 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGRABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
---	---	--

Zadatak 1.

Izračunajte zbroj reda $\sum_{n=1}^{+\infty} \frac{1}{n^2 + n}$.

Zadatak 2.

Izračunajte egzaktnu i približnu (s točnošću od 10^{-5}) vrijednost zbroja reda $\sum_{n=0}^{+\infty} \frac{(-1)^n}{n+1}$.

Zadatak 3.

Izračunajte egzaktnu i približnu (s točnošću od 10^{-5}) vrijednost zbroja Dirichletova reda $\sum \frac{1}{n^p}$ za $p \in \{2, 3, 4\}$.


Zadatak 4.

Neka su $\varphi = \frac{1+\sqrt{5}}{2}$ i $\psi = 1-\varphi$. Neka je $(F_n)_{n \in \mathbb{N}}$ niz čiji je opći član definiran pravilom $F_n = \frac{\varphi^n - \psi^n}{\sqrt{5}}$.

- S točnošću od 10^{-5} izračunajte zbroj reda $\sum_{n=1}^{+\infty} \frac{1}{F_n}$.
- Odredite apsolutnu vrijednost pogreške koju činimo kad vrijednost zbroja reda iz **a)** podzadatka zamijenimo zbrojem prvih 20 članova toga reda. Zapišite dobiveni rezultat u znanstvenom obliku. (Pretpostavite da mantisa rezultata ima točno 6 znamenaka.)
- Odredite najmanji $k \in \mathbb{N}$ takav da je pogreška aproksimacije vrijednosti zbroja reda iz **a)** podzadatka zbrojem prvih k članova toga reda strogo manja od 10^{-10} .

Zadatak 5.

- Odredite apsolutnu vrijednost pogreške koju činimo kad točnu vrijednost zbroja reda $\sum_{n=0}^{+\infty} \frac{1}{n!}$ zamijenimo zbrojem prvih 15 članova toga reda. Zapišite dobiveni rezultat u znanstvenom obliku. (Pretpostavite da mantisa rezultata ima točno 6 znamenaka.)
- Odredite najmanji $k \in \mathbb{N}$ takav da je pogreška aproksimacije egzaktnosti zbroja reda iz **a)** podzadatka zbrojem prvih k članova toga reda strogo manja od 10^{-14} .

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

Zadatak 6.

Odredite najmanji $k \in \mathbb{N}$ takav da je razlika zbroja reda $\sum_{n=1}^{+\infty} \frac{1}{n^2}$ i zbroja prvih k članova toga reda strogo manja od 10^{-5} .

Zadatak 7.


Odredite najmanji $k \in \mathbb{N}$ takav da je zbroj prvih k članova harmonijskoga reda strogo veći od 10.

Zadatak 8.

Odredite najveći $k \in \mathbb{N}$ takav da je zbroj prvih k članova reda $\sum_{n=1}^{+\infty} \frac{1}{\sqrt{n}}$ strogo manji od 100.

Rezultati zadataka:

- 1.
- $\ln 2$; 0.69315.
- $\frac{\pi^2}{6} \approx 1.64493$; $\zeta(3) \approx 1.20206$; $\frac{\pi^4}{90} \approx 1.08232$.
- a) 3.35989;
b) $2.39177 \cdot 10^{-4}$;
c) 51.
- a) $8.15348 \cdot 10^{-13}$;
b) 17.
- 100 000.
- 12 367.
- 2 573.

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--


Komentari i objašnjenja programskih kodova

U ovoj vježbi obrađujemo još jednu korisnu primjenu simboličkoga računa na određivanje zbrojeva konvergentnih numeričkih redova, te – grubo rečeno – ocjenu brzine konvergencije. Pritom izostavljamo teorijske analize brzine konvergencije jer su one predmet interesa numeričke matematike. Zadovoljit ćemo se s utvrđivanjem koliko članova nekoga reda treba zbrojiti kako bismo s određenom točnošću dobili zbroj reda određen simboličkim računom.

