MNUM Exam 2014

Telmo Baptista

January 12, 2020

1 Exercise 1

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx$$

$$m\frac{dy}{dt} + cy + kx = 0 \Rightarrow \frac{dy}{dt} = -\frac{cy + kx}{m}$$

```
from math import sin, cos, exp, log, sqrt

def euler(f, t, x, y, dt):
    return dt*f(t, x, y)

dy = lambda t, x, y: - (y + k*x)/20

dx = lambda t, x, y: y

y0 = 0
t0 = 0
x0 = 1
h = 0.1

n = int(5/0.1)

k = 5
y = y0
t = t0
x = x0
print("K=5")
for i in range(n):
    hx = euler(dx, t, x, y, h)
    hy = euler(dy, t, x, y, h)

    y += hy
    x += hx
    t += h
    print(f"t = {t:.5f} \t X = {x:.5f} \t Y = {y:.5f}")

k = 20
y = y0
t = t0
x = x0
print("K=20")
for i in range(n):
    hx = euler(dx, t, x, y, h)
    hy = euler(dy, t, x, y, h)

    y += hy
    x += hx
    t += h
    print(f"t = {t:.5f} \t X = {x:.5f} \t Y = {y:.5f}")

k = 40
y = y0
t = t0
x = x0
print("K=40")
for i in range(n):
    hx = euler(dx, t, x, y, h)
    hy = euler(dy, t, x, y, h)

    y += hy
    x += hx
```

```
    t += h
    print(f"t = {t:.5f} \t X = {x:.5f} \t Y = {y:.5f}")
```

Program Output:

K=5
t = 0.10000 X = 1.00000 Y = -0.02500
t = 0.20000 X = 0.99750 Y = -0.04988
t = 0.30000 X = 0.99251 Y = -0.07456
t = 0.40000 X = 0.98506 Y = -0.09900
t = 0.50000 X = 0.97516 Y = -0.12313
t = 0.60000 X = 0.96284 Y = -0.14690
t = 0.70000 X = 0.94815 Y = -0.17023
t = 0.80000 X = 0.93113 Y = -0.19309
t = 0.90000 X = 0.91182 Y = -0.21540
t = 1.00000 X = 0.89028 Y = -0.23712
t = 1.10000 X = 0.86657 Y = -0.25819
t = 1.20000 X = 0.84075 Y = -0.27856
t = 1.30000 X = 0.81289 Y = -0.29819
t = 1.40000 X = 0.78307 Y = -0.31702
t = 1.50000 X = 0.75137 Y = -0.33501
t = 1.60000 X = 0.71787 Y = -0.35212
t = 1.70000 X = 0.68266 Y = -0.36831
t = 1.80000 X = 0.64583 Y = -0.38353
t = 1.90000 X = 0.60748 Y = -0.39776
t = 2.00000 X = 0.56770 Y = -0.41096
t = 2.10000 X = 0.52660 Y = -0.42310
t = 2.20000 X = 0.48429 Y = -0.43415
t = 2.30000 X = 0.44088 Y = -0.44408
t = 2.40000 X = 0.39647 Y = -0.45288
t = 2.50000 X = 0.35118 Y = -0.46053
t = 2.60000 X = 0.30513 Y = -0.46701
t = 2.70000 X = 0.25843 Y = -0.47230
t = 2.80000 X = 0.21120 Y = -0.47640
t = 2.90000 X = 0.16356 Y = -0.47930
t = 3.00000 X = 0.11563 Y = -0.48099
t = 3.10000 X = 0.06753 Y = -0.48148
t = 3.20000 X = 0.01938 Y = -0.48076
t = 3.30000 X = -0.02869 Y = -0.47884
t = 3.40000 X = -0.07658 Y = -0.47573
t = 3.50000 X = -0.12415 Y = -0.47143
t = 3.60000 X = -0.17129 Y = -0.46597
t = 3.70000 X = -0.21789 Y = -0.45936
t = 3.80000 X = -0.26383 Y = -0.45162
t = 3.90000 X = -0.30899 Y = -0.44276
t = 4.00000 X = -0.35327 Y = -0.43282
t = 4.10000 X = -0.39655 Y = -0.42183
t = 4.20000 X = -0.43873 Y = -0.40981
t = 4.30000 X = -0.47971 Y = -0.39679
t = 4.40000 X = -0.51939 Y = -0.38281
t = 4.50000 X = -0.55767 Y = -0.36791
t = 4.60000 X = -0.59446 Y = -0.35213
t = 4.70000 X = -0.62968 Y = -0.33551
t = 4.80000 X = -0.66323 Y = -0.31809
t = 4.90000 X = -0.69504 Y = -0.29992
t = 5.00000 X = -0.72503 Y = -0.28104

