Programska rešenja pri proceni broja Pi Monte Carlo metodama uz interaktivne animacije

Ana Savić
Informacioni sistemi
Visoka škola elektrotehnike i računarstva
Beograd, Srbija
ana.savic@viser.edu.rs

Goran Bjelobaba
Narodna banka Srbije
Fakultet organizacionih nauka
Beograd, Srbija
Goran.Bjelobaba@nbs.rs

Hana Stefanović
Informacione tehnologije
Visoka škola strukovnih studija za IT
Beograd, Srbija
hana.stefanovic@its.edu.rs

Sažetak—**U ovom radu izložen je postupak procene vrednosti iracionalnog broja Pi primenom Monte Carlo metoda, pri nasumičnom izboru velikog broja tačaka untar kvadrata u koji je upisana kružnica, procenom verovatnoće da slučajno generisana tačka prirada i unutrašnjosti kruga, respektujući odnos površina kvadrata i upisanog kruga. Urađena je analiza tačnosti estimirane vrednosti broja Pi u zavisnosti od broja slučajno generisanih tačaka, uz očekivani rezultat koji je u skladu sa zakonom velikih brojeva. Prikazano je nekoliko ishoda simulacije, uz diskusiju kvaliteta generatora pseudoslučajnih brojeva u programskim okruženjima MATLAB i Mathematica. Korišćeni su generatori dostupni u okviru ovih alata, bez detaljnije analize mogućnosti poboljšanja njihovih performansi. Animacija kojom je ispraćen rast broja slučajno generisanih tačaka, kao i njegov uticaj na tačnost procene, urađeni su u Scratch programibilnom alatu.**

Ključne reči-generator pseudoslučajnih brojeva; Monte Carlo metode; procena vrednosti broja Pi

I. UVOD

Monte Carlo metode predstavljaju statistički simulacioni postupak koji podrazumeva upotrebu nizova slučajnih brojeva za vršenje simulacije [1]. Upotreba i primena koncepta slučajnosti i procesa ponavljanja u metodi, zbog određene analogije sa aktivnostima u kazinu i nekim igrama na sreću, uticala je da i sam naziv metoda ”Monte Carlo”, popularizovan od strane prvih istraživača, inicijalno bude formiran baš po nazivu čuvenog kazina u Monaku.

S obzirom da su za dobijanje dovoljno tačne ocene tražene veličine potrebna izračunavanja i odgovarajuća statistička obrada za veoma velik broj posebnih slučajeva, efektivna primena Monte Carlo metoda omogućena je tek pojavom elektronskih računara [2], [3]. Monte Carlo metoda zahteva da se fizički sistem opiše funkcijama gustine verovatnoće, da bi se na osnovu toga izvršio proces simulacije slučajnim izborom vrednosti iz funkcija [4]. Nakon realizacije mnogobrojnih eksperimenata ili proba, za rešenje se uzima prosečan rezultat svih simulacija. Monte Carlo metode imaju značajnu primenu prilikom rešavanja različitih problema kod kojih se teško dolazi do analitičkih izraza, pa je pogodno koristiti računske metode zasnovane na modeliranju slučajnih promenljivih [5]-[7].

Osim primene u integraciji, optimizaciji i generisanju uzoraka kod raspodele verovatnoće, Monte Carlo metode su veoma korisne i u fizičkim problemima, u okviru simulacije sistema sa više stepena slobode. Primena Monte Carlo metoda podrazumeva: definisanje domena ulaza, zatim generisanje niza slučajnih ulaza pomoću raspodele verovatnoće, kao i vršenje determinističkih proračuna i sažimanje ili usrednjavanje rezultata [8]-[10].

Prilikom procene broja Pi, koji je zbog mnogih primena interesantan u matematici, korišćene su razne aproksimacije racinalnim brojevima, kao što su 22/7 koja ima tačnost za prve dve decimale, zatim 333/106 za 4 decimale, 355/113 za 5 decimala… Proračun više od 12 triliona decimala broja Pi primenom Chudnovsky formule trajao je skoro 100 dana, zahtevajući izuzetne performance računara na kojem je proračun rađen [11]-[13].

