TVZ TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGRABIENSE Elektrotehnički odjel	**Matematički alati u elektrotehnici** (preddiplomski stručni studij elektrotehnike)	**Vježba 9.** Osnove integralnoga računa u MATLAB-u.

Zadatak 1.

Odredite standardnu antiderivaciju funkcije $f(x) = 15 \cdot \cos^5 x$. Pojednostavnite dobiveni izraz što više možete.

Zadatak 2.

Odredite neodređeni integral $\int 2 \cdot \operatorname{Arth} t \cdot \mathrm{d}t$. Pojednostavnite dobiveni izraz što više možete.

Zadatak 3.

Izračunajte egzaktnu i približnu (s točnošću od 10^{-5}) vrijednost integrala $\int_0^1 \frac{4 \cdot u^2}{e^{2 \cdot u}} \cdot \mathrm{d}u$.

Zadatak 4.

Izračunajte egzaktnu i približnu (s točnošću od 10^{-5}) vrijednost integrala $\int_1^e \left(3 \cdot \sqrt{3} \cdot w \cdot \ln w\right)^2 \cdot \mathrm{d}w$.

Zadatak 5.

Izračunajte površinu ravninskoga lika kojega krivulja $y = A \cdot \sin(\omega \cdot t + \varphi)$ zatvara s osi apscisa na segmentu $\left[-\frac{\varphi}{\omega}, \frac{2 \cdot \pi - \varphi}{\omega}\right]$.

Zadatak 6.

Izračunajte egzaktnu i približnu (s točnošću od 10^{-5}) vrijednost površine lika omeđenoga ravninskom krivuljom $16 \cdot x^2 + 9 \cdot y^2 = 144$.

Zadatak 7.

Izračunajte površinu „omče“ omeđene krivuljom $y^2 = \frac{x^3 + 3 \cdot x^2}{1 - x}$.

Zadatak 8.

Neka je $f : \mathbb{R} \to \mathbb{R}$ funkcija definirana pravilom $f(x) = \begin{cases} x+1, \text{ za } x > 1, \\ 2 \cdot x - 1, \text{ za } x \in [0,1], \\ x^2 - 1, \text{ za } x < 0. \end{cases}$ Izračunajte $\int\limits_{-3}^{5} f(x) \cdot \mathrm{d}x.$

Zadatak 9.

Bez mijenjanja oznake nezavisne varijable izračunajte neprave integrale:

a) $\int\limits_{0}^{+\infty} e^{-x^2} \cdot \mathrm{d}x$;

b) $\int\limits_{-\infty}^{0} \frac{8 \cdot e^t}{\left(1+e^{2 \cdot t}\right)^2} \cdot \mathrm{d}u$;

c) $\int\limits_{-\infty}^{+\infty} \frac{12 \cdot \operatorname{arctg}^2 u}{1+u^2} \cdot \mathrm{d}u$;

d) $\int\limits_{0}^{1} \frac{-\ln\left(w-w^2\right)}{2} \cdot \mathrm{d}w.$

Rezultati zadataka:

1. $3 \cdot \sin^5 x - 10 \cdot \sin^3 x + 15 \cdot \sin x.$

2. $2 \cdot t \cdot \operatorname{Arth} t + \ln(t^2 - 1) + C,\ C \in \mathbb{R}.$