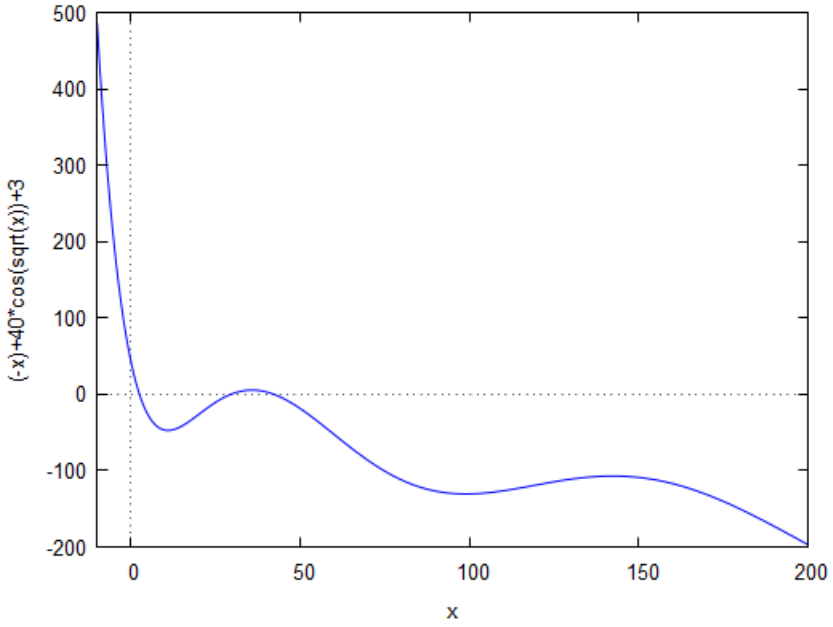
K=20
t = 0.10000 X = 1.00000 Y = -0.10000
t = 0.20000 X = 0.99000 Y = -0.19950
t = 0.30000 X = 0.97005 Y = -0.29750
t = 0.40000 X = 0.94030 Y = -0.39302
t = 0.50000 X = 0.90100 Y = -0.48508

t = 0.60000 X = 0.85249 Y = -0.57276
t = 0.70000 X = 0.79521 Y = -0.65514
t = 0.80000 X = 0.72970 Y = -0.73139
t = 0.90000 X = 0.65656 Y = -0.80070
t = 1.00000 X = 0.57649 Y = -0.86236
t = 1.10000 X = 0.49025 Y = -0.91569
t = 1.20000 X = 0.39868 Y = -0.96014
t = 1.30000 X = 0.30267 Y = -0.99521
t = 1.40000 X = 0.20315 Y = -1.02050
t = 1.50000 X = 0.10110 Y = -1.03571
t = 1.60000 X = -0.00247 Y = -1.04064
t = 1.70000 X = -0.10654 Y = -1.03519
t = 1.80000 X = -0.21005 Y = -1.01936
t = 1.90000 X = -0.31199 Y = -0.99326
t = 2.00000 X = -0.41132 Y = -0.95710
t = 2.10000 X = -0.50703 Y = -0.91118
t = 2.20000 X = -0.59814 Y = -0.85592
t = 2.30000 X = -0.68374 Y = -0.79183
t = 2.40000 X = -0.76292 Y = -0.71949
t = 2.50000 X = -0.83487 Y = -0.63960
t = 2.60000 X = -0.89883 Y = -0.55292
t = 2.70000 X = -0.95412 Y = -0.46027
t = 2.80000 X = -1.00015 Y = -0.36256
t = 2.90000 X = -1.03640 Y = -0.26073
t = 3.00000 X = -1.06248 Y = -0.15579
t = 3.10000 X = -1.07805 Y = -0.04876
t = 3.20000 X = -1.08293 Y = 0.05929
t = 3.30000 X = -1.07700 Y = 0.16729
t = 3.40000 X = -1.06027 Y = 0.27415
t = 3.50000 X = -1.03286 Y = 0.37881
t = 3.60000 X = -0.99498 Y = 0.48020
t = 3.70000 X = -0.94696 Y = 0.57729
t = 3.80000 X = -0.88923 Y = 0.66910
t = 3.90000 X = -0.82232 Y = 0.75468
t = 4.00000 X = -0.74685 Y = 0.83314
t = 4.10000 X = -0.66354 Y = 0.90366
t = 4.20000 X = -0.57317 Y = 0.96549
t = 4.30000 X = -0.47662 Y = 1.01798
t = 4.40000 X = -0.37482 Y = 1.06056
t = 4.50000 X = -0.26877 Y = 1.09274
t = 4.60000 X = -0.15949 Y = 1.11415
t = 4.70000 X = -0.04808 Y = 1.12453
t = 4.80000 X = 0.06437 Y = 1.12371
t = 4.90000 X = 0.17675 Y = 1.11166
t = 5.00000 X = 0.28791 Y = 1.08842
K=40
t = 0.10000 X = 1.00000 Y = -0.20000
t = 0.20000 X = 0.98000 Y = -0.39900
t = 0.30000 X = 0.94010 Y = -0.59301
t = 0.40000 X = 0.88080 Y = -0.77806
t = 0.50000 X = 0.80299 Y = -0.95033
t = 0.60000 X = 0.70796 Y = -1.10618
t = 0.70000 X = 0.59734 Y = -1.24224
t = 0.80000 X = 0.47312 Y = -1.35550
t = 0.90000 X = 0.33757 Y = -1.44334
t = 1.00000 X = 0.19324 Y = -1.50364
t = 1.10000 X = 0.04287 Y = -1.53477
t = 1.20000 X = -0.11061 Y = -1.53567
t = 1.30000 X = -0.26417 Y = -1.50587
t = 1.40000 X = -0.41476 Y = -1.44551
t = 1.50000 X = -0.55931 Y = -1.35533