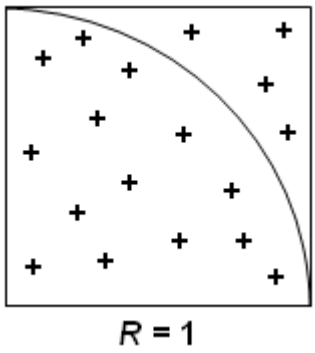
U okviru ovog rada primenom Monte Carlo metoda, na vrlo jednostavan način može se proceniti vrednost broja Pi, pri čemu tačnost i kvalitet estimacije svakako zavise od kvaliteta generisanog niza slučajnih ulaza, kao i od broja izvršenih eksperimenata.

U drugom poglavlju rada izložena je postavka problema i definisan je domen ulaza, dok je realizacija algoritma, uključujući i diskusiju postignutih rezultata, data u trećem poglavlju. U četvrtom poglavlju je prikazana animacija rađena u Scratch programabilnom alatu, sa ciljem da se, prvenstveno u nastavi matematike, verovatnoće i statiske, na interaktivan način, omogući vizuelno praćenje uticaja broja slučajno generisanih tačaka koje se koriste u algoritmu, na tačnost aproksimirane vrednosti broja Pi.

II. POSTAVKA PROBLEMA I DEFINISANJE DOMENA ULAZA

Primena Monte Carlo metoda u cilju procene broja Pi odabirom tačaka unutar kruga koji je upisan u kvadrat, svodi se na određivanje verovatnoće da slučajno izabrana tačka iz kvadrata bude u krugu upisanom u kvadrat.

Površina kvadrata određena je sa $A_{box}=R^2=1$, sa ukupno N_{box} generisanih tačaka, dok je površina od interesa za vršenje eksperimenta četvrtina kruga jediničnog prečnika, određena sa $A_{pie}=1/4*R^2*Pi=Pi/4$, pri čemu se u toj površini nalazi ukupno N_{pie} tačaka, kao što je prikazano na Sl.1.



Slika 1. Definisanje ulaznih podataka i površine od interesa

Odnos površina A_{pie}/A_{box} koji iznosi Pi/4, biće aproksimiran odnosom generisanih tačaka u što većem broju, sa uniformnom raspodelom, a nakon provere koje od njih se nalaze unutar površine od interesa, odnosno unutar četvrtine upisanog kruga, što se može opisati sa:

$$\frac{N_{pie}}{N_{box}} \approx \frac{A_{pie}}{A_{box}} = \frac{\pi}{4} \tag{1}$$

Estimacija broja Pi određena je sa:

$$\hat{\pi} = \frac{4N_{pie}}{N_{box}} \tag{2}$$

Primena ovakve metode zahteva vršenje jednostavnog proračuna za svaki ulaz, u smislu provere da li se tačka nalazi unutar kruga, nakon čega se jednostavno procenjuje vrednost broja Pi. Na tačnost procene svakako utiče kvalitet same raspodele, što je uslovljeno performansama generatora psedoslučajnih brojeva, ali je od velikog značaja i broj ulaznih tačaka koje će se koristiti u samoj proceni.

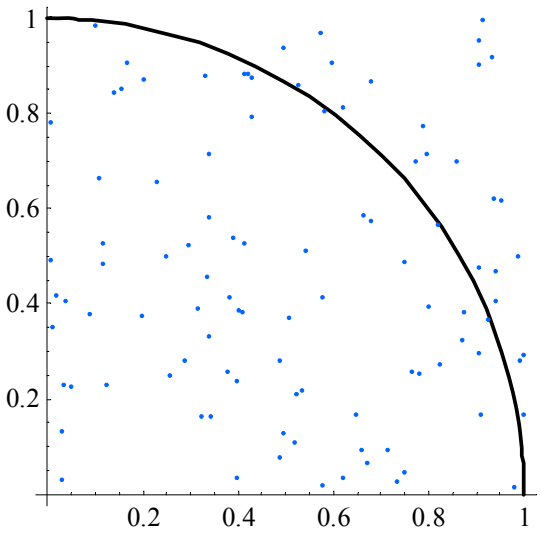
Generatori slučajnih brojeva koriste konačan skup podataka zbog ograničenih mogućnosti algoritama i računara na kojima se izvršavaju [2]. Iz tog razloga generator slučajnih brojeva u nekom trenutku počinje da ponavlja nizove brojeva, a maksimalna dužina tog niza pre početka ponavljanja predstavlja period generatora slučajnih brojeva.

