1)

$\because f(1)=1, f(2)=1, f(4)=2$

$\therefore l_0(x)=\dfrac{(x-2)(x-4)}{3}$

$l_1(x)=\dfrac{(x-1)(x-4)}{-2}$

$l_2(x)=\dfrac{(x-1)(x-2)}{6}$

$\therefore \phi(x)=\dfrac{x^2-3x+8}{6}$

2.a)

对$\phi(x)=1$在$x=x_i$处进行n次lagrange插值

$\therefore \sum y_i l_i(x)=\sum l_i(x)=\phi(x)=1$

2.b)

对$\phi(x)=x^j$在$x=x_i$处进行n次lagrange插值

$\therefore \sum x_i^j l_i(x)=\phi(x)=x^j$

2.c)

利用上题结论，对$(x_i-x)^j$展开可得

$\therefore \sum (x_i-x)^j l_i(x)=\sum(\sum(-1)^{j-k}C_j^k x_i^k x^{j-k})l_i(x)=\sum(-1)^{j-k}C_j^k x^{j-k}x^k=0$

2.d)

利用前几题易知

$\sum l_i(0)x_i^j=1, \quad j=0$

$=0, \quad j=1,\dots,n$

而对于n+1次的$f(x)=x^{n+1}$进行lagrange插值，由余项公式知

$f(x)=\phi(x)+R(x)=\sum x_i^{n+1}l_i(x)+\frac{1}{(n+1)!}f^{(n+1)}(\xi)w_{n+1}(x)=\sum x_i^{n+1}l_i(x)+w_{n+1}(x)=x^{n+1}$

令$x=0$有

$\sum l_i(0)x_i^{n+1}=-w_{n+1}(0)=(-1)^n\prod x_i$

得证

3)

$\because |R(x)| \leq \frac{1}{8}(x_1 - x_0)^2 max|f^{(2)}(x)|$

$\therefore h \leq \sqrt{8 * 10^{-6}/max|f^{(2)}(x)|}$

$\because f^{(1)}(x) = \dfrac{1}{\pi}\displaystyle\int_0^{\pi} -sintsin(xsint)dt$

$f^{(2)}(x) = \dfrac{1}{\pi}\displaystyle\int_0^{\pi} -sin^2tcos(xsint)dt \leq \dfrac{1}{\pi}\displaystyle\int_0^{\pi} sin^2tdt = \dfrac{1}{2}$

$\therefore h \leq 4 * 10^{-3}$

4)

对$f(x)$进行k次插值，由于$f(x)$是n次多项式，$k > n$，所以

$\phi(x) = f(x) = f(x_0) + \sum\limits_{i=1}^{k} f[x_0, \ldots, x_i] \prod\limits_{j=0}^{i-1}(x - x_j) = f(x_0) + \sum\limits_{i=1}^{n} f[x_0, \ldots, x_i] \prod\limits_{j=0}^{i-1}(x - x_j)$

令$k = n + 1$得

$f[x_0, \ldots, x_k] \prod\limits_{j=0}^{n}(x - x_j) = 0$

$\therefore f[x_0, \ldots, x_k] = 0$

同理，由数学归纳法知$k > n$时

$f[x_0, \ldots, x_k] = 0$

5.1)

利用python进行计算

```
def fx(x):
    return math.exp(x*x)

def lagrange(x, y):
    M = len(x)
    p = 0.0
    for j in range(M):
        pt = y[j]
        for k in range(M):
            if k == j:
                continue
            fac = x[j]-x[k]
            pt *= np.poly1d([1.0, -x[k]])/fac
        p += pt
    return p

xi = [0.6, 0.7, 0.8, 0.9, 1.0]
yi = []
for i in range(5):
    ans = fx(xi[i])
```

```
    yi.append(ans)

Lp = lagrange(xi, yi)
print(Lp)
print(Lp(0.82))
print(Lp(0.98))
'''
Out:
   4         3         2
4 x - 8.319 x + 8.929 x - 3.473 x + 1.581
1.9589096433405886
2.6128479259385866
'''
```

所以得到

$$P(0.82) = 1.959, \quad P(0.98) = 2.613$$

5.2)

