/09/2022 15:25	F90 File	8 KB
EulerMethod.cbp	08/09/2022 14:06	CBP File	2 KB
input.txt	04/09/2022 12:37	Text Document	2 KB
mainEuler.f90	08/09/2022 15:27	F90 File	4 KB
newtonLib.f90	08/09/2022 15:01	F90 File	6 KB
RUN.bat	02/09/2022 16:58	Windows Batch File	1 KB



Name	Date modified	Type	Size
######### READ_ME_FIRST.txt	08/09/2022 15:41	Text Document	2 KB
eulerLib.f90	08/09/2022 15:25	F90 File	8 KB
EulerMethod.cbp	08/09/2022 14:06	CBP File	2 KB
input.txt	04/09/2022 12:37	Text Document	2 KB
mainEuler.f90	08/09/2022 15:27	F90 File	4 KB
newtonLib.f90	08/09/2022 15:01	F90 File	6 KB
RUN.bat	02/09/2022 16:58	Windows Batch File	1 KB



newtonLib.f90	08/09/2022 15:01	F90 File	6 KB
newtonlib.mod	08/09/2022 15:27	Movie Clip	2 KB
output.txt	08/09/2022 15:27	Text Document	1 KB
RUN.bat	02/09/2022 16:58	Windows Batch File	1 KB

```
################################################################################
#                                                                              #
# this is the input file for Euler Method                                      #
# please read the 'READ_ME_FIRST.txt' first                                    #
#                                                                              #
################################################################################
# legend
#
# comments have '#' as first word in line
# format: <key> <value>
#
################################################################################
#                                                                              #
# data for the Euler                                                           #
#                                                                              #
################################################################################

# initial condition

# starting of time
t0 0.0d0

# value of y at t0
y0 -0.01d0

# h = delta(t)
h 0.3d0

# t starting point
tStart 0.d0

# t ending point
tEnd 6.d0

# --log and outputs will be added to the respective files--
# --following variables determine printing in console only--
# to see the log on Screen make logOnScreen true
logOnScreen .false.

# to see the output on Screen make outOnScreen true
outOnScreen .true.

################################################################################
#                                                                              #
# data for the Newton method                                                   #
#                                                                              #
################################################################################

# error tolerance
tol 1.0d-6

# max iterations
maxIt 20

# --log and outputs will be added to the respective files--
# --following variables determine printing in console only--
# to print newton iterations on screen
newtOnScreen .false.

# to print newton iterations in output file
newtInOut .false.
```

FORTRAN 90:

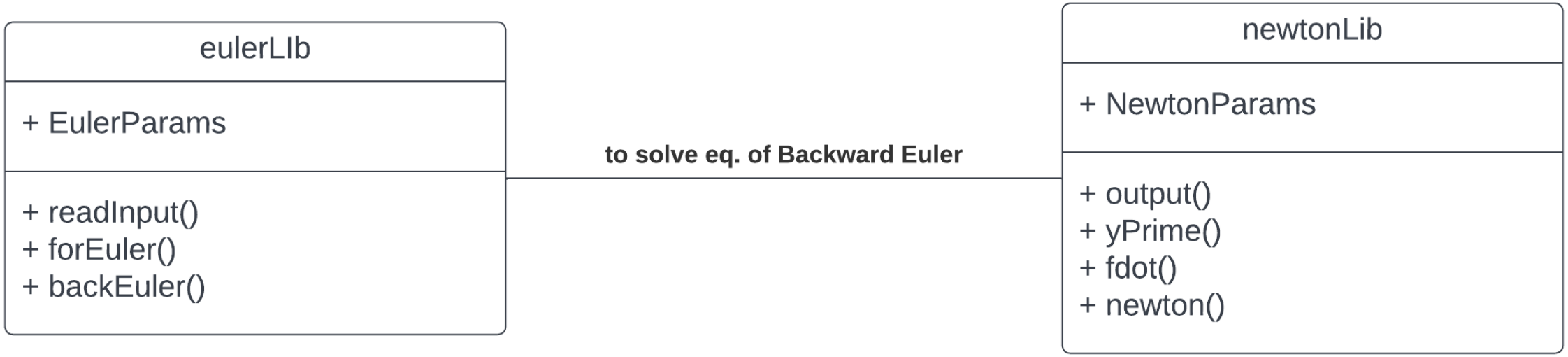
UNIVERSITÄT DUISBURG ESSEN
Open-Minded

- FORmulaTRANsform
  - Made for engineers
- Advantages
  - Compiled
  - Fast and efficient than interpreted languages
- Limitations
  - Not so pupular
  - Resources
  - Pseudo OOPs

