Seminarski rad

Visnja Polic

December 6, 2021

1 Uvod

U ovom radu predstavljeno je resavanje sistema diferencijalnih jednacina numerickom metodom tipa *Runge-Kutta* sa redom tacnosti tri ili cetiri u **programskom jeziku C**.

Program je potrebno pokrenuti komandom:

gcc seminarski.c rad.c -lm

./a.out ime-datoteke.txt p

2 Resavanje sistema diferencijalnih jednacina metodama tipa *Runge-Kutta*

2.1 Objasnjenje rada programa

Program se sastoji iz tri dela:

seminarski.c - sadrzi glavni deo programa main;

rad.c - sadrzi implementaciju svih funkcija koje su koriscene u programu;

rad.h - sadrzi definicije svih funkcija koriscenih u programu.

Deo programa *seminarski.c* najpre ucitava podatke. Prvo se ucitavaju argumenti komandne linije, gde program proverava da li je ucitan dovoljan broj argumenata i vraca odgovarajucu poruku ukoliko je doslo do greske. Zatim se datoteci f dodeljuje prvi argument komandne linije koji predstavlja ime datoteke iz koje se ucitavaju podaci. Ukoliko je doslo do greseke pri otvaranju datoteke, ispisuje se odgovarajuca poruka. Promenljivoj p dodeljuje se vrednost drugog argumenta komandne linije koji predstavlja red metode 3 ili 4. Ukoliko je zadata vrednost razlicita od 3 ili 4, ispisuje se odgovarajuza poruka o gresci na standardni izlaz.

Nakon sto je datoteka uspesno otvorena za ciranje, iz nje se ucitavaju redom sledeci podaci:

n - dimenzija sistema;

A - matrica sistema dimenzije $n \times n$;

b - vektor slobodnih clanova;

x_0 - pocetna tacka (vektor);

u - vrednost funkcije u pocetnoj tacki;

h - korak;

ε - zadata tacnost.

Zatvaramo datoteku.

Nakon ovoga, racunaju se vrednosti funkcija u i v pomocu funkcije *racunaj* u istoj tacki $x + h$, s tim sto se vrednosti racunaju redom za dati korak i za duplo duzi korak. Zatim se proverava kriterijum zaustavljanja pomocu funkcije *uslov*. Sve dok uslov nije zadovoljen, za dato u se racuna popravka:

$u_p = u + \frac{u-v}{2^p-1}$,

racunaju se nove u i v i ponovo proverava uslov.

Kada je uslov ispunjen, *while* petlja se zavrsava i ispisuje se resenje u.

Funkcije koriscene za ova racunanja i ucitavanja su implementirane u segmentu *rad.c*.

Funkcija *ucitaj_matricu_iz_datoteke* ucitava matricu A dimenzije $n \times n$ iz datoteke f. Ukoliko nije uspelo ucitavanje matrice - vraca 1, a inace vrati 0 kao znak da je uspesno ucitano.

Funkcija *ucitaj_vektor_iz_datoteke* ucitava vektor dimenzije n iz datoteke f. Ukoliko nije uspelo ucitavanje vektora - vraca 1, a inace vrati 0 kao znak da je uspesno ucitano.

Funkcija *racunaj* u zavisnosti od reda metode, racuna koeficijente k po odgovarajucim formulama i vraca vrednost funkcije u u k-toj koordinati.

Funkcija *uslov* u zavisnosti od reda metode proverava da li je ispunjen kriterijum zaustavljanja: $\frac{|u-v|}{2^p-1} < \varepsilon$.

Ukoliko uslov nije ispunjen - vraca -1, a inace vrati 1 kao znak da je uslov ispunjen.

2.2 Primer

Tekst fajl datoteka.txt sadrzi sledece podatke:

3
-2 0 5
-1 -1 3
-1 0 2
0 0 0
0 0 0
-0.075092
0.073039
0.063325
0.25
0.001

Na slikama *Figure1* i *Figure2* predstavljena je za dati primer jedna iteracija funkcije *racunaj*, redom za red metode $p = 3$ i $p = 4$. Prikazano je racunanje koeficijenata k, a zatim i funkcije u pomocu izracunatih koeficijenata.

Na slikama *Figure3* i *Figure4* predstavljena je za dati primer jedna iteracija provere uslova zaustavljanja i racunanja popravke, redom za red metode $p = 3$ i $p = 4$.

```
k1 = h * f(x_0, u_prethodno)
k1[0] =0.101973
k1[1] =0.040642
k1[2] =0.043071
k2 = h * f(x_0 + h/2, u_prethodno + k1/2)
k2[0] =0.103399
k2[1] =0.038967
k2[2] =0.041092
k3 = h * f(x_0 + h, u_prethodno -k1 + 2*k2)
k3[0] =0.098452
k3[1] =0.034448
k3[2] =0.036421
za p=3, u se racuna po formuli:
 u_prethodno + h/6 * (k1 + 4*k2 + k3)u[1] = 0.082662
k1 = h * f(x_0, u_prethodno)
k1[0] =0.101973
k1[1] =0.038237
k1[2] =0.043071
k2 = h * f(x_0 + h/2, u_prethodno + k1/2)
k2[0] =0.103399
k2[1] =0.036862
k2[2] =0.041092
k3 = h * f(x_0 + h, u_prethodno -k1 + 2*k2)
k3[0] =0.098452
k3[1] =0.032493
k3[2] =0.036421
za p=3, u se racuna po formuli:
 u_prethodno + h/6 * (k1 + 4*k2 + k3)u[2] = 0.073486
```

