1 在计算机上编程实现 [复化梯形公式] 及 [复化Simpson公式] 的计算，并以课本 例2 检验程序

In [15]:

```
y = [1.0000000, 0.9973978, 0.9896158, 0.9767267, 0.9588510, 0.9361556, 0.9088516, 0.8771925, 0.

def Tn(a, b):
    re = 0.0
    div = 1
    n = (len(y) - 1) / div
    h = (b - a) * 1.0 / n
    for i in range(n * div):
        re += y[i] + y[i + 1]
    return re * h / 2

print 'Tn =', Tn(0, 1)

def Sn(a, b):
    re = 0.0
    div = 2
    n = (len(y) - 1) / 2
    h = (b - a) * 1.0 / n
    for i in range(0, n * div, 2):
        re += y[i] + 4 * y[i + 1] + y[i + 2]
    return re * h / 6

print 'Sn =', Sn(0, 1)
```

```
Tn = 0.94569080625
Sn = 0.946083254167
```

2 在计算机上编程实现 [变步长的梯形法] 的计算，并以课本 例3 检验程序

In [14]:

```
y = [1.0000000, 0.9973978, 0.9896158, 0.9767267, 0.9588510, 0.9361556, 0.9088516, 0.8771925, 0.
T = []

def TXF(a, b):
    h = (b - a) * 1.0
    T.append((y[0] + y[-1]) * h / 2)
```

3 在计算机上编程计算 P88 17、18、20

3-1 17题

In [16]:

```
y = [1.00000, 1.65534, 1.55152, 1.06666, 0.72159]

def Tn(a, b):
    re = 0.0
    div = 1
    n = (len(y) - 1) / div
    h = (b - a) * 1.0 / n
    for i in range(n * div):
        re += y[i] + y[i + 1]
    return re * h / 2

print 'Tn =', Tn(0, 1)

def Sn(a, b):
    re = 0.0
    div = 2
    n = (len(y) - 1) / 2
    h = (b - a) * 1.0 / n
    for i in range(0, n * div, 2):
        re += y[i] + 4 * y[i + 1] + y[i + 2]
    return re * h / 6

print 'Sn =', Sn(0, 1)
```

```
Tn = 1.28357875
Sn = 1.30938583333
```

3-2 18题

In [ ]:

```

```

3-3 20题

In [ ]:

```

```

编程实现 [Romberg方法] 的计算，并以课本 例4 检验程序

In [ ]:

```

```