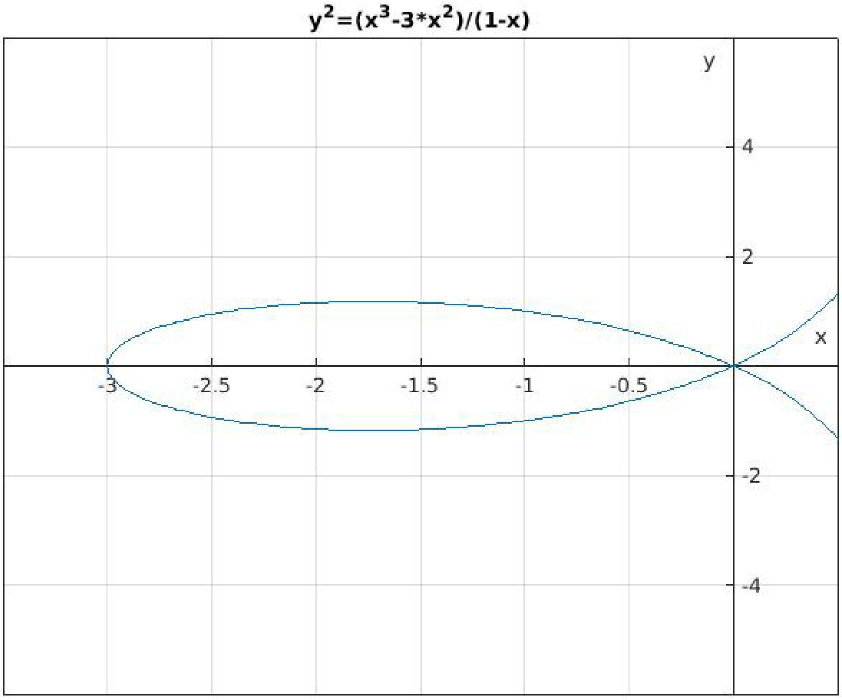
3. $1 - \dfrac{5}{e^2} \approx 0.32332.$

4. $5 \cdot e^3 - 2 \approx 98.42768.$

5. $P = \dfrac{4 \cdot A}{\omega}$ kv. jed.

6. $P = 12 \cdot \pi \approx 37.69911$ kv. jed.

7. Vidjeti sliku 1. $P = 3 \cdot \sqrt{3}$ kv. jed.



Slika 1.

8. 22.

9.

a) $\dfrac{\sqrt{\pi}}{2}$;

b) $\pi + 2$;

c) π^3;

d) 1.

Komentari i objašnjenja programskih kodova

z1.m i z2.m

U ovim zadacima upoznajemo ugrađenu funkciju `int` pomoću koje možemo odrediti standardnu antiderivaciju, određeni integral i nepravi integral. Pritom se određeni i nepravi integral određuju na potpuno jednak način, tj. potpuno jednakom sintaksom (razlika je jedino u vrijednostima ulaznih varijabli). O njima ćemo govoriti malo kasnije. Sad se posvetimo određivanju standardne antiderivacije, odnosno rješavanju zadatka 1.

Pogledajmo strukturu programskoga koda. Prva dva retka izgledaju uobičajeno: najprije deklariramo simboličku varijablu (x), a potom definiramo pravilo funkcije. U trećem retku određujemo standardnu antiderivaciju. Uočimo na koji način to činimo. Funkcija `int` u tom smislu ima sintaksu potpuno analognu onoj koju ima funkcija `diff` koju koristimo za određivanje derivacije funkcije. Njezina prva ulazna varijabla je oznaka funkcije, a druga ulazna varijabla je oznaka varijable po kojoj se integrira. **Zamjena redoslijeda varijabli nije dozvoljena**.

Na taj će način funkcija *f* odrediti standardnu antiderivaciju funkcije *f*. Cilj nam je maksimalno pojednostavniti njezino pravilo. U tu svrhu primijenimo ugrađene funkcije `simplify` i `expand`. Bez primjene potonje ugrađene funkcije dobili bismo izraz rastavljen na faktore, a takav izraz nije maksimalno pojednostavljen izraz koji se traži u zadatku. Ne morate se brinuti: unaprijed ne možemo znati hoće li nam trebati neka od „dodatnih“ funkcija za pojednostavljivanje izraza kao što je `expand`. Upravo zbog takvih slučajeva podesno je pisati rješenja zadataka u obliku programskih kodova. Tako ćemo vrlo lako i jednostavno naknadno moći uvrstiti dodatne potrebne ugrađene funkcije, te iznova pokrenuti dotični kod.

Programski kod prikazan kao rješenje zadatka 1., dakle, korektno određuje standardnu antiderivaciju integrabilne realne funkcije. Međutim, u zadatku 2. tražimo neodređeni integral realne funkcije *f*. Podsjetimo, to je *skup* svih funkcija koje derivirane daju zadanu funkciju *f*. Taj skup ima beskonačno mnogo elemenata. Ti elementi su vrlo slični: imaju zajednički dio koji sadrži nezavisnu varijablu, a razlikuju se po slobodnom članu (tj. članu koji ne sadrži nezavisnu varijablu). U slučaju realne funkcije jedne realne varijable taj slobodni član je realna (**ne**: cjelobrojna!) konstanta.

