Web aplikacija za računanje Cauchyjevog problema Eulerovom metodom

Čorak, Antonio

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:955642

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-06**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Stručni studij računarstva

WEB APLIKACIJA ZA RAČUNANJE CAUCHYJEVOG PROBLEMA EULEROVOM METODOM

Završni rad

Antonio Čorak

Osijek, 2022.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Osijek, 15.09.2022.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Ime i prezime Pristupnika:	Antonio Čorak
Studij, smjer:	Preddiplomski stručni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	AR4656, 25.07.2018.
OIB Pristupnika:	02820620401
Mentor:	Ivan Hrehorović, prof.
Sumentor:	izv. prof. dr.sc. Josip Job
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Tomislav Rudec
Član Povjerenstva 1:	Ivan Hrehorović, prof.
Član Povjerenstva 2:	Anja Šteko, mag. educ. math. et inf.
Naslov završnog rada:	Web aplikacija za računanje Cauchyjevog problema Eulerovom metodom
Znanstvena grana završnog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada	Opisati Cauchyjev problem i objasniti način njegovog rješavanja preko Eulerove metode. Napraviti i objasniti web aplikaciju koja će rješavati Cauchyjev problem Eulerovom metodom. Sumentor s FERIT-a:Josip, Job
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	15.09.2022.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 02.10.2022.

Ime i prezime studenta:	Antonio Čorak
Studij:	Preddiplomski stručni studij Računarstvo
Mat. br. studenta, godina upisa:	AR4656, 25.07.2018.
Turnitin podudaranje [%]:	11

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za računanje Cauchyjevog problema Eulerovom metodom**

izrađen pod vodstvom mentora Ivan Hrehorović, prof.

i sumentora izv. prof. dr.sc. Josip Job

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA

o odobrenju za pohranu i objavu ocjenskog rada

kojom ja Antonio Čorak, OIB: 02820620401, student/ica Fakulteta elektrotehnike,
računarstva i informacijskih tehnologija Osijek na studiju Preddiplomski stručni studij Računarstvo, kao autor/ica ocjenskog rada pod naslovom: Web aplikacija za računanje Cauchyjevog problema Eulerovom metodom**,**
dajem odobrenje da se, bez naknade, trajno pohrani moj ocjenski rad u javno dostupnom digitalnom repozitoriju ustanove Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek i Sveučilišta te u javnoj internetskoj bazi radova Nacionalne i sveučilišne knjižnice u Zagrebu, sukladno obvezi iz odredbe članka 83. stavka 11. *Zakona o znanstvenoj djelatnosti i visokom obrazovanju* (NN 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).
Potvrđujem da je za pohranu dostavljena završna verzija obranjenog i dovršenog ocjenskog rada. Ovom izjavom, kao autor/ica ocjenskog rada dajem odobrenje i da se moj ocjenski rad, bez naknade, trajno javno objavi i besplatno učini dostupnim:

a) široj javnosti

b) studentima/icama i djelatnicima/ama ustanove

c) široj javnosti, ali nakon proteka 6 / 12 / 24 mjeseci (zaokružite odgovarajući broj mjeseci).

**U slučaju potrebe dodatnog ograničavanja pristupa Vašem ocjenskom radu, podnosi se obrazloženi zahtjev nadležnom tijelu Ustanove.*

Osijek, 02.10.2022. _____

(mjesto i datum) (vlastoručni potpis studenta/ice)

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. EULEROVA METODA 2

2.1. Algoritam za Eulerovu metodu 3

2.2. Primjeri rješavanja Cauchyjevog problema Eulerovom metodom 3

2.2.1. Primjer 1. 3

2.2.2. Primjer 2. 5

2.2.3. Primjer 3. 6

2.3. Greške Eulerove metode 6

3. KORIŠTENE TEHNOLOGIJE 7

3.1. HTML 7

3.2. CSS 7

3.3. JavaScript 8

4. OPIS WEB APLIKACIJE 9

4.1. Struktura aplikacije 9

4.2. Izgled web aplikacije 9

4.3. Performanse algoritma 17

4.4. Primjer korištenja web aplikacije 17

5. ZAKLJUČAK 20

6. LITERATURA 21

7. SAŽETAK 22

8. ABSTRACT 22

9. PRILOG 23

1. UVOD

Obične diferencijalne jednadžbe je u nekim situacijama lagano za riješiti jer se nekad rješenje može izraziti pomoću poznatih funkcija, dok neke jednadžbe ne možemo samo tako lagano riješiti te ne možemo naći egzaktno rješenje za takve jednadžbe. Iz tih razloga koristimo numeričke metode za rješavanje diferencijalnih jednadžba koje daju približna rješenja tih jednadžbi. Jedna od metoda za rješavanje je i Eulerova metoda.

Ovim završnim radom se želi prikazati rješavanje zadataka sa Cauchyjevim ili inicijalnim problemom Eulerovom metodom. Pri tome rad obuhvaća teorijsku osnovu Eulerove metode te njegovu primjenu pri rješavanja zadataka sa Cauchyjevim ili inicijalnim problemom. Također ovim radom je u praktičnom dijelu prikazana izrada i korištenje web aplikacije za rješavanje Cauchyjevog problema Eulerovom metodom. Web tehnologije koje se koriste pri izradi web aplikacije su: JavaScript, HTML i CSS. Ovaj rad je strukturiran na više poglavlja. Poglavlje „Eulerova metoda“ opisuje teorijsku osnovu Eulerove metode te primjere rješavanja zadataka, poglavlje „Korištene tehnologije“ opisuje tehnologije i alate koji su korišteni pri izradi web aplikacije te poglavlje „Opis web aplikacije“ daje uvid u strukturu i izradu web aplikacije.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je istražiti i opisati rješavanje Cauchyjevog problema koristeći Eulerovu metodu te izraditi programsko rješenje u obliku web aplikacije za rješavanje takvih tipova zadataka.

