INTEGRAL TENGLAMALARNI SIMPSON VA TRAPETSIYA USULLARI YORDAMIDA YECHISH

Nabiyev A.A.

FarDU talabasi, nabiyevasadbek0808@gmail.com

Annotatsiya: Ushbu maqolada integral tenglamalarni sonli yechish masalasi Simpson va trapetsiya usullari asosida ko'rib chiqiladi. Integralni diskretlashtirish orqali integral tenglamalarni chiziqli algebraik tenglamalar sistemasiga keltirish yondashuvi bayon etiladi. Simpson va trapetsiya usullarining aniqlik darajasi, xato baholari hamda konvergentsiya xossalari tahlil qilinadi. Shuningdek, ushbu usullarni Fredholm va Volterra turidagi integral tenglamalarga tatbiq etish imkoniyatlari yoritiladi. Natijalar shuni ko'rsatadiki, Simpson usuli silliq funksiyalar uchun yuqori aniqlikni ta'minlaydi, trapetsiya usuli esa soddaligi va hisoblash barqarorligi bilan ajralib turadi.

Kalit so'zlar: integral tenglamalar, sonli usullar, Simpson usuli, trapetsiya usuli, diskretlashtirish, Fredholm tenglamasi, Volterra tenglamasi

Integral tenglamalarni taxminiy yechishda amaliyotda keng qo'llaniladigan usullardan biri bu tenglamada qatnashayotgan integralni u yoki bu

$$\int_a^b \varphi(x)dx = \sum_{j=1}^{n} A_j \varphi \ x_j \ + R(\Phi)$$

kvadratura formulasi bilan almashtirishdan iboratdir. Bu yerda $x_1, x_2, \ldots, x_n$ va $A_1, A_2, \ldots, A_n$ lar mos ravishda k.f. ning tugunlari hamda koeffitsientlari bo'lib, ular $\Phi(x)$ funksiyaga bog'liq emas, $R(\Phi)$ esa qoldiq had. Bu formulada $A_j \geq 0$ va $\sum_{j=1}^{n} A_j = b - a$ shartlar bajariladi, deb faraz qilamiz. Misol sifatida 7-bobda ko'rilgan quyidagi formulalarni keltiramiz:

1. Umumlashtirilgan to'g'ri to'rtburchaklar formulasi:

$$x_j = a + (j-1)h, \quad h = \frac{b-a}{n}, \quad A_j = h, \quad j = 1, 2, \ldots, n.$$

2. Umumlashtirilgan trapetsiyalar formulasi:

$$x_j = a + (j-1)h, \quad h = \frac{b-a}{n-1}, \quad A_1 = A_n = \frac{h}{2}, \quad A_j = h, \quad j = 2, \ldots, n-1.$$

3. Umumlashtirilgan Simpson formulasi ($n = 2m + 1$ deb olamiz):

$$x_j = a + (j-1)h, \quad h = \frac{b-a}{2m}, \quad A_1 = A_{2m+1} = \frac{h}{3},$$

$$A_2 = A_4 = \ldots = A_{2m} = \frac{4}{3}h, \quad A_3 = A_5 = \ldots = A_{2m-1} = \frac{2h}{3}.$$

4. Gauss formulasi:

$$x_j = \frac{b-a}{2} x_j^{(n)} + \frac{a+b}{2}, \quad A_j = \frac{b-a}{2} A_j^{(n)},$$

bu yerda $x_j^{(n)}$ va $A_j^{(n)}$ lar $[-1,1]$ segment uchun qurilgan Gauss formulasining tugunlari va koeffitsientlaridir.

Ma'lumki, agar $K(x,s)$ va $f(x)$ funksiyalar $D = \{a \le s \le x \le b\}$ sohasida uzluksiz bo'lsa, u holda Volterraning 2-jins integral tenglamasi

$$u(x) - \lambda \int_a^x K(x,s)u(s)ds = f(x) \qquad (1.1)$$

λ ning ixtiyoriy qiymatida yagona yechimga ega. Mazkur yechimni quyidagicha izlaymiz:

$$u(x) = \sum_{k=0}^{\infty} \lambda^k \varphi_k(x). \qquad (1.2)$$

Bu qatorni (1.1) tenglamaga qo'yib, keyin λ ning oldidagi bir hil darajali koeffitsientlarni tenglashtiramiz, natijada

