Zadatak 1.

Zadane su matrice $A=\begin{bmatrix}-1 & 2\\ 3 & -4\end{bmatrix}$ i $B=\begin{bmatrix}9 & -8\\ -7 & 6\end{bmatrix}$. Odredite matricu $C=\left(-\frac{3}{2}\right)\cdot A^T-\left(\frac{1}{2}\cdot B\right)^T$.

Zadatak 2.

Zadane su matrice $A=\begin{bmatrix}7 & 6\\ 5 & 4\end{bmatrix}$ i $B=\begin{bmatrix}-8 & -7\\ -6 & -5\end{bmatrix}$. Neka su X i Y rješenja jednadžbi $A\cdot X=B$ i $Y\cdot B=A$. Odredite matricu $D=(X\cdot Y)^T-(X+Y)^T$.

Zadatak 3.

Zadane su matrice $A=\begin{bmatrix}1 & -1\\ 2 & -3\end{bmatrix}$ i $B=\begin{bmatrix}0 & 1\\ 1 & -2\end{bmatrix}$. Neka je X rješenje matrične jednadžbe $(B-A)\cdot(2\cdot A-X)=B+X$. **Bez ispisa matrice X** odredite $x_{1,2}-x_{2,1}$.

Zadatak 4.

Zadane su matrice $A=\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$ i $B=\begin{bmatrix}10 & 9\\ 14 & 4\end{bmatrix}$. Izračunajte **determinantu** matrice $F=3\cdot(A.\backslash B-A./B)\cdot(A.*B)$.

Zadatak 5.

Zadane su matrice $A=\begin{bmatrix}-2 & -1\\ 3 & 4\end{bmatrix}$ i $B=\begin{bmatrix}-4 & -3\\ 2 & 1\end{bmatrix}$. Neka su X i Y rješenja sustava jednadžbi $X.*(A+B)=A-B$ i $X.*Y=A.\hat{}B$. Odredite matricu $G=6\cdot(8\cdot Y-5\cdot X)^T$.

Zadatak 6.

Zadane su matrice $A=\begin{bmatrix}\frac{\pi}{4} & \frac{-3}{4}\cdot\pi\\ \frac{-5}{4}\cdot\pi & \frac{7}{4}\cdot\pi\end{bmatrix}$ i $B=\begin{bmatrix}-\log 2 & 2\cdot\log 3\\ -\log 3 & 2\cdot\log 2\end{bmatrix}$. Neka su $X=\operatorname{tg} A,\ Y=10^B$ i $Z=6\cdot(X-Y)'$.

Izračunajte $z_{2,1}^2-z_{1,2}^2$

Zadatak 7.

Zadana je matrica $A=\begin{bmatrix}1 & -2\\ 2 & 3\end{bmatrix}$. Neka je Y matrica takva da svi njezini elementi pripadaju segmentu $[0,3]$ i da vrijedi jednakost $\cos\left(\frac{\pi}{3}\cdot Y\right)+2\cdot E-A=0,$ pri čemu su E i 0 redom jedinična matrica, odnosno nulmatrica. **Bez ispisa matrice Y** izračunajte zbroj svih njezinih elemenata.

Zadatak 8.

Zadana je matrica $A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$. Neka je X matrica takva da vrijedi jednakost $6\cdot\ln(X+E)+A.\wedge 2-A^2=0$, pri čemu su E i 0 redom jedinična matrica, odnosno nulmatrica. **Bez ispisivanja matrice *X*** odredite cijeli broj najbliži umnošku najmanjega i najvećega elementa te matrice.

Zadatak 9.

Zadane su matrice $A=\begin{bmatrix}-1 & 1\\ 0 & 1\end{bmatrix}$ i $B=\begin{bmatrix}1 & -\sqrt{3}\\ 1 & -1\end{bmatrix}$. Neka su $X=\arcsin A$, $Y=\operatorname{arcctg} B$ i $Z=\left(X^2+Y^2\right)'$. Svaki element matrice X zaokružimo na prvi veći cijeli broj, pa dobijemo matricu W_1. Svaki element matrice Y zaokružimo na prvi manji cijeli broj, pa dobijemo matricu W_2. Svaki element matrice Z zaokružimo na najbliži cijeli broj, pa dobijemo matricu W_3. Neka je $W=\sum_{i=1}^{3}W_i$. Odredite aritmetičku sredinu svih elemenata matrice W.

Rezultati zadataka

1. $C=\begin{bmatrix}-3 & -1\\ 1 & 3\end{bmatrix}$.
2. $D=\begin{bmatrix}3 & -2\\ 4 & 1\end{bmatrix}$.
3. -12.
4. 116 298.
5. $G=\begin{bmatrix}1 & 2154\\ 111 & 302\end{bmatrix}$.
6. 2240.
7. 6.
8. 5.
9. 2.

Pojašnjenja programskih kodova

U ovoj ćemo vježbi upoznati osnovne načine zadavanja matrica u MATLAB-u, te implementiranje osnovnih algebarskih operacija s matricama. Podsjetimo, naziv MATLAB je kratica naziva MAT*rix* LAB*oratory*, pa su matrice zapravo osnovni objekti s kojima radimo u MATLAB-u. U to ćemo se uvjeriti i kad vidimo kako se ugrađene MATLAB-ove funkcije primjenjuju na strukture poput matrica.

Iz *Matematike* 1 znamo da je matrica zapravo pravokutna tablica čiji su elementi najčešće realni ili kompleksni brojevi. Ona je jednoznačno određena zadavanjem svojega tipa i navođenjem svih svojih elemenata. Pritom ti elementi mogu biti definirani i nekim pravilom (formulom).

Znamo da svaki element matrice ima svoju jedinstvenu *poziciju* unutar matrice. Ona se sastoji od dva podatka: broja retka matrice u kojemu se nalazi taj element i broja stupca matrice u kojemu se nalazi taj element. Pritom **ne smijemo** zamijeniti poredak tih brojeva – najprije se mora navesti broj retka, a potom broj stupca.