2. EULEROVA METODA

Eulerova metoda je poznata kao najjednostavnija metoda za rješavanje Cauchyjevog problema za obične diferencijalne jednadžbe oblika

$$y' = f(x, y), \quad y(x_0) = y_0 \tag{1-1}$$

gdje je $y(x_0) = y_0$ početni uvjet odnosno inicijalni ili Cauchyjev problem, a x_0 i y_0 inicijalne ili početne vrijednosti za x i y.[1] Autor ove metode je Leonhard Euler koji ju je publicirao 1768. Naš cilj je zapravo pomoću ove metode odrediti (približno) nepoznatu funkciju $y(x)$ za $x \geq x_0$. U zadacima nam je eksplicitno zadan $y(x_0)$, te koristeći jednadžbu (1-1) možemo odrediti brzinu promjene y u točki x_0:

$$y'(x_0) = f\big(x_0, y(x_0)\big) = f(x_0, y_0)$$

Ako bi brzina promjene $y(x)$ ostala $f(x_0, y_0)$ za sve točke x, tada bi $y(x)$ točno glasio $y_0 + f(x_0, y_0)(x - x_0)$. Međutim brzina promjene $y(x)$ ne ostaje $f(x_0, y_0)$ za sve točke x, ali očekujemo da će ostati blizu $f(x_0, y_0)$ za vrijednost x blizu x_0. U tom slučaju bi vrijednost $y(x)$ ostala blizu $y_0 + f(x_0, y_0)(x - x_0)$ za x blizu x_0, za mali broj h :

$$x_1 = x_0 + h$$

$$y_1 = y_0 + f(x_0, y_0)(x_1 - x_0)$$

$$y_1 = y_0 + hf(x_0, y_0)$$

gdje je $h = x_1 - x_0$ koji se naziva veličina koraka. Možemo reći da je $y(x_1) \approx y_1$. Ponavljajući ove korake dolazimo do:

$$x_2 = x_1 + h$$

$$y_2 = y_1 + f(x_1, y_1)(x_2 - x_1)$$

$$y_2 = y_1 + hf(x_1, y_1)$$

gdje možemo reći da je $y(x_2) \approx y_2$ itd. Ako definiramo n= 0,1,2,3,4… onda imamo

$$x_{n+1} = x_n + h$$

$$y(x_n) = y_n + f(x_n, y_n)(x - x_n)$$

$$y(x_{n+1}) \approx y_{n+1} = y_n + hf(x_n, y_n)$$

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{1-2}$$

Jednadžba (1-2) se naziva Eulerova metoda te iz nje možemo dobiti:

$$\frac{y_{n+1}-y_n}{h} = f(x_n, y_n),\ n = 0,1,2,3 \ldots$$

Dobivene vrijednosti y_n su aproksimacije rješenja diferencijalne jednadžbe u točkama x_n. [2]

2.1. Algoritam za Eulerovu metodu

Tipični koraci za rješavanje problema Eulerovom metodom su:

Korak 1: definirati $f(x_n, y_n)$

Korak 2: definirati ulazne inicijalne ili početne varijable x_0 i y_0

Korak 3: definirati veličinu koraka h i broj koraka n

Korak 4: izračunati x i y :

for n = 1 : N-1

$x_{n+1} = x_n + h$

$y_{n+1} = y_n + hf(x_n, y_n)$

Korak 5: izlaz je x_{n+1} i y_{n+1}

2.2. Primjeri rješavanja Cauchyjevog problema Eulerovom metodom

Slijedi par primjera rješavanja Cauchyjevog (inicijalnog) problema Eulerovom metodom.

2.2.1. Primjer 1.

Zadana je jednadžba $y' = y - x$, riješite ju koristeći se Eulerovom metodom ako su zadane inicijalne vrijednosti $y(0) = 2$, sa veličinom koraka $h = 0.1$ i $h = 0.5$ i gdje je $x \in [0,1]$. Egzaktno rješenje je $y(x) = e^x + x + 1$.

Rješenje:

za $n = 0,\ x_1 = x_0 + h => x_1 = 0 + 0.1 = 0.1$

$y(x_1) = y(0.1) = y_1 = y_0 + h \cdot f(x_0, y_0) =$

$= 2 + 0.1 \cdot f(0,2) = 2 + 0.1 \cdot 2 = 2.2$

za $n = 1,\ x_2 = x_1 + h => x_2 = 0.1 + 0.1 = 0.2$

$y\ (x_2) = y(0.2) = y_2 = y_1 + h \cdot f(x_1, y_1) =$

$= 2.2 + 0.1 \cdot f(0.1,2.2) = 2.2 + 0.1 \cdot 2.1 = 2.41$

…

Tablica vrijednosti za 10 iteracija izgleda ovako:

n	x_n	y_n	$y' = \frac{dy}{dx}$	Egzaktno rješenje: $y(x) = e^x + x + 1$
0	0	2	2	2
1	0.1	2.2	2.1	2.2051709
2	0.2	2.41	2.21	2.4214028
3	0.3	2.631	2.331	2.6498588
4	0.4	2.8641	2.4641	2.8918247
5	0.5	3.11051	2.61051	3.1487213
6	0.6	3.371561	2.7716561	3.4221188
7	0.7	3.648717	2.948717	3.7137527
8	0.8	3.943589	3.143589	4.0255409
9	0.9	4.257948	3.356948	4.3596031

Tablica 1. Rješenje jednadžbe $y' = y - x$ Eulerovom metodom za $h = 0.1$

n	x_n	y_n	$y' = \frac{dy}{dx}$	Egzaktno rješenje: $y(x) = e^x + x + 1$
0	0	2	2	2
1	0.5	3	2.5	3.1487213
2	1	4.25	3.25	4.7182818

Tablica 2. Rješenje jednadžbe $y' = y - x$ Eulerovom metodom za $h = 0.5$

2.2.2. Primjer 2.

Zadana je jednadžba $5y' - y^2 = -x^2$, riješite ju koristeći se Eulerovom metodom ako je zadan inicijalni problem $y(0) = 1$, sa veličinom koraka $h = 0.5$ i gdje je $x \in [0,3]$.

Rješenje:

za $n = 0,\ x_1 = x_0 + h => x_1 = 0 + 0.5 = 0.5$

$y(x_1) = y(0.5) = y_1 = y_0 + h \cdot f(x_0, y_0) =$

$= 1 + 0.5 \cdot f(0,1) = 1 + 0.5 \cdot 0.2 = 1.1$

za $n = 1,\ x_2 = x_1 + h => x_2 = 0.5 + 0.5 = 1$

$y(x_2) = y(1) = y_2 = y_1 + h \cdot f(x_1, y_1) =$

$= 1.1 + 0.5 \cdot f(0.5,1.1) = 1.1 + 0.5 \cdot 0{,}192 = 1.196$

…

Tablica vrijednosti za 6 iteracija izgleda ovako:

n	x_n	y_n	$y' = \frac{dy}{dx}$
0	0	1	0.2
1	0.5	1.1	0.192
2	1	1.196	0.086083
3	1.5	1.239042	-0.142955
4	2	1.167564	-0.527359
5	2.5	0.903885	-1.086599
6	3	0.360585	-1.773996

Tablica 3. Rješavanje jednadžbe $5y' - y^2 = -x^2$ Eulerovom metodom

2.2.3. Primjer 3.

Zadana je jednadžba $y' = y + x + 1$. Koristeći se Eulerovom metodom riješite $y(1.3)$, ako su zadane inicijalne vrijednosti $y(1) = 2$, sa veličinom koraka $h = 0.1$.