$$\varphi_0(x) = f(x), \quad \varphi_k(x) = \int_a^x K(x,s)\varphi_{k-1}(s)ds, \quad k = 1,2,\ldots \qquad (1.3)$$

tengliklar kelib chiqadi. dagi belgilashlarda

$$|\varphi_k(x)| \le \frac{N\ M(b-a)^k}{k!} \qquad (1.4)$$

bahoqa ega bo'lamiz. Agar (1.1) tenglamaning taqribiy yechimi sifatida (1.2) qatorning avvalgi n ta hadini olsak, u holda (1.4) tengsizlikka ko'ra katolik uchun quyidagi bahoga ega bo'lamiz:

$$\varepsilon_n = |u(x) - u_n(x)| = \left| \sum_{k=n+1}^{\infty} \lambda^k \varphi_k(x) \right| \le N \sum_{k=n+1}^{\infty} \frac{M(b-a)^k}{k!}. \qquad (1.5)$$

Bu baho ancha keng. Ko'p hollarda absolut katolik bundan ancha kichik bo'lishi mumkin.

Misol. Quyidagi

$$u(x)-\int_0^x (s-x)u(s)ds = x, \quad 0\le x\le 2$$

integral tenglamaning yechimi $\varepsilon = 10^{-5}$ absolut katolik bilan topilsin. Bu yerda $N = x\le 2\,|\,K(x,s)|\le 2$, $\lambda = 1$, $b-a\le 2$ bo‘lganligi uchun (1.4) dan

$$|\varphi_k(x)|\le \frac{2^{k+1}}{k!}$$

bahoga ega bo‘lamiz. Bundan esa

$$\varepsilon_n = 2\sum_{k=n+1}^{\infty}\frac{2^k}{k!} < 10^{-5}$$

tengsizlik bajarilishi uchun $n = 11$ bo‘lishi lozim. Aslida bunday emas. Haqiqatan ham, $\varphi_0(x) = x$ deb olib, ketma-ket quyidagilarni hosil qilamiz:

$$\varphi_1(x) = \int_0^x (s-x)\varphi_0(s)ds = \frac{x^3}{3} - \frac{x\cdot x^2}{2} = -\frac{x^3}{3!},$$

$$\varphi_2(x) = \int_0^x (s-x)\varphi_1(s)ds = -\frac{1}{3!}\left(\frac{x^5}{5} - \frac{x\cdot x^4}{4}\right) = \frac{x^5}{5},$$

……………………………

$$\varphi_n(x) = (-1)^n\frac{x^{2n+1}}{(2n+1)!}.$$

Bundan ko‘ramizki, $e_n = 10^{-5}$ bo‘lishi uchun n=5 yetarli. Shunday qilib, taxminiy yechim sifatida

$$u(x)\equiv u_5(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{1!}$$

ni olishimiz mumkin. Ko‘rinib turibdiki, aniq yechim $u = \sin x$.

Agar (1.3) integrallar aniq olinmasa, u holda kvadratura formulalaridan foydalanishga to‘g‘ri keladi. Masalan, [a, b] oraliqni n ga bo‘lib, umumlashtirilgan trapetsiyalar formulasidan foydalanamiz. Buning uchun

$$h = \frac{b-a}{n}, \quad x_i = a + ih, \quad K_{ij} = K(x_i, x_j), \quad \varphi_{ki} = \varphi_k(x_i)$$

deb belgilaymiz hamda $\varphi_k(x_i)$, $u_a(x_i)$ larning taxminiy qiymatini mos ravishda $\tilde{\varphi}_{ki}, \tilde{y}_{ni}$

orqali belgilab, quyidagiga ega bo‘lamiz:

$$\varphi_{k+1}(x_i)=\int_0^{x_i}K(x_i,s)\varphi_k(s)ds\cong$$

$$\cong\frac{h}{2}\left[K_{i0}\varphi_{k0}+2\ \ K_{i1}\varphi_{k1}+K_{i2}\varphi_{k2}+\cdots+K_{i-1}\varphi_{k,i-1}\ \ +K_{ii}\varphi_{ki}\right]$$

$$\tilde{\varphi}_{k+1,i}=\frac{h}{2}\left[K_{i0}\tilde{\varphi}_{k0}+2\ \ K_{i1}\tilde{\varphi}_{k1}+K_{i2}\tilde{\varphi}_{k2}+\cdots+K_{i,i-1}\tilde{\varphi}_{k,i-1}\ \ +K_{ii}\tilde{\varphi}_{ki}\right],\quad i=1,2,\ldots,n.$$