Najvažnija ugrađena funkcija koju ćemo koristiti u ovoj vježbi je `symsum`. Ta funkcija može određivati zbroj konačno mnogo simboličkih brojeva, ali i zbrojeve redova. Podsjetimo da se zbroj reda definira kao granična vrijednost pripadnoga niza djelomičnih zbrojeva, pa se određivanje zbroja reda zapravo svodi na određivanje granične vrijednosti određenih nizova. Osnovni problem kod nizova djelomičnih zbrojeva je vrlo česta nemogućnost nalaženja eksplicitne (zatvorene) formule za opći član toga niza. Zbog toga se često najprije nastoji dokazati konvergencija nekoga reda (primjenom nekoga od osnovnih kriterija za konvergenciju redova), a potom naći način za određivanje barem približne vrijednosti zbroja toga reda. Drugim riječima, najprije se riješi pitanje *postojanja* (egzistencije) *zbroja reda*, pa, ako se utvrdi da taj zbroj postoji (tj. da red konvergira), onda se nastoji barem približno odrediti taj zbroj.

Sintaksa ugrađene funkcije `symsum` je prilično jednostavna. Ona ima ukupno četiri ulazne varijable. Prva od njih je pravilo funkcije čije vrijednosti ćemo zbrajati (najčešće: pravilo općega člana niza). Druga je oznaka nezavisne varijable dotične funkcije (funkcija može sadržavati i „slova“ koja označavaju realne konstante, parametre i sl.). Treća je početna vrijednost od koje započinjemo zbrajanje (najčešće 0 ili 1). Ta vrijednost **obavezno** mora biti **cjelobrojna**. (Dakle, ne dolaze u obzir razlomci, iracionalni brojevi i sl.) Posljednja ulazna varijabla je završna vrijednost s kojom završavamo zbrajanje. Ona može biti „konkretna“ cijeli broj jednak ili veći od početne vrijednosti. Korak između dvije uzastopne vrijednosti uvijek je jednak 1. Napomenimo da početna vrijednost može biti jednaka konstanti $-\text{Inf}$, dok krajnja vrijednost može biti jednaka konstanti Inf . Prvi slučaj je relativno rijedak, dok se drugi koristi upravo pri računanju zbrojeva redova.

Od funkcije `symsum` ne treba očekivati čuda i tražiti izračunavanje egzaktnoga zbroja bilo kakvoga konvergentnoga reda. Takve (egzaktne) vrijednosti u znatnom broju slučajeva ni ne postoje. Zapravo će nam biti posve dovoljno procijeniti vrijednost zbroja reda za kojega utvrdimo da je konvergentan.

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

Prijedimo na rješavanje zadataka.

z1.m

Zadatak 1. je uvodni i lagani zadatak „za zagrijavanje“. Treba izračunati zbroj zadanoga reda. Konvergenciju zadanoga reda lagano možemo dokazati koristeći Raabeov kriterij. Učinite to sami za vježbu. Pogledajmo koliko iznosi zbroj reda.

Prvi redak koda je jasan: deklariramo simboličku varijablu n .


Potom u drugom retku zadajemo pravilo općega člana reda. Taj opći član moramo zasebno označiti, pa smo odabrali oznaku a .

Oprez: Oznaka općega člana reda ne može biti $a(n)$ jer, kako znamo otprije, taj oznaka se koristi za n -tu poziciju u jednoretdanoj ili jednostupčanoj matrici.

U trećem retku računamo zbroj reda koji je sugestivno označen sa `zbroj`. Primjenjujemo ugrađenu funkciju `symsum`. Njezin prvi argument je a - dakle, pravilo općega člana reda. Njezin drugi argument je n - dakle, varijabla pomoću koje je zadan opći član reda. Njezin treći argument je početna od svih vrijednosti po kojima ćemo zbrajati: to je 1 i naznačena je ispod znaka Σ . Njezin posljednji argument je konstanta `Inf` koja „igra ulogu“ $+\infty$: to je ujedno vrijednost s kojom završavamo zbrajanje i naznačena je iznad znaka Σ .

U četvrtom retku izračunata je vrijednost simboličkoga broja `zbroj`, a u petom retku ta je vrijednost ispisana korištenjem funkcije `fprintf`. Uočite da je varijabla `zbroj` simbolička iako je njezina vrijednost cjelobrojna.

- *Pitanje za razmišljanje:* Kako biste analitički provjerili točnost dobivena rezultata? (*Uputa:* Rastavite opći član reda na parcijalne razlomke, pa riješite rekurziju kojom se zadaje pripadni niz djelomičnih zbrojeva.)