Rješenje:

za $n = 0,\ x_1 = x_0 + h => x_1 = 1 + 0.1 = 1.1$

$y(x_1) = y(1.1) = y_1 = y_0 + h \cdot f(x_0, y_0) =$

$= 2 + 0.1 \cdot f(1, 2) = 2 + 0.1 \cdot 4 = 2.4$

za $n = 1,\ x_2 = x_1 + h => x_2 = 1.1 + 0.1 = 1.2$

$y(x_2) = y(1.2) = y_2 = y_1 + h \cdot f(x_1, y_1) =$

$= 2.4 + 0.1 \cdot f(1.1, 2.4) = 2.\ 4 + 0.1 \cdot 4.5 = 2.85$

za $n = 3,\ x_3 = x_2 + h => x_3 = 1.2 + 0.1 = 1.3$

$y(x_3) = y(1.3) = y_3 = y_2 + h \cdot f(x_2, y_2) =$

$= 2.85 + 0.1 \cdot f(1.2,2.85) = 2.85 + 0.1 \cdot 5.05 = 3.355$

$y(1.3) \approx 3.355$

2.3. Greške Eulerove metode

Prilikom rješavanja jednadžbi pomoću Eulerove metode javlja se dva izvora grešaka:

- Greške odbacivanja: - posljedice su činjenice da su formule koje definiraju Eulerovu metodu temeljene na nekakvoj vrsti aproksimacije

-Greške zaokruživanja: - posljedice su korištenja aritmetike računala umjesto egzaktne aritmetike. Uzrok ovakvih grešaka je neegzaktnost prikaza realnog broja u računalu i aritmetičkih operacija među realnim brojevima.[3]

Rješavanjem primjera 2.2.1 možemo zaključiti da preciznost Eulerove metode možemo povećati smanjenjem koraka h.

3. KORIŠTENE TEHNOLOGIJE

Web aplikacija za rješavanje Cauchyjevog problema Eulerovom metodom je napravljena koristeći se jezicima: HTML, CSS, JavaScript. HTML i CSS se koristi za estetični izgled web aplikacije, dok se JavaScript koristi pri implementaciji algoritma za rješavanje Cauchyjevog problema Eulerovom metodom.

3.1. HTML

HTML je kratica za HyperText Markup Language, što prevedeno sa engleskog jezika znači prezentacijski jezik za kreiranje web stranica. HTML se danas koristi uglavnom za kreiranje „kostura“ web stranice tj. on omogućava kreiranje i strukturiranje sekcija, paragrafa i linkova koristeći HTML elemente kao što su znakovi (eng. *Tags)* koji se unutar HTML dokumenta uglavnom otvaraju sa <naziv_taga> i zatvaraju sa </naziv_taga>, npr. Ovo je nekakav tekst unutar *p* elementa .[4] Za prikaz elemenata korisniku odgovoran je web preglednik koji korisnik upotrebljava, tj. napisan HTML kod upućuje web preglednik kako da prikaže elemente krajnjem korisniku. Za ukrašavanje i estetiku HTML elemenata koristi se CSS standard koji će u nastavku biti detaljnije opisan. HTML stranice su statične, te da bi postale dinamične koristi se razne druge tehnologije poput JavaScript-a, PHP-a, AJAX-a itd.

3.2. CSS

CSS (eng. *Cascading Style Sheets)* je jezik koji specificira kako su dokumenti prezentirani krajnjim korisnicima, tj. kako su stilizirani, posloženi itd. Dokument predstavlja tekstualnu datoteku koja je strukturirana koristeći prezentacijski jezik kao što je HTML, SVG ili XML. CSS se može koristiti za jednostavnu stilizaciju teksta, npr. mijenjanje boje i veličine teksta, naslova itd. Može se koristiti za kreiranje rasporeda elemenata, kreiranje efekata poput animacija itd. CSS je jezik baziran na pravilima. Pravila se definiraju određivanjem grupa stilova koji bi se trebali primijeniti na određene elemente ili grupe elemenata na web stranici.[5] U donjem primjeru je prikazana sintaksa CSS koda.

```
p {
color: blue;
}
```

Slika 3.1. Primjer sintakse CSS koda

U prikazanom primjeru CSS pravilo se otvara sa selektorom koji odabire HTML element koji će se stilizirati. U ovom slučaju se stilizira paragraf (<p>). Nakon toga stavljaju se vitičaste zagrade {}. Unutar tih zagrada može biti jedna ili više deklaracija koji sadrže svojstvo koje se mijenja i vrijednost koja se pripisuje tom svojstvu. U ovom slučaju svojstvu boja (eng. *color)* je dodijeljena vrijednost plava (eng. *blue*). Svojstvo i vrijednost svojstva je podijeljeno sa dvotočkom. Ovim kodom svim paragrafima koje sadrže tekst mijenja se boja teksta u plavu boju.

3.3. JavaScript

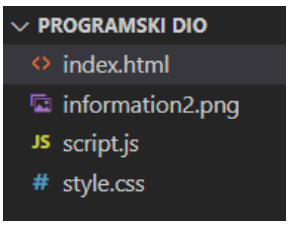
JavaScript je programski jezik koji omogućava implementaciju kompleksnih značajki u web stranicu. Namijenjen je ponajprije razvoju interaktivnih HTML stranica. JavaScript omogućava kreiranje dinamičkog ažuriranja sadržaja, kontrolu multimedije na stranici, animaciju slika itd. Također pomoću JavaScripta može se promijeniti i HTML kod i CSS kod, te može se koristiti pri računanju, manipulaciji te potvrdi podataka. Izvršavanje JavaScript koda se odvija unutar web preglednika korisnika te kodu nije potreban server. JavaScript se dodaje u web stranicu koristeći <script> element u HTML-u, te unutar njega se unosi JavaScript kod.[6]

4. OPIS WEB APLIKACIJE

U ovom poglavlju opisati će se struktura, način izvedbe programa i izgled same web aplikacije.

