Naloge za Matlab – 4. sklop

1. Sestavi preprosto opisno datoteko, v kateri definiraš matriko A velikosti 4x4 in stolpec b s štirimi elementi! V opisni datoteki izračunaj še vektor c, ki reši sistem enačb Ax=b. Z uporabo podpičja prepreči izpis vseh prirejanj in računov! V delovnem oknu poženi opisno datoteko. Z ukazi A, b in c izpiši podatke in rezultat.

2. (**Opisna datoteka ravninskega paličja**) Ravninsko paličje lahko v Matlabu preprosto podamo z nekaj matrikami:

   - matrika koordinat vozlišč (npr T=[0 0; 0 2; 2 0; 2 2; 4 0; 4 2; 6 0; 6 2]);
   - matrika povezav med vozlišči za palice (npr. pq= [1 2; 1 3; 1 4; 2 3; 2 4; 3 4; 3 5; 3 6; 4 5; 4 6; 5 6; 5 7; 5 8]);
   - matriki lastnosti palic (npr. EI=19E+12, A=0.000362);
   - vektor fiksnih prostostnih stopenj (npr. fix=[1 2 17 18]);
   - matrika obtežb (npr. f=[11 1; 12 –4]).

   (a) Za navedene (ali za svoje) podatke sestavi opisno datoteko ravninskega paličja podatki_palicja.m! Datoteka naj bo čimbolj pregledna – vrstice matrik naj torej ne bodo ločene s podpičjem, temveč zapisane v različnih vrsticah datoteke.

   (b) V ukaznem oknu prikliči podatke o paličju in nariši graf vozlišč paličja (to je zgolj graf točk, ne povezav, zato uporabimo prameter; npr '+' ali '*').

   (c) Nariši paličje.

   (d) Nariši podpore.

3. Napiši funkcijo, ki izračuna hipotenuzo pravokotnega trikotnika z znanima katetama. Poskusi zapisati funkcijo tako, da bo delovala tudi za več trikotnikov hkrati (prvi parameter je stolpec prvih katet vseh trikotnikov, drugi parameter je stolpec drugih katet vseh trikotnikov).

4. Napiši funkcijo, ki vrne dolžino tretje stranice trikotnika po kosinusnem pravilu

   $$c^2 = a^2 + b^2 - 2abcos(t),$$

   kjer sta a in b znani stranici in t kot med njima.

5. Napiši funkcijo, ki izračuna vrednost števila π po formuli

   $$\frac{\pi^2 - 8}{16} = \sum_{n=1}^{\infty} \frac{1}{(2n-1)^2\,(2n+1)^2}.$$

   Seveda ne računamo neskončne vsote, temveč seštevamo do nekega velikega naravnega števila N. Za dani veliki N naj torej funkcija vrne oceno za π. To oceno primerjaj z vgrajeno vrednostjo za različne N.

6. (**Koordinatna transformacija**) Eulerjeva kota ϑ in ψ določata koordinatno transformacijo iz referenčnega v lokalni koordinatni sistem. Pripadajoča transformacijska matrika je

$$\mathsf{Q} = \begin{bmatrix} \cos\psi\cos\vartheta & \sin\psi\cos\vartheta & -\sin\vartheta \\ -\sin\psi & \cos\psi & 0 \\ \cos\psi\sin\vartheta & \sin\psi\sin\vartheta & \cos\vartheta \end{bmatrix}.$$

Napiši funkcijo, ki za dana Eulerjeva kota ϑ in ψ ter dani vektor $\mathsf{X} = [X_1, X_2, X_3]^T$, zapisan v referenčni bazi, vrne pripadajoči vektor $\mathsf{x} = \mathsf{QX}$ v lokalni bazi. V ukaznem oknu preveri, kam se preslikajo bazni vektorji ($[1\ 0\ 0]^T$, $[0\ 1\ 0]^T$ in $[0\ 0\ 1]^T$) za nekaj različnih Eulerjevih kotov (npr. $\vartheta = \pi/2$ in $\psi = 0$; $\vartheta = \pi$ in $\psi = -\pi$; $\vartheta = \pi/3$ in $\psi = \pi/6$...).

7. **(Rotacijska matrika)** Rotacija v prostoru je določena z enotskim vektorjem $\mathsf{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}$ in s kotom zasuka ϑ. Če definiramo matriko $\mathsf{N} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$, je s formulo $\mathsf{R} = \mathsf{I} + \sin\vartheta\mathsf{N} + (1-\cos\vartheta)\,\mathsf{N}^2$ določena rotacijska matrika.

   (a) Napiši funkcijo RotM, ki za dani kot in enotski vektor na osi rotacije vrne rotacijsko matriko!