Barcha $\tilde{\varphi}_{ki}(k=\overline{0,n})$ni hisoblab bo'lganidan keyin $u_n(x_i)$ ning $\tilde{y}_{ni}$ taxminiy qiymati

$$\tilde{y}_{ni}=\sum_{k=0}^{n}\lambda^k\tilde{\varphi}_{ki}$$

formula yordamida aniqlanadi.

Boshqa kvadratura formulalarini ham qo'llash mumkin. Masalan, $x_i=a+ih,\quad h=\dfrac{b-a}{2n}$ nuqtalar yordamida $[a,b]$ oraliqni $2n$ bo‘lakka bo‘lib

$$\varphi_{k+1,2i}=\varphi_{k+1}(x_{2i})=\int_0^{x_{2i}}K(x_{2i},s)\varphi_k(s)ds$$

integralga Simpson formulasini qo'llab, taxminiy qiymat uchun quyidagi formulaga ega bo'lamiz:

$$\tilde{\varphi}_{k+1,2i}=\frac{h}{3}\left[K_{2i,0}\tilde{\varphi}_{k0}+4(K_{2i,1}\tilde{\varphi}_{k1}+K_{2i,3}\tilde{\varphi}_{k3}+K_{2i,2i-1}\tilde{\varphi}_{k,2i-1})\right]+$$

$$+\frac{2h}{3}\left[(K_{2i,2}\tilde{\varphi}_{k2}+K_{2i,4}\tilde{\varphi}_{k4}+\cdots+K_{2i,2i-2}\tilde{\varphi}_{k,2i-2})+K_{2i,2i}\tilde{\varphi}_{k,2i}\right],\qquad i=1,2,\ldots,n$$

Toq *i* lar uchun $\tilde{\varphi}_{k+1,i}$ interpolyatsiya yo'li bilan topiladi.

Ketma-ket yaqinlashish jarayonini

$$\frac{\| y_k-y_{k+1}\|}{\| y_k\|}\le\varepsilon$$

shart bajarilguncha davom ettirish kerak, bunda $\| y\| \ =\max_{\alpha\le x\le\delta}|y(x)|,\varepsilon\ >0$ — berilgan nisbiy xatolik. Bu shart shuni ko‘rsatadiki, jarayonni to‘xtatish uchun ikkita qo‘shni ketma-ket yaqinlashishlar natijasini solishtirish kerak. Agar ular yaqin bo‘lsa, u holda kerakli aniqlikka erishilgan deb hisoblash mumkin.

(1.2) tenglamani taqribiy yechish uchun unga kiradigan integralni to‘g‘ridan-to‘g‘ri biror kvadratura formulasiga almashtirish mumkin. Yuqorida ko‘rganimizdek, bu maqsadda umumlashtirilgan trapetsiyalar formulasini qo‘llash maqbuldir. Mazkur formulalarni qo‘llab, quyidagiga ega bo‘lamiz:

$$u(x_i)-\lambda\int_0^{x_i}K(x_i,s)u(s)ds \geq y_i-\frac{\lambda h}{2}\left[K_{i0}y_0+2\left(K_{i1}y_1+K_{i2}y_2+\ldots+K_{i,l-1}y_{l-1}\right)+K_{il}y_l\right]=f$$

yoki

$$y_i-\frac{h\lambda}{2}\left[K_{i0}y_0+2\left(K_{i1}y_1+\ldots+K_{i,l-1}y_{l-1}\right)+K_{il}y_l\right]=f_i,$$

bundan esa

$$y_i=\frac{1}{1-\frac{h\lambda}{2}K_{ii}}\left[f_i+\frac{\lambda h}{2}K_{i0}y_0+\lambda h\sum_{\substack{j=1\\ j\neq i}}^{l-1}K_{ij}y_j\right]$$

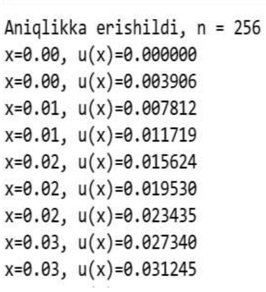
kelib chiqadi. Shunday qilib, biz qadam-ba-qadam y_i larni topib olamiz.