4.1. Struktura aplikacije

Struktura datoteka ove web aplikacije prikazana je na slici 4.1. Najvažnije datoteke su index.html koja prikazuje početnu stranicu kada se pristupi aplikaciji, script.js koja predstavlja JavaScript kod te style.css datoteka koja služi za stiliziranje HTML elemenata iz index.html datoteke.



Slika 4.1. Prikaz strukture web aplikacije

4.2. Izgled web aplikacije

Početna stranica web aplikacije prikazana je slikom 4.2. Glavni zadatak ove web stranice je omogućiti krajnjem korisniku rješavanje Cauchyjevog problema Eulerovom metodom na što jednostavniji način. Korisnik kada pristupi web stranici, unosi podatke potrebne za izračun na mjestima gdje to predviđeno. Ukoliko su korisniku potrebne upute može ih dobiti klikom na ikonu za informacije u desnom dijelu stranice.

Slika 3.3. Početna stranica za rješavanje Cauchyjevog problema Eulerovom metodom

Korisnik prvo unosi podatke za početne uvijete x_0 i y_0, nakon toga unosi funkciju $f(x, y)$ te duljinu koraka h. Na kraju unosi traženi uvjet koji se računa. Na slici 4.4. je prikazan kod koji prihvaća unesene podatke korisnika. Za input odnosno unos podataka u web aplikaciju korišten je HTML element „<input>“ koji je tipa „text“ za unos funkcije $f(x, y)$ te tipa „number“ za unos početnog uvjeta, veličinu koraka h i traženog uvjeta.

```
<p>Unesite početni uvjet za $x_0$ i $y_0$: (<input id="xi" type="number"
class="small" value=1 step=".1">,
<input id="yi" type="number" class="small" value=1 step=".1" >)</p>
<p>Unesite funkciju: $\frac{dy}{dx} = y' = f(x,y)$: <input id="dydx"
type="text" value="x+y" ></p>
<p id="function_error"></p>
<p>Duljina koraka $h$ ili $&Delta;x$: <input id="h" type="number"
class="small" value=0.1 step=".01" ></p>
<p id="last">Za $y($<input id="zadanUvijet" type="number" class="small"
value="0" step=".01"> $) =$ <span id="konacno_rjesenje">?</span></p>
<button class="btn btn-success" id="update">Izračunaj</button>
```

Slika 4.4. Prihvaćanje unesenih podataka korisnika

Nakon što korisnik unese podatke i pritisne gumb „Izračunaj“, JavaScript dohvaća unesene podatke preko „Document“ metode „getElementById()“ i sprema ih u varijable. Ovaj dio koda je prikazan slikom 4.5.

```
var tbody = document.getElementById('tbody'),
    inputs_for_main_table = {
        xi: document.getElementById('xi'),
        yi: document.getElementById('yi'),
        dydx: document.getElementById('dydx'),
        h: document.getElementById('h'),},
    inputs_for_steps_tab = {
        xi: document.getElementById('xi'),
        yi: document.getElementById('yi'),
        dydx: document.getElementById('dydx'),
        h: document.getElementById('h'),},
    zadanUvijet = document.getElementById('zadanUvijet'),
```

Slika 4.5. Dohvaćanje elemenata u JavaScriptu

Nakon dohvaćanja elemenata iz HTML-a, provjerava se vjerodostojnost unesenih podataka. Naime ako je duljina koraka veća od 0 onda traženi uvjet mora biti veći od x_0 u početnom uvjetu i obratno, ako je duljina koraka manja od 0 onda traženi uvjet mora biti manji od x_0 u početnom uvjetu. U ovim slučajevima rezultat za traženi uvjet je nepostojeći. Stoga se korisniku prikazuje greška sa odgovarajućom porukom. Provjera vjerodostojnosti podataka je prikazana slikom 4.6.

```
if(parseFloat(inputs_for_main_table.h.value)> 0 &&
      parseFloat(inputs_for_main_table.xi.value) >
      parseFloat(zadanUvijet.value) ){

      main_errors.innerHTML = 'Ako je duljina koraka veća od 0 onda
      traženi uvjet mora biti veći od x u početnom uvjetu!!'

      main_errors.style.display = "block";

      table.style.display = "none";

      table2.style.display = "none";

      koraci_content.style.display = "none";

      function_error.style.display = "none";

}

else if(parseFloat(inputs_for_main_table.h.value)< 0 &&
      parseFloat(inputs_for_main_table.xi.value) <
      parseFloat(zadanUvijet.value) ){

      main_errors.innerHTML = 'Ako je duljina koraka manja od 0 onda
      traženi uvjet mora biti manji od x u početnom uvjetu!!'

      main_errors.style.display = "block";

      table.style.display = "none";

      table2.style.display = "none";

      koraci_content.style.display = "none";

      function_error.style.display = "none";

}
```

Slika 4.6. Provjera vjerodostojnosti unesenih podataka

Nakon toga dolazi provjera unesenih podataka za funkciju $f(x, y)$ gdje aplikacija dopušta samo unos sljedećih znakova: x, y, e, +, -, *, /, te brojeve od 0-9. Ukoliko korisnik unese bilo koji drugi znak koji ne pripada skupini dopuštenih znakova prikazuje se greška sa odgovarajućom porukom. Za samu provjeru korišteni su regularni izrazi (regex) koji služe za pretraživanje niza znakova koji su uneseni od strane korisnika. Provjera unesenih podataka za funkciju $f(x, y)$ je prikazana slikom 4.7.

```
else if (/[^()xye\d+\-*/^]$/.test(inputs_for_main_table.dydx.value)) {
      function_error.innerHTML = 'Neispravno unesena funkcija' +
            katex.renderToString('f(x,y)')+ '. Pokušajte ponovo! ';
      function_error.style.display = "block";
      table.style.display = "none";
      table2.style.display = "none";
      koraci_content.style.display = "none";
      main_errors.style.display = "none";
}
```

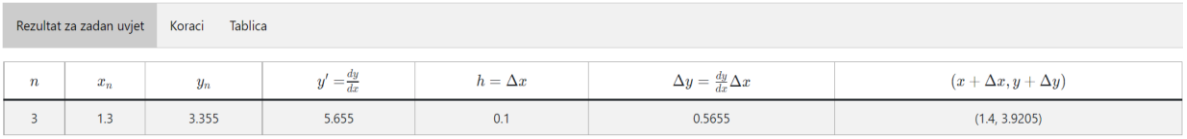
Slika 4.7. Provjera unesenih podataka za funkciju $f(x, y)$

Za uljepšan prikaz matematičkih oznaka u ovoj web aplikaciji korišten je „*Katex Library*“. Na slici 4.8. je prikazana njegova implementacija.

```
var rjesenjefunkcija = document.getElementById('rjesenjefunkcija') ;
rjesenjefunkcija.innerHTML =
katex.renderToString(inputs_for_main_table.dydx.value);
```

Slika 4.8. Korištenje „*Katex*“ biblioteke

Nakon svih provjera izračunavaju se rezultati i ispisuju se unutar tri kartice (eng. tab). Prva kartica se naziva *„Rezultat za tražen uvjet“* koja prikazuje samo rezultate za zadan uvjet u obliku tablice. Kartica postaje aktivna čim se pritisne gumb „*Izračunaj*“. Izgled kartice je prikazan slikom 4.9.