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--


z2.m

Ovaj zadatak ne donosi ništa posebno novoga. Rješenje zadatka 1. bilo je cjelobrojno, pa nismo imali razloga pretvarati ga iz simboličkoga u „običan“ realan broj kako bismo s njim dalje mogli nešto računati. U ovom zadatku, međutim, zbroj zadanoga reda nije cjelobrojan, pa ima smisla odrediti njegovu približnu vrijednost. U tu svrhu koristimo ranije upoznatu funkciju `double` koja pretvara simboličke brojeve u „obične“ realne brojeve zapisane u formatu dvostruke preciznosti.

Kod **z2.m** je standardan za ovaj tip zadatka. U njegovu prvomu retku deklariramo simboličku varijablu (n). U drugomu retku zadajemo pravilo općega člana reda. Koristeći simboličku funkciju `symsum`, u trećem retku određujemo egzaktnu vrijednost zbroja reda. U zadatku je naznačeno da je početna vrijednost varijable n jednaka nuli, pa je treći argument funkcije `symsum` jednak nuli. Koristeći funkciju `double`, u četvrtomu retku „pretvaramo“ simbolički broj egzaktna u „običan“ realan broj približna.

U petomu i šestomu retku ispisujemo dobivene vrijednosti. Uočite da je vrijednost egzaktna ispisana korištenjem opcije `%s` kojom smo MATLAB-u dali do znanja da će na dotično mjesto u rečenici trebati uvrstiti simboličku varijablu. Za razliku od te vrijednosti, vrijednost približna ispisana je korištenjem opcije `%d` kojom smo MATLAB-u dali do znanja da će na dotično mjesto u rečenici trebati uvrstiti broj zapisan u formatu dvostruke preciznosti.

- *Pitanje za razmišljanje:* Provjerite da zadatak ima istovjetna rješenja ako umjesto zadanoga reda promatramo red $\sum_{n=1}^{+\infty} \frac{(-1)^{n-1}}{n}$. Objasnite zašto se dobiju ista rješenja. (*Uputa:* Pokažite da se radi o istim redovima.)

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

z3.m

U ovom zadatku promatramo *Dirichletove* (ili *hiperharmonijske*) redove, te, za prvih nekoliko vrijednosti n , određujemo egzaktnu i približnu vrijednost zbroja reda. Kako znamo iz *Matematike 2*, Dirichletov red konvergira ako i samo ako je $p > 1$, pa je to svojstvo razlog zbog kojega se zbrojevi reda *ne* računaju za $p = 1$.


- *Pitanje za razmišljanje:* Koji se poznati numerički red dobiva za $p = 1$?

Da ne bismo morali triput ispisivati istu rečenicu, u ovom programskom kodu iskoristit ćemo *for*-petlju. Ideja rješavanja je sljedeća. Triput ćemo proći kroz *for*-petlju: prvi put za $p = 2$, drugi put za $p = 3$ i posljednji, treći put za $p = 4$. U svakom pojedinom prolasku izračunat ćemo egzaktnu i približnu vrijednost zbroja Dirichletova reda (za dotični p), te ispisati dobivene rezultate kao dijelove jedne rečenice analogno kao u rješenjima prethodnih zadataka. Uočite da se pritom vrijednost simboličke varijable a mijenja pri svakom prolazu kroz petlju. Iako, zbog relativno maloga broja vrijednosti p za koje trebamo odrediti zbroj reda, korištenjem *for*-petlje nismo postigli posebnu „uštedu“ u duljini koda (tj. ukupnom broju linija u kodu) u odnosu na varijantu prema kojoj se svaki zbroj računa u zasebnom retku koda, ovaj način daje željenu „uštedu“ u slučajevima kad je potrebno ispisati značajno više vrijednosti zbrojeva reda.

Dodatno primijetite kako su u okviru funkcije `fprintf` poredane vrijednosti koje treba ispisati. Prva vrijednost je varijabla p u formatu dvostruke preciznosti, druga vrijednost je simbolička varijabla a , a treća vrijednost je približna vrijednost varijable a ponovno zapisana u formatu dvostruke preciznosti. Tako smo u jednom retku koda i u jednoj rečenici iskoristili mogućnost ispisa različitih tipova varijabli.

Kao egzaktna vrijednost zbroja Dirichletova reda za $p = 3$ pojavljuje se vrijednost tzv. *Riemannove zeta-funkcije*. Vrijednosti te funkcije su definirane upravo kao zbrojevi Dirichletova reda. Vrijednost $\zeta(3)$ poznata je i kao *Apéryjeva konstanta*, a pojavljuje se u brojnim fizikalnim modelima.

