

```

1  from sympy import symbols, zeros, diff, exp
2
3  a = symbols('a')
4  f = exp(a) - 2
5
6  eps = 0.1
7  k = 0.1
8
9  N_max = 50
10 x = zeros(N_max)
11
12 x[0] = -2.
13
14 for n in range(N_max) :
15     x[n + 1] = (x[n] - f.subs(a,x[n])/diff(f,a).subs(a,x[n])).evalf()
16     psi_0 = f.subs(a,x[n])**2 ; psi_1 = f.subs(a,x[n + 1])**2
17     tau_n = (psi_0 + k*psi_1)/(psi_0 + psi_1)
18     x[n + 1] = (x[n] - tau_n*f.subs(a,x[n])/diff(f,a).subs(a,x[n])).evalf()
19     if n >= 1 and abs((x[n+1]-x[n])/(1-(x[n+1]-x[n])/(x[n]-x[n-1])))<eps :
20         k = n + 1
21         break
22
23 print('Найден корень x = {0:.2f}, число итераций - {1}'.format(x[k],k))
24
25 # Листинг программы, реализующей решение нелинейного уравнения f(x)=0
26 # с помощью обобщённого метода Ньютона

```