Slika 4.9. Kartica *„Rezultat za zadan uvjet“*

Kod koji predstavlja ispis rezultata u karticu *„Rezultat za tražen uvjet“* je prikazan na slici 4.10. Kod radi tako da uzima elemente iz specifičnog reda (ovisno o traženom uvjetu) iz tablice koja se nalazi u kartici „*Tablica*“ te ih kopira u elemente tablice koja se nalazi u kartici *„Rezultat za traženi uvjet“* i ispisuje na ekran korisniku.

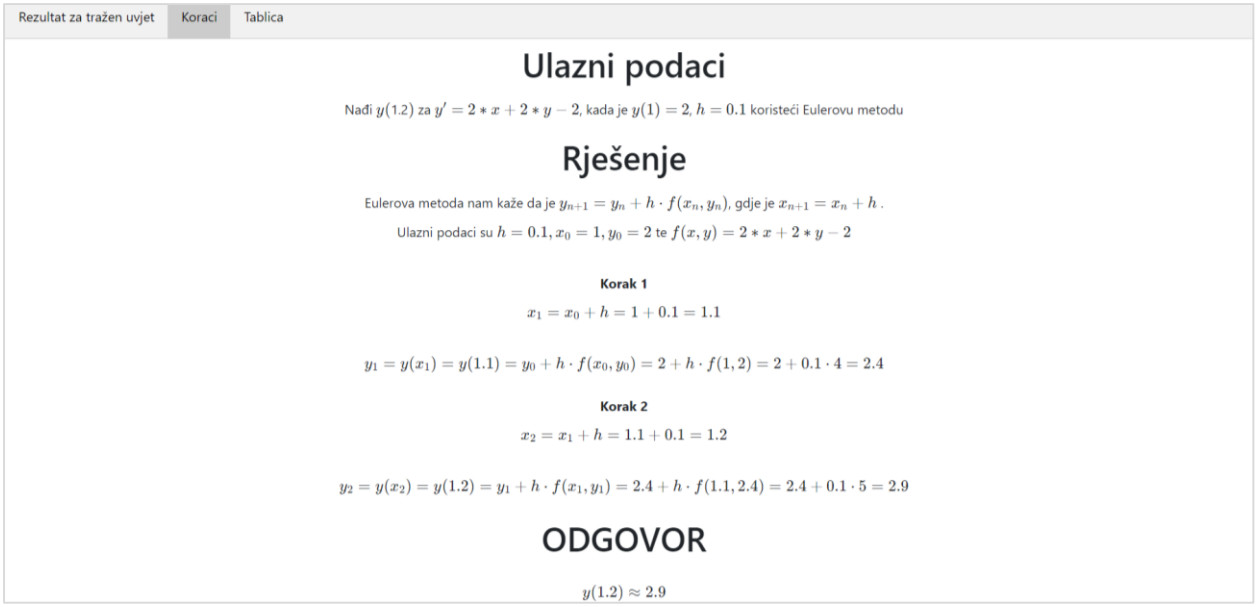
```
for (i = 0; i < 1; i++) {
      var tr = document.createElement('tr');
      var tds = [];
      for (j = 0; j < 7; j++) tds[j] = document.createElement('td');
      tds[0].textContent = tds2[0].innerText ;
      tds[1].textContent =  tds2[1].innerText;
      tds[2].textContent =  tds2[2].innerText;
      tds[3].textContent = tds2[3].innerText;
      tds[4].textContent = tds2[4].innerText;
      tds[5].textContent = tds2[5].innerText;
      tds[6].textContent = tds2[6].innerText;
      for (k = 0; k < 7; k++) tr.appendChild(tds[k]);
      tbody2.appendChild(tr);
}
```

Slika 4.10. Ispis rezultata u kartici *„Rezultat za zadan uvjet“*

Druga kartica koja se naziva *„Koraci“* sadrži detaljno rješenje za određeni zadan uvjet. Rješenje je podijeljeno po koracima. Broj koraka se računa tako da se od zadanog uvjeta oduzme početni x_0 te se ta razlika podijeli sa duljinom koraka h. Nakon toga se radi izračun i ispis rješenja za pojedini korak. Kod za ispis i izračun rješenja za pojedini korak se nalazi na slici 4.11. Prvo se računa broj potrebnih koraka, zatim se kroz svaki korak prolazi preko for petlje. Unutar for petlje se prvo računa y' tj. $f(x, y)$ pomoću JavaScript funkcije *„eval“* te se rezultat sprema u varijablu *„deriv3“*. Nakon toga se računa rezultat za prvi red pojedinog koraka tj. x_n i to tako da se zbroje x_{n-1} i h. Nadalje se računa rezultat za drugi red pojedinog koraka tj. y_n na način da se pomnože h i varijabla *„deriv3“* te se njihov umnožak zbroji sa y_{n-1}. Poslije toga se radi priprema za sljedeći korak. Množi se varijabla *„deriv3“* sa h te se rezultat sprema u varijablu dy koju zbrajamo sa trenutnim y_n da dobijemo y_{n+1}. Također zbrajamo h sa trenutnim x_n da dobijemo x_{n+1} za sljedeći korak. Ako se dođe do uvjeta u kojem gdje je traženi uvjet jednak trenutnom x onda se ispisuje konačno rješenje koje je aproksimacija egzaktnog rješenja te se prekida petlja. Kada se izvrši for petlja ispisuju se rezultati na ekran korisnika. Slika 4.12. prikazuje karticu *„Koraci“* nakon ispisa rješenja jednog primjera zadatka.

```
var punoRjesenje = document.getElementById('punoRjesenje');

var htmlElements = "";

var counter_tab = (zadanUvijet.value - pretty(scope2.x)) / h_tab;

for (i = 0; i < counter_tab; i++) {

      var deriv3 = func.eval(scope2);

      var first_row_result = pretty(scope2.x) + h_tab;

      var second_row_result = pretty(scope2.y) + h_tab * pretty(deriv3);

      //dio koda odgovoran za ispis rezultata u kartici „Koraci“

      htmlElements += '<strong id="tab_step_title">Korak'+(i+1)+'</strong>
      <p>'+katex.renderToString('x_'+(i+1)+'= x_'+i+'+ h
      ='+pretty(scope2.x+ '+'+h_tab+'='+pretty(first_row_result)+'')+'</p>
      <p>'+katex.renderToString('y_'+(i+1)+'= y(x_'+(i+1)+') =
      y('+pretty(first_row_result)+') = y_'+i+'+ h \\cdot
      f(x_'+i+',y_'+i+') = '+pretty(scope2.y)+'+ h \\cdot
      f('+pretty(scope2.x)+','+pretty(scope2.y)+') =
      '+pretty(scope2.y)+'+'+h_tab+'\\cdot'+pretty(deriv3)+'='+pretty(seco
      nd_row_result)+'')+'</p>';

      //dio koda odgovoran za računanje rješenja za pojedini korak

      var dy = pretty(deriv3) *  h_tab;

      scope2.x += parseFloat(h_tab);

      scope2.y += dy;

      if(zadanUvijet.value == pretty(scope2.x)){

            htmlElements += '<h1 id="tab_answer_title">ODGOVOR</h1>

            <p id="tab_answer">
            '+katex.renderToString('y('+pretty(first_row_result)+')

            \\approx '+pretty(second_row_result)+'')+'</p>'

            break;

       }

}

punoRjesenje.innerHTML = htmlElements;
```