Istaknimo da se matrica u MATLAB-u zadaje *retčano*, tj. najprije se unose svi elementi njezina prvoga retka (u poretku kojim su navedeni u matrici), potom svi elementi njezina drugoga retka itd. Između dvaju elemenata u *istom* retku stavljamo prazninu. Simbolom ; (točka-zarez) odvajamo posljednji element jednoga retka od prvoga elementa neposredno sljedećega retka. Simbolima [i] redom označavamo početak, odnosno kraj unosa elemenata matrice.

z1.m

Osnovna ideja rješavanja ovoga zadatka je jednostavna: u MATLAB-u najprije moramo zadati obje matrice, a potom pomoću njih izračunati traženu matricu. Pogledajmo kako ćemo to učiniti.

U prvom retku koda zadajemo matricu A. Najprije pišemo ime te matrice (A), potom stavljamo znak = i otvaramo uglatu zagradu [. Time dajemo MATLAB-u do znanja da ćemo u varijabli A pohraniti neku matricu. Nakon što smo otvorili uglatu zagradu, unosimo elemente prvoga retka matrice: -1 2. Iza 2 stavljamo znak ; jer smo upisali sve elemente prvoga retka matrice. Potom upisujemo elemente drugoga retka matrice: 3 –4. Time smo završili unos svih elemenata matrice, pa zatvaramo uglatu zagradu] i unosimo znak ; jer ne želimo da MATLAB ispiše matricu A u komandnom prozoru.

- *Pitanje za razmišljanje*: Izvršite ovaj redak koda, pa promotrite što se dogodilo sa sadržajem radnoga prostora. Precizno objasnite nastalu promjenu.

Potpuno analogan postupak ponavljamo i u drugom retku koda zadajući matricu B.

U posljednjem retku koda određujemo traženu matricu C. Prema zahtjevu zadatka, matricu A najprije trebamo transponirati, pa dobiveni rezultat pomnožiti s $\left(-\frac{3}{2}\right)$. Transponiranje matrica u MATLAB-u može se napraviti na dva načina: koristeći ugrađenu funkciju `transpose` čiji je jedini argument matrica koju treba transponirati ili koristeći znak ' (jednostruki navodnik). Drugi je način bitno kraći i jednostavniji, pa je primijenjen u ovom kodu. Od tako dobivene matrice treba oduzeti transponat matrice $\frac{1}{2}\cdot \boldsymbol{B}$. Budući da jednostruki navodnik zamjenjuje ugrađenu funkciju ', matricu $\frac{1}{2}\cdot \boldsymbol{B}$ najprije stavljamo u okruglu zagradu (jer bismo to napravili i ako bismo koristili funkciju `transpose`), pa potom transponiramo tu matricu navodeći znak '.

- *Pitanja za razmišljanje*: Preradite posljednji redak koda tako da izbrišete sve okrugle zagrade napisane u njemu (a ostale znakove ostavite nepromijenjene). Usporedite dobiveni rezultat s prvotnim rješenjem zadatka. Što zaključujete? Jesmo li mogli predvidjeti rezultat koristeći osnovna svojstva matričnih operacija poznata iz *Matematike* 1? Precizno obrazložite svoje odgovore.

z2.m

U ovom zadatku upoznat ćemo dvije nove matrične operacije: *matrično lijevo dijeljenje* i *matrično desno dijeljenje*. Osnova obiju operacija je *rješavanje* (konzistentnih) *sustava linearnih jednadžbi*. Pritom izraz „konzistentan“ znači da zadani sustav ima barem jedno rješenje. Zbog toga nećemo promatrati nekonzistentne (nemoguće) sustave, pa u daljnjem tekstu pretpostavljamo da se radi o konzistentnim sustavima.

Podsjetimo da je svakom sustavu linearnih jednadžbi *jednoznačno* možemo pridružiti matricu slobodnih članova, matricu nepoznanica i matricu slobodnih koeficijenata. Preciznije, ako je

$$
\begin{aligned}
&a_{11} \cdot x_1 + a_{12} \cdot x_2 + ... + a_{1n} \cdot x_n = b_1 \\
&\vdots \\
&a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + ... + a_{mn} \cdot x_n = b_m
\end{aligned}
$$

zadani sustav m linearnih jednadžbi s n nepoznanica, onda taj sustav možemo zapisati u matričnom obliku $A \cdot X = B$, gdje su

$$A = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \; X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \text{ i } B = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

U numeričkoj matematici (4. semestar) detaljno se obrađuju metode rješavanja takvoga sustava, pa ih ovdje nećemo navoditi i opisivati. Rješenje ovoga sustava određuje se *matričnim lijevim dijeljenjem* (oznaka: \):

$$X = A \backslash B.$$

Promotrimo vrlo jednostavan primjer. Treba riješiti sustav jedne linearne jednadžbe s dvije nepoznanice $x_1 + x_2 = 1$. On očito ima beskonačno mnogo međusobno različitih rješenja. Zapišemo li ga u matričnom obliku, dobijemo:

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix}.$$

Dakle, $A = \begin{bmatrix} 1 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 \end{bmatrix}$. Utipkamo li u novi redak MATLAB-ova komandnoga prozora

```
A=[1 1]; B=[1]; X=A\B
```

i pritisnemo Enter, MATLAB će ispisati:

```
X =
      1
      0
```

Kako vidimo, MATLAB nam nije rekao ništa o tome da sustav ima beskonačno mnogo rješenja! Štoviše, iz ispisanoga rezultata možemo jedino, ali ipak potpuno pogrešno zaključiti da zadana jednadžba ima jedinstveno rješenje.

Promotrimo još jedan primjer. Treba riješiti sustav dviju linearnih jednadžbi s dvije nepoznanice

$$x_1 + x_2 = 1,$$
$$2 \cdot x_1 + 2 \cdot x_2 = 2.$$

Lako vidimo da se druga jednadžba dobije množenjem prve s 2 (ili, ekvivalentno, da je prva jednadžba dobije dijeljenjem druge s 2), što znači da drugu jednadžbu možemo zanemariti. Tako ponovno dobijemo jednu linearnu jednadžbu s dvije nepoznanice, pa i ovaj sustav ima beskonačno mnogo međusobno različitih rješenja.

