Tecnológico de Monterrey

Numerical methods in Engineering

Prof. Alberto Alcaraz Paz

Final Project

Martin Alegria - A01022216

Diego Moreno - A01022113

Luis Garcia - A01021865

Julio Villazón - A01370190

INDEX

INSTRUCTIONS:

PREREQUISITES:

- HAVE A TEXT EDITOR INSTALLED
- HAVE GFORTRAN INSTALLED
    - *http://www.lapk.org/gfortran/gfortran.php?OS=7
    - *https://gcc.gnu.org/wiki/GFortranBinaries
- PREFERABLY HAVE A UNIX BASED TERMINAL

1.- IN ORDER TO LOAD A FUNCTION INTO THE PROGRAM YOU NEED TO OPEN THE
'final.f90' FILE WITH THE TEXT EDITOR OF YOUR CHOICE.
(NOTE: THE PROGRAM COMES PRELODED WITH A FUNCTION)

2.- AT THE BOTTOM OF THE PROGRAM YOU WILL FIND TWO FUNCTIONS CALLED
'f' and 'f_prime'. THEY LOOK SOMETHING LIKE THIS:

```
DOUBLE PRECISION FUNCTION f(x)
```

```
        IMPLICIT NONE
        DOUBLE PRECISION :: x
        f =  <---------------------- YOUR FUNCTION GOES HERE
      END FUNCTION f

      DOUBLE PRECISION FUNCTION f_prime(x)
        IMPLICIT NONE
        DOUBLE PRECISION :: x
        f_prime = <------------------ THE DERIVATIVE OF YOUR FUNCTION GOES
HERE
      END FUNCTION f_prime
```

3.- AFTER YOU INPUT YOUR FUNCTION AND THE DERIVATIVE OF YOUR FUNCTION, SAVE THE FILE AND EXIT YOUR TEXT EDITOR.

4.- THEN, OPEN YOUR TERMINAL AND NAVIGATE TO THE DIRECTORY IN WHICH THE PROGRAMS ARE LOCATED

5.- IN YOUR TERMINAL TYPE THE FOLLOWING COMMANDS (omit the '$')

```
      $gfortran -o final final.f90

      #THIS COMMAND WILL COMPILE THE FILE AND CREATE ANOTHER
ONE YOU CAN
      EXECUTE

      $./final

      #THIS COMMAND WILL EXECUTE THE PROGRAM.
```

INSIDE THE PROGRAM

*INSIDE THE PROGRAM YOU WILL HAVE A MENU WHERE YOU CAN SELECT THE DIFFERENT METHODS USED TO SOLVE LINEAR AND NONLINEAR EQUATIONS, MATRICES AND CURVE FITTING.

```
 ################ NUMERICAL METHODS ################
 PLEASE BE SURE TO READ THE userManual.pdf FILE BEFORE USING THE
PROGRAM
 WHAT DO YOU WANT TO SOLVE ?
 1.) NO LINEAR EQUATIONS
 2.) SYSTEMS OF LINEAR EQUATIONS
```

3.) INTERPOLATION
4.) REGRESSION
5.) NUMERICAL INTEGRATION
6.) ORDINARY DIFFERENTIAL EQUATIONS

*THEN, YOU TYPE THE NUMBER OF THE METHOD YOU WANT TO CHOOSE AND CLICK ENTER

THEN FOR THE 6 DIFFERENT OPTIONS YOU WILL FIND THE METHODS THAT CAN HELP YOU TO SOLVE WHAT YOU SELECTED.

FOR NONLINEAR

INPUT THE NUMBER OF THE METHOD TO USE

WHAT METHOD DO YOU WANT TO USE ?
1.) BISECTION
2.) FALSE POSITION
3.) NEWTON-RHAPSON
4.) SECANT

IN BISECTION AND FALSE POSITION YOU WILL BE ASKED TO INPUT YOUR LOWER AND UPPER BOUND OF INTERVAL. IN CASE YOUR INTERVAL IS NOT CORRECT THE PROGRAM SHOWS THIS MESSAGE “No interval, try again
----------”

ON THE OTHER HAND IF ITS CORRECT YOU NEED TO INPUT THE TOLERANCE AND NUMBER OF ITERATIONS THE PROGRAM WILL DO.

Lower bound of the interval:
-2
Upper bound of the interval:
0
-43.000000000000000
5.0000000000000000
Perfect!
Input the tolerance
0.001

Input the number of iterations
50
The result is: -0.75928280261679293
Iters: 19.0000000

NOTE: AT THE END OF EVERY METHOD THE PROGRAM WILL ASK YOU IF YOU WANT TO USE ANOTHER METHOD OR NOT. IN CASE YOU TYPE 1 YOU WILL BE REDIRECTED TO THE MAIN MENU TO SELECT WHAT YOU WANT TO SOLVE.

FOR NEWTON-RHAPSON AND SECANT THE INPUTS YOU NEED ARE THE GUESS, TOLERANCE AND NUMBER OF ITERATIONS, THEN THE METHOD WILL SOLVE IT FOR YOU AND THEN IT DISPLAYS THE RESULTS.

Enter your guess:
0
Enter the tolerance:
0.01
Enter the number of iterations desired:
100
P: 0.0000000000000000
F(P): 25.000000000000000
F'(P): 10.000000000000000
New P: -2.5000000000000000
Iters: 0.00000000
****
P: -2.5000000000000000
F(P): 0.0000000000000000
F'(P): 10.000000000000000
The result is: -2.5000000000000000
Iters: 1.00000000
ERROR -- 0.0000000000000000

FOR SYSTEM OF LINEAR EQUATIONS

INPUT THE NUMBER OF THE METHOD TO USE

WHAT METHOD DO YOU WANT TO USE ?
1.) GAUSSIAN ELIMINATION
2.) LU DECOMPOSITION
3.) GAUSS-SEIDEL

FOR THE ALL 3 METHOD THE PROGRAM NEEDS YOU TO INPUT THE NAME OF THE FILE WHERE YOUR MATRIX IS LOCATED.

NOTE: ALL THE FILES TO READ NEED TO BE IN THE SAME FOLDER AS THE MAIN PROGRAM.

THE FILE USES THE FOLLOWING FORMAT (INCLUDING COMMAS):

```
n
a11,  a12,  a13,  b1
a21,  a22,  a23,  b2
a31,  a32,  a33,  b3
```

n = GRADE OF MATRIX (2,3,4,etc.)

THEN YOU NEED TO INPUT THE NAME OF THE FILE WHERE THE RESULTS WILL BE SHOWN. THE FORMAT OF THE FILE COULD BE .txt or .csv

1)GAUSSIAN ELIMINATION

```
        3
  3.0000000000000000      -5.0000000000000000       4.0000000000000000
4.0000000000000000
  -2.0000000000000000      -3.0000000000000000       3.0000000000000000
-4.0000000000000000
  1.0000000000000000       4.0000000000000000      -6.0000000000000000
-2.0000000000000000
 ------------------ New matrix ------------------
  1.0000000000000000      -1.6666666666666667       1.3333333333333333
1.3333333333333333
  0.0000000000000000      -6.3333333333333339       5.6666666666666661
-1.3333333333333335
  0.0000000000000000       5.6666666666666670      -7.3333333333333330
-3.3333333333333330
 ------------------ New matrix ------------------
  1.0000000000000000      -1.6666666666666667       1.3333333333333333
1.3333333333333333
  -0.0000000000000000       1.0000000000000000      -0.89473684210526294
0.21052631578947370
  0.0000000000000000       0.0000000000000000      -2.2631578947368425
-4.5263157894736841
```

```
------------------Final matrix ------------------
  1.0000000000000000       -1.6666666666666667       1.3333333333333333
1.3333333333333333
 -0.0000000000000000        1.0000000000000000      -0.89473684210526294
0.21052631578947370
  0.0000000000000000        0.0000000000000000       -2.2631578947368425
-4.5263157894736841
 ********** RESULTS EXPORTED TO CSV **********


2)LU DECOMPOSITION
 ********** Read Matrix **********
  3.0000000000000000       -5.0000000000000000        4.0000000000000000
 -2.0000000000000000       -3.0000000000000000        3.0000000000000000
  1.0000000000000000        4.0000000000000000       -6.0000000000000000
 ********** L matrix **********
  3.0000000000000000        0.0000000000000000        0.0000000000000000
 -2.0000000000000000       -6.3333333333333339        0.0000000000000000
  1.0000000000000000        5.6666666666666670       -2.2631578947368425
 ********** U matrix **********
  1.0000000000000000       -1.6666666666666667        1.3333333333333333
  0.0000000000000000        1.0000000000000000      -0.89473684210526294
  0.0000000000000000        0.0000000000000000        1.0000000000000000
 ********** B matrix **********
  4.0000000000000000       -4.0000000000000000       -2.0000000000000000
 Y MATRIX **********
  1.3333333333333333        0.21052631578947370       1.9999999999999996
  1.9999999999999996
 ********** RESULTS EXPORTED TO CSV **********


3) GAUSS SEIDEL

  3.0000000000000000       -5.0000000000000000        4.0000000000000000
 -2.0000000000000000       -3.0000000000000000        3.0000000000000000
  1.0000000000000000        4.0000000000000000       -6.0000000000000000
 Enter your tolerance
0.01
 ********** RESULTS EXPORTED TO CSV **********
```

FOR INTERPOLATION

INPUT THE NUMBER OF THE METHOD TO USE

WHAT METHOD DO YOU WANT TO USE ?
1.) POWER SERIES
2.) LAGRANGE
3.) NEWTON DIVIDED DIFFERENCES

FOR THE ALL 3 METHODS THE PROGRAM NEEDS YOU TO INPUT THE NAME OF THE FILE WHERE YOUR MATRIX IS LOCATED

NOTE: ALL THE FILES TO READ NEED TO BE IN THE SAME FOLDER AS THE MAIN PROGRAM.

THE FILE USES THE FOLLOWING FORMAT (INCLUDING COMMAS):

n
x0,y0
x1,y1
x2,y2
.
.
.
xn,yn

n = NUMBER OF X POINTS

THEN YOU NEED TO INPUT THE NAME OF THE FILE WHERE THE RESULTS WILL BE SHOWN. THE FORMAT OF THE FILE COULD BE .txt or .csv

1)POWER SERIES
THE POWER SERIES USES LU DECOMPOSITION TO SOLVE THE MATRIX AND THEN AFTER THE METHOD YOU INPUT A POINT TO EVALUATE IN THE FUNCTION.

********** Read Data **********
1.0000000000000000 0.0000000000000000
4.0000000000000000 1.3862000000000001
6.0000000000000000 1.7917000000000001
********** Number to interpolate: 0.0000000000000000

```
 ***********************************************
 ************ MATRIX TO SOLVE *************
   1.0000000000000000        1.0000000000000000        1.0000000000000000
   1.0000000000000000        4.0000000000000000        16.000000000000000
   1.0000000000000000        6.0000000000000000        36.000000000000000
 ************ L matrix *************
   1.0000000000000000        0.0000000000000000        0.0000000000000000
   1.0000000000000000        3.0000000000000000        0.0000000000000000
   1.0000000000000000        5.0000000000000000        10.000000000000000
 ************ U matrix *************
   1.0000000000000000        1.0000000000000000        1.0000000000000000
   0.0000000000000000        1.0000000000000000        5.0000000000000000
   0.0000000000000000        0.0000000000000000        1.0000000000000000
 ************ B matrix *************
   0.0000000000000000        1.3862000000000001        1.7917000000000001
 Y MATRIX *********************
   0.0000000000000000       0.46206665039062500       -5.1863324642181394E-002
  -5.1863324642181394E-002
 AT WHAT POINT DO YOU WANT TO EVALUATE
6
 Point    6.0000000000000000       RES =    1.7916999936103821
 DO YOU WANT TO EVALUATE ANOTHER POINT ?
 [ 1 == YES, 0 == NO]
0
 ************ RESULTS EXPORTED TO CSV *************
```