- *Pitanje za razmišljanje:* Kako biste riješili zadatak koristeći *while*-strukturu? Preinačite programski kod na odgovarajući način tako da dobijete ispravno rješenje.
- *Pitanje za razmišljanje:* Kako biste riješili zadatak ako bi trebalo pohraniti (a ne samo ispisati) određene egzaktne, odnosno približne vrijednosti?

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

Preinačite programski kod na odgovarajući način tako da dobijete ispravno rješenje.

(*Uputa:* Za pohranu vrijednosti koristite jednoretčane ili jednostupčane matrice. Pokušajte indeksirati pozicije u matrici koristeći postojeću varijablu p , odnosno bez uvođenja novih varijabli.)

z4.m

U ovom zadatku promatramo red čiji je opći član jednak recipročnoj vrijednosti n -toga Fibonaccijeva broja. Podsjetimo se, Fibonaccijev niz definiran je rekurzivnom relacijom $F_n = F_{n-1} + F_{n-2}$, za $n \geq 3$, te početnim uvjetima $F_1 = F_2 = 1$. U ovom smo ga predmetu već susreli u zadatku 5. vježbe 6. U *Matematici 2* (u okviru točke 2.1. *Rekurzije*) izveli smo tzv. *Binetovu formulu* za eksplicitno (nerekurzivno) određivanje n -toga člana toga niza:


$$F_n = \frac{1}{\sqrt{5}} \cdot \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right).$$

Upravo tu formulu koristimo u ovom zadatku. Dodatno koristimo i lako dokazivi identitet:

$$\frac{1-\sqrt{5}}{2} = 1 - \frac{1+\sqrt{5}}{2}.$$

Pogledajmo najprije strukturu prvih sedam redaka koda koji predstavljaju rješenje zadatka 4.a). U prvomu retku koda deklariramo simboličku varijablu (n). U drugomu retku koda deklariramo konstantu $a := \frac{1+\sqrt{5}}{2}$. U trećemu retku koda koristimo gore navedeni identitet, pa konstantu b računamo prema formuli $b = 1 - a$. U četvrtomu retku koda definiramo Fibonaccijev niz koristeći Binetovu formulu. Primijetite da je F simbolička funkcija (varijable n). U petomu retku definiramo opći član reda kao recipročnu vrijednost n -toga člana Fibonaccijeva niza. Jasno je da smo četvrti i peti redak mogli spojiti u jedan, tj. definirati opći član reda bez korištenja funkcije F , ali ovaj način je jasniji i pregledniji.

Može se pokazati da ne postoji dovoljno jednostavna zatvorena formula za zbroj promatranoga reda. To smo već istaknuli i u uvodu: mnoge zbrojeve redova efektivno možemo odrediti samo približno. Takav je i promatrani red. Zbog toga ne računamo zasebno egzaktnu vrijednost zbroja reda kao u prethodnim zadacima,


 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

nego u šestom retku koda odmah računamo (samo) približnu vrijednost toga zbroja, a u sedmomu je retku ispisujemo. Time je **a)** podzadatak riješen.

Kako znamo, zbroj konvergentnoga reda je zapravo *granična vrijednost* (niza djelomičnih zbrojeva), a ne broj nastao zbrajanjem beskonačno mnogo pribrojnika. Zbog toga se postavlja pitanje: možemo li tu graničnu vrijednost sa *dovoljnom točnošću* izračunati zbrajanjem *konačno mnogo* članova reda? Pokazuje se da je to moguće učiniti. Pritom ni u kojem slučaju „konačno mnogo“ ne treba shvatiti kao npr. zbroj prvih 5, 10, 20 i dr. (dakle, relativno malo) članova niza. U mnogim slučajevima vrijednost zbroja reda s točnošću od 10^{-5} možemo izračunati zbrojimo li prvih nekoliko tisuća članova reda. Međutim, svakako je riječ o zbroju *konačno mnogo* pribrojnika, pa se u tom slučaju mogu primijeniti dobro nam poznata svojstva zbrajanja prirodnih brojeva (komutativnost, asocijativnost i dr.).