Zapišemo li taj sustav u matričnom obliku, dobit ćemo:

$$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Dakle, $A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$. Utipkamo li u novi redak MATLAB-ova komandnoga prozora

```
A=[1 1;2 2]; B=[1;2]; X=A\B
```

slijedi novi šok: MATLAB će ispisati

```
Warning: Matrix is singular to working precision.

X =

    NaN
    NaN
```

Prva rečenica je upozorenje: Matrica (ali ne piše koja) je singularna do na preciznost računala. Što sad to znači? Jedina matrica koja može biti singularna je matrica A jer su pojmovi regularnosti i singularnosti matrica definirani samo za kvadratne matrice. Ova rečenica zapravo znači: uzimajući u obzir preciznost računala pri numeričkim izračunima, MATLAB zaključuje da je matrica A singularna. Time MATLAB ne isključuje mogućnost da je A regularna matrica (tj. da je njezina determinanta različita od nule), ali napominje da je njezina determinanta možda tako blizu nuli da je – opet zbog preciznosti računala – nije moguće razlikovati od nule, pa je MATLAB zbog toga izjednačava s nulom. Tipičan primjer takvoga slučaja je npr. kvadratna matrica

$$C = \begin{bmatrix} 10^{-2021} & -10^{-2021} \\ 10^{-2021} & 10^{-2021} \end{bmatrix}.$$

Njezina determinanta je jednaka $2 \cdot 10^{-4042}$ (dokažite!), što znači da je C regularna matrica. Međutim, zbog nemogućnosti preciznoga izračuna s „malim“ strogo pozitivnim brojevima, MATLAB (i mnogi drugi računalni programi koji imaju implementiranu mogućnost određivanja determinanti) će utvrditi da je determinanta matrice C jednaka nuli i pogrešno „proglasiti“ C singularnom matricom uz gornje upozorenje.

- *Pitanje za razmišljanje*: Kako biste se uvjerili da MATLAB doista smatra da je determinanta matrice C jednaka 0?

Ispis

```
X =

   NaN
   NaN
```

zapravo znači da nijedna komponenta matrice nepoznanica X nije kompleksan broj (podsjetnik: NaN je kratica za „Not a Number“). To bismo efektivno protumačili da zadani sustav nema rješenja, što očito nije točno.

Sad smo već u ozbiljnom iskušenju napisati prosvjednu notu autorima MATLAB-a sa zahtjevom za ispriku zbog navođenja na potpuno pogrešne zaključke. Ipak, pričekajmo malo s tom aktivnošću. Operacija matričnoga lijevoga dijeljenja pripada u tzv. *operacije sa simboličkim objektima*. O njima će nešto više riječi biti u drugom dijelu ovoga predmeta. Osnovna ideja je da se, umjesto s „konkretnim“ brojevima, radi sa simboličkim objektima, što se u ovom slučaju pokazuje kao dobitna kombinacija. Naime, utipkamo li u novi redak MATLAB-ova komandnoga prozora

```
A=sym([1 1]); B=sym([1]); X=A\B
```

i potom pritisnemo *Enter*, MATLAB će biti ponešto razgovorljiviji negoli maloprije i, između ostaloga, ispisati:

```
Warning: Solution is not unique because the system is rank-
deficient.
```

```
X =

1
0
```

Ovdje samo kratko napomenimo da MATLAB-ova ugrađena funkcija `sym` „pretvara" svoj argument u simbolički objekt. Bitno zanimljiviji je ispisani rezultat. MATLAB je sad ispisao zaključak kojega smo dobili analitičkim rješavanjem sustava, tj. da rješenje sustava nije jedinstveno jer matrica sustava nema puni rang i da je jedno moguće rješenje sustava $X = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Potpuno analogno, utipkamo li u novi redak MATLAB-ova komandnoga prozora

```
A=sym([1 1;2 2]); B=sym([1;2]); X=A\B
```

i pritisnemo Enter, dobit ćemo (doslovno) isti (ispravan) rezultat kao i maloprije. Što iz svega ovoga možemo zaključiti? Pri rješavanju različitih sustava linearnih jednadžbi treba biti oprezan, a naročito u slučajevima kad broj jednadžbi nije jednak broju nepoznanica. U takvim je slučajevima bolje raditi sa simboličkim objektima na gore opisani način. U „tehničke" pojedinosti ovdje nećemo ulaziti.

Praktično najčešći slučaj ipak predstavljaju tzv. *Cramerovi sustavi*, tj. sustavi n linearnih jednadžbi s n nepoznanica koji imaju jedinstveno rješenje. Ako su svi elementi matrice sustava i matrice slobodnih članova „dovoljno dobri", tj. takvi da determinanta matrice sustava i sve „pomoćne" determinante nisu jednake nuli do na preciznost računala, onda nije nužno raditi sa simboličkim objektima. Upravo takav slučaj imamo u ovom zadatku.

U prva dva retka programskoga koda redom zadajemo matrice A i B. Potom u trećem retku koda računamo matricu X kao „lijevi količnik" matrica A i B. Ta će matrica biti rješenje sustava $A \cdot X = B$.

Priča potpuno analogna dosadašnjoj vrijedi i za tzv. *desno dijeljenje* matrica. U tom se slučaju promatra matrična jednadžba $X \cdot A = B$. Iz *Matematike* 1 znamo da množenje matrica općenito *nije komutativno*, tj. da općenito vrijedi $A \cdot X \neq X \cdot A$. Zbog toga jednadžba $X \cdot A = B$ općenito *nije ekvivalentna* jednadžbi $A \cdot X = B$. Rješenje te matrične jednadžbe je „*desni količnik*" matrica B i A (oznaka: /), tj.

$$X = B / A.$$

Sve analize koje smo ranije proveli za matrično lijevo dijeljenje vrijede i za desno dijeljenje, pa ih ovdje nećemo navoditi. U četvrtom retku koda određena je matrica Y kao „desni količnik" matrica A i B.

Oprez: Rješenje matrične jednadžbe $X \cdot A = B$ je $X = B / A$, a ne $X = A / B$. $X = A / B$ je rješenje matrične jednadžbe $X \cdot B = A$.