MATLAB *ne* može odrediti neodređeni integral. Rezultat „najbliži“ neodređenom integralu kojega MATLAB može odrediti jest standardna antiderivacija. Je li to loše?

Nije nimalo. Naime, standardna antiderivacija pripada neodređenom integralu, a njezina realna konstanta jednaka je nuli. Kako onda dobiti cijeli skup, odnosno neodređeni integral? Na papiru bi rješenje bilo jednostavno: uzeli bismo olovku, nadopisali bismo *C*, napomenuli da je $C \in \mathbb{R}$ i dovršili zadatak. Međutim, u MATLAB-u ne možemo uzeti olovku i žvrljati po zaslonu. Znači li to da ni na koji način ne možemo dobiti ispravno rješenje zadatka? Ne, naravno. U pomoć će nam priteći ugrađena funkcija `fprintf` koja će nam omogućiti ispis ispravnoga rješenja.

Pogledajmo strukturu programskoga koda **z2.m**. Prva tri retka su potpuno analogna onima u kodu **z1.m**. To je potpuno logično jer izvršavanjem tih triju redaka određujemo standardnu antiderivaciju polazne funkcije. Ni u ovom zadatku ne trebamo koristiti funkciju `expand`.

U četvrtom retku omogućujemo korektan ispis neodređenoga integrala. Pogledajmo kako koristimo funkciju `fprintf`. Unutar okruglih zagrada smještavamo argument te funkcije. To će biti rečenica *Traženi integral je jednak I* = (standardna antiderivacija *F*) + *C*. Tu rečenicu stavljamo pod znakove apostrofa. Jedino je pitanje kako u nju uvrstiti pravilo standardne antiderivacije određeno u trećem retku koda. To pravilo shvatimo kao niz znakova, tj. varijablu tipa *string*. Zbog toga iza znaka = pišemo `%s`. Što time „zapovijedamo“ MATLAB-u? Kažemo mu neka normalno ispiše sve znakove koji tvore rečenicu do znaka =. (U te znakove uključene su i praznine.) Kad ispiše znak = i prazninu iza njega, umjesto %s treba uvrstiti *prvi sljedeći* (tj. drugi) *argument* funkcije `fprintf`. Nakon što uvrsti taj argument MATLAB treba nastaviti s ispisom znakova pod apostrofima sve dok ne dođe ili do sljedećega izraza oblika %*nešto* ili do desnoga apostrofa. U ovom slučaju nemamo više ništa za uvrstiti u rečenicu, pa se nakon `%s` ne pojavljuje nijedan izraz oblika %*nešto*.

Kad smo zatvorili desni apostrof, navodimo što to treba uvrstiti umjesto `%s` u netom dovršenu rečenicu. To je pravilo standardne antiderivacije kojega smo u trećem retku koda označili s *F*. Zbog toga iza znaka apostrofa stavljamo znak zareza (kojim, kao što znamo, odvajamo nezavisne varijable), upisujemo *F*, zatvaramo zagradu i završavamo zadatak. Pokretanjem programskoga koda dobivamo ispravno rješenje zadatka.

- *Pitanje za razmišljanje*: Što radi opcija `\n` unutar opcije `fprintf`? Zasebno izbrišite svaku od triju opcija `\n` u četvrtom retku koda, pokrenite tako dobiveni kod i uočite promjenu u odnosu na početno rješenje.

- *Pitanje za razmišljanje*: Kako biste ova dva zadatka riješili analitički (bez korištenja MATLAB-a)?

z3.m i z4.m

U ovim dvama zadacima određujemo egzaktnu i približnu vrijednost određenoga integrala. Odmah pojasnimo na što mislimo pod tim pojmovima.

