

$Lv(x_i) = f(x_i)$, Integralne jednačine

Sadržaj

Fredholmove integralne jednačine druge vrste.....	1
Metoda kvadrature formula.....	1
Varijacione metode.....	4
Ric-Galerkinova metoda.....	5
Metoda kolokacije.....	8
Metoda najmanjih kvadrata	10

Fredholmove integralne jednačine druge vrste

Bavićemo se integralni jednačinama oblika

$$u(x) - \lambda \int_a^b K(x, t)u(t)dt = f(x), \quad a \leq x \leq b,$$

gde je jedino $u(x)$ nepoznata funkcija. Pretpostavka je da je $f : [a, b] \rightarrow \mathbb{R}$ neprekidna funkcija, $K : [a, b] \times [a, b] \rightarrow \mathbb{R}$ neprekidna funkcija koju još nazivamo jezgrom i $\lambda \neq 0$.

Metoda kvadrature formula

Ova metoda kao što joj i ime kaže zasniva se na aproksimaciji integrala nekom kvadrature formulom.

Aproksimacija $v(x)$ se dobija na diskretnom skupu tačaka rešavanjem sistema linearnih jednačina.

Pretpostavimo da su čvorovi kvadrature formule ekvidistantno zadati.

$$x_i = x_0 + ih, \quad i = 0, 1, \dots, n,$$

gde je $x_0 = a$, $h = (b - a)/n$.

Na ovom skupu tačaka definišimo mrežnu funkciju $v(x_i) = v_i$, a zatim integral u integralnoj jednačini aproksimiramo nekom kvadrature formulom čiji su čvorovi baš x_i , $i = 0, \dots, n$,

$$\int_a^b K(x, t)u(x, t)dt \approx \sum_{i=0}^n c_i K(x, t_i)u(t_i),$$

gde su nam t_i zapravo čvorovi x_i a c_i težinski koeficijenti.

Tada ćemo dobiti aproksimaciju u tački x kao $v(x) - \lambda \sum_{i=0}^n c_i K(x, t_i)v(x_i) = f(x)$. No da bismo uspeali da izračunamo $v(x)$, zamenimo x sa x_j . Tako dobijamo

$v_j - \lambda \sum_{i=0}^n c_i K(x, t_i)v_j = f(x)$, $j = 0, \dots, n$. Ovako dobijamo jedan sistem linearnih jednačina sa matricom

$$\begin{bmatrix} 1 - \lambda c_0 K(x_0, t_0) & -\lambda c_1 K(x_0, t_1) & -\lambda c_2 K(x_0, t_2) & \cdots & -\lambda c_n K(x_0, t_n) \\ -\lambda c_0 K(x_1, t_0) & 1 - \lambda c_1 K(x_1, t_1) & -\lambda c_2 K(x_1, t_2) & \cdots & -\lambda c_n K(x_1, t_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\lambda c_0 K(x_n, t_0) & -\lambda c_1 K(x_n, t_1) & -\lambda c_2 K(x_n, t_2) & \cdots & 1 - \lambda c_n K(x_n, t_n) \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

Koristićemo kvadrature formule:

Trapeza:

$$\int_a^b f(x)dx \approx \frac{h}{2}[f_0 + 2(f_1 + f_2 + \dots + f_{n-1}) + f_n];$$

$$\text{Simpsona: } \int_a^b f(x)dx \approx \frac{h}{3}[f_0 + 4(f_1 + f_3 + \dots + f_{2m-1}) + 2(f_2 + f_4 + \dots + f_{2m-2}) + f_{2m}], \quad n = 2m.$$

Primer1

Uzmimo jedan prost primer

$$u(x) + \int_0^1 x(e^{xt} - 1)u(t)dt = e^x - x. \text{ Rešenje ovo integralne jednačine je } u \equiv 1.$$

Podatke o ovoj Fredholmovoj integralnoj jednačini ćemo čuvati u posebnom script jalu **PrimerF21.m**

```
type PrimerF21
```

```
%primerF21

%interval integracije [a,b]
a=0;
b=1;

%jezgro K(x,t)
lambda=-1;
K=@(x,t) x.*(exp(x.*t)-1);

%desna strana jednacine
f=@(x) exp(x)-x;

%tacno resenje
ue=@(x) ones(size(x));
```

