



Einführung in die Informatik I

Übung 5

1 Elementare Vektoroperationen

MATLAB stellt über verschiedene spezialisierte Operatoren und eingebaute Funktionen eine Reihe von elementaren Vektoroperationen bereit:

- Löschen des i -ten Eintrags des Vektors x .
- Einfügen eines neuen Eintrags a an Stelle i in einen Vektor x .
- Finden des Minimums oder Maximums der Einträge eines Vektors x .
- Umkehrung der Reihenfolge der Elemente eines Vektors x (Spiegeln).
- Zyklisches Weiterschieben der Elemente eines Vektors x (Länge n) um k Stellen:

$$y_i = x_{i-k} \quad \text{für } i > k$$
$$y_i = x_{n-k+i} \quad \text{für } i \leq k$$

- Differenz je zweier aufeinanderfolgender Einträge des Vektors x :

$$y_i = x_{i+1} - x_i$$

- Summe und Produkt aller Vektoreinträge.
- Löschen eines bestimmten Wertes (oder einer bestimmten Zahl) aus einem Vektor.

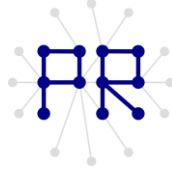
Aufgabenstellung:

- Finden Sie heraus, wie sich die obigen Aufgaben in einer Zeile mit den Vektoroperationen oder eingebauten Funktionen von MATLAB lösen lassen.
- Programmieren Sie alternativ jede der Operationen als Funktion, wobei nur der Zugriffsoperator $v[i]$ und Schleifen verwendet werden dürfen.

2 Verfolgen und analysieren Sie den Programmablauf

Welches Ergebnis liefern die folgenden Funktionen? Falls sie kein Ergebnis liefern, beschreiben Sie kurz das Problem.

<pre>function a = f1(x,y,b) if x + y < b a = floor(x); elseif y + b > x a = ceil(y); else a = fix(b^12); end</pre>	<pre>f1(1.75,1.05,0.5) = f1(1.5,2.05,0.57) =</pre> <p style="text-align: right;">1</p>
--	--



<pre>function v = f2(x,n) v = zeros(n,1); for i = 1:round(n/2)+x v(i) = n * x; end</pre>	<pre>f2(2,-1) = f2(1,4) =</pre> <p style="text-align: right;">2</p>
<pre>function x = f3(m) while m > 0 m = round(m/3) - 2; end x=m;</pre>	<pre>f3(36) = f3(-26.75) =</pre> <p style="text-align: right;">3</p>
<pre>function x = f4(v) x = 1; for i = length(v):-1:2 x = x + v(i) - v(i+1); end</pre>	<pre>f4([1,2,3,4]) = f4([1]) =</pre> <p style="text-align: right;">4</p>
<pre>function x = f5(M) i = M(1,1); j = M(1,2); if i>size(M,1) j>size(M,2) x = 0; else x = M(i,j); end</pre>	<pre>f5([-1 47 11;0 8 15]) = f5([-10 1;42 69]) =</pre> <p style="text-align: right;">5</p>

3 Transponieren einer Matrix

Schreiben Sie eine Funktion namens **transp**, die eine quadratische Matrix A transponiert, indem sie ihre Elemente an der Diagonalen (von links oben nach rechts unten) spiegelt. Die Aufgabe soll allein durch Vertauschungen in der Matrix gelöst werden. Es empfiehlt sich die Verwendung von zwei ineinander verschachtelten `for`-Schleifen! Verwenden Sie das folgende Gerüst:

```
1 function A = transp(A)
2
3 % Vertauschungen in A (zu programmieren):
4 for ...
5     for ...
6         ...
7     end;
8 end;
```

Hinweis: Transponieren einer Matrix erfolgt im MATLAB auch durch das Hochkomma z.B. A'