Egzaktna vrijednost je, grubo i neprecizno rečeno, točna vrijednost neke veličine. Ako vrijednost neke veličine iznosi π, onda trebamo ispisati π (a ne 3.14, 3.141 i sl.) Ako vrijednost neke veličine iznosi e, onda trebamo ispisati e (a ne 2.7, 2.71 i sl.). Određeni integral je *realan broj*, pa među moguće vrijednosti određenoga integrala pripadaju razlomci (racionalni brojevi) i iracionalni brojevi. Zbog toga će egzaktna vrijednost određenoga integrala biti *simbolički broj*. Time ćemo omogućiti da, ako je ta vrijednost jednaka npr. π, MATLAB ispiše `pi`, ako je ta vrijednost jednaka ln 2, MATLAB ispiše `log 2` i sl.

Međutim, u brojnim praktičnim problemima iz različitih je razloga korisno odrediti približne vrijednosti necjelobrojnih veličina. Npr. podatak da je razmak između dviju osoba jednak π metara je teorijski korektan, ali praktično „neupotrebljiv“ osim ako ne znamo napamet približnu vrijednost broja π. U takvim je situacijama korisno znati približnu vrijednost, tj. aproksimaciju (i)racionalnoga broja uz određen broj značajnih znamenaka. Tako razmak od π metara znači da taj razmak iznosi približno 3.14 metara, odnosno 3 metra i 14 centimetara, što je obično dovoljno za praktične potrebe.

Dakle, u ovim dvama zadacima odredit ćemo egzaktnu vrijednost svakoga određenoga integrala (i to kao simbolički broj), a potom aproksimirati tu egzaktnu vrijednost decimalnim brojem s točnošću od 10^{-5}. Pogledajmo kako to učiniti.

Prva dva retka u obama zadacima su standardna: najprije deklariramo simboličku varijablu korištenu u zadavanju pravila podintegralne funkcija, a potom i sâmu podintegralnu funkciju. Međutim, iako su zadaci jednaki do na podintegralnu funkciju i granice segmenta integracije, što znači da bi se u MATLAB-u trebali rješavati jednakim algoritmom, treći i četvrti redak promatranih kodova se značajno razlikuju. Pogledajmo te razlike i objasnimo ih.

U trećem retku koda **z3.m** određujemo egzaktnu vrijednost zadanoga određenoga integrala. U vježbi 7. vidjeli smo da prilikom rada s eksponencijalnom i logaritamskom funkcijom treba „uključiti“ vrlo opciju `IgnoreAnalyticConstraints` koja će omogućiti pojednostavljivanje izraza bez korištenja standardnih analitičkih uvjeta pohranjenih u MATLAB-ovoj „knjižnici uvjeta“. Tako ćemo postupiti i u ovom slučaju. Nepojednostavljenu egzaktnu vrijednost dobivamo primjenom ugrađene funkcije `int` koja sad ima točno četiri argumenta: podintegralnu funkciju, varijablu integracije, donju granicu segmenta integracije i gornju granicu segmenta integracije. **Zamjena redoslijeda tih argumenata nije dozvoljena!** Dobivenu egzaktnu vrijednost pojednostavljujemo primjenom ugrađene funkcije `simplify` i spomenute opcije `IgnoreAnalyticConstraints`. Vidimo da ugrađena funkcija `simplify` može značajno pojednostavniti i simboličke brojeve.

U četvrtom retku koda **z3.m** određujemo približnu vrijednost primjenom ugrađene funkcije double koja „pretvara“ simboličke brojeve u „obične“ kompleksne brojeve (odnosno, u našem slučaju, u „običan“ realan broj). Koristeći ugrađenu funkciju `round`, dobiveni rezultat zaokružujemo na pet decimalnih mjesta.

Naposljetku, u petom i šestom retku koda **z3.m** ispisujemo dobivene rezultate.

Rješavajući zadatak 4. postupamo ponešto drugačije. Razlog su ponovno „nesretne kombinacije“ eksponencijalne i logaritamske funkcije. U trećem retku koda **z4.m** najprije definiramo varijablu `e` kao *simbolički broj* jednak graničnoj vrijednosti niza $a_n = \left(1+\frac{1}{n}\right)^n$. (Iz *Matematike* 1, ta je granična vrijednost doista jednaka *e*.)