Implementiraćemo funkciju $v = \text{mkvadF2}(\text{primer}, n)$ koja trapeznom kvadraturnom formulom rešava

Fredholmovu integralu jednačinu zadatu script fajlom **primer** sa n čvorova. Izlazni argument je aproksimacija rešenja u čvorovima v . Pretpostavimo da se u scipr fajlu **primer** nalazi i tačno rešenje problema pa će funkcija uporediti aproksimaciju i tačno rešenje iscrtavanjem grafika tačnog rešenja i aproksimacije.

```
type mkvadF2.m
```

```
function v=mkvadF2(primer,n)
% Funkcija v=mkvadF2(primer,n) odredjuje priblizno
% resenje Fredholmove integralne jednacine druge vrste
%
% u(x)-lambda*int_a^b K(x,t)*u(t)dt=f(x)
%
% koristeci metodu kvadraturnih formula.
%
% Ulazne vrednosti:
%         primer - naziv skript fajla u kome se
%                 nalaze podaci a,b,lambda, K(x,t),f
%         n - broj cvorova
% Izlazna vrednost:
%         v - priblizna vrednost u cvorovima
% Test: mkvadF2('primerF21',5)
```

```

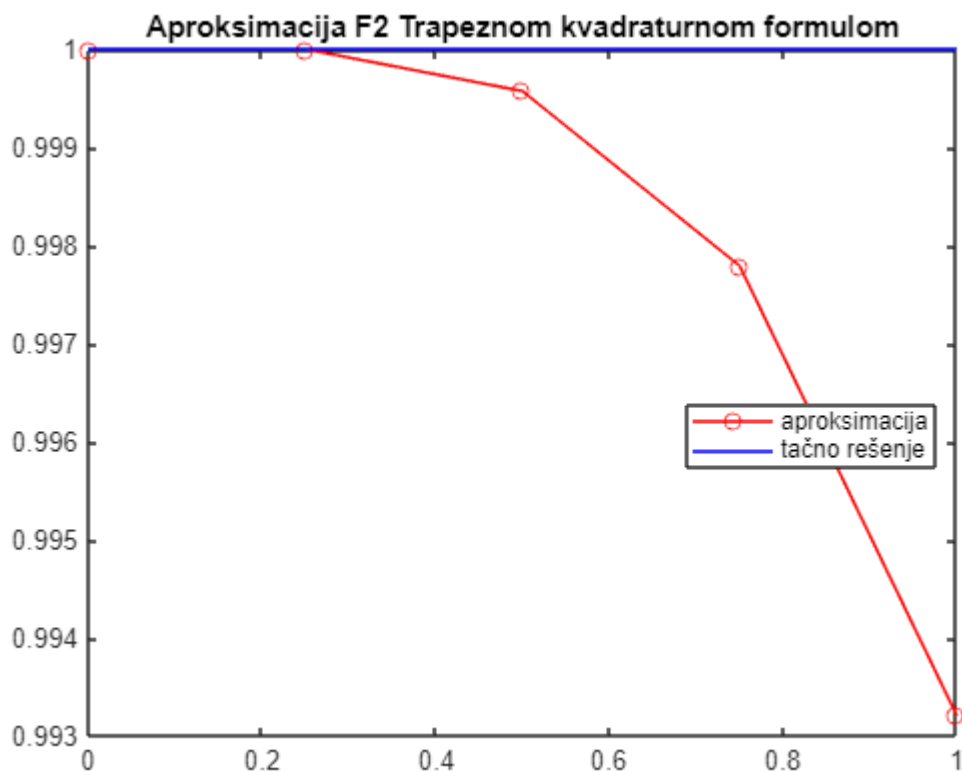
run(primer)
%% formiranje mreze
x=linspace(a,b,n);%cvorovi
h=(b-a)/(n-1);

%% formiranje matrice sistema
A=eye(n);%koeficijenti sistema, postavljamo 1 na dijagonalu za vi
for i=1:n
    Ki=@(t) lambda*K(x(i),t); %radi lakse citljivosti koda uvedimo pomocnu funkciju
    %primenjujemo trapezno kvadraturno pravilo na integral
    pom=h/2*[Ki(x(1)) 2*Ki(x(2:n-1)) Ki(x(n))];
    A(i,:)=A(i,:)-pom;
end
%% desna strana sistema
F=f(x');
%% resenje
v=A\F;

%% grafik
plot(x,v,'o-r', x,ue(x),'b')
legend('aproksimacija', 'tačno rešenje','Location','best')
title("Aproksimacija F2 Trapeznom kvadraturnom formulom")

