Лабораторная работа №3
Ильин Денис, z3243

Задание 1

Второй способ редукции плох, так как возрастает относительная погрешность возрастает при суммировании, к тому же получается грубое вычисление, способное убить первое слагаемое. Первый вариант лучше, взятие корней квадратных.

```
Мы хотим вычислить ln(x), x>>1 (в коде есть "защита от дурака"),
Вычислим его с помощью редукции к аргументу близкому к
единице

def red(x,eps):
   i=0
   if x==1:
      return [x,0]
   else:
      if x>1:
         y=x
      else: y=1/x
      while y-1 > eps:
         y=np.sqrt(y)
         i=i+1
      return [y,i]

eps=желаемая точность
```

Задание 2

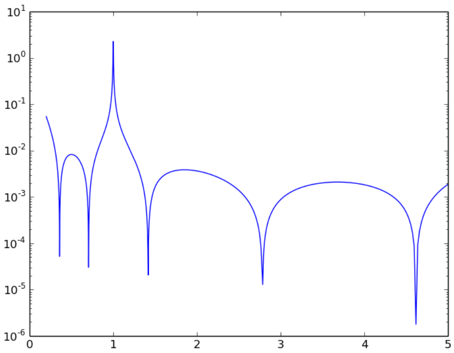
Выполним сначала редукцию аргумента, так чтобы a<1, если изначально a<0, то заменим ln(1/x) на -ln(x), короче говоря свежем все к случаю 0<a<1, тогда предпоследние звено сходится:

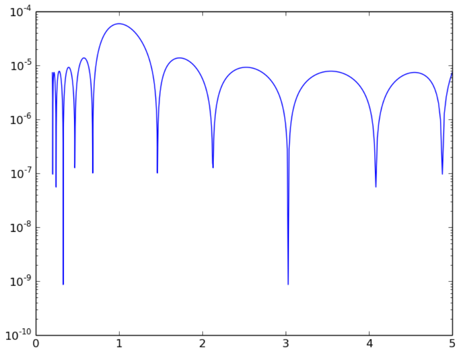
$$|R_N| = |\frac{a^{N+1}}{(N+1)!} * \frac{\mathrm{d}^{N+1}}{\mathrm{d}a^{N+1}} ln(\theta a)| = |\frac{a^{N+1}}{N+1} * \frac{1}{(1+\theta a)^{N+1}}| \leq |\frac{a^{N+1}}{N+1}| \leq \delta$$

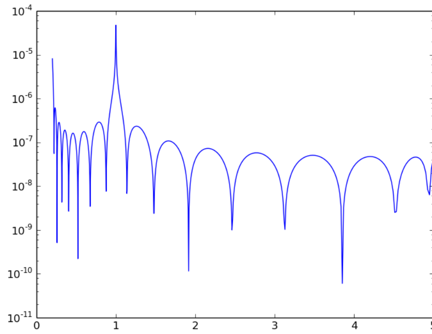
Задание 3

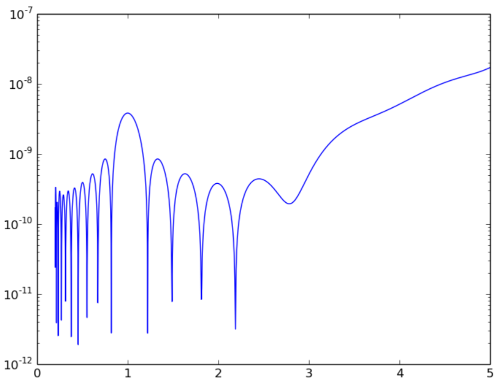
```
# Узлы итерполяции
N_u = 5
un_u = np.cos(np.pi * (np.arange(N_u) + .5) / (N_u + 1))
yn_u = np.log((1 + 2*un_u/3) / (1 - 2*un_u/3))
# Тестовые точки
u_u=np.linspace(-1,1,1000)
# Многочлен лагранжа

L_u=scipy.interpolate.lagrange(un_u,yn_u)

x_u = (1 + 2 * u_u / 3) / (1-2 * u_u / 3)
y_u=np.log(x_u)
yl_u=L_u((3*(x_u-1))/(2*(x_u+1)))
```