Ovdje se otvara i još jedan problem. (Konačan) zbroj prvih n članova nekoga niza/rede možemo izračunati zbrajajući simboličke brojeve (korištenjem funkcije `symsum`) ili zbrajajući „obične“ realne brojeve zapisane u formatu dvostruke preciznosti. Teorijski, ta dva načina su potpuno ravnopravna. Praktično, oni, nažalost, nisu ravnopravni. Naime, u slučajevima zbrajanja npr. nekoliko tisuća simboličkih brojeva MATLAB radi bitno sporije negoli kad te brojeve zbraja kao „obične“ brojeve zapisane u formatu dvostruke preciznosti. Dakle, prema kriteriju brzine izvođenja programa (što može biti vrlo bitno na kolokvijima i praktičnim ispitima), ti načini nisu ravnopravni. Zbog toga ćemo, kad god to bude primjereno, u ovakvim situacijama preferirati nesimbolički račun, odnosno rad s „običnim“ realnim brojevima.

U **b)** podzadatku, nasreću, ne zbrajamo puno pribrojnika, pa nema razloga „prebaciti“ se na nesimbolički račun. U osmomu retku koda računamo zbroj prvih 20 članova reda i „prebacujemo“ ga u format dvostruke preciznosti, pa u devetom retku koda računamo apsolutnu vrijednost razlike „stvarnoga“ zbroja reda i zbroja prvih 20 članova reda. Svi članovi promatranoga reda su pozitivni brojevi, pa smo sigurni da će zbroj prvih 20 članova reda biti strogo manji od zbroja reda. Zbog toga smo u ovom slučaju mogli izostaviti apsolutnu vrijednost. No, u slučaju *bilo kojega* konvergentnoga reda, a posebno alternirajućih redova (koji sadrži naizmjenice pozitivne i negativne članove), moguće je da, za neke n , zbroj prvih n članova reda bude strogo veći od zbroja reda. Da se ne bismo morali pitati o kojemu se slučaju radi u svakom pojedinom zadatku, najbolje je koristiti apsolutnu vrijednost. To smo i učinili u devetom retku koda.


 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

U retcima 10. – 17. riješen je **c)** podzadatak. Odmah naglasimo: unaprijed *ne znamo* koliko ćemo prvih članova reda trebati zbrojiti. Možda će to biti prvih deset članova, možda prvih sto, a možda i prvih milijun. Teorijski, svejedno nam je jer će posao određivanja traženoga broja ionako napraviti MATLAB, ali praktično nije nimalo svejedno. Upravo zato jer ne znamo ukupan broj pribrojnika „prebacujemo“ se sa simboličkih na „obične“ realne brojeve. Tako ćemo dobiti na brzini izvođenja toga dijela programa.

Sama struktura redaka 10. – 17. je relativno jednostavna. Koristimo *while*-strukturu i zbrajamo sve više i više članova reda sve dok je apsolutna vrijednost razlike „stvarnoga“ zbroja reda i zbroja prvih k članova jednaka ili veća od 10^{-10} . Primijetite da je početna vrijednost varijable *zbroj* jednaka nuli i da u svakom prolazu kroz *while*-strukturu računamo k -ti član pripadnoga niza djelomičnih zbrojeva. Radi ubrzavanja izvođenja programa, koristeći ugrađenu funkciju `matlabFunction` simboličku funkciju g definiranu u petomu retku pretvorili smo u „običnu“ funkciju g . Tako vrijednosti funkcije g možemo jednostavno računati kao $g(k)$, tj. bez korištenja ugrađene funkcije `subs`.

U nekom trenutku apsolutna vrijednost razlike „stvarnoga“ zbroja reda i zbroja prvih k članova reda postat će strogo manja od 10^{-10} . Tada će se *while*-struktura prestati izvršavati. Rješenje **c)** podzadatka je upravo vrijednost varijable k za koju će se to dogoditi, pa tu vrijednost ispisujemo u posljednjem retku koda.

- *Pitanja za razmišljanje:* Što bi se dogodilo da smo u desetomu retku koda deklarirali $k = 1$? Kako bi trebalo preinačiti retke 11. – 17. da se i u tom slučaju dobije ispravno rješenje **c)** podzadatka? Provedite potrebne preinake i precizno obrazložite svaku od njih.

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

z5.m


U odnosu na prethodni zadatak, ovaj ne donosi ništa posebno novoga, osim jedne nove ugrađene funkcije i jedne skrivene zamke u rješavanju. Komentirajmo zasebno svaki „novitet“.