```

```
mkvadF2('PrimerF21',5)
```



```

ans = 5x1
    1.0000
    1.0000
    0.9996
    0.9978
    0.9932

```

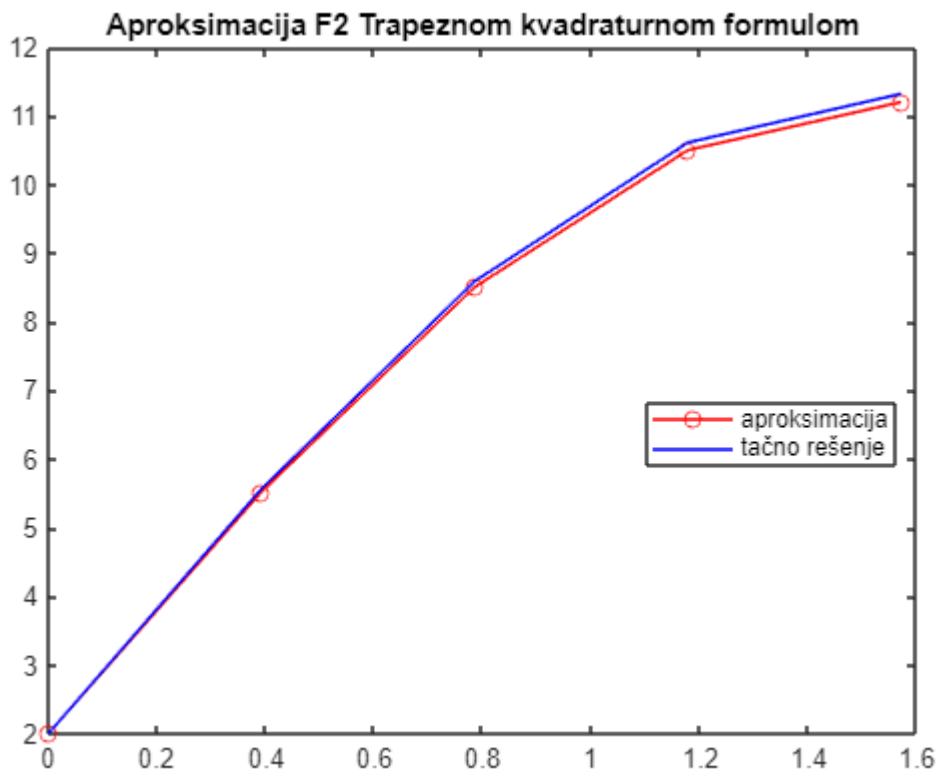
Primer2

Testiraćemo kod na još jednom primeru:

$u(x) - \int_0^{\pi/2} \sin(x)\sin(t)u(t)dt = 2$. Tačno rešenje ovog problema je $u(x) = 2 + 8/(4 - \pi) \sin(x)$.

Podatke o ovoj jednalini sačuvajmo u script fajlu **PrimerF22.m**.

```
mkvadF2('PrimerF22',5)
```



```
ans = 5×1  
2.0000  
5.5205  
8.5050  
10.4992  
11.1995
```

U ovom slučaju aproksimacija je dosta bolja.

Varijacione metode

Zapišimo sada jednačinu u operatorskom obliku

$Lu = f$, gde je $L(u)(x) = u(x) - \lambda \int_a^b K(x, \xi) dx d\xi$.

Neka je sa $(\cdot, \cdot)$ definisan skalarni proizvod u $V = L_2(a, b)$. Ako je u rešenje jednačine date jednačine tada za proizvoljnu funkciju $v \in V$ koju jo[nazivamo i test funkcijom važi

$$(Lu, v) = (f, v), \forall v.$$

Rešenje gornje jednačine se naziva **slabim rešenjem**.

Prilikom konstrukcije varijacionih i projekcionih metoda test funkcije biramo iz nekog prostora V a Neka je V_n konačnodimenzioni potprostor prostora V . Rešenje tražimo u obliku

$$\tilde{u} = f + \sum_{i=1}^n c_i \phi_i, \text{ gde su } \phi_i, i = 1, \dots, n \text{ bazne funkcije prostora } V_n.$$

Ric-Galerkinova metoda

Sada ćemo za test funkcije birati baš funkcije ϕ_i .

Tako dobijamo sistem jednačina

$$(Lv, \phi_i) = (f, \phi_i), i = 1, \dots, n \text{ gde smo sa } v \text{ označili aproksimaciju rešenja.}$$

Iz ove jednačine neposredno sledi da je greška $Lv - f$ ortogonalna na prostor V_n .

Ako raspišemo sada operator L dobićemo

$$\sum_{j=1}^n c_j \left[(\phi_j, \phi_i) - \lambda \int_a^b \int_a^b K(x, t) \phi_j(t) \phi_i(x) dt dx \right] = \lambda \int_a^b \int_a^b (K(x, t) f(t) \phi_i(x)) dx dt.$$

Ovo je jedan sistema linearnih jednačina

$$Ac = F.$$

Matrica $A = (a_{ij})$ je dimenzije $n \times n$, $a_{ij} = \alpha_{ij} - \lambda \beta_{ij}$

$$\alpha_{ij} = (\phi_i, \phi_j), \beta_{ij} = \int_a^b \int_a^b K(x, t) \phi_j(t) \phi_i(x) dx dt.$$

$$F = [F_1, F_2, \dots, F_n]^T.$$

$$F_i = \lambda \int_a^b \int_a^b K(x, t) f(t) \phi_i(x) dx dt.$$

Baze koje ćemo koristiti su polinomijalne $1, x, x^2, \dots$ i trigonometrijske $1, \cos(x), \sin(x), \cos(2x), \sin(2x), \dots$.

Implementirajmo sada *Ric-Galerkinovu* metodu funkcijom $c = RGF2(\text{primer}, n, \text{baza})$, gde je *primer* imer script fajla u kome se nalaze podaci o samoj jednačini, n broj bazisnih funkcija i *bazanam* je promenjiva kojom korisnik određuje koju bazu će koristiti. Ako je *baza* = 0 onda koristimo polinomijlanu u suprotnom trigonometrijsku. Izlazno c su koeficijenti uz bazisne funkcije.

Bazisne funkcije ćemo čuvati u niz ćelija.

Kada je baza polinomijalna konstrukcija niza ćelija je jednostavna.

n=3

n =
3

```

phi=cell(1,n);
for k=1:n
    phi{k}=@(x) x.^(k-1);
end
phi

```

phi = 1x3 cell

	1	2	3
1	@(x)x.^(k-1)	@(x)x.^(k-1)	@(x)x.^(k-1)

Kada je baza trigonometrijska situacija je malo složenija.

n=2

n =
2

```

phi=cell(1,n);
for k=1:n/2
    phi{2*k-1}=@(x) sin(k*x);
    phi{2*k}=@(x) cos(k*x);
end
if mod(n,2)~=0 % ako je baza sa neparnim brojem clanova onda moramo dodati jos
jedan clan.
    if n==1 % ako je n=1 onda u prvu petlju i ne ulazimo pa moramo ovde da
formiramu bazu.
        k=n;
    else
        k=k+1;
    end
    phi{2*k-1}=@(x) sin(k*x);
end
phi