2)LAGRANGE
SAME AS POWER SERIES. AFTER THE PROGRAM DOES THE METHOD YOU INPUT A POINT TO EVALUATE IN THE FUNCTION.

```
 ************ Read Data *************
   1.0000000000000000        0.0000000000000000
   4.0000000000000000        1.3862000000000001
   6.0000000000000000        1.7917000000000001
 ************ Number to interpolate:    4.9406564584124654E-324
 AT WHAT POINT DO YOU WANT TO EVALUATE
6
 Point    6.0000000000000000       RES =    1.7917000000000001
 DO YOU WANT TO EVALUATE ANOTHER POINT ?
 [ 1 == YES, 0 == NO]

0
```

********** RESULTS EXPORTED TO CSV **********

3)NEWTON DIVIDED DIFFERENCES
SAME AS POWER SERIES AND LAGRANGE. AFTER THE PROGRAM DOES THE METHOD YOU INPUT A POINT TO EVALUATE IN THE FUNCTION.

********** Read Data **********
1.0000000000000000 0.0000000000000000
4.0000000000000000 1.3862000000000001
6.0000000000000000 1.7917000000000001
WHICH NUMBER DO YOU WANT TO INTERPOLATE:
6
X = 6.0000000000000000 Y = -0.28283333333333349
DO YOU WANT TO TRY ANOTHER NUMBER ?:
1 == YES, 0 == NO
0
********** RESULTS EXPORTED TO CSV **********

FOR REGRESSION

INPUT THE NUMBER OF THE METHOD TO USE

WHAT METHOD DO YOU WANT TO USE ?
1.) POLYNOMIAL
2.) EXPONENTIAL
3.) LOGARITHMIC

FOR THE ALL 3 METHODS THE PROGRAM NEEDS YOU TO INPUT THE NAME OF THE FILE WHERE YOUR MATRIX IS LOCATED.

NOTE: ALL THE FILES TO READ NEED TO BE IN THE SAME FOLDER AS THE MAIN PROGRAM.

THE FILE USES THE FOLLOWING FORMAT (INCLUDING COMMAS):

n
x0,y0
x1,y1
x2,y2

.
.
.
xn,yn

n = NUMBER OF X POINTS

THEN YOU NEED TO INPUT THE NAME OF THE FILE WHERE THE RESULTS WILL BE SHOWN. THE FORMAT OF THE FILE COULD BE .txt or .csv

1)POLYNOMIAL

********** Read Data **********
  0.0000000000000000 3.3000000000000000E-004
  25.364100000000001 1.9000000000000001E-004
  60.000000000000000 1.6670000000000001E-004
  94.641000000000005 1.9000000000000001E-004
  120.00000000000000 3.3300000000000002E-004
 INPUT THE DEGREE
3
  300.00510000000003 27600.256449810004 2808009.4382457752
300960279.06825668 33264004104.339680 3751487586290.4839
  5.0000000000000000 300.00510000000003 27600.256449810004
2808009.4382457752
  300.00510000000003 27600.256449810004 2808009.4382457752
300960279.06825668
  27600.256449810004 2808009.4382457752 300960279.06825668
33264004104.339680
  2808009.4382457752 300960279.06825668 33264004104.339680
3751487586290.4839
  3.3000000000000000E-004 141.03619719154722 4498.3680471915477
31376.330726715358
 Y MATRIX **********
  6.6000000000000005E-005 1.4681501624542962E-002
2.6243231948444703E-005 2.4027615044770242E-006
  2.4027615044770242E-006
 RESULTS --- 6.6000000000000005E-005 3.0230705381700224E-002
-4.0598085552365963E-004 2.4027615044770242E-006