Nova ugrađena funkcija koja se pojavljuje u ovom zadatku je funkcija `factorial`. Ona računa vrijednosti funkcije $n!$, pa zbog toga ima točno jednu ulaznu varijablu: prirodan broj n . Pokušate li kao vrijednost te varijable upisati neki realan broj koji nije prirodan, MATLAB će javiti grešku. Naziv ugrađene funkcije je engleska inačica riječi *faktorijel* kojom obično čitamo oznaku $n!$. U njezinoj primjeni doista nema ničega teškoga: sve i da želite, ne možete unijeti barem dvije ulazne varijable, pa nema smisla pamtit i redoslijed ulaznih varijabli. Ima smisla zapamtiti da ulazna varijabla ove funkcije mora biti prirodan broj (ili, eventualno, nula).

Prva četiri retka koda su manje-više uobičajena: najprije deklariramo simboličku varijablu n , potom računamo egzaktnu vrijednost zbroja reda i odmah je pretvaramo u „običan“ realan broj radi daljnjih računa s tom vrijednošću, pa računamo zbroj prvih 15 članova reda. Čini se da tu nema ničega neobičnoga. Ipak, na kraju četvrtoga retka koda skrila se zamka. Doista, tražimo zbroj prvih 15 članova reda, ali ne smijemo previdjeti da je inicijalna vrijednost varijable n u izrazu kojim je definiran red jednaka nuli. Dakle, prvi član reda dobijemo za $n=0$, drugi član reda dobijemo za $n=1$ itd., pa 15. član reda dobijemo za $n=14$. Zbog toga varijabla po kojoj se zbraja u četvrtom retku koda poprima sve cjelobrojne vrijednosti između (uključivo) 0 i 14. Dakle, vrlo je bitno pripaziti na *početnu* vrijednost te varijable jer ona u nekim slučajevima može biti jednaka 0.

Ostatak koda je potpuno analogan odgovarajućem dijelu prethodnoga zadatka, pa ga nećemo dodatno komentirati. Primijetiti ćemo tek da, za razliku od prethodnoga zadatka, rješenje zadatka predstavlja vrijednost varijable k određena posljednjim izvršavanjem *while*-strukture, a ne vrijednost $k-1$. Uočite da je inicijalizirana vrijednost te varijable 0 (jer se prvi član reda dobiva za $n=0$), pa nakon prvoga prolaza strukturom varijabla k ima vrijednost 1, nakon drugoga vrijednost 2, a nakon posljednjega (k -toga) vrijednost k . Zbog toga upravo tu vrijednost treba ispisati kao rješenje **b)** podzadatka.

- *Pitanje za razmišljanje:* Analitički odredite egzaktnu vrijednost zbroja zadanoga reda. Potom provjerite svoje rješenje koristeći MATLAB. Utvrdite s kojom je preciznošću određen taj zbroj u trećem retku koda.

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

- *Pitanje za razmišljanje:* Je li moguće promijeniti *pravilo* općega člana reda tako da se prvi član reda uobičajeno dobije za $n=1$, drugi za $n=2$ itd.? Ako jest, preinačite kod na odgovarajući način tako da ispisuje ispravan rezultat. U suprotnom, precizno objasnite svoj odgovor.


z6.m

Ovaj zadatak se, prema svojoj postavci, ni po čemu ne razlikuje od zadataka **4.c)** i **5.c)**. Štoviše, ima isti algoritam za rješavanje, pa je jedino što ga efektivno razlikuje od navedenih zadataka pravilo općega člana reda. Zbog čega je onda zadan ovaj zadatak i što se njime želi postići?

U **4.c)** i **5.c)** podzadatku željeni cilj postigli smo već nakon nekoliko desetaka zbrojenih članova reda. Time smo zapravo pokazali da red prilično brzo konvergira prema svojem zbroju. No, kako smo ranije istaknuli, ta konvergencija ne mora biti toliko brza čak i kad ne zahtijevamo relativno malo odstupanje zbroja prvih k članova reda od zbroja reda. Ovaj zadatak će nam pokazati primjer upravo takvoga reda.

Struktura koda je manje-više standardna. Najprije deklariramo simboličku varijablu, potom pravilo općega člana reda (riječ je o Dirichletovu redu za $p=2$, pa znamo da taj red konvergira), te izračunamo zbroj reda i zapišemo ga u formatu dvostruke preciznosti kako bismo s njim dalje mogli brže i jednostavnije računati. Potom, analogno kao u rješenjima zadataka **4.c)** i **5.c)** stvorimo *while*-strukturu pomoću koje odredimo traženi broj. Dakle, u samoj strukturi nema ništa posebno u odnosu na kodove koje smo vidjeli u rješenjima prethodnih podzadataka.