```

phi = 1x2 cell

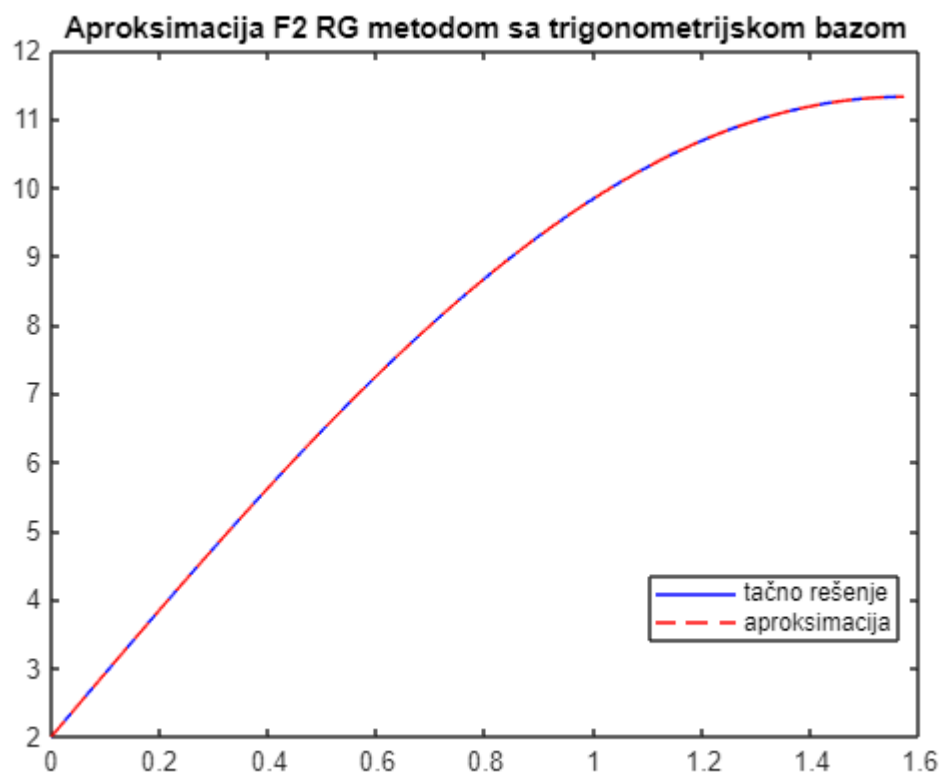
	1	2
1	@(x)sin(k*x)	@(x)cos(k*x)

Kraći način za formiranje niza je sledeći

```
phi=@(x,k) mod(k,2)*sin((k+1)/2*x)+mod(k+1,2)*cos(k/2*x);
```

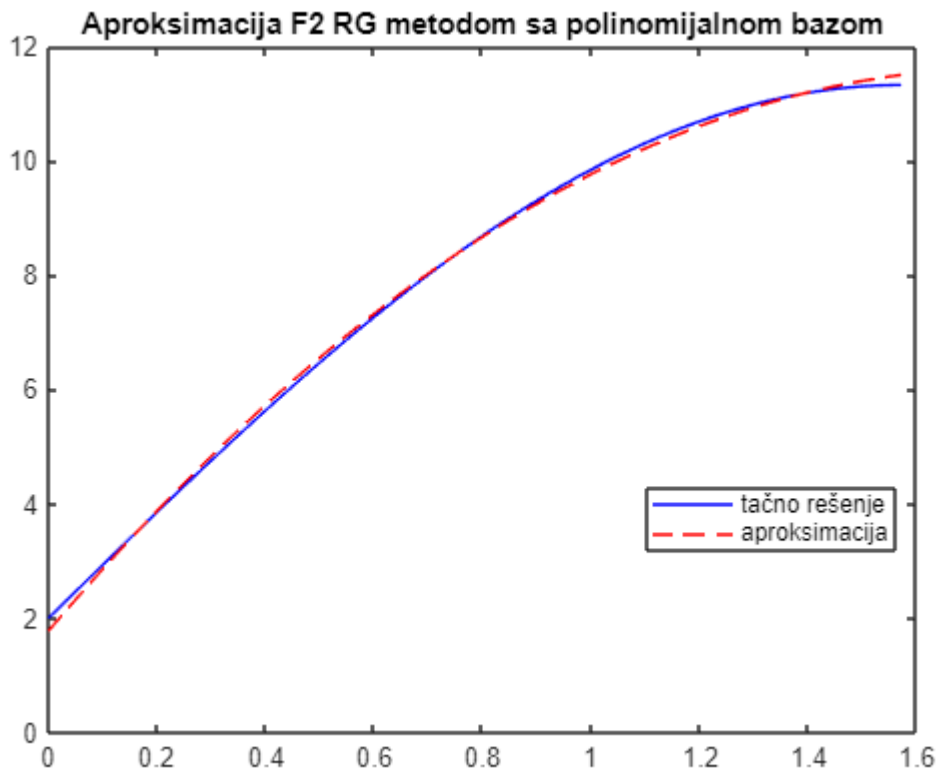
Testirajmo sada na nekim primerima funkciju $c = RGF2(primer, n, baza)$

```
RGF2('PrimerF22',3,1)
```



```
ans = 3×1
      9.3196
      0.0000
     -0.0000
```

```
RGF2('PrimerF22',3,0)
```



```
ans = 3×1
-0.2266
11.1381
-3.1506
```

Metoda kolokacije

U metodi kolokacije za test funkcije uzimamo *Dirakove distribucije* $\psi_i = \delta(x - x_i)$, $i = 1, 2, \dots, n$, gde su nam tačke x_i zadate tačke iz segmenta $[a, b]$. Kako je

$$\int_a^b f(x) \delta(x - x_i) dx = f(x_i) \text{ to iz sistema dobijamo}$$

$$(Lv, \psi_i) = (f, \psi_i), \quad i = 1, 2, \dots, n.$$

Ako sada napišemo eksplicitan izraz operatora L dobijamo

$$Lv(x_i) = v(x_i) - \lambda \int_a^b K(x_i, t) v(t) dt = f(x_i) + \sum_{j=1}^n c_j \varphi_j(x_i) - \lambda \int_a^b K(x_i, t) f(t) dt - \lambda \sum_{j=1}^n c_j \int_a^b K(x_i, t) \varphi_j(t) dt$$

To nam daje linearan sistem jednačina

$$Ac = F$$

gde je

$$a_{ij} = \varphi_j(x_i) - \lambda \int_a^b K(x_i, t) \varphi_j(t) dt,$$

$$F_i = \lambda \int_a^b K(x_i, t) f(t) dt.$$

```
type PrimerF23.m
```

```
%primerF23
```

```
%interval integracije [a,b]
```

```
a=0;
```

```
b=1;
```

```
lambda=1/2;
```

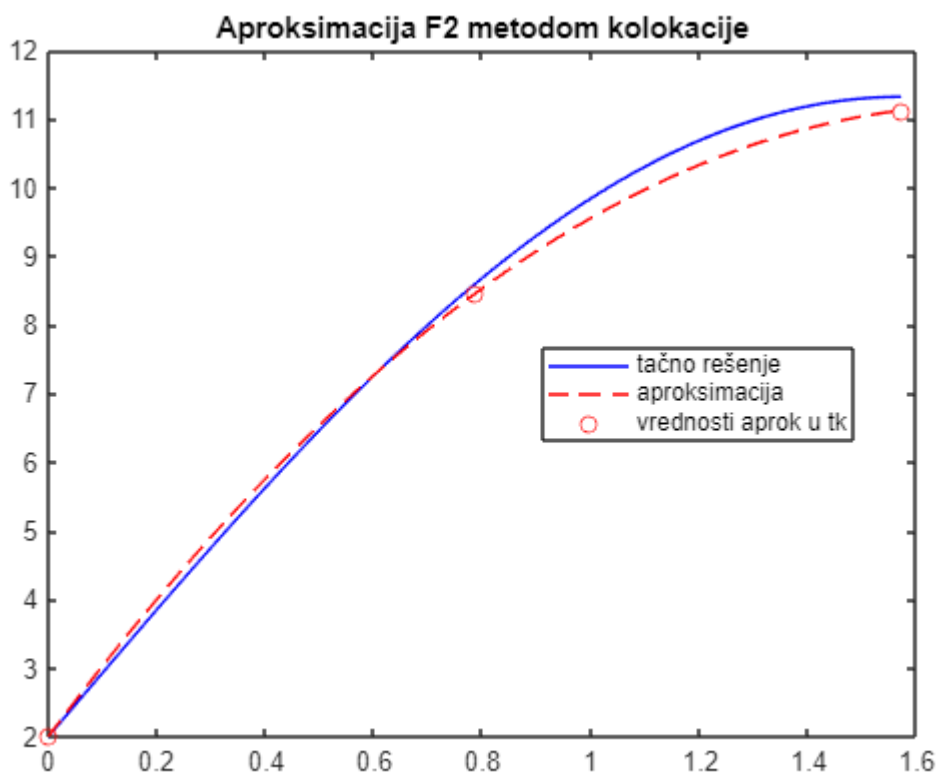
```
K=@(x,t) x;
```

```
%desna strana jednacine
```