Potom u četvrtom retku koda **z4.m** određujemo egzaktnu vrijednost zadanoga određenoga integrala analogno kao u rješenju zadatka 3., pri čemu kao četvrtu varijablu ugrađene funkcije `int` uvrštavamo deklariranu varijablu `e` i *ne* koristimo ugrađenu funkciju `simplify`. Tako smo svojevrsnim lukavstvom izbjegli ispis racionalnih brojeva s relativno velikim brojnicima i nazivnicima koji bi se pojavili da smo varijablu `e` deklarirali koristeći ugrađenu funkciju `exp`.

Analogno kao u kodu **z3.m**, u petom retku koda **z4.m** određujemo približnu vrijednost zadanoga integrala, a u posljednja dva retka koda ispisujemo dobivene rezultate u komandni prozor koristeći ugrađenu funkciju `fprintf`.

- *Pitanja za razmišljanje*: Preradite dobivene programske kodove tako da se u određivanju egzaktne vrijednosti određenoga integrala **ne** koriste funkcija `simplify` i varijabla `e` definirana kao granična vrijednost niza. Potom izvršite oba dobivena koda i uočite razliku.

- *Pitanja za razmišljanje*: Mogu li se dobivene i maksimalno pojednostavljene egzaktne vrijednosti odrediti na još koji način (bez korištenja funkcije `simplify` i granične vrijednosti niza)? Ako mogu, preradite svaki programski kod tako da se dobiju maksimalno pojednostavljene egzaktne vrijednosti i objasnite svaki redak tako dobivenih kodova.

z5.m

U sljedećim trima zadacima ne upoznajemo nove ugrađene funkcije u MATLAB-u, nego obrađujemo primjenu određenoga integrala na zadatke u kojima se traži računanje površine ravninskoga lika. U zadatku 5. jednadžba krivulje zadana je eksplicitno, dok su u zadacima 6. i 7. krivulje zadane svojim implicitnim jednadžbama.

Strategija rješavanja svih triju zadataka je praktički ista. Površina ravninskoga lika bit će iskazana pomoću određenoga integrala. Znamo da se ta dva pojma, grubo rečeno, mogu poistovjetiti samo za funkcije nenegativne na segmentu integracije. Međutim, ako funkcija nije nenegativna na segmentu integracije, nego je ili strogo negativna ili djelomično nenegativna, a djelomično nepozitivna, onda treba biti vrlo oprezan. Teorija nam kaže da u takvim slučajevima treba uzeti apsolutnu vrijednost, ali, zbog svoje definicije po dijelovima, funkcija apsolutne vrijednosti je najčešće „nezgodna“ za integriranje, pa je treba izbjegavati kad god je to moguće.

U zadatku 5. zapravo tražimo površinu koju „obična“ sinusoida zatvara s osi apscisa na svojemu osnovnomu segmentu. Na tomu se segmentu točno polovica sinusoide nalazi iznad osi apscisa, a druga polovica ispod osi apscisa (i to upravo tim redoslijedom: najprije dolazi dio iznad osi apscisa, a potom dio ispod osi apscisa). Kad bismo jednostavno išli zbrajati pripadne određene integrale, dobili bismo nulu, a taj broj očito nije ispravno konačno rješenje zadatka. Naime, promatrani ravninski lik sigurno ima površinu i ta površina je neki strogo pozitivan realan broj. Zbog toga u rješenju zadatka 5. moramo postupiti lukavije.

Da bismo odredili traženu površinu, dovoljno je odrediti površinu lika kojega polovica sinusoide iznad osi apscisa zatvara s osi apscisa, pa tu površinu pomnožiti s 2. Početak te „gornje“ polovice sinusoide je u lijevom kraju osnovnoga segmenta, a kraj točno na sredini toga segmenta. Ta sredina je zapravo aritmetička sredina donje i gornje granice osnovnoga segmenta. Zbog toga je površina lika kojega polovica sinusoide iznad osi apscisa zatvara s osi apscisa jednaka određenom integralu funkcije koja zadaje sinusoidu računatom u granicama od lijevoga kraja osnovnoga segmenta do sredine toga segmenta.