Slika 4.11. Kod za ispis i izračun rješenja za pojedini korak u kartici „Koraci“



Slika 4.12. Kartica „*Koraci*“ nakon ispisa rješenja primjera zadatka

Posljednja kartica se naziva „*Tablica*“. Algoritam za ovu karticu je sličan algoritmu za karticu „*Koraci*“ samo je razlika u tome što su rješenja u tabličnom obliku zbog lakšeg pregleda. Kod je prikazan na slici 4.13. Algoritam počinje tako da se prvo odredi broj koraka kao i u kartici „*Koraci*“. Nakon toga se preko for petlje obrađuje korak po korak za unesene vrijednosti. Jedan korak zapravo predstavlja jedan red u tablici pri ispisu rezultata. U prvom stupcu tablice se nalazi broj koraka n. U drugom stupcu tablice se nalazi x_n koji u nultom koraku predstavlja početni uvjet za x te koji se za svaki novi korak poveća za duljinu koraka h. U trećem stupcu se nalazi y_n koji također kao x_n u nultom koraku predstavlja početni uvjet y te koji se za svaki novi korak poveća za Δy. U četvrtom stupcu se nalazi y' koji je rješenje za unesenu funkciju s obzirom na početni uvjet. U petom stupcu se nalazi duljina koraka odnosno h. U šestom stupcu se nalazi $\Delta y = h * y'$. U sedmom stupcu se nalazi par $(x + h, y + \Delta y)$. Slika 4.14. predstavlja karticu „*Tablica*“ nakon ispisa rješenja primjera zadatka.

```
//određivanje broja koraka
var counter = (zadanUvijet.value - pretty(scope.x)) / h_tab + 1;
//računanje i ispis rezultata u tablicu
for (i = 0; i <= counter; i++) {
      var tr = document.createElement('tr');
      var tds = [];
      for (j = 0; j < 7; j++) tds[j] = document.createElement('td');
      tds[0].textContent = i;
      tds[1].textContent =  pretty(scope.x);
      tds[2].textContent =  pretty(scope.y);
      var deriv2 = func.eval(scope);
      tds[3].textContent = pretty(deriv2);
      tds[4].textContent = seed_for_main_table.h;
      var dy = deriv2 * seed_for_main_table.h;
      tds[5].textContent = pretty(dy);
      tds[6].textContent = '('+pretty(scope.x+parseFloat(seed_for_
      main_table .h ) ) + ', ' + pretty(scope.y+dy) + ')';
      //ukoliko je traženi uvjet jednak x dodaj elemente tog koraka u
      tablicu za karticu „Rezultat za tražen uvijet"
      if(zadanUvijet.value == pretty(scope.x)){
            tds2 = tds;
       }
      for (k = 0; k < 7; k++) tr.appendChild(tds[k]);
      tbody.appendChild(tr);
      scope.x += parseFloat(seed_for_main_table.h);
      scope.y += dy;
}
```

Slika 4.13. Kod za ispis i izračun rješenja za pojedini korak u kartici „Tablica"

Rezultat za tražen uvjet | Koraci | Tablica

n	x_n	y_n	$y' = \frac{dy}{dx}$	$h = \Delta x$	$\Delta y = \frac{dy}{dx}\Delta x$	$(x + \Delta x, y + \Delta y)$
0	1	2	4	0.1	0.4	(1.1, 2.4)
1	1.1	2.4	4.5	0.1	0.45	(1.2, 2.85)
2	1.2	2.85	5.05	0.1	0.505	(1.3, 3.355)