$\because f^{(5)}(x) = e^{x^2}(32x^5 + 160x^3 + 120x)$

易知其上界在$x = 1$时取到，利用python计算余项得

```
def f5(x):
    ans = fx(1) * (32+160+120)
    return ans

def cal02(x):
    ans = f5(x)
    for i in range(5):
        ans *= x-xi[i]
        ans /= i+1.0
    return ans

print(cal02(0.82))
print((Lp(0.82)-fx(0.82))/fx(0.82))
print(Lp(0.82)-fx(0.82))
print(cal02(0.98))
print((Lp(0.98)-fx(0.98))/fx(0.98))
print(Lp(0.98)-fx(0.98))
'''
Out:
5.3735865035163305e-05
-1.1985898523732654e-05
-2.3479573626028483e-05
-0.00021657181968717416
4.0786648120852294e-05
0.00010656496252092751
'''
```

由此可知

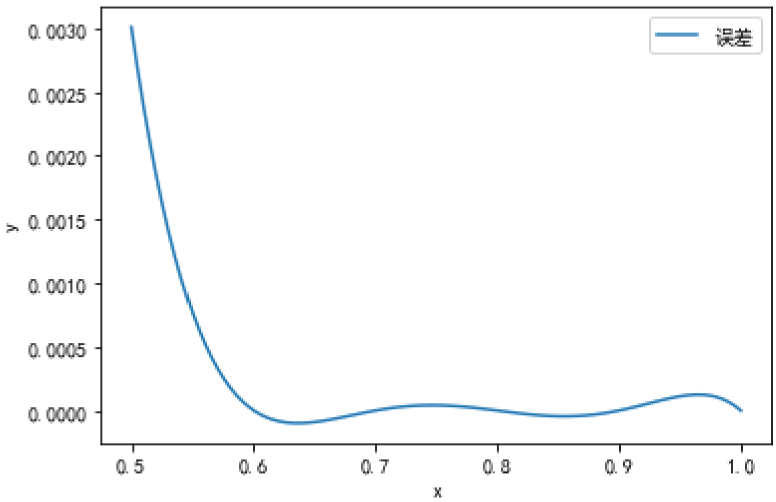
	误差上界	相对误差界	实际误差
x=0.82	5.374e-5	1.199e-5	2.348e-5
x=0.98	2.166e-4	4.079e-5	1.066e-4

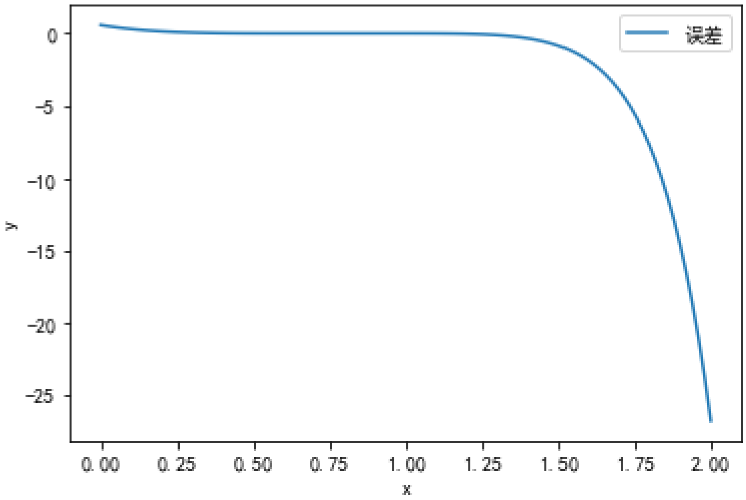
5.3)

利用python画图

```
import matplotlib.pyplot as plt
from pylab import *

def draw(a, b):
    mpl.rcParams['font.sans-serif'] = ['SimHei']
    mpl.rcParams['axes.unicode_minus'] = False
    xj = np.linspace(a, b, 100)
    y = []
    for xjj in xj:
        y.append(Lp(xjj)-fx(xjj))
    plt.plot(xj, y, label='误差')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.legend()

    plt.show()

draw(0.5, 1.0)
draw(0.0, 2.0)
```

得到误差如图





误差
0
−5
−10
−15
−20
−25
y
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
x