t = 1.60000 X = -0.69484 Y = -1.23669
t = 1.70000 X = -0.81851 Y = -1.09154
t = 1.80000 X = -0.92766 Y = -0.92238
t = 1.90000 X = -1.01990 Y = -0.73223
t = 2.00000 X = -1.09312 Y = -0.52459
t = 2.10000 X = -1.14558 Y = -0.30334
t = 2.20000 X = -1.17592 Y = -0.07271
t = 2.30000 X = -1.18319 Y = 0.16284
t = 2.40000 X = -1.16690 Y = 0.39866
t = 2.50000 X = -1.12704 Y = 0.63005
t = 2.60000 X = -1.06403 Y = 0.85231
t = 2.70000 X = -0.97880 Y = 1.06085
t = 2.80000 X = -0.87272 Y = 1.25131
t = 2.90000 X = -0.74759 Y = 1.41960
t = 3.00000 X = -0.60563 Y = 1.56202
t = 3.10000 X = -0.44943 Y = 1.67533
t = 3.20000 X = -0.28189 Y = 1.75684
t = 3.30000 X = -0.10621 Y = 1.80443
t = 3.40000 X = 0.07424 Y = 1.81665
t = 3.50000 X = 0.25590 Y = 1.79272
t = 3.60000 X = 0.43517 Y = 1.73258
t = 3.70000 X = 0.60843 Y = 1.63688
t = 3.80000 X = 0.77212 Y = 1.50701
t = 3.90000 X = 0.92282 Y = 1.34505
t = 4.00000 X = 1.05733 Y = 1.15376
t = 4.10000 X = 1.17270 Y = 0.93653
t = 4.20000 X = 1.26635 Y = 0.69731
t = 4.30000 X = 1.33608 Y = 0.44055
t = 4.40000 X = 1.38014 Y = 0.17113
t = 4.50000 X = 1.39725 Y = -0.10575
t = 4.60000 X = 1.38668 Y = -0.38468
t = 4.70000 X = 1.34821 Y = -0.66009
t = 4.80000 X = 1.28220 Y = -0.92643
t = 4.90000 X = 1.18956 Y = -1.17824
t = 5.00000 X = 1.07173 Y = -1.41026

By analysis of the output, it's possible to conclude that the value of k used is 20.

2 Exercise 2



The function has 3 roots by analysing the graph of the function.

```
(%i5)g:(-x)+40*cos(sqrt(x))+3;
```

(%o5) $$-x+40\cos\sqrt{x}+3$$

```
(%i14)dg:diff(g,x);
```

(%o14) $$-\frac{20\sin\sqrt{x}}{\sqrt{x}}-1$$

```
(%i15)nt:x-g/dg;
```

(%o15) $$x-\frac{-x+40\cos\sqrt{x}+3}{-\frac{20\sin\sqrt{x}}{\sqrt{x}}-1}$$

```
(%i16)ratsimp(%);
```

(%o16) $$\frac{20\sin\sqrt{x}\,x+(40\cos\sqrt{x}+3)\,\sqrt{x}}{\sqrt{x}+20\sin\sqrt{x}}$$

```
from math import sin, cos, exp, log, sqrt

g = lambda x: -x + 40*cos(sqrt(x)) + 3

newton = lambda x: (20*sin(sqrt(x))*x+(40*cos(sqrt(x))+3)*sqrt(x))/(sqrt(x)+20*sin(sqrt(x)))

x = 1.7
for i in range(2):
    print(f"G({x:.5f}) = {g(x):.5f}")

    x = newton(x)

    print(f"X = {x:.5f}")
```

Program Output:

```
G(1.70000) = 11.85185
X = 2.45031
G(2.45031) = 0.76769
X = 2.50603
```

As the the difference between the two last iterations is greater than 0.1 we can present the results with 0 exact decimal places.

