Table of Contents

Aufgabe 4	. 1
sum	
symsum	
Echtes Ergebnis	
naechster versuch mit abschaetzung $M = 8333332$. 2

Aufgabe 4

summe konvergiert. maximale maschinegenauigkeit ist erricht wenn f(t) == f(t-1) also wenn $1/(n^2) == 0$ um das zu beschleunigen (approximation), kann als abbruchbedingung round $(1/(n^2),x) == 0$ genommen werden

```
f = @(x) 1/(x.^2);
n = 1;
% Runden an Nachkommastelle
r = 10;

sum_f = 0;

while round(f(n),r) ~= 0
    sum_f = sum_f + f(n);
    n = n + 1;
end

fprintf('Ergebnisse (Iterationen):\n');
fprintf('Eigene Implementierung (%d):\t%e\n', n, sum_f);
% Alternitiv Matlab Style und wahrscheinlich genauer und schneller:
Ergebnisse (Iterationen):
Eigene Implementierung (141422): 1.644927e+00
```

sum

```
num_iter = 10^6;
n = 1 : num_iter;
sum_f = sum(1./(n.^2));
fprintf('Matlab sum (%d):\t\t\e\n', num_iter, sum_f);
Matlab sum (1000000): 1.644933e+00
```

symsum

```
syms k
s_sum = symsum(1/k^2, k, 1, num_iter);

fprintf('Matlab symsum (%d):\t\t%e\n', num_iter, s_sum);
```

```
Matlab symsum (1000000): 1.644933e+00
```

Echtes Ergebnis

```
fprintf('Echtes Ergebnis (PI^2)/6:\t\t%e\n', (pi.^2)/6); 
 Echtes Ergebnis (PI^2)/6: 1.644934e+00
```

naechster versuch mit abschaetzung M = 8333332

kleine Zahlen zuerst Addieren, wegen floating point Ungenauigkeiten

```
M = single(8333332);
j = M;
sum_f = single(0);
while j >= 1
    sum_f = sum_f + f(j);
    j = j - 1;
end

fprintf('Ergebnis mit Abschaetzung (%d):\t%e\n', M, sum_f);
Ergebnis mit Abschaetzung (8333332): 1.644934e+00
```

Published with MATLAB® R2016a