

MATLAB AFLEVERING 1

Mathias Bruun Houmøller – 202006837

26. september 2022



"It's not a bug; it's an undocumented feature" — Anonymous

Opgave 2

a)

Main script

```
1 p = [-0.8 -1.1 2.5 1.7 0.3];
2 q = [-1.3 -0.9 -0.2 0.8
      -1.3];
3
4 %test af skalprod funktion
5 skalprod(p,q)
6
7 %for prik produktet kan
      kommandoen dot(a,b) også
      bruges
8 dot(p,q)
```

Output

```
1 ans = 2.5000
2
3 ans = 2.5000
```

Matlab function

```
1 function sp = skalprod(a,b)
2 % skalprod - skalar produktet af 2
      vektorer
3 % Kald: sp = skalprod(a,b)
4 % Input:
5 % a = Første vektor
6 % b = Anden vektor
7 % Output:
8 % sp = skalar, prik eller indre
      produktet af de to vektorer
9 m = length(a); n = length(b);
10 if m ~= n, error('Vektorerne skal v
      ære af samme størrelse'); end
11 sp = 0;
12 for i = 1:n
13     sp = sp + a(i) * b(i);
14 end
```

Bemærk at hvis det forsøgs at bruge to vektorer af forskellig længde kommer der en advarsel.

b)

Main script

```

1  z = [5 6 9]; %7 30 de
    udkommenterede tal og
    strings giver de errors
    som vi forventede
2  b = ["rejsekort", "kontant",
    "rejsekort"]; % "kort",
    "kontant"
3  a = ["barn", "voksen", "voksen"]; %"voksen",, "barn"
4
5  for i = 1:length(z)
6      billettakst(z(i), b(i), a(i))
7  end

```

Der benyttes forloop til at kontrollere alle værdierne, errors tages ud undervejs.

$z = 30$ er uden for vores antal zoner og vi får fejlen:

Error using billettakst Zoner skal være ml. 2 og 26

$b = \text{'kort'}$ er ikke en gyldig billet type og vi får fejlen:


Error using billettakst Ugyldig billetttype

Output

```

1  ans = 14.2500
2  ans = 60
3  ans = 51.5000

```



Matlab function

```

1  function pris = billettakst(zoner,
    billetttype, alder)
2  % billettakst - Skriver prisen på
    en bestemt billet type
3  % Kald: pris = billettakst(zoner,
    billetttype, alder)
4  % Input:
5  % zoner = antal zoner passageren
    skal køre
6  % billetttype = billetttypen der skal
    rejses med
7  % alder = alderen på passageren
8  % Output:
9  % pris = prisen på billetten
10 if zoner < 2 || 26 < zoner, error('
    Zoner skal være ml. 2 og 26'),
    end
11 if strcmp(billetttype, 'kontant')
12     if zoner <= 4,
13         pris = 10*zoner + 2;
14     else
15         pris = 10*zoner;
16     end
17 elseif strcmp(billetttype, 'rejsekort')
18     if zoner <= 3
19         pris = 5*zoner + 2.5;
20 elseif zoner <= 6
21     5.5*zoner + 1;
22 elseif zoner <= 8
23     pris = 6*zoner - 2;
24 elseif zoner <= 10
25     pris = 6*zoner - 2.5;
26 else
27     pris = 5.5*zoner + 2.5;
28 end
29 else
30 error('Ugyldig billetttype')
31 end
32 if strcmp(alder, 'barn')
33     pris = ans/2;
34 end

```

c)

Main script

```

1 syms x;
2 g_sym = x^2+2*x-1;
3
4 g_sym_diff = matlabFunction(
5     diff(g_sym, x));
6 g_sym_diff_sol = g_sym_diff
7     (2)
8 %Vi sammenligner med matlab
9     diff funktionen
10 g = @(x) x^2+2*x-1;
11 g_num_sol = numdiff(g,2)
12 forskel = g_sol -
13     g_num_diff_sol;
14 proc = forskel/
15     g_num_diff_sol * 100;
16 %forskellen er
17 fprintf(['Forskellen imellem
18     symbolsk og numerisk er
19     således: %12.4f' ...
20     '\ndette er en forskel i
21     procent på: %30.4f
22     %%'], forskel, proc)