Struktura programskoga koda **z5.m** jasno proizlazi iz gornjega razmatranja. U prvom retku toga koda moramo popisati sve simboličke varijable koje ćemo koristiti. MATLAB ne dozvoljava korištenje grčkih slova u programskim kodovima, pa umjesto ω i φ pišemo redom `omega` i `fi`. U drugom retku definiramo jednadžbu sinusoide. U trećem retku definiramo donju granicu osnovnoga segmenta. U četvrtom retku definiramo gornju granicu osnovnoga segmenta. U petom retku određujemo sredinu segmenta kao aritmetičku sredinu donje i gornje granice. U šestom retku određujemo traženu površinu u skladu s gornjim razmatranjima.

Uočite da je ovdje bilo vrlo bitno navesti i varijablu po kojoj se integrira. Naime, u prvom retku koda deklarirali smo točno četiri simboličke varijable. U samom kodu zapravo je bitna samo varijabla t, dok sve ostale varijable predstavljaju konstante. Međutim, MATLAB bi potpuno korektno proveo postupak integriranja da smo kao varijablu naveli bilo koju od tri konstante, a ne t. Uvjerite se u to promijenivši na odgovarajući način šesti redak koda.

- *Pitanje za razmišljanje*: Kako bismo izračunali traženu površinu da smo promatrali površinu ravninskoga lika kojega polovica sinusoide **ispod** osi apscisa zatvara s osi apscisa? Preinačite programski kod tako da određuje traženu površinu koristeći taj ravninski lik.

- *Pitanje za razmišljanje*: Teorijski, tražena površina jednaka je određenom integralu *apsolutne vrijednosti* funkcije koja zadaje sinusoidu na osnovnom segmentu. Možemo li u ovom zadatku odrediti traženu površinu koristeći funkciju apsolutne vrijednosti? Precizno objasnite svoj odgovor.

z6.m

U zadatku 6. treba odrediti površinu lika omeđenoga zadanom elipsom. Zbog toga u prvom retku koda **z6.m** najprije crtamo tu elipsu koristeći ugrađenu funkciju `fimplicit`. Da nam bude lakše odrediti granice integracije, pravokutni koordinatni sustav postavljamo tako da se koordinatne osi sijeku u ishodištu i da jedinični razmaci na koordinatnim osima budu jednaki. (Dodatno postavljamo i raspon vrijednosti na svakoj pojedinoj koordinatnoj osi.) Izvršenjem prvih 11 redaka dotičnoga programskoga koda dobivamo grafički prikaz elipse.

U 12. i 13. redak uvrštene su naredbe za zaustavljanje izvršavanja programskoga koda sve dok ne pritisnemo bilo koju tipku na tipkovnici. To je učinjeno zato da možemo dobro promotriti dobivenu sliku i odrediti granice integracije. Zbog simetrije ravninskoga lika s obzirom na *obje* koordinatne osi, dovoljno je odrediti površinu dijela lika koji se nalazi u prvom kvadrantu, pa tu površinu pomnožiti s 4.

Da bismo mogli odrediti određeni integral, iz jednadžbe elipse moramo izraziti jednu od varijabli. Odaberemo npr. varijablu y, pa je u 14. retku koda izrazimo pomoću varijable x. Tako je y postala simbolička funkcija varijable x. Tu funkciju u 15. retku koda integriramo na segmentu $[0, 3]$ koristeći ugrađenu funkciju `int`. Dobiveni rezultat potom pomnožimo s 4 i dobit ćemo traženu površinu.

Napomenimo da je u MATLAB-u moguće iz implicitno zadane jednadžbe izraziti jednu varijablu pomoću druge (u ovom slučaju varijablu y pomoću varijable x). U tu se svrhu koristi ugrađena funkcija `solve`. Međutim, zbog parnoga eksponenta uz y ne dobiva se jednoznačna funkcija varijable x, pa bi za dobivanje jednoznačnosti rješenja trebali postaviti dodatne uvjete na vrijednost varijable x. Imajući pred sobom dobiveni ravninski lik, brže je i jednostavnije analitički („ručno“) izraziti y pomoću x, nego „natjerati“ MATLAB da obavi taj posao.