Struktura slučajnih tačaka je takođe veoma važna, a takođe je bitno da se generator lako implementira, odnosno da je programski kod kratak i jednostavan. Uticaj povećanja broja ulaznih tačaka koje će se koristiti u procesu procene, na kvalitet same procene analiziran je u narednom poglavlju.

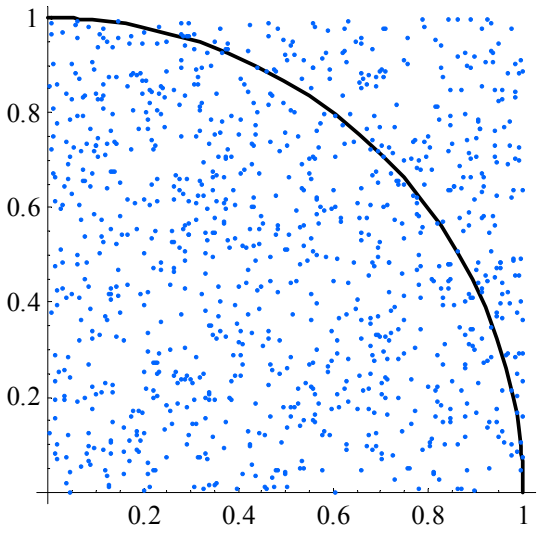
III. REALIZACIJA ALGORITMA I PRIKAZ REZULTATA

Upotrebom RandomReal [ ] generatora psedoslučajnih brojava u programskom okruženju Mathematica [14], za različite vrednosti broja ulaznih tačaka, dobijeni su rezultati prikazani na Sl.2, Sl.3. i Sl.4. Procena je vršena za n=100, n=1000 i n=10000 slučajno generisanih tačaka, što je ilustrovano na Sl.2, Sl.3. i Sl.4., respektivno.

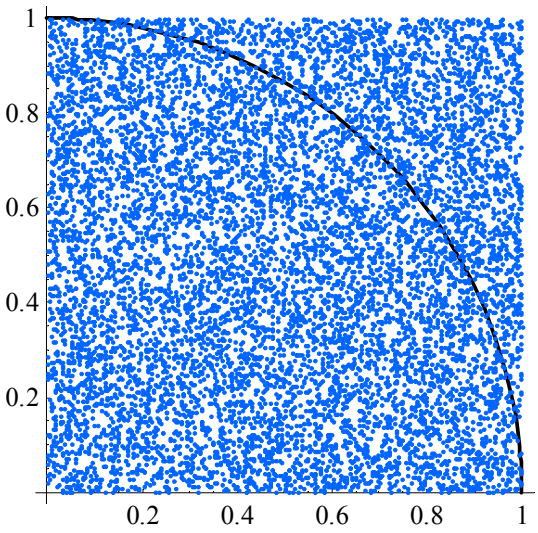
Primenom generatora pseudoslučajnih brojeva u MATLAB programskom okruženju [15], sa izdvojenim tačkama koje zadovoljavaju uslov pripadnosti krugu, prikazan je na Sl.5, Sl.6, Sl.7. i Sl.8. respektivno, za različite vrednosti broja generisanih tačaka, n=500, n=1000, n=5000 i n=10000, respektivno. U slučaju 500 generisanih tačaka, procenjena vrednost iznosi 3.1680, za 1000 tačaka iznosi 3.2240, za 5000 tačaka iznosi 3.1280, dok za 10000 tačaka iznosi 3.1496.



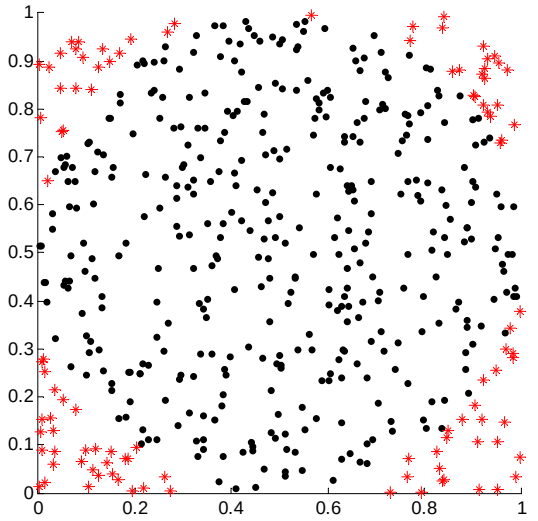
Slika 2. Određivanje broja Pi slučajnim odabirom n=100 tačaka