Python dasturlash tilida yuqoridagi tenlgama uchun Simpson usulida yechim

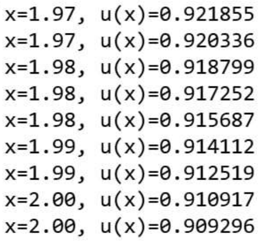
```
import numpy as np
eps = 1e-5
n = 4
max_iter = 10
def solve_volterra(n):
    h = 2 / n
    x = np.linspace(0, 2, n+1)
    u = np.zeros(n+1)
    for i in range(n+1):
        s = 0.0
        for j in range(i+1):
            if j == 0 or j == i:
                w = 1
            elif j % 2 == 1:
                w = 4
            else:
                w = 2
            s += w * (x[j] - x[i]) * u[j]
        u[i] = x[i] + (h/3) * s
```

```
    return x, u
for k in range(max_iter):
    x1, u1 = solve_volterra(n)
    x2, u2 = solve_volterra(2*n)
    if np.max(np.abs(u2[::2] - u1)) < eps:
        break
    n *= 2
print("Aniqlikka erishildi, n =", n)
for xi, ui in zip(x2, u2):
    print(f"x={xi:.2f}, u(x)={ui:.6f}")
```

Natija:



```
Aniqlikka erishildi, n = 256
x=0.00, u(x)=0.000000
x=0.00, u(x)=0.003906
x=0.01, u(x)=0.007812
x=0.01, u(x)=0.011719
x=0.02, u(x)=0.015624
x=0.02, u(x)=0.019530
x=0.02, u(x)=0.023435
x=0.03, u(x)=0.027340
x=0.03, u(x)=0.031245
```



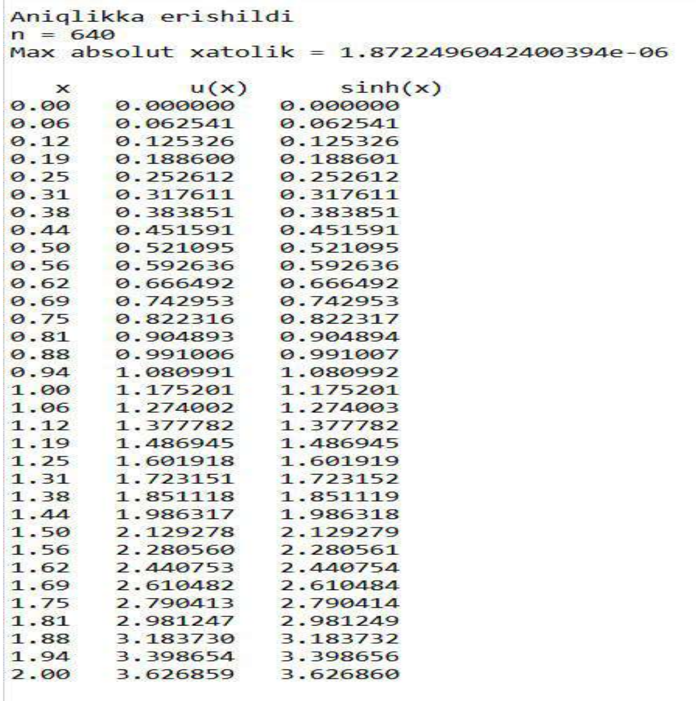
```
x=1.97, u(x)=0.921855
x=1.97, u(x)=0.920336
x=1.98, u(x)=0.918799
x=1.98, u(x)=0.917252
x=1.98, u(x)=0.915687
x=1.99, u(x)=0.914112
x=1.99, u(x)=0.912519
x=2.00, u(x)=0.910917
x=2.00, u(x)=0.909296
```