Slika 4.14. Kartica *„Tablica"* nakon ispisa rješenja primjera zadatka

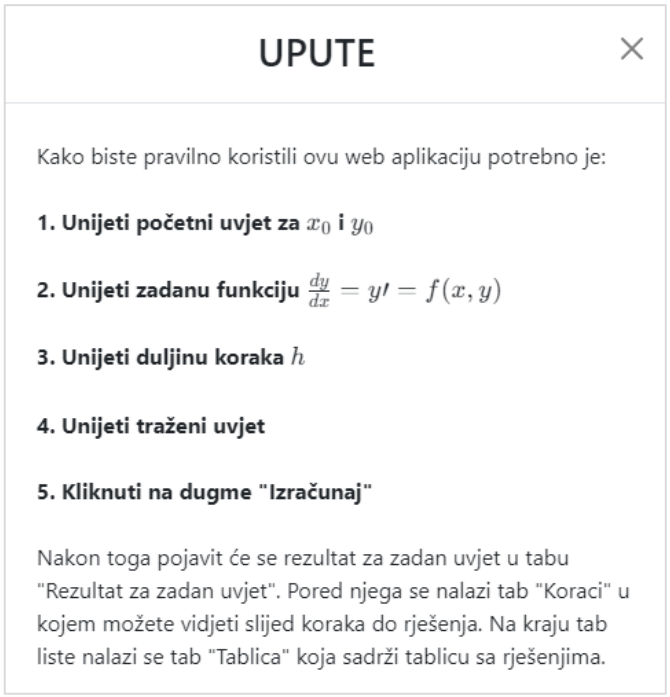
4.3. Performanse algoritma

Mjerenjem performansi algoritama se može odrediti koji je algoritam bolji i efikasniji. Performanse algoritma je uglavnom predstavljena pomoću veliko *O* notacije (eng. *Big O Notation*). Ona opisuje ograničavajuće ponašanje funkcije kada argument teži određenoj vrijednosti ili beskonačnosti, odnosno veliko *O* notacija opisuje kompleksnost koda koristeći algebarske izraze.[7] Algoritam za rješavanje Cauchyjevog problema Eulerovom metodom ima vremensku složenost *O*(n), jer Eulerova metoda nije jako precizna te kako bi rješenje bilo prihvatljivo mora se koristiti jako mala duljina koraka što automatski implicira da je potreban veći broj koraka za rješavanje tj. više vremena za izračun. Vrijeme raste linearno stoga je vremenska složenost *O*(n).

4.4. Primjer korištenja web aplikacije

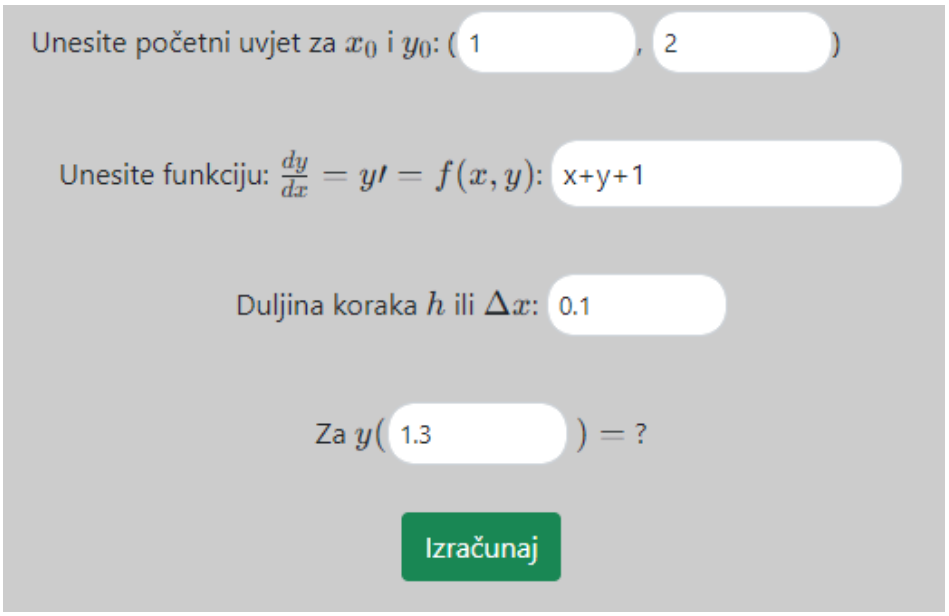
Zadana je jednadžba $y' = y + x + 1$. Koristeći se Eulerovom metodom riješite $y(1.3)$, ako su zadane inicijalne vrijednosti $y(1) = 2$, sa veličinom koraka $h = 0.1$.

Za novog korisnika je najprije potrebno pročitati upute koje se prikazuju klikom na ikonu za informacije koja se nalazi na desnoj strani web stranice. Upute su prikazane slikom 4.15.



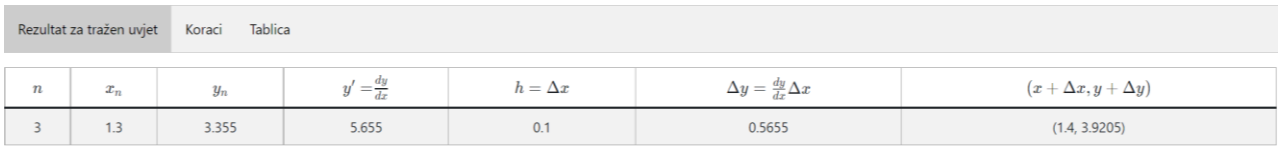
Slika 4.15. Upute za korištenje web aplikacije

Nakon toga korisnik unosi podatke unutar mjesta koja su predviđena za to. Unosi početne uvijete za x_0 i y_0, funkciju $f(x,y)$ ili y', duljinu koraka h i traženi uvjet. Unosi za ovaj slučaj prikazani su na slici 4.16.



Slika 4.16. Dio web stranice za unos podataka

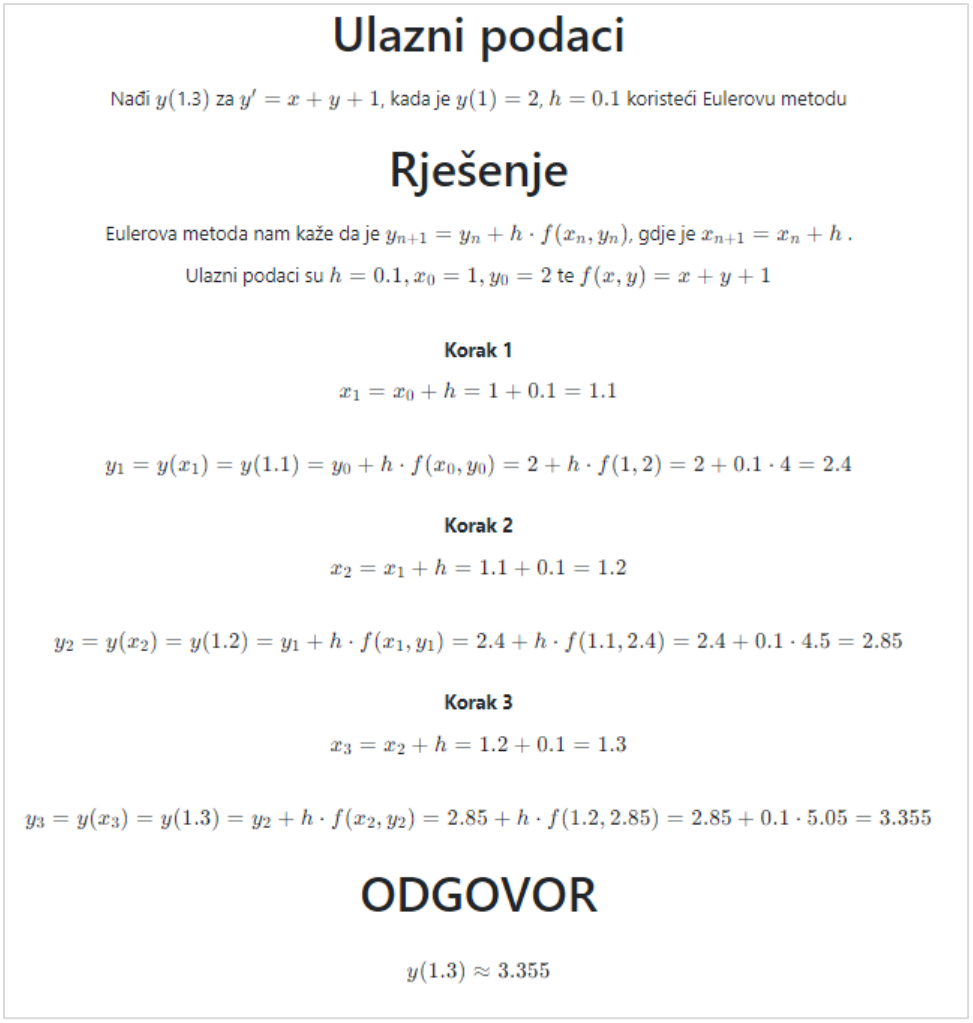
Nakon toga korisnik mora kliknuti na gumb *„Izračunaj“* . Nakon klika na gumb pojavljuju se rezultati u kartici *„Rezultat za tražen uvjet“*. Rezultati su prikazani na slici 4.17.



