Algorithmische Methoden in der Numerik - Uebung1

Felix Dreßler (k12105003) Elisabeth Köberle (k12110408) Eva Zahn (k12110309)

20. Mai 2022

1 Funktion - estimateAbsError

Im folgenden steht die MatLab Funktion zur experimentellen Fehlerabschätzung einer beliebigen Funktion f und einem gegebenen Punkt x0 mit Schrittweite h.

Die Funktion erstellt Matrizen mit den jeweiligen Größen pxp, eine mögliche Optimierung dieser Vorgehensweise wäre die Speicherung der Werte in nur einen Vektor der Größe p.

```
%The function estimateAbsError takes as input a function
2
   % f, a point x0 and an increment h (if h is not provided
3
   % it defaults to (10^{-6})*((10^{-4})+abs(x0)))
4
    % It returns the estimated absolute Error near x0
6
    function [eps_A, exit_flag] = estimateAbsError(f,x0,h)
8
   exit_flag = 0;
9
   %argument checks:
   if nargin < 2
13
        error('more arguments required');
14
   end
   if nargin > 3
17
        error('too many arguments');
18
   end
20
   if nargin == 2 || h==0
21
      h = (10^{-6}) * ((10^{-4}) + abs(x0));
22
   end
23
24
    %steps from 7 to 11 instead of 6 to 10 because the array starts with 1 (not
    응()
    for p = 7:11
26
27
        temp = zeros(p,p);
28
        for k = 1:p
            %calculates the first coloum of the matrix usign
30
            %the functions output
            temp(k,1)=f(((k-1)*h)+x0);
32
        end
34
        for i = 2:p
            for j = 2:p-i+2
36
               %calculates the matrix-entries accordingly
               temp(j-1,i) = temp(j,i-1) - temp(j-1,i-1);
38
        end
40
41
        k=-1;
        for i = 2:p
42
            corr = 0;
43
            for j = 2:p-i+1
44
               if sign(temp(j-1,i)) \sim= sign(temp(j,i))
45
46
                   %counts the number of sign changes
47
                   corr = corr +1;
               end
            %checks all the requirements for k
            if ((corr >= p-1-i && i>=p-1-3) || (corr >= floor(0.8*(p-1-i)) && i < p-1-3)) &&
```

2 TESTS Page 2

```
(i \ge 2) \&\& (i \le p-1-2)
                 k = i+1;
52
                 break
54
            end
        end
        %if k was set (and thus satisfies the requirements) we break the loop
56
57
58
            break
        end
60
    end
61
    if p > 10 \mid \mid k == -1
62
63
        exit_flag = 1;
64
    end
65
    %the k that is used in this specific program is k-1 from
66
   %the assignment / script
67
68
   eps_A = max(abs(temp(:,k)))/sqrt(nchoosek(2*(k-1),(k-1)));
69
   end
```

2 Tests

2.1 Testfunktionen

Obiges Programm wurde mit folgenden Funktionen getestet:

```
function [f] = f1(x)
    if x == 1
        f = 3;
else
        y = x*x*x;
        f = (y-1)/(x-1);
end
```

```
function [f] = f2(x)
f = x*x+x+1;
end
```

2.2 Ergebnisse

Untenstehend ist der output des Programs mit den angegebenen Funktionen an den Stellen

```
x\theta = 0.1

x\theta = 10^{-3}

x\theta = -1,0000001

x\theta = 1,0000001
```

2 TESTS Page 3

```
7
8
         >> arrayfun(@(x0) estimateAbsError(@f1,x0),x0s)
9
         ans =
11
12
         1.326970321773750e-16
                                    9.930136612989092e-17
13
         5.307881287095000e-17
                                    4.784171007815724e-11
14
15
         >> arrayfun(@(x0) estimateAbsError(@f2,x0),x0s)
16
17
         ans =
18
19
         1.592364386128500e-16
                                    9.930136612989092e-17
20
         7.829083830833991e-18
                                 1.061576257419000e-16
```