

# **Teme de implementare in Matlab pentru Laboratorul de Metode Numerice**

As. Ruxandra Barbulescu  
Septembrie 2017

Orice nelamurire asupra enunturilor/implementarilor se rezolva in cadrul laboratorului de MN, in cursul zilei de luni, oricand intre orele 8 - 16, in afara zilelor in care se dau colocviile.

## 1 Setul 1 (pregatire pentru colocviul 1)

### ① Adunarea a doua numere.

Scrieti un script care aduna doua numere intregi/reale si afiseaza suma lor in consola. Salvati scriptul creat si executati-l.

### ② Compara numere.

Scrieti o functie care compara doua numere reale  $a$  si  $b$ .

(a) Daca  $a > b$  afiseaza  $a$  cu un text sugestiv.

(b) Daca  $a < b$  afiseaza  $b$  cu un text sugestiv.

Salvati functia si executati-o. In cate moduri poate fi executata aceasta functie?

### ③ Suma elementelor a doi vectori

(a) Pe hartie: Generati doi vectori de lungime 5 cu valori alese de voi.

(b) Pe hartie: Determinati vectorul ale carui elemente reprezinta suma elementelor cu acelasi index ale celor doi vectori.

(c) Pe calc: Scrieti o functie care calculeaza vectorul-suma de la punctul anterior si afiseaza rezultatul.

(d) Pe calc: Scrieti un program principal sub forma de script care initializeaza cei doi vectori, apeleaza functia si afiseaza rezultatul ei. Indicatie: Ce modificari va suferi functia scrisa initial?

(e) Pe calc: Executati programul principal.

### ④ Reprezentarea grafica a vectorilor

(a) Pe calc: Generati automat un vector care sa contina valorile 20, 40, 60, 80, 100. Acest vector va reprezenta dimensiunea problemei. Indicatie: *linspace*

(b) Pe hartie: Generati un vector de lungime 5 cu valori reale intre 20 si 100, in ordine crescatoare. Acest vector va reprezenta timpul masurat pentru rezolvarea unei probleme oarecare.

(c) Pe calc: Initializati vectorul cu valorile de la punctul anterior.

(d) Pe calc: Reprezentati grafic cei doi vectori (timp vs. dimensiunea problemei) prin puncte. Adaugati graficului un titlu relevant, denumiti axele si stabiliti limitele axelor. Indicatie: *plot, title, xlabel, ylabel, axis*.

(e) Pe calc: Initializati un nou vector, cu valorile: 0.00006, 0.0003, 0.004, 0.02, 0.5. Acest vector contine erorile calculate pentru fiecare dimensiune a problemei.

(f) Pe calc: Reprezentati pe acelasi grafic si acest vector (erori vs. dimensiunea problemei), prin puncte unite de o linie. Puteti spune doar uitandu-va pe grafic care este valoarea erorii pentru dimensiune 40 ?

### ⑤ Produsul scalar a doi vectori

(a) Pe hartie: Generati doi vectori de lungime 4 cu valori alese de voi.

(b) Pe hartie: Calculati produsul scalar al celor doi vectori.

(c) Pe hartie: Incercati sa va imaginati structura codului care ar calcula produsul scalar, scrieti pseudocodul.

(d) Pe calc: Implementati o functie care realizeaza produsul scalar a doi vectori de dimensiune  $N$ .

- ce date de intrare ar trebui sa aiba functia?
  - ce date de iesire?
  - este nevoie de variabile auxiliare?
  - de cati indecsi este nevoie, intre ce valori?
  - ce structura repetitiva veti folosi?
- (e) Pe calc: Scrieti un program principal care apeleaza functia si testati-l pentru vectorii generati de voi la punctul a)

⑥ Ce face algoritmul urmator?

```
functia nume (n, x, a)
  intreg n
  tablou real x(n), a(n, n), rez(n)
  intreg i, j
  pentru i = 1, n
    rezi = 0
    pentru j = 1, n
      rezi = rezi + aij · xj
intoarce rez
```