```

Matlab function

```

1 function dfdx = numdiff(f,x)
2 % numdiff - Numerisk
3     differentiation
4 % Kald: dfdx = numdiff(f,x)
5 % Input:
6 % f = funktion hvis
7     differentialkvotient skal findes
8 % x = værdi af x for hvilken
9     differentialkvotienten skal
10     findes
11 % Output:
12 % dfdx = differentialkvotient
13 h = 10^-3;
14 dfdx = (f(x + h) - f(x))/h;
15 end

```

Output

```

1 g_sym_diff_sol = 6
2 g_num_sol = 6.0010
3 Forskellen imellem symbolsk
4     og numerisk er således:
5     0.0010
6 dette er en forskel i
7     procent på:
8
9     0.0167 %

```

Opgave 4

a)

Main script

```

1 %Vi tester funktion som skal
   give 1010
2 ruteafstand([2 4 5 7 3 1 2])
3
4 ruteafstand([2]) % her fåes
   0 da vi bliver i vores
   start by.
5 % Vi køre altså ikke fra
   start byen til en ny by.

```


Når vi kun har en værdi i vores vektor har vi kun vores start punkt og dermed kan der ikke findes en afstand til et slutpunkt.

Output

```

1 ans = 1010
2 ans = 0

```




Matlab function

```

1 function afstand = ruteafstand(bynr
   )
2 % ruteafstand - afstand imellem
   byer
3 % Kald: afstand = ruteafstand(bynr)
4 % Input:
5 % bynr = By nr under byen du
   starter i samt by nr på byerne
6 % du vil igennem i rækkefølge.
7 % Output:
8 % afstand = afstanden der skal
   rejses på ruten
9 atbl = [ 0 307 168 417 298 336 232
   266 237 35
10 307 0 145 119 165 39 99 52 72 273
11 168 145 0 255 136 174 70 105 75 134
12 417 119 255 0 274 80 208 161 181
   382
13 298 165 136 274 0 195 72 125 96 264
14 336 39 174 80 195 0 129 81 102 302
15 232 99 70 208 72 129 0 59 29 197
16 266 52 105 161 125 81 59 0 29 230
17 237 72 75 181 96 102 29 29 0 206
18 35 273 134 382 264 302 197 230 206
   0];
19
20 n = length(bynr);
21 afstand = 0;
22 for i = 1:n-1
23     afstand = afstand + atbl(bynr(i
   ),bynr(i+1));
24 end

```



b)

Main script

```

1  xs = [1 3 3.5 5 5.2 5.7
        6.2];
2  ys = [1 2.5 5.8 3 4.6 3 2];
3
4  for i = 1:length(xs)
5      svar = iomr(xs(i), ys(i))
6      fprintf('Er koordinatsæt
              %i i området?\t %s \n', i, svar)
7  end

```

Vi benytter et forloop til at teste alle svarende.

Output

```

1  Er koordinatsæt 1 i området?
    nej ✓
2  Er koordinatsæt 2 i området?
    ja ✓
3  Er koordinatsæt 3 i området?
    ja ✗
4  Er koordinatsæt 4 i området?
    ja ✓
5  Er koordinatsæt 5 i området?
    ja ✗
6  Er koordinatsæt 6 i området?
    ja ✓
7  Er koordinatsæt 7 i området?
    nej ✓

```

Matlab function

```

1  function i = iomr(x,y)
2  % iomr - Bestemmelse om punktet er
   i vores område
3  % Kald: i = iomr(x,y)
4  % Input:
5  % x = x-koordinat
6  % y = y-koordinat
7  % Output:
8  % i = 'ja' hvis punktet er i vores
   område
9  % i = 'nej' hvis punktet er uden
   for området
10 if x < 2
11     i = 'nej';
12 elseif (-x/2+4 <= y || y <= x/2+4)
   && (2 <= x && x < 4)
13     i = 'ja';
14 elseif (2 <= y || y <= -x^2 + 8*x -
   10) && (4 <= x && x <= 6)
15     i = 'ja';
16 else
17     i = 'nej';
18 end

```

Det bemærkes også at testen også kontrollerer kanterne af området som er blevet inkluderet med ' \leq ' tegnet så værdien medtages.