3 Exercise 3

```
(%i26)DA:da+zeromatrix(4,4);
```

(%o26)
$$\begin{pmatrix} da & da & da & da \\ da & da & da & da \\ da & da & da & da \\ da & da & da & da \end{pmatrix}$$

```
(%i27)DB:db+zeromatrix(4,1);
```

(%o27)
$$\begin{pmatrix} db \\ db \\ db \\ db \end{pmatrix}$$

```
(%i28)X:[x0,x1,x2,x3];
```

(%o28)
$$[x_0, x_1, x_2, x_3]$$

```
(%i29)DB-DA . X;
```

(%o29)
$$\begin{pmatrix} -da\,x_3 - da\,x_2 - da\,x_1 - da\,x_0 + db \\ -da\,x_3 - da\,x_2 - da\,x_1 - da\,x_0 + db \\ -da\,x_3 - da\,x_2 - da\,x_1 - da\,x_0 + db \\ -da\,x_3 - da\,x_2 - da\,x_1 - da\,x_0 + db \end{pmatrix}$$

```
from math import sin, cos, exp, log, sqrt
from copy import deepcopy

def gauss(A, b):

    rows, cols = len(A), len(A[0])

    for i in range(rows):

        pivot = A[i][i]

        for j in range(i, cols):
            A[i][j] /= pivot

        b[i] /= pivot

        for i2 in range(i+1, rows):
            coef = A[i2][i]

            for j in range(i, cols):
                A[i2][j] -= A[i][j] * coef

            b[i2] -= b[i]*coef

    for i in range(rows):
        for j in range(cols):
            print(f"{A[i][j]:.5f}  ", end='\t')
        print(f"{b[i]:.5f}")

    sols = []
    for i in range(rows-1, -1, -1):
        sol = b[i]
        for j in range(cols-1, i, -1):
            sol -= A[i][j]*sols[cols-1-j]
```

```
        sols.append(sol)
    sols.reverse()

    return sols

A = [[0.1, 0.5, 3, 0.25], [1.2, 0.2, 0.25, 0.2], [-1, 0.25, 0.3, 2], [2, 0.00001, 1, 0.4]]

b = [0, 1, 2, 3]

sols = gauss(deepcopy(A), deepcopy(b))

for i in range(len(sols)):
    print(f"X({i}) = {sols[i]:.5f}")

x0, x1, x2, x3 = sols
da = db = 0.3

new_b = [-da*x3-da*x2-da*x1-da*x0+db,
                -da*x3-da*x2-da*x1-da*x0+db,
                -da*x3-da*x2-da*x1-da*x0+db,
                -da*x3-da*x2-da*x1-da*x0+db]

internal_stab = gauss(A, new_b)

for i in range(len(sols)):
    print(f"dX({i}) = {internal_stab[i]:.5f}")
```

Program Output:

$$\begin{array}{ccccc} 1.00000 & 5.00000 & 30.00000 & 2.50000 & 0.00000 \\ 0.00000 & 1.00000 & 6.16379 & 0.48276 & -0.17241 \\ 0.00000 & 0.00000 & 1.00000 & -0.95417 & -1.41034 \\ 0.00000 & 0.00000 & 0.00000 & 1.00000 & 1.82038 \end{array}$$

$X(0) = 0.97263$

$X(1) = -3.06443$

$X(2) = 0.32662$

$X(3) = 1.82038$

$InternalStability$

$$\begin{array}{ccccc} 1.00000 & 5.00000 & 30.00000 & 2.50000 & 2.83442 \\ 0.00000 & 1.00000 & 6.16379 & 0.48276 & 0.53756 \\ 0.00000 & 0.00000 & 1.00000 & -0.95417 & -0.14353 \\ 0.00000 & 0.00000 & 0.00000 & 1.00000 & 0.13439 \end{array}$$

$dX(0) = 0.12249$

$dX(1) = 0.56700$

$dX(2) = -0.01530$

$dX(3) = 0.13439$

4 Exercise 4

Starting by calculating the Newton's expressions for each of the equations:

```
(%i38)a:x^m-R;
```

(%o38) $$x^m - R$$

```
(%i39)b:1-R/x^m;
```

(%o39) $$1 - \frac{R}{x^m}$$

```
(%i40)da:diff(a,x);
```

(%o40) $$m\,x^{m-1}$$

```
(%i41)db:diff(b,x);
```

(%o41) $$R\,m\,x^{-m-1}$$

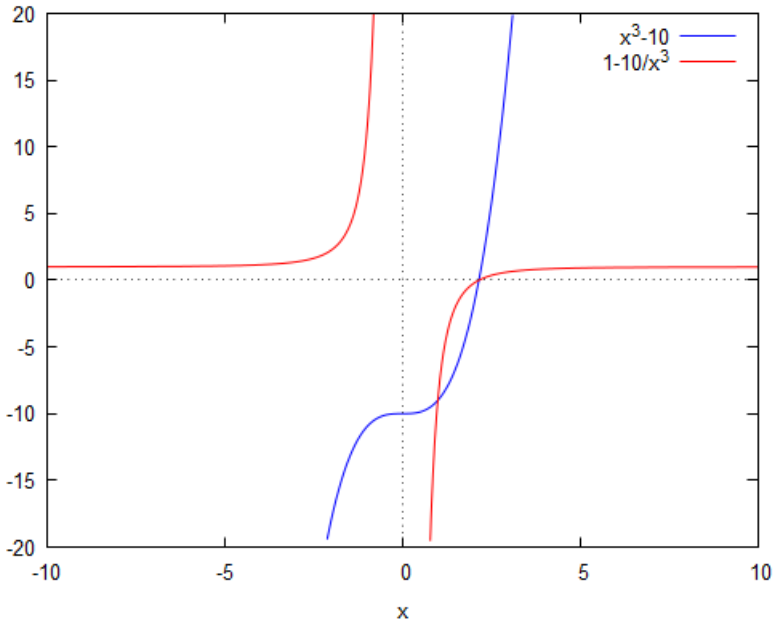
```
(%i42)nta:ratsimp(x-a/da);
```

(%o42) $$\frac{(m-1)\,x^{m+1} + R\,x}{m\,x^m}$$

```
(%i43)ntb:ratsimp(x-b/db);
```

(%o43) $$-\frac{x^{m+1} + (-R\,m - R)\,x}{R\,m}$$

Comparing both expressions obtained, *nta* and *ntb*, we can note the expression for the equation b is simpler than the one for equation a, as it doesn't involve divisions by the variable x, and so the it represents a continuous function. Expression b requires a division as well, but as it is by a constant we just optimize this process by calculating the dividend just one time or even transforming the division on a multiplication.
Comparing now the graphs of the equations, and assuming arbitrary values for $R = 10$ and $m = 3$:



The magnitude of the slope of the function influences the performance of the method because slopes with high values cause the convergence to slow down, while slopes near 0 cause fast convergences.
If the initial guess is at the right side of the root, than the expression b is better as the slope of the curve isn't big, compared to the expression a whose slope is almost a vertical line. On the other hand, if the initial guess is at the left side of the root, the expression a is slightly better as the slope is less than the slope of the expression b.
Overall the expression b gives us more advantages, and would be the one I would pick to solve this problem.

5 Exercise 5

```
from math import sin, cos, exp, log, sqrt
from copy import deepcopy


f = lambda x: 5*cos(x) - sin(x)

g_ratio = (sqrt(5)-1)/2

a = 2
b = 4
c = 2.76393
d = 3.23606

fc = f(c)
fd = f(d)

for i in range(2):

    if fc < fd:
        b = d
        d = c
        fd = fc
        c = b - g_ratio*(b-a)
        fc = f(c)
    else:
        a = c
        c = d
        fc = fd
        d = a + g_ratio*(b-a)
        fd = f(d)

    print(f"a = {a:.5f} \t b = {b:.5f} \t c = {c:.5f} \t d = {d:.5f}")
    print(f"fa = {f(a):.5f} \t fb = {f(b):.5f} \t fc = {fc:.5f} \t fd = {fd:.5f}")

if fc < fd:
    print(f"Interval Lenght: {d-a:.5f}")
else:
    print(f"Interval Lenght: {b-c:.5f}")
```

Program Output:

Iteration 1
a = 2.00000 b = 3.23606 c = 2.47213 d = 2.76393
fa = -2.99003 fb = -4.88338 fc = -4.54135 fd = -5.01639
Iteration 2
a = 2.47213 b = 3.23606 c = 2.76393 d = 2.94427
fa = -4.54135 fb = -4.88338 fc = -5.01639 fd = -5.09902
Interval Lenght: 0.47213