U posebnom slučaju, tj. kad se radi o Cramerovim sustavima kod kojih je matrica sustava regularna, *interpretaciju* matričnoga lijevoga i desnoga dijeljenja možemo još više pojednostavniti. Naime, iz *Matematike* 1 znamo da, ako je A regularna matrica i ako su sva množenja dobro definirana, onda vrijede ekvivalencije:

$$(A \cdot X = B) \Leftrightarrow (X = A^{-1} \cdot B),$$
$$(Y \cdot A = B) \Leftrightarrow (Y = B \cdot A^{-1}).$$

Dakle, *u tim posebnim slučajevima* matrično lijevo i desno dijeljenje možemo *interpretirati* i ovako:

$$(X = A \setminus B) \Leftrightarrow (X = A^{-1} \cdot B),$$
$$(Y = A / B) \Leftrightarrow (Y = A \cdot B^{-1}).$$

Oprez: Ponovno naglasimo da u gornjim ekvivalencijama primjenjujemo pretpostavku da su pripadni sustavi Cramerovi. Međutim, oba matrična dijeljenja (lijevo i desno) su definirana i u slučajevima kad matrice sustava nisu kvadratne matrice, pa su u takvim slučajevima gornje ekvivalencije besmislene (jer ne postoji inverz matrice sustava).

Naposljetku, u petom retku koda određujemo matricu D korištenjem netom određenih matrica X i Y. Time je zadatak u potpunosti riješen.

- *Pitanja za razmišljanje*: Neka su $a, b \neq 0$ (kompleksni) brojevi. Znamo da u tom slučaju vrijede ekvivalencije $(a \cdot x = b) \Leftrightarrow (x \cdot a = b) \Leftrightarrow \left(x = \frac{b}{a}\right) \Leftrightarrow (x = b \cdot a^{-1})$. Izdvojimo ekvivalenciju $(x \cdot a = b) \Leftrightarrow (x = b \cdot a^{-1})$. Pretpostavimo li da su a, x i b *matrice* i da je izdvojena ekvivalencija istinita, kojemu je matričnom dijeljenju i uz koje uvjete „slična“ operacija standardnoga dijeljenja kompleksnih brojeva? Koji od tih uvjeta je svojevrstan analogon uvjeta $a, b \neq 0$?

z3.m

U ovom ćemo zadatku iskoristiti jedno od netom definiranih matričnih dijeljenja za rješavanje malo složenijih matričnih jednadžbi. Kao što smo naučili u *Matematici* 1, pri rješavanju takvih jednadžbi moramo biti vrlo oprezni jer operacija množenja matrica nije komutativna, pa prilikom množenja ne smijemo mijenjati poredak faktora.

Osnovna ideja rješavanja ovoga zadatka je vrlo jednostavna: iz zadane jednadžbe trebamo izraziti matricu X koristeći osnovna svojstva algebarskih operacija s matricama. Zbog toga najprije imamo:

$$(B-A)\cdot 2\cdot A-(B-A)\cdot X=B+X,$$
$$-(B-A)\cdot X-X=B-2\cdot(B-A)\cdot A,$$
$$(A-B-E)\cdot X=B-2\cdot(B-A)\cdot A,$$

a odatle primjenom matričnoga lijevoga dijeljenja slijedi

$$X=(A-B-E)\backslash(B-2\cdot(B-A)\cdot A).$$

U gornjem se izrazu pojavila i jedinična matrica reda 2 označena slovom E. Njezine elemente, naravno, možemo unijeti i „ručno“, ali brže i jednostavnije (posebno kod jediničnih matrica nešto većih redova) je primijeniti ugrađenu funkciju `eye`. Njezin naziv najvjerojatnije potječe od engleskoga izgovora uobičajene oznake za jediničnu matricu I (čitati: [aj:]). Ta ugrađena funkcija ima točno jedan argument: red jedinične matrice koju želimo generirati. Taj broj je nužno prirodan broj, pa će – u slučaju da taj zahtjev nije ispunjen – MATLAB ispisati poruku o pogrešci. Izlazna vrijednost je jedinična matrica čiji je red jednak argumentu funkcije `eye`.

Sâm programski kod je vrlo jednostavan. U prvomu retku zadajemo matricu A. U drugomu retku zadajemo matricu B. U njegovu trećemu retku generiramo jediničnu matricu reda 2 koristeći funkciju `eye`. U četvrtom retku određujemo matricu X prema gornjem izrazu. U petom retku računamo traženu vrijednost izraza.

Primijetite na koji način „pozivamo“ elemente matrice u programskom kodu. Zadavanje „na papiru“ je jasno: $x_{1,2}$ je element na presjeku prvoga retka i drugoga stupca. U MATLAB-u taj element pozivamo navodeći `X(1,2)`. Dakle, najprije treba navesti ime matrice, potom otvoriti okruglu zagradu, pa navesti broj retka, pa navesti broj stupca i naposljetku zatvoriti okruglu zagradu. *Redoslijed navođenja podataka se pritom ne smije mijenjati.* Broj retka i broj stupca treba odvojiti zarezom. Ako nenamjerno pogrešno pokušamo pozvati element na poziciji koja u matrici ne postoji (npr. x_{34} u gornjem slučaju), MATLAB će ispisati poruku o pogrešci.

- *Pitanje za razmišljanje*: Preradite dobiveni programski kod tako da umjesto matričnoga lijevoga dijeljenja koristite invertiranje matrice.

z4.m

U prethodnim smo zadacima upoznali osnovne matrične operacije. Osnovno svojstvo tih operacija je da su njihovi argumenti matrice. Sada ćemo upoznati matrične operacije koje djeluju „po točkama“, odnosno zasebno na svaki pojedini element matrice (a ne na matricu u cijelosti).

U *Matematici* 1 je vjerojatno „najgora“ operacija u radu s matricama bila množenje matrica. Najprije smo morali provjeriti jesu li matrice ulančane (jer inače množenje nije definirano), pa, ako jest, onda zasebno računati svaki element umnoška kao skalarni umnožak pripadajućega retka prvoga faktora i pripadajućega stupca drugoga faktora. Taj postupak nije bio pretjerano tehnički težak, ali je bio poprilično spor u slučaju relativno velikoga broja skalarnih umnožaka koje je trebalo računati. Zbrajanje, oduzimanje i množenje matrice skalarom bili su vrlo jednostavni: zbrajali/oduzimali smo elemente na istim pozicijama unutar matrice, dok smo pri množenju matrice skalarom jednostavno svaki element matrice pomnožili tim skalarom.

