Задача 1

Найти точку минимума и минимум функции $f(x) = -e^{-x} ln x + a \bullet x$ на интервале [0,5; 2,5] с точностью ε=0,1; 0,01; 0,001 в зависимости от параметра α.

Применить для решения задачи все рассмотренные методы минимизации функции.

Вариант 3

α = 0.02

In [ ]:
```
import math
import numpy as np
```

In [ ]:
```
x1 = 0.5
x2 = 2.5
A = 0.02
e = 0.1
```

In [ ]:
```
def f(x):
    return -1*math.exp(-1*x) * math.log(x) + A * x
```

Метод перебора

In [4]:
```
n = (x2 - x1) / e
X = np.arange(x1, x2, e)
Y = np.zeros(len(X))
y_min = f(x1)
x_min = x1

for i in range( int(n) ):
    Y[i] = f(X[i])
    print(f"f({X[i] : .3f}) = {Y[i] : .4f}")
    if Y[i] < y_min:
        y_min = Y[i]
        x_min = X[i]

print()
print(f"min: f({x_min : .3f} ) = {y_min : .4f}")
```

```
f( 0.500) =  0.4304
f( 0.600) =  0.2923
f( 0.700) =  0.1911
f( 0.800) =  0.1163
f( 0.900) =  0.0608
f( 1.000) =  0.0200
f( 1.100) = -0.0097
f( 1.200) = -0.0309
f( 1.300) = -0.0455
f( 1.400) = -0.0550
f( 1.500) = -0.0605
f( 1.600) = -0.0629
f( 1.700) = -0.0629
f( 1.800) = -0.0612
f( 1.900) = -0.0580
f( 2.000) = -0.0538
f( 2.100) = -0.0489
f( 2.200) = -0.0434
f( 2.300) = -0.0375
f( 2.400) = -0.0314

min: f( 1.700 ) = -0.0629
```

Метод дихотомии

In [5]:
```
d = 0.02
a = x1
b = x2

while e < (b - a)/2 :
    x1 = (b + a - d) / 2;
    x2 = (b + a + d) / 2;

    if (f(x1) < f(x2)):
        b = x1;
    else:
        a = x2
        print(f'a = {a :0.3f} | b = {b :0.3f} | x1 = {x1 :0.3f} | x2 = {x2 :0.3f} | f(x1) = 
{f(x1) :0.4f} | f(x2) = {f(x2) :0.4f}')

x_min = (a + b ) / 2
y_min = f(x_min)

print()
print(f"min: f( {x_min :0.3f} ) = {y_min :0.4f}")
```

```
a = 1.510 | b = 2.500 | x1 = 1.490 | x2 = 1.510 | f(x1) = -0.0601 | f(x2) = -0.0608
a = 1.636 | b = 1.742 | x1 = 1.616 | x2 = 1.636 | f(x1) = -0.0630 | f(x2) = -0.0632

min: f( 1.689 ) = -0.0630
```

Метод средней точки

In [6]:
```
e = 0.01
a = x1
b = x2
x_min = (a+b) / 2

def df(x):
    return math.exp(-1*x) * math.log(x) - (math.exp(-1*x)/x) +A

while e < abs( df(x_min) ):
    if df(x_min) > 0:
        print(f"a = {a :0.3f} | b = {b :0.3f} | x_cp = {x_min :0.3f} | f'(x_cp) = {df(x_mi
n) :0.4f} | + ")
        b = x_min
    else:
        print(f"a = {a :0.3f} | b = {b :0.3f} | x_cp = {x_min :0.3f} | f'(x_cp) = {df(x_mi
n) :0.4f} | - ")
        a = x_min
    x_min = (a+b) / 2

print(f"a = {a :0.3f} | b = {b :0.3f} | x_cp = {x_min :0.3f} | f'(x_cp) = {df(x_min) :0.4f}
| Условие точности выполнено ")
print(f"min: f( {x_min :0.3f} ) = {f(x_min) :0.4f}")
```

```
a = 1.616 | b = 1.636 | x_cp = 1.626 | f'(x_cp) = -0.0053 | Условие точности выполнено
min: f( 1.626 ) = -0.0631
```

Метод Ньютона

In [7]:
```
def ddf(x):
    return math.exp(-1*x) * ( -1* math.log(x) + 2/x + 1/x**2)

def xk_1(xk):
    return xk - df(xk) / ddf(xk)

xk = 0.7
while e < abs( df(xk) ):
    print(f"xk = {xk :0.3f} | f'(xk) = {df(xk) :0.4f}")
    xk = xk_1(xk)

print(f"xk = {xk :0.3f} | f'(xk) = {df(xk) :0.5f} | Условие точности выполнено")
print(f"min: f( {xk :0.3f} ) = {f(xk) :0.4f}")
```

```
xk = 0.700 | f'(xk) = -0.8665
xk = 1.032 | f'(xk) = -0.3139
xk = 1.342 | f'(xk) = -0.0979
xk = 1.556 | f'(xk) = -0.0224
xk = 1.640 | f'(xk) = -0.00229 | Условие точности выполнено
min: f( 1.640 ) = -0.0632
```