FORTRAN 90:

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

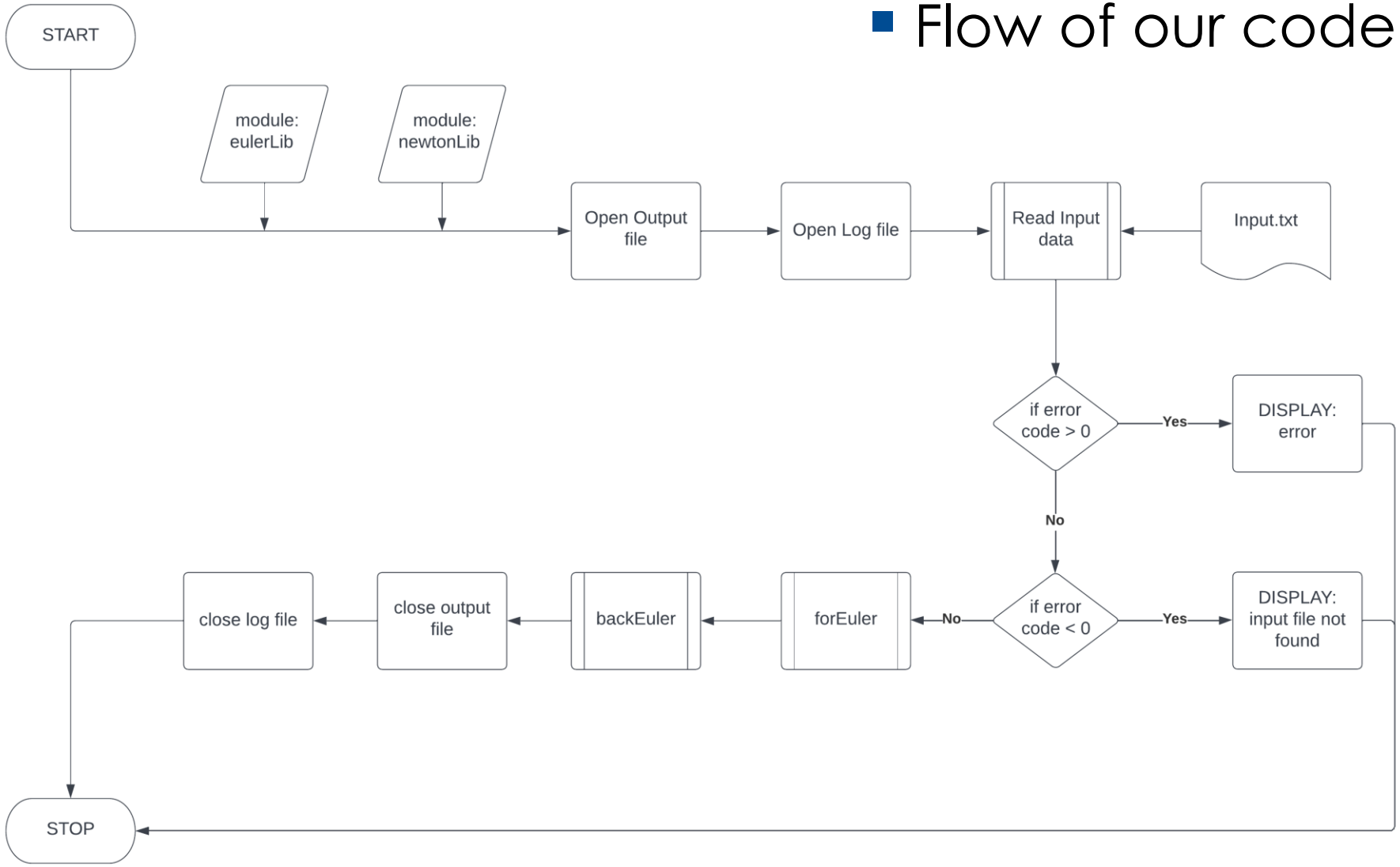
- OOPs in our code:



FORTRAN 90:

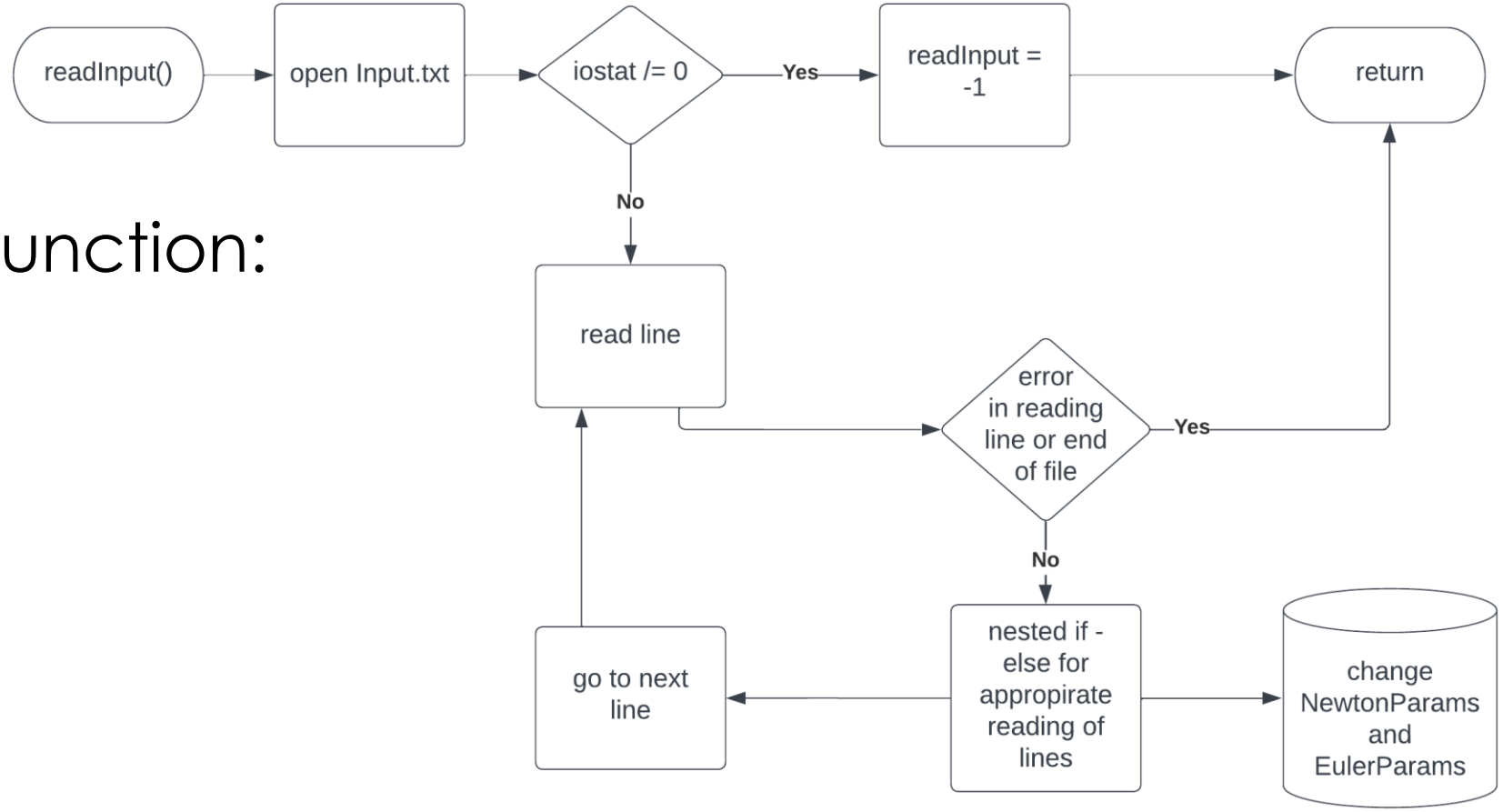
UNIVERSITÄT DUISBURG ESSEN
Open-Minded

- Flow of our code:



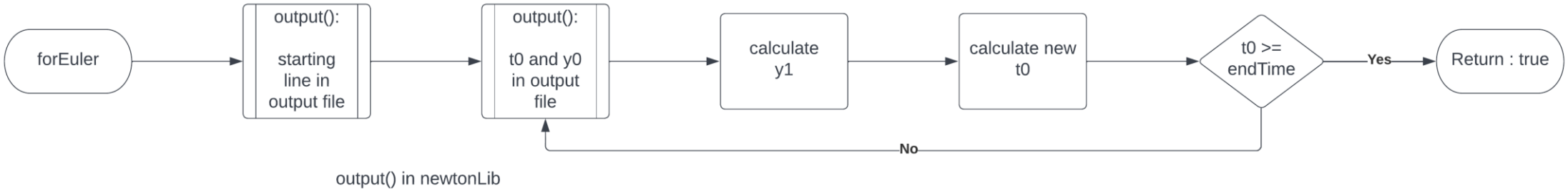
FORTRAN 90:

- Read function:

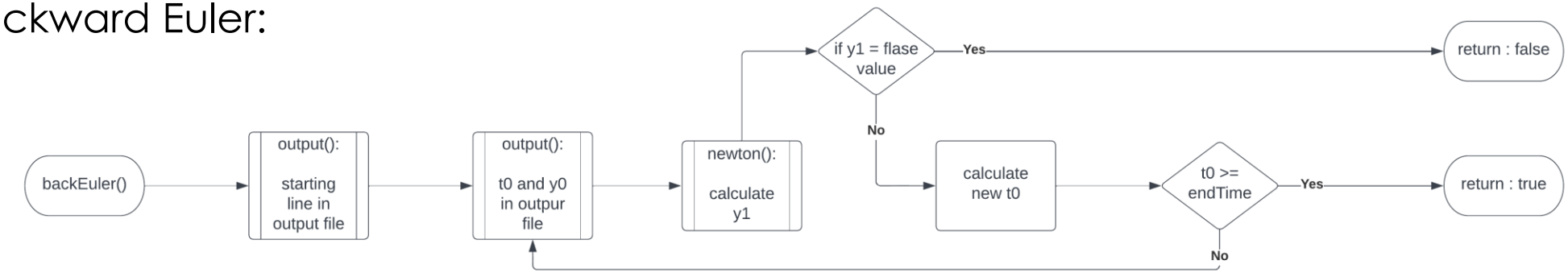


FORTRAN 90:

- Forward Euler:

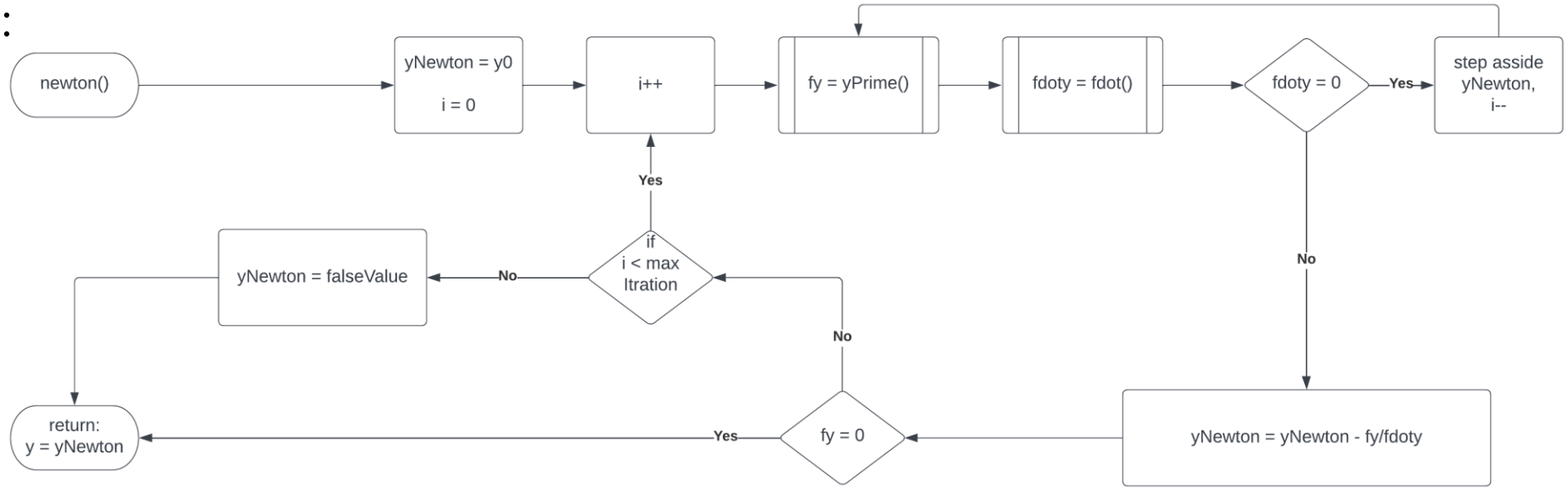


- Backward Euler:

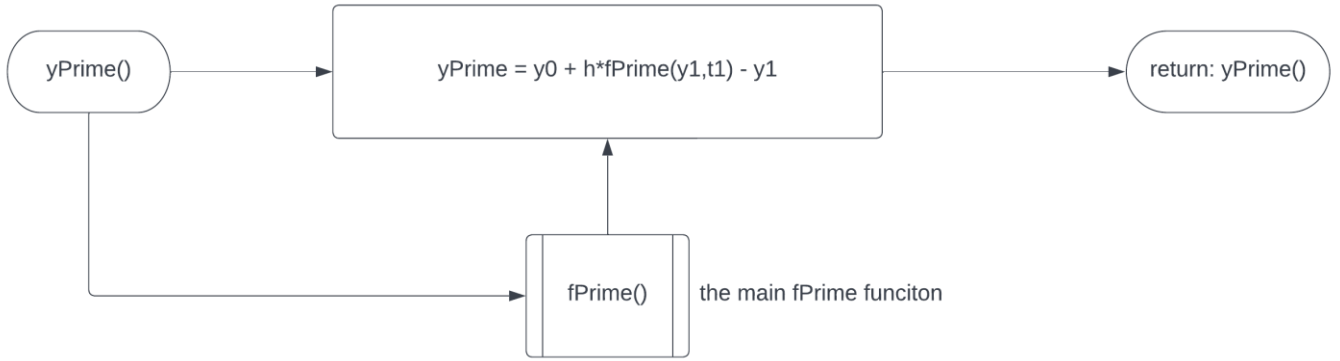


FORTRAN 90:

- Newton:



- yPrime:



Outputs:

- Forward Euler – output.txt

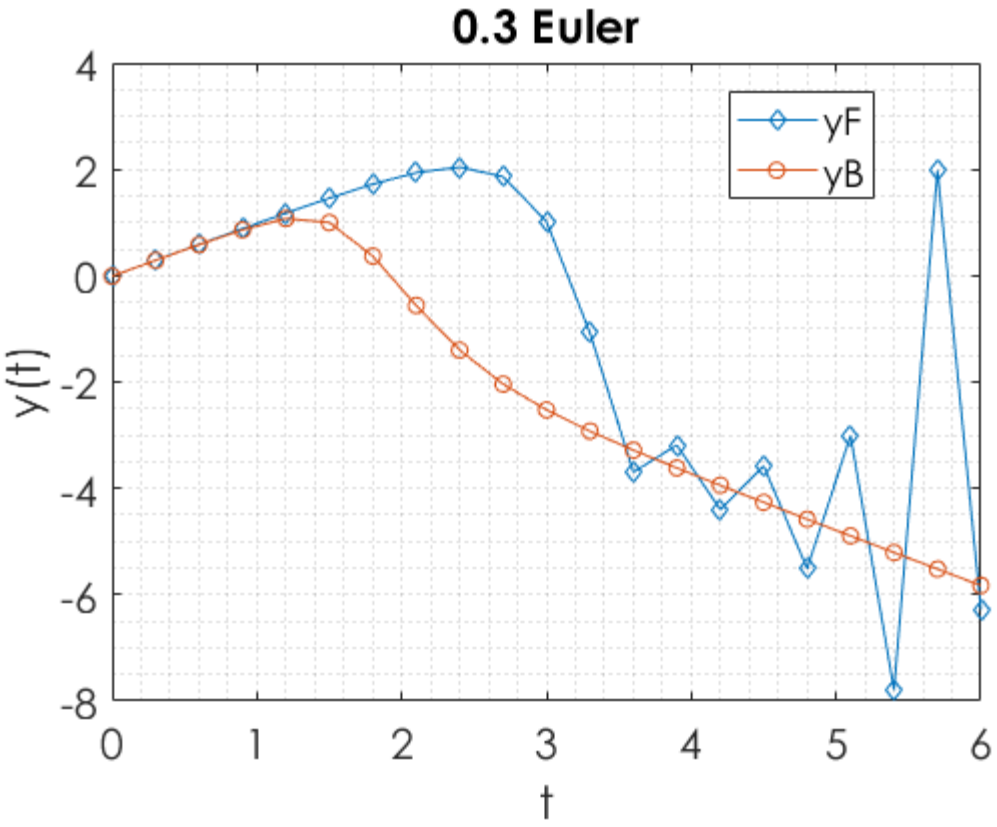
```
**** Forward Euler Method ****
---t(i)|-------y(t)
     0.00  -0.0100
     0.30   0.2900
     0.60   0.5883
     0.90   0.8841
     1.20   1.1756
     1.50   1.4581
     1.80   1.7210
     2.10   1.9376
     2.40   2.0408
     2.70   1.8623
     3.00   1.0157
     3.30  -1.0749
     3.60  -3.6953
     3.90  -3.1868
     4.20  -4.4031
     4.50  -3.5789
     4.80  -5.5114
     5.10  -3.0108
     5.40  -7.7943
     5.70   1.9830
     6.00  -6.2843
```