- *Pitanje za razmišljanje*: Kako možemo odrediti traženu površinu ako iz jednadžbe elipse izrazimo x pomoću y? Preinačite programski kod tako da određuje traženu površinu koristeći pravilo kojim je x definiran kao funkcija varijable y.

z7.m

U ovom zadatku postupamo analogno kao u prethodnom. U prvih devet redaka programskoga koda crtamo zadanu krivulju. Radi određivanja segmenta integracije, postavljamo koordinatne osi tako da se sijeku u ishodištu, a raspon vrijednosti na osi apscisa podešavamo na [-4, 1]. (Uočite da za $x=1$ izraz nije definiran, tj. krivulja ima uspravnu asimptotu $x=1$.) Tako dobivamo ravninski lik čiju površinu tražimo.

Uočavamo da je taj lik simetričan s obzirom na os apscisa. Zbog toga je dovoljno izračunati površinu polovice „omče“ iznad osi apscisa, pa dobivenu vrijednost pomnožiti s 2. Površina polovice „omče“ iznad osi apscisa jednaka je određenom integralu izraza koji se dobije kad se varijabla y izrazi pomoću varijable x. Prisjetimo se da promatramo dio „omče“ iznad osi apscisa, što znači da ćemo izraz za y dobiti kao pozitivan drugi korijen desne strane jednakosti kojom je zadana krivulja. To je učinjeno u 13. retku koda. Dalje je jednostavno: u 14. retku koda određujemo površinu polovice „omče“ iznad osi apscisa kao određeni integral funkcije y na segmentu [-3, 0], pa tu površinu pomnožimo s 2.

- *Pitanje za razmišljanje*: Odredite traženu površinu promatranjem polovice „omče“ ispod osi apscisa. Odgovarajuće preinačite programski kod tako da se dobije točan rezultat.

z8.m

U ovom zadatku upoznat ćemo kako u MATLAB-u zadati, pa integrirati simboličku funkciju zadanu po dijelovima. U tu svrhu koristimo ugrađenu funkciju `piecewise`. Pogledajmo kako se primjenjuje ta ugrađena funkcija.

Svaka po dijelovima zadana funkcija ima domenu podijeljenu na konačan broj dijelova. Na svakom pojedinom dijelu funkcija ima zasebno pravilo. Upravo ta svojstva tvore osnovu primjene ugrađene funkcije `piecewise`. Kao njezine argumente, navodit ćemo dijelove domene i pravila koja vrijede na tim dijelovima. Pritom nećemo komplicirati: dijelovi domene bit će intervali, a pravila analitički izrazi.

Posebnu pozornost moramo obratiti na zadavanje intervala. Nećemo moći napisati npr. $x \in \langle 0,1 \rangle$, nego ćemo tu relaciju morati zapisati koristeći znakove $<$, $\leq$, $>$ i $\geq$. U tu će nam svrhu korisno poslužiti sljedeća tablica.

Relacija	*Ekvivalentni zapis*	*Zapis u* MATLAB-*u*
$x \in \langle a,b \rangle$	$a < x < b$	`(a<x)&(x<b)`
$x \in \langle a,b]$	$a < x \leq b$	`(a<x)&(x<=b)`
$x \in [a,b \rangle$	$a \leq x < b$	`(a<=b)&(x<b)`
$x \in [a,b]$	$a \leq x \leq b$	`(a<=x)&(x<=b)`
$x \in \langle a,+\infty \rangle$	$x > a$	`x>a`
$x \in [a,+\infty \rangle$	$x \geq a$	`x>=a`
$x \in \langle -\infty,a \rangle$	$x < a$	`x<a`
$x \in \langle -\infty,a]$	$x \leq a$	`x<=a`

Tablica 1.

Primijetimo kako u MATLAB-u zadajemo relaciju pripadnosti intervalu s objema konačnim granicama. Koristimo logički operator AND zapisan u „skraćenom“ obliku: &. Npr. $x \in \langle a,b \rangle$ zapravo znači „x je strogo veći od a **i** x je strogo manji od b“, pa je zapis u MATLAB-u doslovni „prijevod“ te rečenice. Analogne tvrdnje vrijede i za druge relacije.

