Решение одномерного интегрального уравнения II рода методом механических квадратур

Постановка задачи

Решить данное интегральное уравнение II рода методом механических квадратур, пользуясь формулой Гаусса.

$$u(x) - \int_a^b K(x,t)u(t)dt = f(x) \tag{1}$$

Условие задачи:

$$\begin{aligned} &K(x,t) = \frac{1-\alpha e^{-(x+t)}}{1+\alpha e^{-(x+t)}} \\ &\alpha = 0.25 \\ &f(x) = 1 \\ &[a,b] = [0,1] \end{aligned} \tag{2}$$

Вывод формул

Метод состоит в использовании для вычисления интегралов квадратурной формулы Гаусса:

$$\int_a^b f(x)dx \approx \sum_{i=1}^n A_i f(\xi_i) \tag{3}$$

Узлы квадратурной формулы ξ_i - корни полинома Лежандра степени n.
Корни можно найти методом итераций:

$$\xi_i^{(0)} = cos(\pi\frac{4i-1}{4n+2}); \quad \xi_i^{(k+1)} = \xi_i^{(k)} - \frac{P_n(\xi_i^{(k)})}{P_n'(\xi_i^{(k)})} \tag{4}$$

$$P_n'(x) = \frac{n}{1-x^2}(P_{n-1}(x) - xP_n(x)) \tag{5}$$

Веса рассчитываются по формуле:

$$A_i = \frac{2}{(1-\xi_i^2)(P_n'(\xi_i))^2} \tag{6}$$

В результате получаем уравнение:

$$\tilde{u}(x) - \sum_{i=1}^n A_i K(x,x_i)\tilde{u}(x_i) = f(x) \tag{7}$$

Полагая $x = x_k$, $u_k = \tilde{u}(x_k)$, имеем:

$$u_k - \sum_{k=1}^n A_k K(x_k,x_i)u_k = f(x_k) \tag{8}$$

Текст программы

```
eps = 0.000000001; alp = 0.01
n = raw_input('n = '); n = int(n)
f = ones(n); AA = zeros((n,n))
def poli_lezh(x,n):
        p0=1; p1=x
        for i in range(2,n+1):
                pi = 1.0/float(i) * ( (2.0*i - 1.0)*x*p1 - (i - 1.0)*p0 )
                p0 = p1; p1 = pi
        pp = n / (1 - x**2) * ( p0 - x*p1)
        return pi, pp
def roots_lezh(n):
        t = []
        for i in range(1,n+1):
                tk = -2
                t0 = np.cos (np.pi * (4.0*i - 1.0) / (4.0 * n + 2) )
                while abs(tk - t0) > eps:
                        tk = t0 - poli_lezh(t0,n)[0] / poli_lezh(t0,n)[1]
                        t0 = tk; t.append(tk)
        return t
def weights_lezh(n):
        a = []; r = roots_lezh(n)
        for i in range(n):
                a.append ( 2.0 / (1 - r[i]**2) / poli_lezh(r[i],n)[1]**2 )
        return a
roots = roots_lezh(n); weights = weights_lezh(n)
for i in range(n):
        roots[i] = 0.5*roots[i] + 0.5
        weights[i] = weights[i]/2.0
for i in range(n):
        for k in range(n):
                AA[i,k] = weights[k]*(1.0-alp*np.exp(-(roots[k]+roots[i])))/(1.0+alp*np
    .exp(-(roots[k]+roots[i])))
        AA[i,i] = AA[i,i] + 1.0
B = linalg.inv(AA)
u = dot (B,f)
```

Результаты

x	u
0.98696288791	0.500376123146
0.932533213944	0.500507828426
0.839705137015	0.500749569572
0.716697758125	0.501106273226
0.57443717155	0.501576998263
0.42556282845	0.502146202778
0.283302241875	0.50277448527
0.160294862985	0.503393819481
0.0674667860562	0.503913681275
0.0130371120903	0.504241414095

График решения

Решение на сетке из 10 узлов: