Table of Contents

Aufgabe 3

a)

Anonymous Function for 2^(-x)

```
epsilonFunc=@(k) 2.^(-k);

i = 0;
epsilon=epsilonFunc(i);
while 1 + epsilonFunc(i) > 1
    epsilon=epsilonFunc(i);
    i = i + 1;
end

fprintf('\nsmallest epsilon=%e\n',epsilon);
fprintf('\nk = %e\n\n',i);

smaller_epsilon = epsilon - 0.000001e-16;

if 1 + smaller_epsilon > 1
    fprintf('smaller epsilon exists: %e\n',smaller_epsilon);
else
    fprintf('smaller epsilon does not exist\n');
end
```

```
smallest epsilon=2.220446e-16

k = 5.300000e+01

smaller epsilon exists: 2.220445e-16
```

b)

Helper Functions

```
dbtype('topExp.m');
dbtype('bottomFactorial.m');
dbtype('topFactorial.m');
dbtype('bottomAddition.m');
dbtype('topAddition.m');
fprintf('\n');
```

```
iterations = 0;


M = 2;
top = inf;
bottom = 0;

% fast top approximation
[top,exp_iter] = topExp(M,1.01);
fprintf('Iterationen exponentielle Aproximation: %d\n', exp_iter);
M = top;

% fprintf('Eingrenzung nach oben: %d\n', M);


% factorial approximation
change_factor = 10;
reduce = 1;
t = 1;
change_factor_func = @(x) 2.^(-0.1*x)*5+1;
change_factor = 10;

fact_iter = 0;

while round(change_factor,12) ~= 1
    change_factor = change_factor_func(t);
    if reduce
        [bottom, iter] = bottomFactorial(M, change_factor);
        fact_iter = fact_iter + iter;
        M = bottom;
        reduce = 0;
    else
        [top,iter] = topFactorial(M, change_factor);
        fact_iter = fact_iter + iter;
        M = top;
        reduce = 1;
    end
    t = t + 1;
end

fprintf('Iterationen Approximation mit Faktor: %d\n', fact_iter);

% approximation with addition
add_iter = 0;

if reduce
    [bottom,add_iter] = bottomAddition(M,2);
    M = bottom;
    %fprintf('Eingrenzung nach unten: %d\n', M);
else
    [top,add_iter] = topAddition(M,2);
    M = top-2;
    %fprintf('Eingrenzung nach oben: %d\n', M);
```

```
end

fprintf('Iterationen Approximation durch Addition: %d\n', add_iter);

% check result
C = M+2;
if (M+1)-M == 1 && (C+1)-M ~= 1
    total_iter = exp_iter + fact_iter + add_iter;
    fprintf('Supremum fuer M: %d\n',M);
    fprintf('Anzahl Schleifendurchlaeufe zur Approximation: %d\n',
 total_iter);
else
    fprintf('Fehler im Programm\n');
end
```

```
function [y,i] = topExp(M,k)
    y = M;
    i = 0;
    while (y+1)-y==1
        y = y.^k;
        i = i + 1;
    end
end
```

```
function [y,i] = bottomFactorial(M, k)
% Function approximate with factor against the condition (y+1)-
y ~= 1
    y = M;
    i = 0;
    while (y+1)-y ~= 1
        y = y/k;
        i = i + 1;
    end
end
```

```
function [y,i] = topFactorial(M, k)
    y = M;
    i = 0;
    while (y+1)-y == 1
        y = y * k;
        i = i + 1;
    end
end
```

```
function [y,i] = bottomAddition(M,k)
    y = M;
    i = 0;
    while (y+1)-y ~= 1
        y = y - k;
        i = i + 1;
    end
end
```

```
function [y,i] = topAddition(M,k)
    y = M;
    i = 0;
    while (y+1)-y == 1
        y = y + k;
        i = i + 1;
    end

end
```

```
Iterationen exponentielle Aproximation: 47
Iterationen Approximation mit Faktor: 685
Iterationen Approximation durch Addition: 11
Supremum fuer M: 9007199254740990
Anzahl Schleifendurchlaeufe zur Approximation: 743
```

Published with MATLAB® R2016a