- (a) Pe hartie: Scrieti ce reprezinta fiecare variabila de intrare (ce tip de date, ce dimensiuni, eventual desenate).
- (b) Pe hartie: Scrieti ce reprezinta variabila/ele de iesire (ce tip de date, ce dimensiuni, eventual desenate).
- (c) Pe hartie: Identificati structurile de decizie (care sunt conditiile?) si repetitive (care sunt indecsii?).
- (d) Pe hartie: Comentati fiecare linie de cod.
- (e) Pe hartie: Identificati ce face algoritmul.
- (f) Pe calc: Implementati algoritmul si testati-l intr-un program principal.
- (g) Pe hartie: Care este complexitatea algoritmului din punct de vedere al timpului de calcul ( $T=O(?)$ ) in functie de dimensiunea problemei (care este aceasta)?
- (h) Pe hartie: Care este complexitatea algoritmului din punct de vedere al necesarului de memorie ( $M=O(?)$ ) in functie de dimensiunea problemei?
- (i) Pe calc: Puteti optimiza algoritmul prin reducerea timpului de calcul? Daca da, modificati programul implementat.

7 Evaluarea unui polinom

$$P(x) = a_0 + a_1x^1 + a_2x^2 + \dots + a_nx^n = a_0 + \sum_{i=1}^n a_ix^i$$

- Pe hartie: Generati un polinom de gradul 3.
- Pe hartie: Evaluati polinomul generat pentru  $x=2$ .
- Pe hartie: Incercati sa va imaginati structura codului care ar evalua polinomul, scrieti pseudocodul.
- Pe calc: Implementati o functie care evalueaza un polinom de gradul  $N$ .
  - ce date de intrare ar trebui sa aiba functia?
  - ce date de iesire?
  - este nevoie de variabile auxiliare?
  - de cati indecsi este nevoie, intre ce valori?
  - ce structura repetitiva veti folosi?
- Pe calc: Scrieti un program principal care apeleaza functia si testati-l pentru polinomul generat de voi la punctul a)

8 Ce face algoritmul urmator?

```
procedura nume (n, x)
  intreg n, i
  tablou real x(n)
  i = 1
  cat timp (i ≤ n)
    daca (xi ≥ 0)
      scrie xi
    i = i + 1
retur
```

- Pe hartie: Scrieti ce reprezinta fiecare variabila de intrare (ce tip de date, ce dimensiuni, eventual desenate).
- Pe hartie: Scrieti ce reprezinta variabila/ele de iesire (ce tip de date, ce dimensiuni, eventual desenate).
- Pe hartie: Identificati structurile de decizie (care sunt conditiile?) si repetitive (care sunt indecsii?).
- Pe hartie: Comentati fiecare linie de cod.
- Pe hartie: Identificati ce face algoritmul.
- Pe calc: Implementati algoritmul si testati-l intr-un program principal.
- Pe hartie: Care este complexitatea algoritmului din punct de vedere al timpului de calcul ( $T=O(?)$ ) in functie de dimensiunea problemei (care este aceasta)?
- Pe hartie: Care este complexitatea algoritmului din punct de vedere al necesarului de memorie ( $M=O(?)$ ) in functie de dimensiunea problemei?
- Pe calc: Puteti optimiza algoritmul prin reducerea timpului de calcul? Daca da, modificati programul implementat.