Python dasturlash tilida yuqoridagi tenlgama uchun Trapetsiya usulda yechim

```
import numpy as np
eps = 1e-5
n = 10
max_iter = 12
def solve_volterra_trap(n):
    h = 2 / n
    x = np.linspace(0, 2, n+1)
    u = np.zeros(n+1)
    for i in range(n+1):
        S = 0.0
        for j in range(1, i):
            S += (x[i] - x[j]) * u[j]
        u[i] = x[i] + h * (0.5 * x[i] * u[0] + S)
    return x, u
for _ in range(max_iter):
```

```
    x1, u1 = solve_volterra_trap(n)
    x2, u2 = solve_volterra_trap(2*n)
    if np.max(np.abs(u2[::2] - u1)) < eps:
        break
    n *= 2
u_exact = np.sinh(x2)
err = np.max(np.abs(u2 - u_exact))
print("Aniqlikka erishildi")
print("n =", n)
print("Max absolut xatolik =", err)
print("\n   x        u(x)      sinh(x)")
for xi, ui, ue in zip(x2[::40], u2[::40], u_exact[::40]):
    print(f"{xi:4.2f}   {ui:8.6f}   {ue:8.6f}")
```

Natija:



```
Aniqlikka erishildi
n = 640
Max absolut xatolik = 1.8722496042400394e-06

   x        u(x)      sinh(x)
0.00   0.000000   0.000000
0.06   0.062541   0.062541
0.12   0.125326   0.125326
0.19   0.188600   0.188601
0.25   0.252612   0.252612
0.31   0.317611   0.317611
0.38   0.383851   0.383851
0.44   0.451591   0.451591
0.50   0.521095   0.521095
0.56   0.592636   0.592636
0.62   0.666492   0.666492
0.69   0.742953   0.742953
0.75   0.822316   0.822317
0.81   0.904893   0.904894
0.88   0.991006   0.991007
0.94   1.080991   1.080992
1.00   1.175201   1.175201
1.06   1.274002   1.274003
1.12   1.377782   1.377782
1.19   1.486945   1.486945
1.25   1.601918   1.601919
1.31   1.723151   1.723152
1.38   1.851118   1.851119
1.44   1.986317   1.986318
1.50   2.129278   2.129279
1.56   2.280560   2.280561
1.62   2.440753   2.440754
1.69   2.610482   2.610484
1.75   2.790413   2.790414
1.81   2.981247   2.981249
1.88   3.183730   3.183732
1.94   3.398654   3.398656
2.00   3.626859   3.626860
```

Xulosa. Mazkur ishda Volterra ikkinchi tur integral tenglamalarini taqribiy yechishda kvadratura formulalariga asoslangan trapetsiya va Simpson usullarining samaradorligi tahlil qilindi. Integralni diskretlashtirish orqali tenglama algebraik koʻrinishga keltirilib, yechim ketma-ket yaqinlashish usuli yordamida qurildi.

Tadqiqot natijalari trapetsiya usulining soddaligi va hisoblash barqarorligi tufayli Volterra tenglamalari uchun qulay ekanligini, Simpson usuli esa silliq funksiyalar uchun yuqori aniqlik va tez konvergentsiyani ta’minlashini ko‘rsatdi. Amaliy hisoblashlarda nazariy xatolik baholari ko‘pincha ortiqcha bo‘lib, real yechim kutilganidan tezroq yaqinlashadi. Shuning uchun, masala xususiyatlarini hisobga olgan holda usul tanlash yoki ularni birgalikda qo‘llash integral tenglamalarni sonli yechishda optimal natija beradi.

FOYDALANILGAN ADABIYOTLAR

1. Abdullayev R.A., Isroilov M.I. "Sonli usullar" – T.: "O‘qituvchi", 2018. – 245 b.
2. Chapra S.C., Canale R.P."Numerical Methods for Engineers", 7th ed. – McGraw-Hill, 2015. – 960 p.
3. Xudoyberganov G‘., Tojiboyev O. "Dasturlash asoslari: C# va Python misollarida" – T.: "Innovatsiya", 2021. – 318 b.
4. Quarteroni A., Saleri F. "Scientific Computing with MATLAB and Octave", 4th ed. – Springer, 2014.
5. Forsythe G.E., Malcolm M.A., Moler C.B. "Computer Methods for Mathematical Computations" – Prentice-Hall, 1977.
6. M.Isroilov “ Hisoblash metodlari 2-qism ” – 2008 .
7. https://replit.com/languages/python3
8. https://share.google/9QTfown5mPmWVdYSO