NOTE: THE PROGRAM IS NOT WORKING PROPERLY AFTER DOING THE METHOD

2)EXPONENTIAL

```
 ************* Read Data *************
          1
   1.0000000000000000        810.00000000000000
          2
   2.0000000000000000        1250.0000000000000
          3
   3.0000000000000000        2100.0000000000000
          4
   4.0000000000000000        3150.0000000000000
          5
   5.0000000000000000        5250.0000000000000
          6
   6.0000000000000000        8600.0000000000000
        sumx                  sumy                  sumx2                 sumxy
   21.000000000000000        47.158284271321293        91.000000000000000
173.31556237805037
 original values for
 a0=    6.20740080      a1 =   0.472089559
        Sr                  St                  R2                  R
   3.4479946965208768E-003   374.19402441186048       0.999990761
0.999995351
  Values for the logarithmic equation: a0 =    496.409302      a1=   0.472089559
 So the equations are:    496.409302     *e  0.472089559
 press 1 if you want to calculate the regression value for a point, 0 if not
1
 what value do you want to calculate
8
 the value for x =    8.00000000      is    21679.3887
```

3)LOGARITHMIC

```
************* Read Data *************
   1.0000000000000000        810.00000000000000
   2.0000000000000000        1250.0000000000000
   3.0000000000000000        2100.0000000000000
   4.0000000000000000        3150.0000000000000
   5.0000000000000000        5250.0000000000000
   6.0000000000000000        8600.0000000000000
        log(sumx)               log(sumy)               log(sumx2)              log(sumxy)
```

```
  2.8573324964312681        20.480582635059751        1.7748184185876701
10.285471721721995
 original values for
 log(a0)=    633.025085      a1 =    1.28513467
        Sr                  St                  R2                  R
  20.466760210704784        70.615852265567412       0.710167646
0.842714429
  Values for the logarithmic equation: a0 =    633.025085      a1=    1.28513467
 So the equations is:    633.025085     *e   1.28513467
 Sr =    20.466760210704784       St =     70.615852265567412       R2 =
0.710167646      R =   0.842714429
           x             y
 press 1 if you want to calculate the regression value for a point, 0 if not
1
 what value do you want to calculate
8
 the value for x =    8.00000000      is    18468690.0
 press 1 if you want to calculate the regression value for a point, 0 if not
```

FOR NUMERICAL INTEGRATION

INPUT THE NUMBER OF THE METHOD TO USE

```
  WHAT METHOD DO YOU WANT TO USE ?
  1.) TRAPEZOID
  2.) SIMPSON 1/3
  3.) SIMPSON 3/8
```

IN THE NUMERICAL INTEGRATION YOU CAN USE DATA OF A FILE OR USE THE FUNCTION GIVEN IN THE MAIN PROGRAM.

FOR THE ALL 3 METHODS THE PROGRAM NEEDS YOU TO INPUT THE NAME OF THE FILE WHERE YOUR MATRIX IS LOCATED.

THE FILE USES THE FOLLOWING FORMAT (INCLUDING COMMAS):

```
n
x0,y0
x1,y1
x2,y2
.
```

.
.
xn,yn

WHILE USING FUNCTION THE PROGRAM ASKS YOU FOR LOWER, UPPER INTERVAL AND NUMBER OF TRAPEZOIDS. ALSO FOR SIMPSON YOU NEED TO INPUT TOLERANCE.
NOTE: ALL THE FILES TO READ NEED TO BE IN THE SAME FOLDER AS THE MAIN PROGRAM.