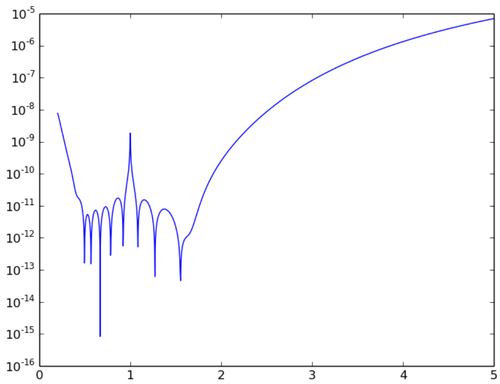
N=5,10,15,25







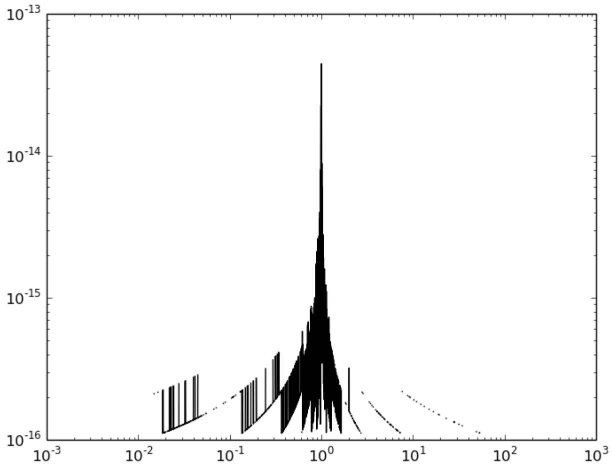


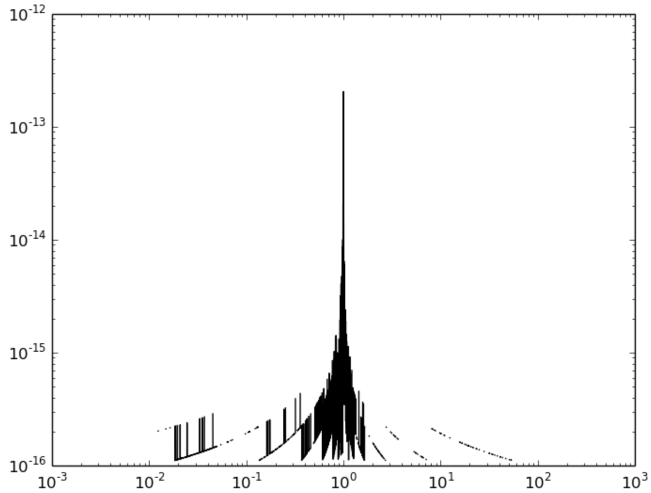


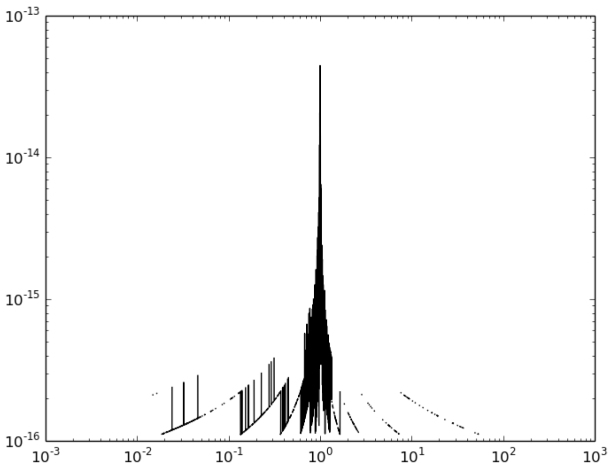
Задание 4

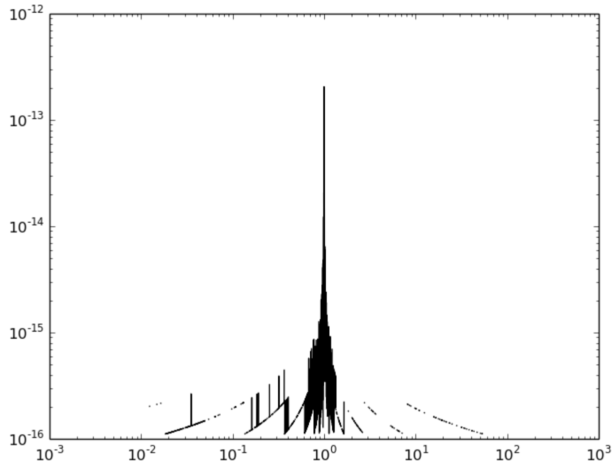
```
Используем экспоненту числа
def log_newton(x, N=5):
    y=np.frexp(x)[1]*np.log(2)
    for j in range(N):
        y=y-1+x/np.exp(y)
    return y
```

N=5,6,7,10





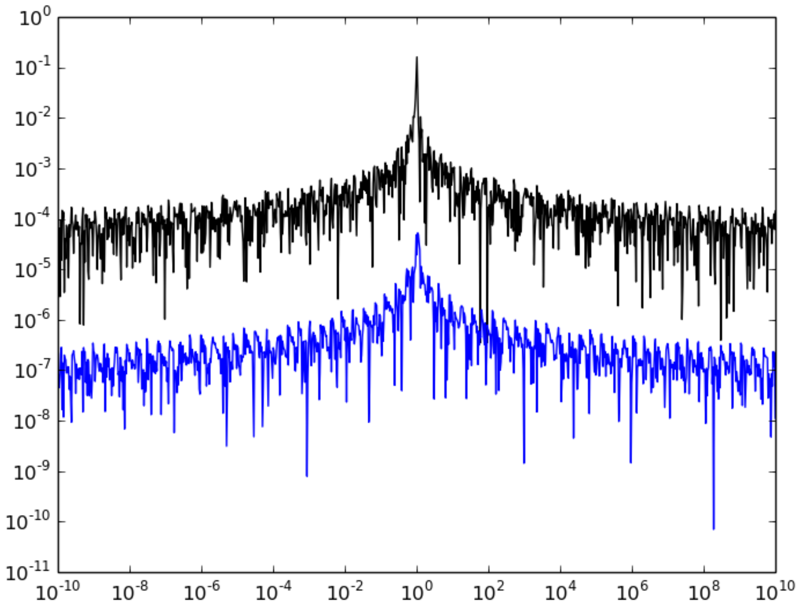




Наибольшая точность достигается при N=6, затем наблюдается ухудшение точности. При больших N большая часть погрешности привносится за счет большого количества сумм и разностей слагаемых. В Единице мы наблюдаем четко выраженный экстремум, так как при y~1, операция y-1 убивает всю точность, к тому же вычисление exp(y) также портит картину.

Задание 5

Линейная аппроксимация
```
table2=np.log((np.arange(1,2**B+1,
dtype=np.double))/(2**B))

def log_table2(x):
    M,E=np.frexp(x)
    arg = M*2**B
    num = arg.astype(int)
    delta = arg - num
    return log2*E+(table2[num-1] + (table2[num] -
table2[num-1]) * delta)
```



При увеличении степени точность увеличивается, до тех пор пока погрешности сумм слагаемых не вносит значительный вклад