Ove tri operacije zapravo su operacije „po točkama“, odnosno prema pozicijama. Da bi prve dvije bile dobro definirane, matrice moraju biti *istoga tipa*. Ta pretpostavka će ujedno biti i pretpostavka za definiranje množenja i dijeljenja „po točkama“.

Ukratko i u bitnom, ako su matrice $A=\left[a_{ij}\right]$ i $B=\left[b_{ij}\right]$ tipa $(r,\ s)$, onda je njihov *umnožak „po točkama“* matrica C tipa $(r,\ s)$ čiji su elementi definirani pravilom:

$$c_{ij}=a_{ij}\cdot b_{ij},\ \forall i=1,2,\ldots,r,\ \forall j=1,2,\ldots s.$$

U tom slučaju pišemo: $C=A.*B.$ Budući da je množenje kompleksnih brojeva komutativno, vrijedi jednakost:

$$A.*B=B.*A.$$

Dakle, evo napokon jednoga matričnoga množenja kod kojega poredak faktora nije bitan! Znak .* ima svoju prirodnu motivaciju: * označava množenje, a . označava da se radi o operaciji „po točkama“. Analogni zapisi .+ i .– su formalno smisleni, ali zapravo potpuno nepotrebni: obje operacije se *prema definicijama* ionako izvode po točkama, pa nije potrebno naglašavati da se radi o „točkovnim“ operacijama.

U zadatku 2. upoznali smo matrično lijevo, odnosno desno dijeljenje. I ove operacije imaju svoj analogon „po točkama“. Ako su matrice $A=\left[a_{ij}\right]$ i $B=\left[b_{ij}\right]$ tipa $(r,\ s)$, onda je njihov *lijevi količnik „po točkama“* matrica C tipa $(r,\ s)$ čiji su elementi definirani pravilom:

$$c_{ij}=\frac{b_{ij}}{a_{ij}},\ \forall i=1,2,\ldots,r,\ \forall j=1,2,\ldots s.$$

U tom slučaju pišemo: $C=A.\backslash B.$ Jasno je da se može dogoditi da je neki od elemenata matrice A jednak nuli i da u tom slučaju pripadni element matrice B treba dijeliti s nulom, što je nemoguće. MATLAB će tada na pripadnoj poziciji kao rezultat „dijeljenja“ ispisati `Inf`.

Naposljetku, ako su matrice $A=\left[a_{ij}\right]$ i $B=\left[b_{ij}\right]$ tipa $(r,\ s)$, onda je njihov *desni količnik „po točkama“* matrica C tipa $(r,\ s)$ čiji su elementi definirani pravilom:

$$c_{ij}=\frac{a_{ij}}{b_{ij}},\ \forall i=1,2,\ldots,r,\ \forall j=1,2,\ldots s.$$

U tom slučaju pišemo: $C=A./B.$ I ovdje vrijedi napomena potpuno analogna gornjoj: ako je neki element matrice B jednak nuli, onda će MATLAB na pripadnoj poziciji kao rezultat dijeljenja ispisati `Inf`.

Rekli smo da matrično množenje općenito nije komutativna operacija, pa ne možemo uspostaviti nikakvu vezu između matričnoga lijevoga i desnoga dijeljenja. Za operacije „po točkama“ ta tvrdnja ne vrijedi. Naime, uz pretpostavku da su oba količnika dobro definirana, iz gornjih definicija matričnoga lijevoga, odnosno desnoga dijeljenja „po točkama“ izravno slijedi:

$$A.\backslash B=B./A.$$

Zbog toga se ponekad matrično lijevo dijeljenje „po točkama“ definira koristeći matrično desno dijeljenje „po točkama“. Ovdje to nismo učinili jer smatramo da su gornje definicijske formule dovoljno prihvatljive i razumljive.

Oprez: Potpuno analogno kao i „na papiru“, prilikom zadavanja operacija „po točkama“ ne smijemo zaboraviti navesti . u okviru svake od tih operacija. Ako su matrice A i B istoga reda (pa k tome još i regularne), onda su izrazi $A\backslash B$, $A.\backslash B$, A/B, $A./B$, $A\cdot B$ i $A.*B$ dobro definirani, ali općenito daju potpuno različite rezultate.

Zaključno, preostaje navesti MATLAB-ovu ugrađenu funkciju koja određuje determinantu kvadratne matrice. To je funkcija `det`. Njezin jedini argument je matrica čiju determinantu treba odrediti. Tu matricu pritom možemo ili upisati kao argument funkcije `det` bez navođenja imena matrice ili zadati u zasebnom retku, pa onda kao argument funkcije `det` navesti ime matrice. Prvi način je praktičan ako je riječ o matrici s kojom – nakon određivanja njezine determinante – više ništa nećemo raditi. U svim ostalim slučajevima praktičniji je drugi način, pa se upravo on i najčešće koristi.

Preostaje komentirati programski kod. U prvom retku zadajemo matricu A. U drugom retku zadajemo matricu B. U trećem retku određujemo matricu F koristeći netom definirane operacije „po točkama“. Uočite da znak * ovdje ima dva značenja: onaj iza broja 3 označava množenje sa skalarom, a onaj između dvije okrugle zagrade označava standardno matrično množenje (dakle, *ne* množenje „po točkama“). Ime matrice namjerno smo označili slovom F jer je slovo E „rezervirano“ za označavanje jedinične matrice. Naposljetku, u petom retku određujemo traženu determinantu matrice F koristeći funkciju `det`.