Figure 1: Primer jedne iteracije funkcije racunaj - RK3

```
k1 = h * f(x_0, u_prethodno)
k1[0] =0.101975
k1[1] =0.040643
k1[2] =0.043072
k2 = h * f(x_0 + h/2, u_prethodno + k1/2)
k2[0] =0.103401
k2[1] =0.038968
k2[2] =0.041093
k3 = h * f(x_0 + h/2, u_prethodno + k2/2)
k3[0] =0.101807
k3[1] =0.038257
k3[2] =0.040420
k4 = h * f(x_0 + h, u_prethodno + k3)
k4[0] =0.101596
k4[1] =0.035942
k4[2] =0.037830
za p=4, u se racuna po formuli:
 u_prethodno + h/6 * (k1 + 2*k2 + 2*k3 + k4)u[1] = 0.082665
k1 = h * f(x_0, u_prethodno)
k1[0] =0.101975
k1[1] =0.038237
k1[2] =0.043072
k2 = h * f(x_0 + h/2, u_prethodno + k1/2)
k2[0] =0.103401
k2[1] =0.036862
k2[2] =0.041093
k3 = h * f(x_0 + h/2, u_prethodno + k2/2)
k3[0] =0.101807
k3[1] =0.036114
k3[2] =0.040420
k4 = h * f(x_0 + h, u_prethodno + k3)
k4[0] =0.101596
k4[1] =0.034071
k4[2] =0.037830
za p=4, u se racuna po formuli:
 u_prethodno + h/6 * (k1 + 2*k2 + 2*k3 + k4)u[2] = 0.073489
```

Figure 2: Primer jedne iteracije funkcije racunaj - RK4

```
Racunaju se nove vrednosti za u i v:
u_novo[0] = 0.168156
v_novo[0] = 0.176969
u_novo[1] = 0.084750
v_novo[1] = 0.080509
u_novo[2] = 0.084748
v_novo[2] = 0.080508
Provera uslova: |u-v|/(2^p-1)>=eps
0.001259 >= 0.001000 ?
jeste:
up = u + (u-v)/(2^p-1)
= 0.168156 + (0.168156 - 0.176969)/0.168156
koord 0: u_popravka[0] = 0.166897
Provera uslova: |u-v|/(2^p-1)>=eps
0.000606 >= 0.001000 ?
nije
Provera uslova: |u-v|/(2^p-1)>=eps
0.000606 >= 0.001000 ?
nije
Nakon popravke:
u_p[0] = 0.166897
u_p[1] = 0.084750
u_p[2] = 0.084748
Racunaju se nove vrednosti za u i v:
u_novo[0] = 0.171156
v_novo[0] = 0.178015
u_novo[1] = 0.083985
v_novo[1] = 0.079055
u_novo[2] = 0.083983
v_novo[2] = 0.079054
Stajemo!Postignuta je trazena tacnost.
Trazeno resenje je
u:
0.171156
0.083985
0.083983
```

Figure 3: Primer jedne iteracije - RK3

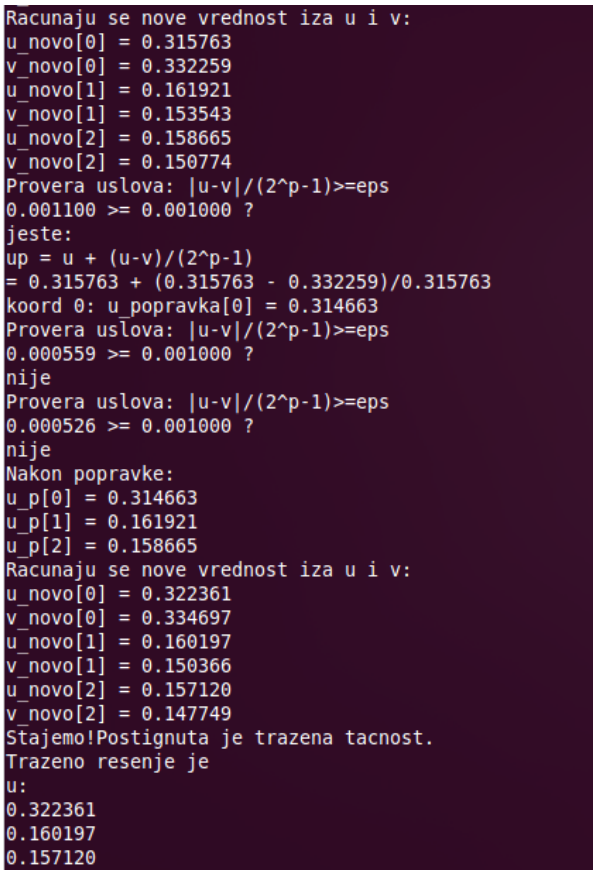

Figure 4: Primer jedne iteracije - RK4