1.) TRAPEZOID
A)USING DATA
INPUT THE NAME OF THE FILE TO BE EXPORTED
trapez.txt
0.0000000000000000 3.3000000000000000E-004
25.364100000000001 1.9000000000000001E-004
60.000000000000000 1.6670000000000001E-004
94.641000000000005 1.9000000000000001E-004
120.00000000000000 3.3300000000000002E-004
RESULT = 2.10768003E-02

B)USING FUNCTION
INPUT THE NAME OF THE FILE TO BE EXPORTED
trap.txt
Lower bound of the interval:
-10
Upper bound of the interval:
10
HOW MANY TRAPEZOIDS DO YOU WANT ?
TIP: SOMETIMES USING LARGER INTERVALS GIVE A BETTER RESULT
5
RESULT = 500.000000

2.) SIMPSON ⅓
A)USING DATA
INPUT THE NAME OF THE FILE TO BE EXPORTED
sim13.txt
0.0000000000000000 3.3000000000000000E-004
25.364100000000001 1.9000000000000001E-004
60.000000000000000 1.6670000000000001E-004
94.641000000000005 1.9000000000000001E-004

```
  120.00000000000000        3.3300000000000002E-004
 THE ANSWER IS =    2.46383995E-02
```

B)USING FUNCTION

```
 Lower bound of the interval:
-10
 Upper bound of the interval:
10
 HOW MANY ITERATIONS:
4
 TOLERANCE:
0.01
 INITIAL:    500.00000000000000
 CONVERGED --
 THE ANSWER IS =    500.000000
```

3.) SIMPSON ⅜

A)USING DATA

sim38.txt

```
  0.0000000000000000        3.3000000000000000E-004
  25.364100000000001        1.9000000000000001E-004
  60.000000000000000        1.6670000000000001E-004
  94.641000000000005        1.9000000000000001E-004
  120.00000000000000        3.3300000000000002E-004
 THE ANSWER IS =    2.51675993E-02
```

B)USING FUNCTION

```
 Lower bound of the interval:
-10
 Upper bound of the interval:
10
 HOW MANY ITERATIONS:
4
 TOLERANCE:
0.01
 INITIAL:    500.00000000000000
  1.0000000000000000          4.0000000000000000
  0.0000000000000000
 CONVERGED
 THE ANSWER IS =    500.000000
```

FOR ORDINARY DIFFERENTIAL EQUATIONS

INPUT THE NUMBER OF THE METHOD TO USE

WHAT METHOD DO YOU WANT TO USE ?
1.) EULER
2.) MODIFIED EULER
3.) RUNGE KUTTA 3rd ORDER
4.) RUNGE KUTTA 4th ORDER

THEN YOU NEED TO INPUT THE NAME OF THE FILE WHERE THE RESULTS WILL BE SHOWN. THE FORMAT OF THE FILE COULD BE .txt or .csv
FOR THE METHODS OF EULER AND MODIFIED EULER THE PROGRAM ASKS YOU FOR THE INITIAL X, INITIAL Y, H, APPROX X AND THE TOLERANCE.

1.) EULER
################## EULER METHOD ##################
INPUT THE NAME OF THE FILE TO BE EXPORTED
eul.txt

GIVE ME THE INITIAL Y
250
GIVE ME THE INITIAL X
5
GIVE ME THE H
0.2
GIVE ME THE APROX X
7
GIVE ME THE TOLERANCE
0.001
-101.85916357881302
ERR = 1.00000000 --- 1.0000000000000000E-003
X: 5.2000000000000002 Y: 229.62816728423741
-97.941503441166361
ERR = 9.32599157E-02 --- 1.0000000000000000E-003
X: 5.4000000000000004 Y: 210.03986659600412
-94.314040350752791
ERR = 9.86666903E-02 --- 1.0000000000000000E-003

```
 X:    5.6000000000000005       Y:   191.17705852585357
  -90.945681766797335
 ERR =   0.105146855      ---   1.0000000000000000E-003
 X:    5.8000000000000007       Y:   172.98792217249411
  -87.809623774838798
 ERR =   0.112992197      ---   1.0000000000000000E-003
 X:    6.0000000000000009       Y:   155.42599741752636
  -84.882636315677502
 ERR =   0.122618943      ---   1.0000000000000000E-003
 X:    6.2000000000000011       Y:   138.44947015439087
  -82.144486757107259
 ERR =   0.134640396      ---   1.0000000000000000E-003
 X:    6.4000000000000012       Y:   122.02057280296941
  -79.577471545947660
 ERR =   0.149997473      ---   1.0000000000000000E-003
 X:    6.6000000000000014       Y:   106.10507849377987
  -77.166033014252278
 ERR =   0.170209423      ---   1.0000000000000000E-003
 X:    6.8000000000000016       Y:   90.671871890929410
  -74.896443807950732
 ERR =   0.197896391      ---   1.0000000000000000E-003
 X:    7.0000000000000018       Y:   75.692583129339255
 SOLUTION *****************
 X:    7.0000000000000018      Y:    75.692583129339255
 ***************** EXPORTED TO FILE *****************
```