- *Pitanja za razmišljanje*: Preradite dobiveni programski kod tako da 3. i 4. redak zamijenite jednim retkom. Možete li traženu determinantu alternativno izračunati kao umnožak determinanti faktora? Ako možete, preradite kod tako da funkcija `det` bude primijenjena na svaki faktor posebno. Ako ne možete, precizno obrazložite svoj odgovor.

z5.m

Ovaj zadatak ne donosi ništa posebno novoga, osim što najprije moramo definirati novu operaciju *matričnoga potenciranja „po točkama“*. Ponovno, ako su matrice $A=\left[a_{ij}\right]$ i $B=\left[b_{ij}\right]$ tipa $(r,\ s)$, onda je njihova *potencija „po točkama“* matrica C tipa $(r,\ s)$ čiji su elementi definirani pravilom:

$$c_{ij}=a_{ij}^{b_{ij}},\ \forall i=1,2,...,r,\ \forall j=1,2,...s.$$

U tom slučaju pišemo: $C=A.\wedge B.$

Zadatak je zapravo bitno jednostavniji negoli se čini na prvi pogled. Promotrimo prvu jednadžbu (s nepoznanicom X). Pretpostavimo da smo izračunali matrice $A+B$ i $A-B$. (To smijemo učiniti jer su nam matrice A i B zadane.) Što radimo na lijevoj strani jednadžbe? *Svaki* element nepoznate matrice X *množimo* s odgovarajućim elementom (na istoj poziciji) matrice $A+B$. Što treba biti rezultat toga množenja? Element na *istoj poziciji* u matrici $A-B$. Kako onda dobijemo nepoznati element matrice X? Jednostavno, dobijemo ga tako da element na uočenoj poziciji u matrici $A-B$ podijelimo s elementom na istoj poziciji u matrici $A+B$. Ovaj postupak provodimo za svaki element matrica $A-B$, odnosno $A+B$. Kako to naznačiti najjednostavnije moguće? Odgovor je, naravno, matrično desno dijeljenje po točkama:

$$X=(A-B)./(A+B).$$

- *Pitanje za razmišljanje*: Zapišite gornju jednakost koristeći matrično *lijevo* dijeljenje po točkama.

Na opisani smo način odredili matricu X, pa iz druge jednadžbe odredimo matricu Y. Kako to učiniti? Ponovno zasebno promotrimo svaku stranu te jednadžbe. Na lijevoj strani imamo matrično množenje po točkama. Što to zapravo znači? To znači da svaki element matrice X množimo s elementom matrice Y koji se nalazi na istoj poziciji. Što trebamo dobiti kao rezultat? Element koji se nalazi *na istoj poziciji* u matrici $A.\wedge B$. Tu matricu možemo izračunati jer znamo matrice A i B. Kako onda odrediti nepoznati element matrice Y? Jednostavno: svaki element matrice $A.\wedge B$ podijelimo s elementom matrice X koji se nalazi na istoj poziciji. Kako to možemo najjednostavnije naznačiti? Ponovno možemo koristiti matrično desno dijeljenje po točkama:

$$Y=(A\wedge .B)./X.$$

- *Pitanja za razmišljanje*: Zapišite i ovu jednakost koristeći matrično *lijevo* dijeljenje po točkama. Utvrdite možemo li izostaviti okruglu zagradu napisanu u ovom izrazu. Ako možemo, precizno objasnite zašto to možemo učiniti. Koja od dviju operacija ima veći rang: potenciranje matrica po točkama ili matrično lijevo/desno dijeljenje po točkama? Usporedite dobiveni zaključak s rangom analognih operacija definiranih na skupu realnih brojeva.

Preostaje komentirati programski kod. Njegova je struktura vrlo jednostavna. U prvim dvama retcima zadajemo matrice A i B. U trećemu retku određujemo matricu X iz jednakosti $X = (A - B)./(A + B)$. U četvrtomu retku određujemo matricu Y iz jednakosti $Y = (A\,\hat{}\,.B)./X$. Naposljetku, u petomu retku određujemo traženu matricu G, pri čemu za operaciju transponiranja koristimo ranije uvedenu oznaku '.

- *Pitanja za razmišljanje*: Preradite dobiveni programski kod tako da se umjesto *svakoga* matričnoga desnoga dijeljenja „po točkama“ pojavljuje matrično lijevo dijeljenje „po točkama“. Potom preradite dobiveni kod tako da matrica Y bude izražena samo pomoću matrica A i B, a ne pomoću matrice X.

z6.m

U ovom ćemo zadatku upoznati „djelovanje“ nekih ugrađenih matematičkih funkcija u MATLAB-u na matrice. I njihov osnovni princip djelovanja na matrice je „točkovni“: ugrađena funkcija „djeluje“ na matricu tako da djeluje na svaki element te matrice. Elementi matrice bit će (općenito) realni brojevi, pa zapravo imamo „obične“ realne funkcije jedne realne varijable. Zbog toga će rezultat „djelovanja“ ugrađenih funkcija na matrice biti *matrice* istoga tipa kao i matrica na koju „djeluje“ funkcija. Međutim, odmah napomenimo da postoje brojne ugrađene funkcije u MATLAB-u čijim „djelovanjem“ na matricu nastaje realan broj, matrica čiji je tip različit od tipa zadane matrice itd. Neke od tih funkcija upoznat ćemo u sljedećim zadacima i nastavnim cjelinama.

Ilustrirajmo rečeno na primjeru matrica $A=\begin{bmatrix} \frac{\pi}{4} & \frac{-3}{4}\cdot\pi \\ \frac{-5}{4}\cdot\pi & \frac{7}{4}\cdot\pi \end{bmatrix}$ i $X=\operatorname{tg} A.$ Pogledajmo kako dobijemo elemente matrice X:

$$x_{1,1}=\operatorname{tg}\left(\frac{\pi}{4}\right)=1 \qquad x_{1,2}=\operatorname{tg}\left(\frac{-3}{4}\cdot\pi\right)=1$$
$$x_{2,1}=\operatorname{tg}\left(\frac{-5}{4}\cdot\pi\right)=-1 \qquad x_{2,2}=\operatorname{tg}\left(\frac{7}{4}\cdot\pi\right)=-1$$

Dakle, na *svaki* element matrice A „djelovali“ smo funkcijom tg. Rezultat svakoga takvoga djelovanja je „konkretan“ realan broj. Taj broj pišemo na istoj poziciji na kojoj je bio i polazni element matrice A. Tako dobijemo matricu X. Potpuno analogno dobivamo i matricu Y pomoću matrice B.