Međutim, nakon što pohranimo i pokrenemo napisani kod, čeka nas malo iznenađenje. Želimo li da razlika zbroja reda i zbroja prvih k članova reda bude strogo manja od 10^{-5} , potrebno je zbrojiti prvih (barem) 100 000 članova reda. Dakle, konvergencija reda je značajno sporija od konvergencije redova iz prethodnih zadataka. Ta spora konvergencija ima značajan utjecaj i na brzinu rada programskoga koda. Kao što smo i rekli u uvodu, budući da unaprijed ne znamo koliko članova reda treba zbrojiti, a želimo relativno brzo dobiti točan rezultat, nećemo raditi sa simboličkim brojevima, nego sa „običnim“ realnim brojevima zapisanima u formatu dvostruke preciznosti. Rješenje ovoga zadatka mogli smo dobiti rabeći samo simboličke brojeve. Međutim, pripadni programski kod bi se tada izvršio u bitno duljem vremenu u odnosu na vrijeme izvršavanja našega programskoga koda. Da smo još više smanjili razliku (npr. na 10^{-8}), razlika u duljini

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--


vremena izvršavanja bila bi još izraženija jer bi traženi broj članova bio nešto veći od 90 milijuna.

- *Pitanje za razmišljanje.* Što bi se dogodilo da smo u četvrtomu retku deklarirali $k = 1$? Kako bi trebalo preraditi retke 5. – 11. tako da se dobije ispravan rezultat? Provedite potrebne preinake i precizno obrazložite svaku od njih.
- *Pitanje za razmišljanje.* Što se događa s brzinom konvergencije Dirichletova reda ako povećavamo eksponent p ? Riješite ovaj zadatak u slučajevima $p \in \{3, 4, 5\}$, usporedite dobivene rezultate i izvedite zaključak.

z7.m i z8.m

U posljednjim dvama zadacima ove vježbe promatramo *divergentne* redove. Oba ta reda su Dirichletovi redovi za $p \leq 1$ i divergiraju prema $+\infty$. Ako je $p = 1$, dobiva se harmonijski red iz zadatka 7., a ako je $p = \frac{1}{2}$, dobiva se red iz zadatka 8. Zbog divergencije, nema smisla računati zbrojeve redova, ali ima smisla utvrditi brzinu divergencije pojedinoga reda. Konkretno, za svaki $x \in \mathbb{R}$ sigurno postoje najveći $k \in \mathbb{N}$ za koji je zbroj prvih k članova reda strogo manji od x (ili najviše jednak x), te najmanji $l \in \mathbb{N}$ takav da je zbroj prvih l članova reda strogo veći od x (ili barem jednak x). Ako red sadrži samo pozitivne članove (a Dirichletovi redovi su takvi), onda je problem vrlo trivijalan za $x \leq 0$. U takvim su slučajevima, naime, $k = l = 1$ jer je već prvi (pozitivan) član reda strogo veći od x . Da bi se izbjegli ovakvi slučajevi, u pravilu se uzima $x > 0$, što je i učinjeno u ovim zadacima.


Strukture samih kodova su praktički identične sve do pretposljednje retke. U kodu **z7.m**, koristeći *while*-strukturu, zbrajamo sve više i više članova harmonijskoga reda. U nekom će trenutku taj zbroj premašiti 10, pa će se struktura tada prestati izvršavati. Brojač ukupnoga broja pribrojnika koje smo uzeli (tj. varijabla k) bit će – kako smo već ranije vidjeli – jednak traženom broju. Dakle, nema ničega novoga u odnosu na rješenja prethodnih podzadataka, osim što u ovom slučaju promatramo *divergentan* red, pa zadatak ima smisla (štoviše, ima i jedinstveno rješenje!) za *bilo koji* strogo pozitivan realan broj. Da je promatrani red bio konvergentan, morali bismo paziti koju donju granicu za veličinu vrijednosti zbroja prvih k članova zadajemo.

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

U zadatku 7. namjerno je odabrana donja granica $m=10$ jer se vrlo brzo dobije da treba uzeti prvih 12 367 članova reda. Međutim, uzmemo li $m=20$, dobijemo da treba uzeti prvih 272 400 600 članova reda. To nas navodi na zaključak da harmonijski red vrlo sporo divergira prema $+\infty$. U skladu s tim je i brzina izvršavanja programskoga koda: za $m \in \{10, 20\}$, relativno brzo se dobiju konačni rezultati, ali već za $m=25$ potrebno je malo više pričekati da se dobije ispravan rezultat 40 427 833 562.