2.) MODIFIED EULER

```
 ################## MOD EULER METHOD ##################
 INPUT THE NAME OF THE FILE TO BE EXPORTED
mode.txt
  GIVE ME THE INITIAL Y
250
  GIVE ME THE INITIAL X
5
  GIVE ME THE H
0.2
  GIVE ME THE APROX X
7
  GIVE ME THE TOLERANCE
0.01
```

```
 ERR =    1.00000000      ---   1.0000000000000000E-002
 X:    5.2000000000000002       Y:   231.32582001055096
 ERR =    8.44216868E-02  ---   1.0000000000000000E-002
 X:    5.4000000000000004       Y:   213.31722099072360
 ERR =    8.87527317E-02  ---   1.0000000000000000E-002
 X:    5.6000000000000005       Y:   195.92806980105354
 ERR =    9.38558653E-02  ---   1.0000000000000000E-002
 X:    5.8000000000000007       Y:   179.11689832294857
 ERR =    9.99138951E-02  ---   1.0000000000000000E-002
 X:    6.0000000000000009       Y:   162.84629156466963
 ERR =   0.107177481      ---   1.0000000000000000E-002
 X:    6.2000000000000011       Y:   147.08237339175810
 ERR =   0.115998894      ---   1.0000000000000000E-002
 X:    6.4000000000000012       Y:   131.79437168974090
 ERR =   0.126888275      ---   1.0000000000000000E-002
 X:    6.6000000000000014       Y:   116.95424861765876
 ERR =   0.140612170      ---   1.0000000000000000E-002
 X:    6.8000000000000016       Y:   102.53638455446952
 ERR =   0.158376694      ---   1.0000000000000000E-002
 X:    7.0000000000000018       Y:   88.517306610930021
 SOLUTION *****************
 X:    7.0000000000000018      Y:    88.517306610930021
 ***************** EXPORTED TO FILE *****************
```

3.) RUNGE KUTTA 3rd ORDER

```
 ################## RUNGE KUTTA 3rd ORDER ##################
  GIVE ME THE INITIAL Y
250
  GIVE ME THE INITIAL X
5
  GIVE ME THE H
0.2
  GIVE ME THE APROX X
7
  0.20000000000000001
   5.0000000000000000        230.02505442236500
   5.2000000000000002        210.80406288982769
   5.4000000000000004        192.28217353696232
   5.6000000000000005        174.41031088688288
   5.8000000000000007        157.14439220829360
   6.0000000000000009        140.44467241419389
```

```
  6.2000000000000011        124.27519300210126
  6.4000000000000012        108.60331581637018
  6.6000000000000014        93.399326431867593
  6.8000000000000016        78.636095044409416
 RES:   7.0000000000000018        78.636095044409416

  4.) RUNGE KUTTA 4th ORDER

  ################## RUNGE KUTTA 4th ORDER ##################
  GIVE ME THE INITIAL Y
250
  GIVE ME THE INITIAL X
5
  GIVE ME THE H
0.2
  GIVE ME THE APROX X
7
         10
 RES:   7.0000000000000018        78.636095044409416
```

Extra notes

THE USER CAN CREATE THEIR OWN FILES OR YOU CAN DOWNLOAD OUR FILES IN THE NEXT LINK https://github.com/MartinAlegria/MetodosNumericos