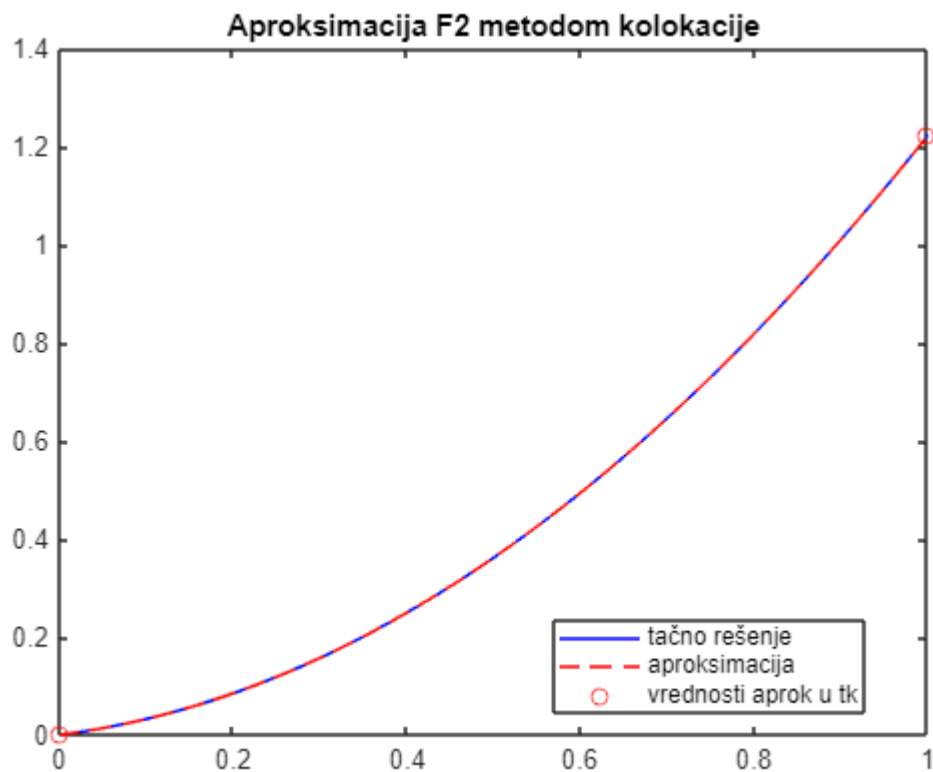
```
f=@(x) x.^2;
```

```
ue=@(x) x.^2+2/9*x;
```

```
mkolF2('PrimerF22',[0,pi/4,pi/2]);
```



```
mkolF2('PrimerF23',[0:1:1]);
```



Metoda najmanjih kvadrata

Kod ove metode minimizujemo funkcional

$$\int_a^b (Lv - f)^2 dx \rightarrow \min.$$

Tako se dobija sistem linearnih jednačina sa matricom $A = a_{ij}$, $i, j = 1, \dots, n$,

$$a_{ij} = (L\psi_i, L\psi_j)$$

$$F_i = (f, L\psi_i)$$