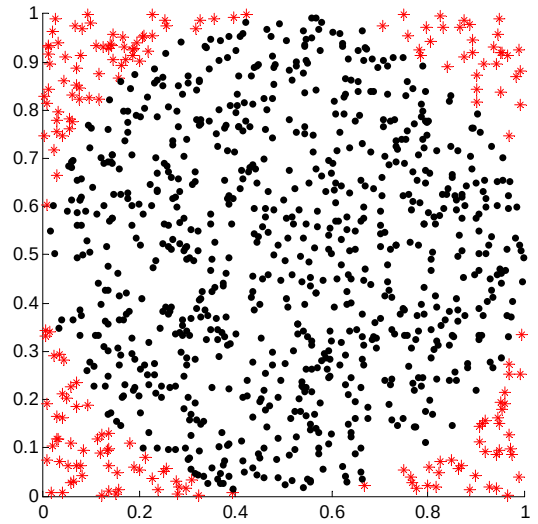
Slika 3. Određivanje broja Pi slučajnim odabirom n=1000 tačaka



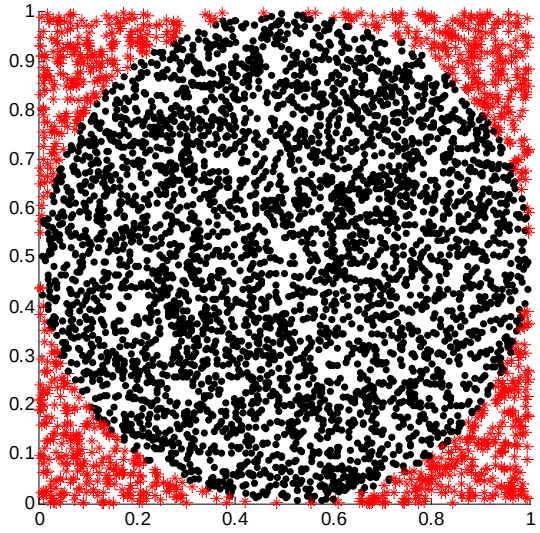
Slika 4. Određivanje broja Pi slučajnim odabirom n=10000 tačaka



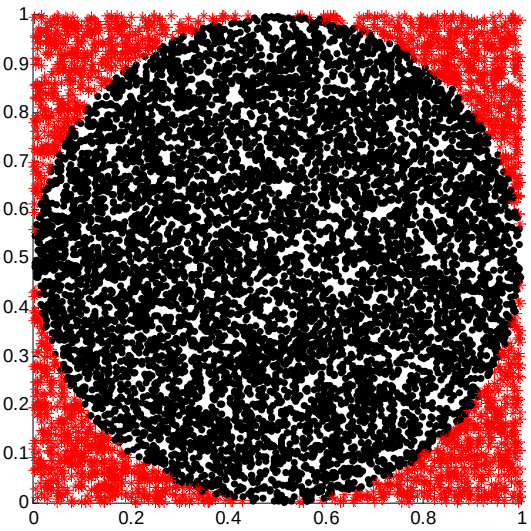
Slika 5. Ilustracija ulaznog skupa slučajnim odabirom n=500 tačaka



Slika 6. Ilustracija ulaznog skupa slučajnim odabirom n=1000 tačaka



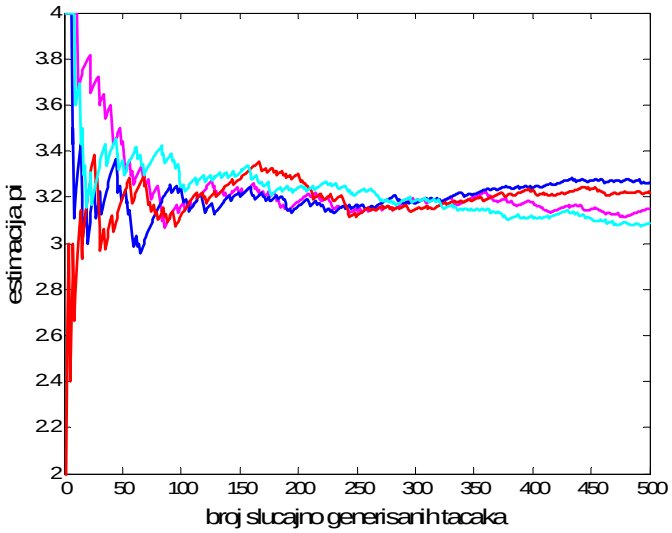
Slika 7. Ilustracija ulaznog skupa slučajnim odabirom n=5000 tačaka