Struktura programskoga koda je sada vrlo jednostavna. U prvom retku zadajemo matricu A, pri čemu konstantu **π** pišemo kao ugrađenu konstantu `pi`. U drugom retku zadajemo matricu B, pri čemu dekadski logaritam (log) pišemo koristeći ugrađenu funkciju `log10`. U trećem retku računamo matricu X, a u četvrtom retku matricu Y. Pritom moramo biti pažljivi: pravilo 10^B je ispravno napisano „na papiru“, ali u MATLAB-u ga moramo naznačiti kao operaciju „po točkama“, tj. kao `10.^B`. Ne učinimo li to, MATLAB će javiti pogrešku. U petom retku računamo matricu Z, pri čemu za transponiranje ponovno koristimo znak '. Konačno, u šestom retku računamo traženu razliku kvadrata, pri čemu $z_{2,1}^2$ zapisujemo kao `Z(2,1)^2` (i analogno za $z_{1,2}^2$). Pritom *nije potrebno* pisati `(Z(2,1))^2` jer MATLAB najprije očita vrijednost koja se u matrici Z nalazi na poziciji (2,1), a potom kvadrira tu vrijednost. Kad malo bolje razmislite, jedino takav način izvršenja operacija je i moguć jer je izraz `(2,1)^2` besmislen i pokušaj njegova izvršavanja rezultirao bi „dojavom“ pogreške. (Uvjerite se u to.)

Oprez: (2,1) u ovakvom zapisu *nije jednoretčana matrica*, nego uređeni par prirodnih brojeva, pa bi i izvršenje izraza `(2,1).^2` u MATLAB-u rezultiralo „dojavom“ pogreške.

- *Pitanje za razmišljanje*: Dobije li se isti rezultat ako se uzmu $A = \begin{bmatrix} 45 & -135 \\ -225 & 315 \end{bmatrix}$ i $X = \text{tgd}(A)$? Objasnite svoj odgovor.

- *Pitanja za razmišljanje*: Izrazite matricu *Z* samo pomoću matrica *A* i *B*, tj. bez korištenja matrica *X* i *Y*. Potom preradite dobiveni programski kod tako da izbrišete njegov 3. i 4. redak, a u 5. retku implementirate dobiveni izraz.

z7.m

U ovom zadatku upoznat ćemo ugrađenu funkciju `sum` pomoću koje ćemo odrediti zbroj svih elemenata matrice. Ta je funkcija primjer funkcije čiji je argument matrica, a rezultat kompleksan broj (a ne matrica, kao što je to bio slučaj u prethodnom zadatku).

Osnovna ideja rješavanja zadatka je: izraziti matricu Y pomoću matrica A i E (nulmatrica 0 nam neće biti potrebna), pa potom koristeći spomenutu funkciju `sum` odrediti traženi zbroj. Pritom se trebamo prisjetiti jednakosti:

$$\arccos(\cos x) = x, \ \forall x \in [0, \pi].$$

Oprez: Ta jednakost općenito *nije istinita* jer je npr. $\arccos(\cos(2\cdot\pi)) = 0 \neq 2\cdot\pi$. Upravo zbog toga je u zadatku navedena pretpostavka da svi elementi matrice Y pripadaju segmentu $[0,3]$. Naime, iz gornje jednakosti lako slijedi:

$$\arccos\left(\cos\left(\frac{\pi}{3}\cdot x\right)\right) = \frac{\pi}{3}\cdot x, \ \forall x \in [0,3].$$

Slijedom navedenoga, izrazimo najprije matricu Y pomoću matrica A i E:

$$\cos\left(\frac{\pi}{3}\cdot Y\right) = A - 2\cdot E, \qquad /\arccos$$

$$\frac{\pi}{3}\cdot Y = \arccos(A - 2\cdot E), \qquad /\cdot\frac{3}{\pi}$$

$$Y = \frac{3}{\pi}\cdot\arccos(A - 2\cdot E).$$

Struktura programskoga koda je sada jasna. U prvom retku zadajemo matricu A. U drugom retku zadajemo jediničnu matricu E reda 2 koristeći ugrađenu funkciju `eye`. Matrica A je reda 2, pa i argument funkcije `eye` mora biti jednak 2. U trećem retku određujemo matricu Y koristeći ugrađenu funkciju `acos`. Na kraju toga retka moramo staviti znak ; jer zadatak zahtijeva da matrica Y ne bude ispisana.

Preostaje navesti kako odrediti traženi zbroj pomoću funkcije `sum`. Ta će funkcija u ovom slučaju imati točno dva argumenta. Prvi argument bit će matrica čiji zbroj svih elemenata želimo izračunati. Drugi argument bit će rezervirana riječ `all` (engl.: svi) *obavezno* navedena pod znakovima jednostrukih navodnika. (Izostavimo li jednostruke navodnike, MATLAB će javiti pogrešku.) Tako ćemo MATLAB-u dati do znanja da treba izračunati zbroj svih elemenata matrice Y. Traženi zbroj pohranjujemo u varijablu `rezultat` i ispisujemo pohranjenu vrijednost.

- *Pitanja za razmišljanje*: Utvrdite što bi se dogodilo ako bismo u posljednjemu retku izostavili rezerviranu riječ `all`. Precizno objasnite sve svoje tvrdnje.

z8.m

U ovome ćemo zadatku vidjeti kako pomoću ugrađenih funkcija `max` i `min` možemo odrediti najveći, odnosno najmanji element neke matrice. Budući da kompleksne brojeve općenito ne možemo uspoređivati po veličini, tj. poredati ih od najmanjega do najvećega (ili obratno), pretpostavljamo da su svi elementi matrice realni brojevi.

Početak postupka rješavanja je potpuno analogan početku postupka rješavanja prethodnoga zadatka. Iz zadane jednakosti izrazimo matricu X, pri čemu koristimo identitet

$$e^{\ln x} = x, \ \forall x > 0.$$

Imamo redom:

$$\begin{aligned}
&6 \cdot \ln(X+E) = A^2 - A.\hat{}\,2, \quad /:6\\
&\ln(X+E) = \frac{1}{6}\cdot\left(A^2 - A.\hat{}\,2\right), \quad /\,e\\
&X+E = e^{\frac{1}{6}\cdot\left(A^2 - A.\hat{}\,2\right)},\\
&X = e^{\frac{1}{6}\cdot\left(A^2 - A.\hat{}\,2\right)} - E.
\end{aligned}$$

Napomena: Ne moramo provjeravati je li svaki element matrice $X+E$ strogo pozitivan realan broj, odnosno je li izraz $\ln(X+E)$ dobro definiran. Naime, logaritamska jednadžba $\ln x = a$ ima jedinstveno rješenje za svaki $a \in \mathbb{R}$, i to rješenje je nužno strogo pozitivan realan broj e^a. Dakle, *svaki* element *matrice* $e^{\frac{1}{6}\cdot\left(A^2 - A.\hat{}\,2\right)}$ je nužno strogo pozitivan realan broj, a kako ta matrica mora biti jednaka $X+E$, spomenuta provjera nije potrebna.