Vratimo se na funkciju `piecewise`. **Obavezan** redoslijed nizanja njezinih ulaznih varijabli je: interval 1, pravilo na intervalu 1, interval 2, pravilo na intervalu 2, ... Dakle, **uvijek** najprije navodimo interval koji je dio domene, a potom pravilo funkcije koje vrijedi na tom intervalu i tako redom. Intervale i pravila međusobno odvajamo zarezom.

Ovdje odmah treba upozoriti na to što će se dogoditi ako intervali na koje je podijeljena domena zadane funkcije *nisu* međusobno disjunktni. U matematičkom zapisu oni moraju biti disjunktni jer inače funkcija nije dobro definirana. No, pogledajmo što će se dogoditi ako u MATLAB-u upišemo sljedeći niz naredbi:

```
syms x
f=piecewise(x>2,x^2,x<5,x+1);
y=subs(f,4)
```

Očekujemo da će MATLAB javiti grešku jer ne zna prema kojemu od dvaju pravila bi računao $f(4)$. Naime, jasno je da je $4>2$ i $4<5$, pa se zapravo smiju primijeniti oba pravila koja, međutim, daju posve različite rezultate. No, MATLAB ispisuje:

```
y =

16
```

i ne javlja nikakvu grešku. Razlog za ovakav postupak je jednostavan. MATLAB „uzima“ vrijednost 4, prolazi kroz intervale na koje je podijeljena domena (zapravo, provjerava upisane logičke uvjete) i kad „naleti“ na *prvi* po redu interval takav da vrijednost 4 pripada tom intervalu, računa vrijednost $f(4)$ prema pripadnom pravilu te završava račun *bez provjere valjanosti ostalih logičkih uvjeta*. Upravo zbog ovoga svojstva treba biti posebno oprezan.

Ukratko komentirajmo programski kod **z8.m.** U njegovu prvomu retku deklariramo simboličku varijablu. U drugomu retku zadajemo funkciju f koristeći izraze iz tablice 1. i pripadajuća pravila. U trećemu retku na uobičajen način određujemo određeni integral funkcije f na zadanom segmentu *bez obzira što je riječ o po dijelovima zadanoj funkciji.*

- *Pitanje za razmišljanje*: Kako biste riješili zadatak bez primjene ugrađene funkcije `piecewise`? Preradite programski kod na odgovarajući način tako da se dobije ispravno rješenje.

z9.m

Posljednji zadatak zapravo ne donosi ništa nova. U njemu određujemo četiri neprava integrala, pri čemu prva tri imaju barem jednu beskonačnu granicu. Već ranije smo vidjeli da za „zapis“ beskonačnosti u MATLAB-u koristimo konstantu `Inf`. Tako će biti i ovoga puta. Sa zadanim integralima postupamo kao da su „obični“ određeni integrali, pa primjenjujemo ugrađenu funkciju `int`. Ako je donja granica $-\infty$, kao treću ulaznu varijablu funkcije `int` uvrštavamo `-Inf`. Ako je gornja granica $+\infty$, kao četvrtu ulaznu varijablu funkcije `int` uvrštavamo `Inf`. Sva četiri integrala konvergiraju, pa ćemo kao konačne rezultate dobiti „konkretne“ realne brojeve. (Da je neki od njih divergentan prema $+\infty$, MATLAB bi kao rezultat ispisao `Inf`. Analogna tvrdnja vrijedi za divergenciju prema $-\infty$.)

- *Pitanje za razmišljanje*: Zašto je posljednji (četvrti) integral nepravi? Precizno objasnite svoj odgovor.

- *Pitanje za razmišljanje*: Kako biste odredili zadane integrale **prema** odgovarajućoj **definiciji** nepravoga integrala koju ste naučili na predmetu *Matematika* 2? Provjerite sve dobivene rezultate tako da **svaki** od zadanih integrala izračunate (u MATLAB-u) koristeći odgovarajuću definiciju nepravoga integrala.