n	x_n	y_n	$y' = \frac{dy}{dx}$	$h = \Delta x$	$\Delta y = \frac{dy}{dx}\Delta x$	$(x + \Delta x, y + \Delta y)$
3	1.3	3.355	5.655	0.1	0.5655	(1.4, 3.9205)

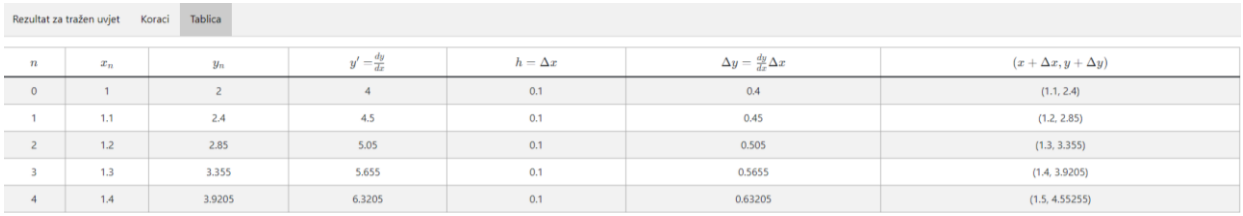
Slika 4.17. Prikaz rezultata u kartici *„Rezultat za tražen uvjet“*

Klikom na karticu *„Koraci“* korisnik dobiva uvid u svaki korak rješenja zadatka. Slika 4.18. prikazuje rezultate za ovaj specifičan primjer.



Slika 4.18. Prikaz rezultata u kartici *„Koraci“*

Klikom na karticu *„Tablica“* prikazuje se slični sadržaj kao i u kartici *„Koraci“* samo u tabličnom obliku. Slika 4.19. prikazuje rješenje za ovaj primjer.



n	x_n	y_n	$y' = \frac{dy}{dx}$	$h = \Delta x$	$\Delta y = \frac{dy}{dx}\Delta x$	$(x + \Delta x, y + \Delta y)$
0	1	2	4	0.1	0.4	(1.1, 2.4)
1	1.1	2.4	4.5	0.1	0.45	(1.2, 2.85)
2	1.2	2.85	5.05	0.1	0.505	(1.3, 3.355)
3	1.3	3.355	5.655	0.1	0.5655	(1.4, 3.9205)
4	1.4	3.9205	6.3205	0.1	0.63205	(1.5, 4.55255)

Slika 4.19. Prikaz rezultata u kartici *„Tablica“*

5. ZAKLJUČAK

Eulerova metoda je jedna od jednostavnijih metoda rješavanja običnih diferencijalnih jednadžbi sa početnim uvjetom. Kroz navedene primjere je pokazano rješavanje takvih tipova zadataka. Može se primijetiti da Eulerova metoda uzrokuje dvije vrste grešaka te da se preciznost ove metode povećava ako se smanjuje veličina koraka h. Za izradu web aplikacije korišteni su HTML, CSS i JavaScript tehnologije. Uz pomoć fakulteta web stranica je dostupna svima na internetu.

6. LITERATURA

[1] R.Scitovski, Numerička matematika, Osijek, 2015.

[2] S. Singer, Numerička analiza, Predavanja i vježbe, Poglavlje 10 Obične diferencijalne jednadžbe, Zagreb, 2003

[3] D. F. Griffiths, D. J. Higham, Numerical Methods for Ordinary Differential Equations, Springer-Verlag London Limited , 2010

[4] https://www.invictum.hr/sto-je-html/, [25.8.2022.]

[5] https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS, [25.8.2022.]

[6]https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript, [25.8.2022.]

[7]https://www.freecodecamp.org/news/big-o-notation-why-it-matters-and-why-it-doesnt-1674cfa8a23c/, [25.8.2022.]

7. SAŽETAK

Tema ovog završnog rada je istražiti Eulerovu metodu i rješavanje Cauchyjevog problema Eulerovom metodom te izrada web aplikacije na temelju istraženog. Korištene web tehnologije u izradi ove web aplikacije su: HTML,CSS i JavaScript. JavaScript je korišten za implementaciju algoritama za izračunavanje rezultata pomoću Eulerove metode. Postupci korištenja web aplikacije se nalazi u ovom radu. Link na ovu aplikaciju je:
http://www.etfos.unios.hr/~antonio.corak /

Ključne riječi: Cauchyjev problem, CSS, Eulerova metoda, HTML, JavaScript

8. ABSTRACT

A WEB APPLICATION FOR SOLVING THE CAUCHY PROBLEM USING THE EULER METHOD

The topic of this final paper is to investigate Euler's method and the solution of the Cauchy problem with Euler's method, as well as the creation of a web application based on the research. The web technologies used in the creation of this web application are: HTML, CSS and JavaScript. JavaScript was used to implement the algorithms to calculate the results using Euler's method. Procedures for using the web application can be found in this paper. The link to this application is: http://www.etfos.unios.hr/~antonio.corak /

Keywords: Cauchy's problem, CSS, Euler's method, HTML, JavaScript

9. PRILOG

Web stranica se nalazi na sljedećem linku: http://www.etfos.unios.hr/~antonio.corak/