

Tentamen Programmeren in MATLAB

Docenten: Dr. H. M. Cuppen en Prof. dr. ir. G. C. Groenenboom

28 januari 2015, 8:30-11:30
Maak iedere opgave op een apart blad

Opgave 1 Matlab basis

Deze vraag is gebaseerd op de onderstaande code.

```
1 function [average, stdev] = stat(x)
2     if (nargin ~= 1)
3         if (nargin > 1)
4             error('Too many input arguments')
5         else
6             error('Missing argument')
7         end
8     end

9     [n1, n2] = size(x);
10    if (n1 ~= 1 && n2 ~= 1) && (length(x) == 1)
11        error('x is not a vector')
12    end

13    n      = max(n1, n2);
14    average = avg(x, n);
15    stdev   = sqrt(sum((x - average).^2)/(n-1));

16    function [average] = avg(x, n)
17        average = sum(x)/n;
18    end
19 end
```

a) Schrijf een help-beschrijving voor deze code. Waar moet deze worden ingevoegd (geef regelnummers) en hoe kan deze worden opgevraagd?

b) Wat geeft MATLAB terug voor de volgende input statements?

- (1) `y = ones(5,1);`
 `a = stat(y)`
- (2) `y = ones(5,1);`
 `[a,s] = stat(y)`
- (3) `x = rand(5,3);`
 `a = stat(x)`
- (4) `x = rand(5,3);`
 `a = stat(x, y)`
- (5) `y = ones(5,1);`
 `a = avg(y, 5)`

c) Je krijgt twee databestanden `data1.dat` en `data2.dat`. `data1.dat` bevat een grote collectie aan meetwaarden aan bodemonsters uit een bepaald gebied. Bestand `data2.dat` bevat dezelfde type meetgegevens van grondmonsters afkomstig uit een onbekend gebied. Schrijf een script om te testen of `data2.dat` uit hetzelfde gebied kan komen als `data1.dat`. Gebruik hiervoor de bovenstaande functie `stat` en de functie `ttest`. Deze laatste heeft de volgende help

```

1  TTEST One-sample and paired-sample t-test.
2  H = TTEST(X) performs a t-test of the hypothesis that the data in the
3  vector X come from a distribution with mean zero, and returns the
4  result of the test in H. H=0 indicates that the null hypothesis
5  ("mean is zero") cannot be rejected at the 5% significance level. H=1
6  indicates that the null hypothesis can be rejected at the 5% level.
7  The data are assumed to come from a normal distribution with unknown
8  variance.

9  X can also be a matrix or an N-D array. For matrices, TTEST performs
10 separate t-tests along each column of X, and returns a vector of
11 results. For N-D arrays, TTEST works along the first non-singleton
12 dimension of X.

13 TTEST treats NaNs as missing values, and ignores them.

14 H = TTEST(X,M) performs a t-test of the hypothesis that the data in
15 X come from a distribution with mean M. M must be a scalar.

```

d) Wat is de waarde van a, b en c. Als we beginnen met de volgende vijf startwaarden voor a: 1, 2, 2.5, 3, 4

```

1  b = a + 3;
2  while b < 6
3      b = b + 1;
4  end
5  if b + a == 9
6      c = a
7  elseif b + a < 8
8      c = 4
9  else
10     c = 5
11 end

12 for j = c:4
13     a = a + 1;
14 end
15 disp (a)

```

Opgave 2 Verbeter de code

a) Vind de vijf fouten in de onderstaande code. Geef voor ieder van de fouten (i) wat de fout is (plus regelnummer), (ii) het type fout (syntax, runtime, logic) en (iii) een vervangend stukje code.

```

1  function absx = absolute(x, display)
2  % ABSOLUTE determines the absolute value of x
3  % absx = absolute(x, display)
4  % if display is true (1), x and absx will be displayed
5      absx = x;
6      for k = 0:length(x)
7          if x(k) > 0
8              absx(k) = -x(k);
9          end
10     end
11     if display = 1
12         disp(X)
13         disp(absx)
14     end
15 end

```

b) Geef vervangende code voor regels 5 tot en met 10 met behulp van het `find`-statement.

Opgave 3 Uitwerken van een probleem

- a) De sneeuwval in Zwitserland (p) hangt kwadratisch af van de hoogte (a) op de berg

$$p = c_0 + c_1 a + c_2 a^2.$$

Schrijf een script dat de coëfficiënten c_0 , c_1 en c_2 berekent gegeven de kolomvectoren $\mathbf{p}$ en $\mathbf{a}$. De vector $\mathbf{p}$ bevat de gemeten sneeuwval over een seizoen. De corresponderende hoogte waar deze sneeuwval gemeten is, is gegeven in vector $\mathbf{a}$.

- b) We willen een programma ontwerpen dat de onderstaande situatie kan simuleren. Dit programma zal bestaan uit een script en verschillende functies. Schrijf op (i) hoe je dit probleem aan zou pakken, (ii) welke functies je zou schrijven, (iii) wat de functies moeten kunnen, (iv) wat hun in- en outputparameters moeten zijn en (v) wat er in je script zou komen waarmee je de functies aanroept. Je hoeft dus niets te programmeren. Maak je programma zo algemeen mogelijk.

Klassieke simulatie van moleculen in een doos. Posities en snelheden van de deeltjes staan in een bestand. De versnelling kan berekend worden uit de kracht die op ieder deeltje werkt en vervolgens kan in kleine tijdstappen de nieuwe positie en snelheid worden bepaald. De kracht wordt bepaald uit de afgeleide van de potentiële energie naar de plaats. Neem aan dat de potentiële energie berekend kan worden met behulp van analytische uitdrukkingen als functie van de positie. Na een bepaald aantal stappen willen we de nieuwe posities weg kunnen schrijven en bijvoorbeeld ook de gemiddelde kinetische energie van het systeem en de gemiddelde potentiële energie. We willen het aantal deeltjes, het aantal tijdstappen en hun lengte en de schrijffrequentie makkelijk kunnen variëren.