   (b) Izračunaj rotacijsko matriko za vektor rotacije $\mathsf{n} = \frac{1}{\sqrt{3}}[1\ {-1}\ \ 1]^T$ in kot rotacije $\vartheta = \pi/6$.

   (c) Funkcijo RotM izboljšaj, da bo delovala tudi za neenotske vektorje na osi rotacije. (Dodaj vrstico, kjer vektor osi rotacije normiraš tako, da postane enotski).

   (d) Z izboljšano funkcijo izračunaj rotacijsko matriko za vektor rotacije $\mathsf{n} = [1\ {-1}\ \ 1]^T$ in kot rotacije $\vartheta = \pi/6$ ter primerjaj rezultate s tistimi iz točke (b).

   (e) Razširi funkcijo RotM tako, da bo vrnila tudi matriko N. Matrika N naj bo drugi izhodni parameter. V ukaznem oknu z uporabo razširjene funkcije izračunaj še N.

8. Z ukazoma who in whos preveri stanje spremenljivk v delovnem oknu. Prepričaj se, da so lokalne spremenljivke v funkcijah res neznane v delovnem okolju.

9. Napiši dve preprosti funkciji. Vhodni parameter naj bo n, izhodni pa m. Telo obeh funkcij je predpisano!

   (a) Telo prve funkcije:
```
if n > 1
   m = n+1;
else
   m = n − 1;
end
```
   (b) Telo druge funkcije:
```
if n < 5
   m = 2*n;
elseif n < 10
   m = 9 − n;
elseif n < 100
   m = sqrt(n);
else
   m = n;
end
```

   Napovej rezultate in jih preveri za n=–10, 0, 1, 7, 80, 300.

10. Napiši funkcijo, ki za dano vrednost x izračuna

$$\mathsf{y(x)} = \begin{cases} 2 \quad \text{za } x < 6 \\ x - 4 \quad \text{za } 6 <= x < 20 \\ 36 - x \quad \text{za } 20 <= x \end{cases} .$$

11. Permutacijski simbol e_{ijk} je definiran takole

$$\begin{aligned} e_{123} = e_{231} = e_{312} = 1 \\ e_{132} = e_{213} = e_{321} = -1, \end{aligned}$$

   za vse preostale nabore indeksov pa je enak nič. Napiši funkcijo, ki za dano permutacijo števil 1, 2 in 3 vrne vrednost permutacijskega simbola! Namig: nabor [i j k] je enak [1 2 3], če ukaz all([i j k]==[1 2 3]) vrne 1.

12. Napiši funkcijo, ki za dani vektor x izračuna

   (a) vsoto elementov x (uporabi zanko for);
   (b) vektor delnih vsot s (s(j) je vsota elementov v x z indeksi od 1 do j).

   Funkcijo testiraj vsaj na vektorju x = [1 8 3 9 0 1]!

13. Za dana vektorja x in y napiši funkcijo, ki vrne naslednje količine:

    (a) matriko A z elementi A(i,j)=x(i)/y(j);
    (b) matriko B z elementi B(i,j)=x(i)*y(N–j), kjer je N velikost vektorja y;
    (c) matriko C z elementi C(i,j)=x(j)*y(i)/(2+x(i)+y(j)).

    Rezultate za x = [4 1 6] in y = [6 2 7] izračunaj "peš" in z napisano funkcijo.

14. Napiši funkcijo, ki za dano matriko pregleda vse elemente (dve zanki for) in vse elemente, ki so manjši kot 0.3 postavi na 0, vse ostale pa na 1. Funkcijo testiraj za naključno matriko poljubne dimenzije! Ali znaš nalogo rešiti brez zank for? (Dodatno: uporabi funkcijo s for zankama in funkcijo brez na zelo veliki naključni matriki. Z uporabo funkcije cputime (help cputime) se prepričaj o njunih hitrostih)

15. Število π lahko računamo tudi po naslednjem algoritmu (http://www.netcom.com/˜hjsmith/Pi/Gauss_L.html):

    (a) definiraj a = 1, b = 1/sqrt(2), t = 1/4 in x = 1;
    (b) ponavljaj naslednje ukaze, dokler razlika med a in b ni dovolj majhna:
        y = a;
        a = (a + b)/2;
        b = sqrt(b*y);
        t = t – x*(y – a)^2;
    (c) ocena za pi je moj_pi = ((a + b)^2)/(4*t).

    Napiši funkcijo ki za dano natančnost (razliko med a in b) oceni število π po opisanem algoritmu!

16. Napiši funkcijo, ki za dano število n izvaja naslednja ukaza, dokler je n večji od 1:

    (a) za sode n naj n postane n/2;
    (b) za lihe n pa naj n postane 3*n+1.

    Funkcija naj vrne vektor števil, ki jih je zavzel n, dokler se je zanka izvajala.