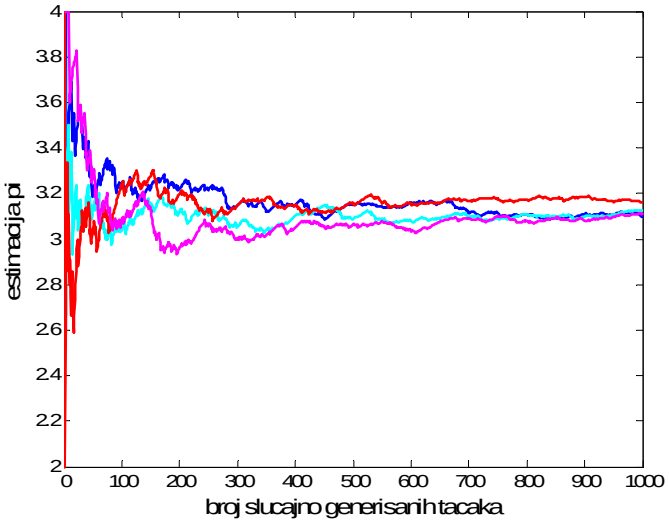
Slika 8. Ilustracija ulaznog skupa slučajnim odabirom n=10000 tačaka

Može se zaključiti da bi dalje povećanje broja slučajno generisanih tačaka povećalo kvalitet i tačnost estimacije, ali i vreme izvršenja simulacije.

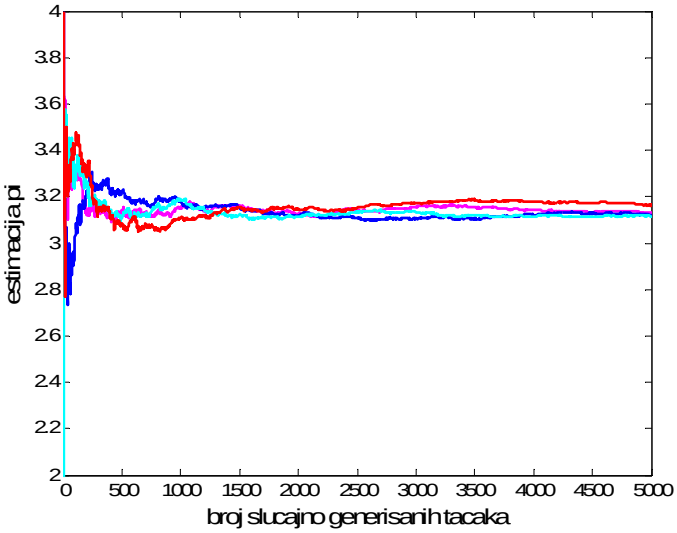
U cilju analize tačnosti Monte Carlo metode, treba uzeti u obzir da različita pokretanja ove metode daju različite rezultate pri aproksimaciji određenog izraza, usled čega postoji stohastička greška. Povećanje (srednje) tačnosti grube Monte Carlo ocene za jednu cifru (tj. smanjenje njene standardne devijacije za faktor 0.1) zahteva povećanje broja simulacija 100 puta, što može u nekim sutuacijama značajno povećati vreme izvršenja simulacije. Generisanje većeg broja podataka kroz nezavisne eksperimente, pri istom broju slučajno generisanih tačaka, kao i rezultat njihovog usrednjavanja, takođe rezultuje boljim kvalitetom estimacije. Realizacija kroz četiri nezavisna eksperimenta, za određenu fiksnu vrednost broja generisanih tačaka, ilustrovana je na Sl.9. Zatim je isti postupak sproveden i sa većim brojem generisanih tačaka, što je ilustrovano na Sl.10. Sl.11. i Sl.12., respektivno.