Zbog toga u prvom retku koda zadajemo matricu A, dok u drugom retku koda zadajemo jediničnu matricu E. Budući da je A matrica reda 2, i E mora biti matrica reda 2.

U trećem retku određujemo matricu X koristeći gornji izraz. Potenciranje matrice bazom e provodimo koristeći ugrađenu funkciju `exp`. Kvadrat matrice A, tj. matrični umnožak $A \cdot A$, zadajemo kao `A^2` (on je dobro definiran jer je A kvadratna matrica), dok matricu dobivenu kvadriranjem svakoga elementa matrice A zadajemo kao `A.^2`.

Uočite sličnost u zadavanju ovih operacija (razlika je samo u točki .) i razliku u njihovim rezultatima.

U četvrtom retku određujemo traženi cijeli broj. Najveći element matrice X određujemo koristeći ugrađenu funkciju `max`. Ona će u ovom slučaju imati točno tri argumenta. Prvi argument bit će ime matrice čiji najveći element tražimo. Drugi argument bit će „prazan“ par uglatih zagrada []. Treći argument bit će rezervirana riječ `all` stavljena u jednostruke navodnike (analogno kao u rješenju prethodnoga zadatka). Poredak argumenata *ne smijemo* mijenjati – učinimo li to, MATLAB će javiti pogrešku. Budući da se funkcija `max` primjenjuje u još nekolicini drugih situacija, ovdje se nećemo zadržavati na razlogu zbog kojega je kao drugi argument nužno zapisati par uglatih zagrada.

Potpuno analogno razmatranje vrijedi i za određivanje najmanjega elementa matrice X. Njega određujemo koristeći ugrađenu funkciju `min` s doslovno istim argumentima poredanim u istom poretku kao u slučaju funkcije `max`.

Naposljetku, kao što smo vidjeli u prethodnoj vježbi, cijeli broj najbliži realnom broju x određujemo korištenjem ugrađene funkcije `round`. U ovom slučaju argument funkcije `round` bit će umnožak `max(X,[],'all')*min(X,[],'all')`. Ponovno se ne moramo brinuti: MATLAB će najprije izvršiti ugrađene funkcije `max` i `min`, potom odrediti umnožak dobivenih realnih brojeva i naposljetku zaokružiti taj umnožak na najbliži cijeli broj.

- *Pitanja za razmišljanje*: Neka je $A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$. Utvrdite što se dobiva izvršavanjem naredbi `B=max(A)` i `C=min(A)`. Potom odredite `B1=max(B)` i `C1=min(C)`. Što uočavate? Koristeći dobiveni rezultat, objasnite kako alternativno možemo odrediti najveći i najmanji element neke matrice bez korištenja `[]` i `'all'`. Potom preradite dobiveni programski kod prema prethodnom objašnjenju.

z9.m

U ovome je zadatku jedina nova ugrađena funkcija koju ćemo koristiti prilikom rješavanja funkcija koja određuje aritmetičku sredinu svih elemenata neke matrice. Riječ je o funkciji `mean` (engl.: srednja vrijednost). Potpuno analogno kao i ugrađena funkcija `sum` koju smo koristili u rješenju zadatka 7., i funkcija `mean` imat će točno dva argumenta. Prvi je matrica čiju aritmetičku sredinu elemenata želimo odrediti, a drugi rezervirana riječ `all` napisana pod znakovima jednostrukih navodnika.

Komentirajmo programski kod koji predstavlja rješenje zadatka. U prvome retku zadajemo matricu A. U drugom retku zadajemo matricu B, pri čemu koristimo funkciju `realsqrt` za određivanje drugoga korijena.

U trećemu retku određujemo matricu X tako da na svaki element matrice A „djelujemo“ funkcijom arcsin, odnosno – „prevedeno“ na MATLAB-ov jezik – ugrađenom funkcijom `asin`.

U četvrtomu retku određujemo matricu Y tako da na svaki element matrice B „djelujemo“ funkcijom arcctg, odnosno – „prevedeno“ na MATLAB-ov jezik – ugrađenom funkcijom `acot`.

U petomu retku određujemo matricu Z kao transponat zbroja kvadrata matrica X i Y. Pritom je, dakako, $X^2 = X \cdot X$ matrica dobivena standardnim matričnim množenjem matrice X sa samom sobom (i analogno za Y^2). Za zapis transponiranja ponovno koristimo jednostruki navodnik '.

U šestom, sedmom i osmom retku provodimo tražena zaokruživanja. Koristeći ugrađenu funkciju `floor`, najprije svaki element matrice X zaokružujemo na prvi manji cijeli broj. Potom koristeći ugrađenu funkciju `ceil` svaki element matrice Y zaokružujemo na prvi veći cijeli broj. Naposljetku, koristeći ugrađenu funkciju `round`, svaki element matrice Z zaokružujemo na najbliži cijeli broj.

Ostatak koda je standardan. U devetomu retku zbrojimo matrice dobivene u šestom, sedmom i osmom retku, a u posljednjem, desetomu, retku odredimo traženu aritmetičku sredinu matrice dobivene u devetomu retku koda.

- *Pitanje za razmišljanje*: Riješite zadatak tako da matricu W – dobivenu u devetomu retku koda - izrazite pomoću zadanih matrica A i B bez korištenja pomoćnih matrica X, Y, Z, W_1, W_2 i W_3. Je li konačno rješenje zadatka jednako onome dobivenomu izvršavanjem gornjega programskoga koda? Precizno objasnite svoj odgovor.