9) Dezvoltarea in serie Taylor a functiei  $y = \cos(x)$

$$y = \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

- (a) Pe hartie: Alegeti o valoare pentru  $x \neq 0$  (in radiani, ex.  $\pi/4, \pi$ ).
- (b) Pe hartie: Incercati sa va imaginati structura codului care ar evalua functia cosinus. Care va fi valoarea initiala a sumei? Cati termeni veti adauga la suma pentru a va asigura ca va apropiati de valoarea exacta?
- (c) Pe calc: Implementati o functie care evalueaza functia cosinus pentru o valoare  $x$  si o eroare de trunchiere impusa.
- ce date de intrare ar trebui sa aiba functia?
  - ce date de iesire?
  - este nevoie de variabile auxiliare?
  - ce structura repetitiva veti folosi?
  - puteti calcula termenii recursiv?
- (d) Pe calc: Scrieti un program principal care apeleaza functia si afiseaza valoarea intoarsa. Pentru verificare, afisati si valoarea intoarsa de functia din Matlab  $\cos(x)$ . Testati programul pentru cateva valori, inclusiv cea aleasa de voi la punctul a).
- (e) Pe calc: Modificati codul astfel incat eroarea de trunchiere sa fie mai mica decat eroarea de rotunjire. Indicatie: se modifica eroarea impusa.

10) Ce face algoritmul urmator?

```
procedura who_am_i( $n, a, b, x$ )  
  tablou real  $a(n, n), b(n), x(n)$   
  pentru  $k = 1, n - 1$   
    pentru  $i = k + 1, n$   
      pentru  $j = k + 1, n$   
         $p = a_{ik}/a_{kk}$   
         $a_{ij} = a_{ij} - a_{kj}p$   
       $b_i = b_i - b_k p$   
   $x_n = b_n/a_{nn}$   
  pentru  $i = n - 1, 1, -1$   
     $s = b_i$   
    pentru  $j = n, i + 1, -1$   
       $s = s - a_{ij}x_j$   
     $x_i = s/a_{ii}$   
retur
```

- (a) Pe hartie: Scrieti ce reprezinta fiecare variabila de intrare (ce tip de date, ce dimensiuni, eventual desenati).
- (b) Pe hartie: Scrieti ce reprezinta variabila/ele de iesire (ce tip de date, ce dimensiuni, eventual desenati).
- (c) Pe hartie: Identificati structurile de decizie (care sunt conditiile?) si repetitive (care sunt indecsiile?).
- (d) Pe hartie: Comentati fiecare linie de cod.

- (e) Pe hartie: Identificati ce face algoritmul.
- (f) Pe calc: Implementati algoritmul si testati-l intr-un program principal.
- (g) Pe hartie: Care este complexitatea algoritmului din punct de vedere al timpului de calcul ( $T=O(?)$ ) in functie de dimensiunea problemei (care este aceasta)?
- (h) Pe hartie: Care este complexitatea algoritmului din punct de vedere al necesarului de memorie ( $M=O(?)$ ) in functie de dimensiunea problemei?
- (i) Pe calc: Puteti optimiza algoritmul prin reducerea timpului de calcul? Daca da, modificati programul implementat.

11) Produsul a doua matrice

- (a) Pe hartie: Generati doua matrice de dimensiune  $3 \times 3$  cu valori alese de voi.
- (b) Pe hartie: Calculati produsul celor doua matrice. Pe calc: Verificati rezultatul in Matlab.
- (c) Pe hartie: Incercati sa va imaginati structura codului care ar calcula produsul celor doua matrice, desenati, scrieti pseudocodul.
- (d) Pe calc: Implementati o functie care realizeaza produsul a doua matrice de dimensiuni  $N \times N$ .
  - ce date de intrare ar trebui sa aiba functia?
  - ce date de iesire?
  - este nevoie de variabile auxiliare?
  - ce structura repetitiva veti folosi?
  - de cati indecsi este nevoie, intre ce valori?
- (e) Pe calc: Scrieti un program principal care apeleaza functia si testati-l pentru matricele generate de voi la punctul a)
- (f) Pe calc: Completati programul principal astfel incat sa masoare si sa afiseze timpul de calcul al produsului celor doua matrice. Indicatie: tic, toc
- (g) Pe calc: Modificati codul astfel incat sa calculeze timpul de calcul pentru produsul a doua matrice de dimensiuni  $100 \times 100$ ,  $500 \times 500$ ,  $1000 \times 1000$ ,  $5000 \times 5000$  generate automat (valori aleatorii) si sa afiseze graficul Timp vs. Dimensiunea problemei (nr. de linii/coloane).

=====