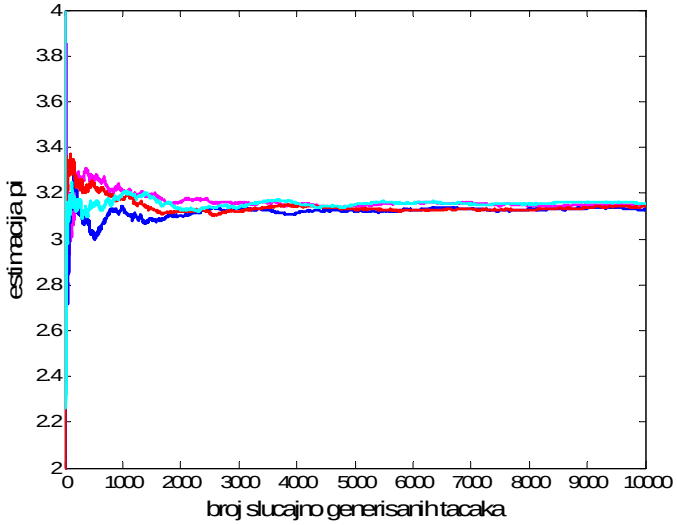
Slika 9. Estimacija Pi nakon slučajnog odabira *n*=500 tačaka



Slika 10. Estimacija Pi nakon slučajnog odabira *n*=1000 tačaka



Slika 11. Estimacija Pi nakon slučajnog odabira *n*=5000 tačaka

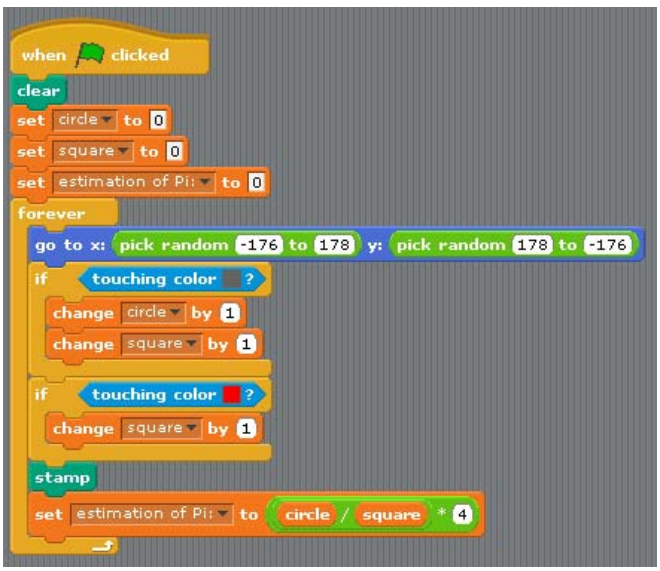


Slika 12. Estimacija Pi nakon slučajnog odabira *n*=10000 tačaka

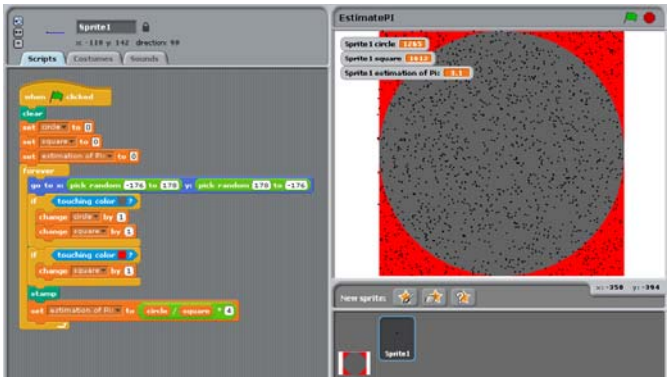
IV. KREIRANJE ANIMACIJE U PROGRAMABILNOM ALATU SCRATCH

Animacija kojom je ilustrovan uticaj povećanja broja slučajno generisanih tačaka na kvalitet estimacije, urađena je u Scratch programabilnom alatu, koji omogućava vrlo efikasnu primenu programskih blokova umesto pisanja programskog koda [16]-[18]. Iz tih razloga ovakva animacija pogodna je za primenu u nastavi, jer se ne zahteva poznavanje sintakse nekog konkretnog programskog jezika.

Osim što je besplatan, Scratch programabilni alat ima vrlo jednostavan i intuitivan korisnički interfejs [17], koji omogućava da se programiranje svede na prevlačenje i uklapanje blokova, kao što je prikazano na Sl.13 i Sl.14. Postoje blokovi za sve značajne programske strukture koje se mogu sresti u svakom savremenom programskom jeziku, a postoje i blokovi za napredne koncepte poput događaja, poruka, delegata.



