为求 $I=\frac{2}{\sqrt{\pi}}\int_0^1 e^{-x}dx$，输入函数句柄 f=@(x)2*exp(-x)/sqrt(pi)并运行函数计算，结果如图：

```
>> FTrapezoid_zty(f,0,1,10000)

ans =

   0.713384508186018

>> FSimpson_zty(f,0,1,5)

ans =

   0.713272065465711

>> TGauss_zty(f,0,1)

ans =

   0.713271327590424

>> Romberg_zty(f,0,1,10^-5)

ans =

   0.713271669814180

>> fprintf('10182499周天翊\n')
10182499周天翊
```

已知 $I=0.713271669674\cdots\cdots$，对以上四种算法分析如下：

1. 复化梯形公式

将求积区间 n 等分，对每段区间上使用梯形公式然后求和，算法简单，但精度较低，经过 10000 次分割后精度仍只有 4 位，难以满足 10^-5 的精度要求。

```
function S = FTrapezoid_zty( f,a,b,n )

S=0;
for i=0:n-1
    S=S+2*feval(f,a+i*(b-a)/n);
end
S=(b-a)*(S+feval(f,a)+feval(f,b))/(2*n);

end
```

2. 复化辛普森公式

将求积区间 n 等分，对每段区间上使用辛普森公式然后求和，算法同样较为简单，而精度相比复化梯形公式相当高，仅 5 等分区间就能满足 5 位精度要求，但继续和另外两种算法对比来说相同计算量下精度仍不高。

```
function S = FSimpson_zty( f,a,b,N )

h=(b-a)/N;
fa=feval(f,a);
fb=feval(f,b);
S=fa+fb;
x=a;
for i=1:N-1
    x=x+h/2;
    fx=feval(f,x);
    S=S+4*fx;
    x=x+h/2;
    fx=feval(f,x);
    S=S+2*fx;
end
S=S+4*feval(f,x+h/2);
S=h*S/6;
end
```

3. 三点高斯公式

数学推导得出的高斯公式在取三点近似计算的情况下有最高的精度，本题中得到了 6 位精度，但求得高斯公式的过程相当繁琐，这使得高斯公式应用较为有限。

```
function G =TGauss_zty( f,a,b )

x1=(a+b)/2-sqrt(3/5)*(b-a)/2;
x2=(a+b)/2+sqrt(3/5)*(b-a)/2;
G=(b-a)*(5*feval(f,x1)/9+8*feval(f,(a+b)/2)/9+5*feval(f,x2)/9)/2;

end
```

4. 龙贝格方法

龙贝格加速算法对牛顿-科特斯公式利用松弛技术进行数据加工，在数据满足精度要求时算法停止，可以在非常小的计算量代价下求得高精度的积分值，是四种算法中最实用的。

```
function [ quad,R ] = Romberg_zty( f,a,b,eps )

h=b-a;
R(1,1)=h*(feval(f,a)+feval(f,b))/2;
M=1; J=0; err=1;
while err>eps
    J=J+1;
    h=h/2;
    S=0;
    for p=1:M
        x=a+h*(2*p-1);
        S=S+feval(f,x);
    end
    R(J+1,1)=R(J,1)/2+h*S;
    M=2*M;
    for k=1:J
        R(J+1,k+1)=R(J+1,k)+(R(J+1,k)-R(J,k))/(4^k-1);
    end
    err=abs(R(J+1,J)-R(J+1,J+1));
end
quad=R(J+1,J+1);
end
```