- Backward Euler – output.txt

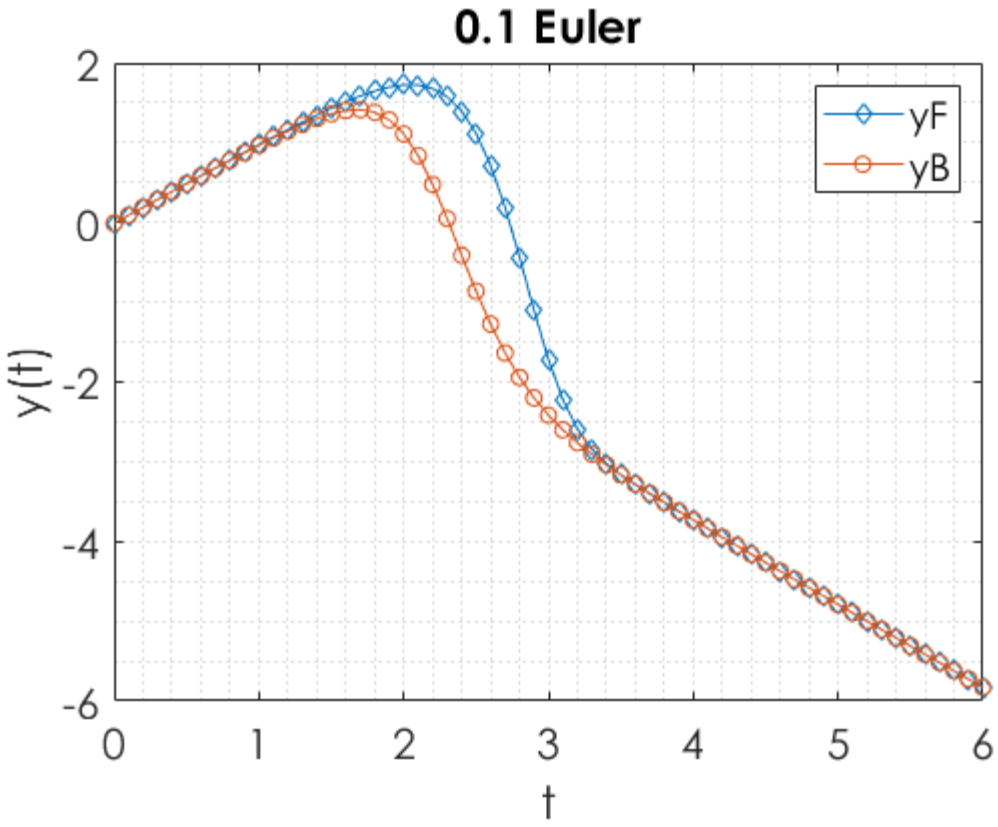
```
**** Backward Euler Method ****
---t(i)|-------y(t)
     0.00  -0.0100
     0.30   0.2879
     0.60   0.5812
     0.90   0.8602
     1.20   1.0746
     1.50   0.9989
     1.80   0.3674
     2.10  -0.5611
     2.40  -1.4006
     2.70  -2.0396
     3.00  -2.5258
     3.30  -2.9254
     3.60  -3.2820
     3.90  -3.6180
     4.20  -3.9438
     4.50  -4.2641
     4.80  -4.5808
     5.10  -4.8951
     5.40  -5.2076
     5.70  -5.5185
     6.00  -5.8282
```

Outputs:

- Plot 0.3 value of Δh

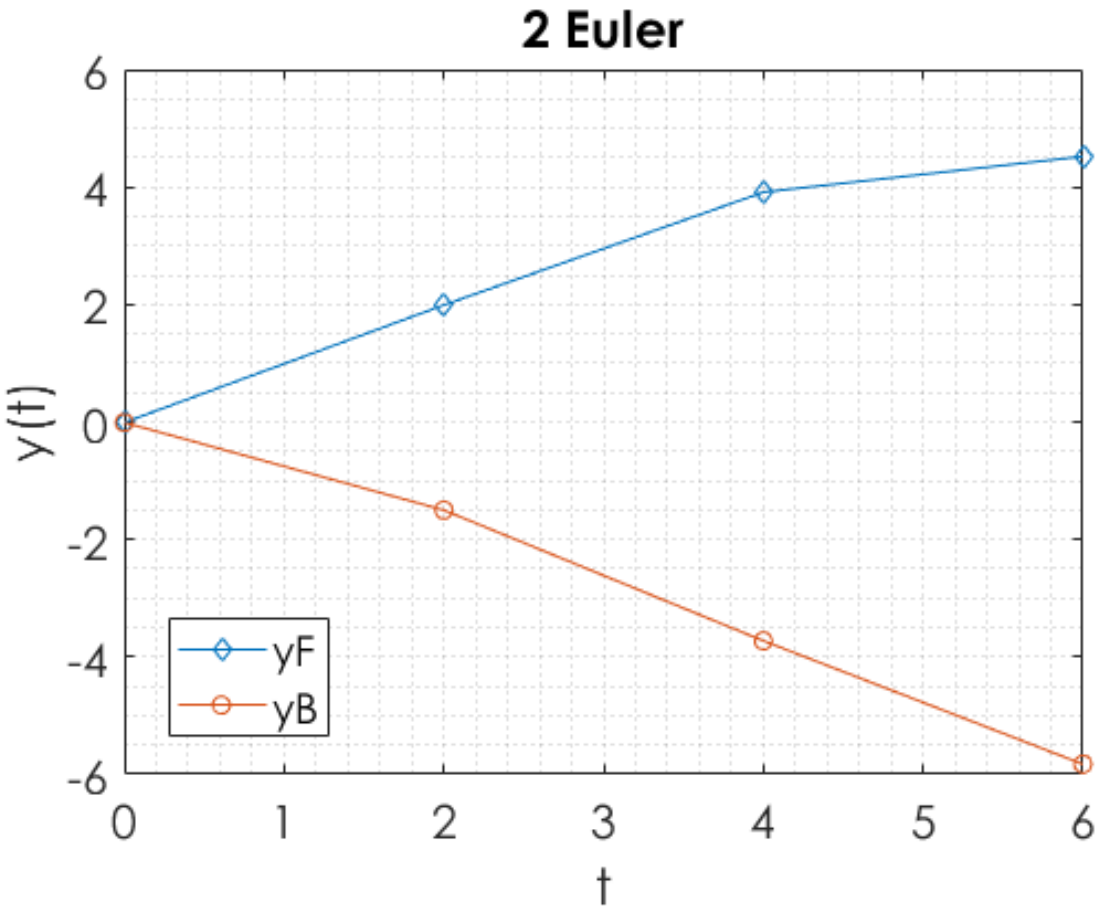


- Plot 0.1 value of Δh



Outputs:

- Plot 2 value of Δh



References:

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

- Lecture notes
- Wikipedia
  - https://en.wikipedia.org/wiki/Euler_method
  - https://en.wikipedia.org/wiki/Backward_Euler_method
  - https://en.wikipedia.org/wiki/Newton%27s_method
- https://web.mit.edu/10.001/Web/Course_Notes/Differential_Equations_Notes/node3.html
- https://www.codesansar.com/numerical-methods/disadvantages-newton-raphson-method.htm
- https://ocw.metu.edu.tr/pluginfile.php/3961/mod_resource/content/12/ch3/3-9.htm
- https://www.lucidchart.com
- https://yuml.me/

Thank you!

Form:

- Bhargav Sanchania (3106382)
- Vashishth Shukla (3098673)

UNIVERSITÄT DUISBURG ESSEN

Open-Minded