U zadatku 8. imamo naizgled vrlo sličan, ali ipak dovoljno različit zadatak. Tražimo koliko najviše prvih članova reda treba zbrojiti da zbroj ne premaši 100. Posljednji redak koda sada je malo drugačiji, pa treba pojasniti zašto se ispravno rješenje dobije tako da se posljednja vrijednost varijable k smanji za 1. Dakle, *while*-struktura će se prestati izvršavati u trenutku kad vrijednost varijable zbroj prvi put postane jednaka ili veća od 100. U tom trenutku vrijednost brojača k predstavlja odgovor na pitanje: Koliko *najmanje* članova zadanoga reda trebamo zbrojiti tako da zbroj bude *barem jednak* 100?. Znamo i da se *while*-struktura u tom trenutku izvršila točno k puta. Međutim, posljednje izvršavanje te strukture je višak (tj. bilo je nepotrebno) jer smo njime premašili zadanu vrijednost, a to ne želimo. Zbog toga je željeni broj izvršavanja strukture za jedan manji od broja njezina stvarnoga izvršavanja. Dakle, traženi broj je jednak $k-1$, pa zbog toga u posljednjemu retku koda piše ta vrijednost.

- *Pitanja za razmišljanje:* Može li se preinačiti programski kod **z8.m** tako da zaključna vrijednost brojača k bude jednaka traženoj vrijednosti? Ako može, preinačite ga na odgovarajući način tako da ispisuje ispravan rezultat. U suprotnom, precizno objasnite svoj odgovor.
- *Pitanja za razmišljanje:* Što bi sve trebalo promijeniti u obama programskim kodovima da smo umjesto strogih nejednakosti razmatrali slabije nejednakosti (tj. slučaj „najviše jednak“ u zadatku 7. i slučaj „barem jednak“ u zadatku 8.)? Kako dodavanje znaka jednakosti utječe na rješenje zadatka? Preinačite dobivene programske kodove na odgovarajući način tako da ispisuju ispravne rezultate.

 TEHNIČKO VELEUČILIŠTE U ZAGREBU POLYTECHNICUM ZAGABIENSE Elektrotehnički odjel	Matematički alati u elektrotehnici (preddiplomski stručni studij elektrotehnike)	Vježba 10. Numerički redovi u MATLAB-u.
--	---	--

Domaća zadaća

1. Izračunajte egzaktnu vrijednost zbroja reda $\sum_{n=1}^{+\infty} \frac{(-1)^{n-1}}{2 \cdot n - 1}$. Pojednostavnite dobiveni izraz što više možete.
2. Izračunajte egzaktnu i približnu (s točnošću od 10^{-5}) vrijednost zbroja reda $\sum_{n=0}^{+\infty} \frac{(-1)^n}{(2 \cdot n)!}$
3. a) Izračunajte egzaktnu vrijednost zbroja reda $\sum_{n=0}^{+\infty} \frac{89 \cdot F_n}{10^{n+1}}$, gdje je F_n n -ti Fibonaccijev broj definiran u zadatku 4.
 b) Procijenite apsolutnu vrijednost pogreške koju činimo ako egzaktnu vrijednost zbroja zadanoga reda zamijenimo zbrojem prvih 10 članova toga reda.
 c) Odredite najmanji $k \in \mathbb{N}$ takav da je razlika egzaktnosti zbroja zadanoga reda i zbroja prvih k članova reda strogo manja od konstante ϵ .
4. Odredite najmanji $k \in \mathbb{N}$ takav da je zbroj prvih k članova reda $\sum_{n=0}^{+\infty} \frac{1}{\sqrt{n^2 + 1}}$ jednak ili veći od 10.
5. Odredite najveći $k \in \mathbb{N}$ takav da je zbroj prvih k članova reda $\sum_{n=0}^{+\infty} \frac{1}{\sqrt[3]{n^2 + 2}}$ jednak ili manji od 50.

Rezultati zadataka za domaću zadaću

1. $\frac{\pi}{4}$.
2. Egzaktna vrijednost je jednaka $\cos 1$. Približna vrijednost je jednaka 0.5403.
3. a) 1; b) $5.84 \cdot 10^{-8}$; c) 21.
4. 6669.
5. 5243.