Slika 13. Prikaz algoritma realizovanog u Scratch programabilnom alatu



Slika 14. Prikaz animacije realizovane u Scratch programabilnom alatu

ZAKLJUČAK

U ovom radu prezentovana je primena Monte Carlo metoda za izračunavanje približne vrenosti broja Pi, koristeći jednostavan princip aproksimacije. Rezultati dobijeni izvršavanjem relativno jednostavnih računarskih programa pisanih u različitim programskim okruženjima, daju dobru aproksimaciju za velik broj realizovanih eksperimenata, što može biti vremenski vrlo zahtevno. Implementacije koje uključuju generisanje slučajnih brojeva, crtanje grafikona, usrednjavanje rezultata po nezavisnim eksperimentima, kao i animaciju, izvedene su u MATLAB, Scratch i Mathematica programskim okruženjima.

LITERATURA

[1] G.S. Fishman, Monte Carlo: Concepts, Algorithms, and Applications, Knoxville, New York: Springer, 1995.

[2] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, CBMS-NSF eISBN: 978-1-61197-008-1, Philadelphia, 1992.

[3] R.Y. Rubinstein, D.P. Kroese, Simulation and the Monte Carlo Method, 2nd Ed, New York: John Wiley & Sons, 2007.

[4] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, 1991.

[5] A.B. Berg, Markov Chain Monte Carlo Simulations and Their Statistical Analysis, Hackensack, NJ: World Scientific, 2004.

[6] R.E. Caflisch, "Monte Carlo and quasi-Monte Carlo methods", Acta Numerica, No.**7**., Cambridge University Press, 1998, pp. 1–49.

[7] H. Stark, J.W. Woods, Probability and Random Processes with Applications to Signal Processing, Prentice Hall PTR, 2002.

[8] M.K. Simon, Probability Distributions Involving Gaussian Random Variables, Kluwer, 2002.

[9] W.H. Tranter, K.S. Shanmugan. T.S. Rappaport, K.L Kosbar, Principles of Communication Systems Simulation with Wireless Applications, Prentice Hall, Upper Saddle River, NJ, 2004.

[10] J. Proakis, M. Salehi, G. Bauch, Contemporary Communication Systems Using MATLAB, 3rd Ed, Global Engineering: C.M. Shortt, 2013.

[11] D.V Chudnovsky, G.V. Chudnovsky, "Computation and Arithmetic Nature of Classical Constants", IBM Research Report, IBM T. J. Watson Research Center, RC14950 (66818), 1989.

[12] E.W. Weisstein, "Pi Approximations", From *MathWorld*--A Wolfram Web Resource: http://mathworld.wolfram.com/PiApproximations.html

[13] http://www.numberworld.org/misc_runs/pi-12t/

[14] http://reference.wolfram.com/language/ref/RandomReal.html

[15] https://www.mathworks.com/help/matlab/random-number-generation.html

[16] S. Strbac-Savic, A. Miletic, H. Stefanovic, "The estimation of Pi using Monte Carlo technique with interactive animations", Int. Conf. Science and Higher Education in Function of Sustainable Development-SED 2015, pp. 2-15 – 2-20.

[17] http://llk.media.mit.edu

[18] http://scratch.mit.edu

ABSTRACT

In this paper some basic concepts of the Monte Carlo technique for estimating the value of a parameter are given, and one simple Monte Carlo estimator is proposed, in order to approximate the value of Pi. The random experiment is defined as taking random samples over the bounding box, after defining the event of interest, while the estimation is based on different numbers of replications of the underlying experiment. The uniform random number generator included in the MATLAB library is used to generate random points in a square around a circle of radius 1, while the points falling within the inscribed circle are counted. The probability of data point landing within the circle presents a ratio of the circle's area to the area of the square, which is used for estimation the value of Pi. Some results are obtained using functions available in Mathematica, while some interactive animations, realized using Scratch programming language, are also provided.

PI APPROXIMATION USING MONTE CARLO METHOD WITH INTERACTIVE ANIMATIONS REALIZED IN DIFFERENT PROGRAMMING ENVIRONMENTS

Ana Savic, Goran Bjelobaba, Hana Stefanovic