Homework 3

自硕 21 崔晏菲 2021210976

注：因为我不会 R 语言，所以代码都是用 Python 写的。代码文件见 homework3-code.ipynb

5.1

解：

证明：

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{1}{2} f''(x_i)(x_{i+1} - x_i)^2 + O(n^{-3})$$

故

$$\begin{aligned}
& p_i(x) \\
&= f(x_i) + (x - x_i)\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \\
&= f(x_i) + (x - x_i)\frac{f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{1}{2} f''(x_i)(x_{i+1} - x_i)^2 + O(n^{-3}) - f(x_i)}{x_{i+1} - x_i} \\
&= f(x_i) + (x - x_i)f'(x_i) + \frac{1}{2}(x - x_i)(x_{i+1} - x_i)f''(x_i) + \frac{x - x_i}{x_{i+1} - x_i} O(n^{-3}) \\
&= f(x_i) + (x - x_i)f'(x_i) + \frac{1}{2}(x - x_i)(x_{i+1} - x_i)f''(x_i) + O(n^{-3})
\end{aligned}$$

证毕

5.3

解：

a.

$$f(\mu, x) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(\mu - \bar{x})^2}{\sigma^2}} \cdot \frac{1}{\pi} \frac{2}{4 + (\mu - 5)^2}$$

故

```python
k = 1/normalcauchy.riemann_integrate(-1e6, 1e6, delta = 1e-2)
print("比例常数k = %f"%(k))
```

✓ 14.0s

```
比例常数k = 7.846538
```

b. Dasds

```
当最慢的方法收敛到0.0001的误差以内时，三种方法的结果如下：
黎曼积分：0.9959602686，与0.99605的差距为0.0000897314
梯形积分：0.9960533589，与0.99605的误差为0.0000033589
辛普森积分：0.9960546648，与0.99605的误差为0.0000046648
```

5.6

解：

a. 证明：

对于高斯-厄米积分规则，有

$$H_0(x) = 1$$
$$H_1(x) = x$$
$$H_k(x) = xH_{k-1}(x) - (k-1)H_{k-2}(x)$$

故

$$H_2(x) = x^2 - 1$$
$$H_3(x) = x^3 - 3x$$
$$H_4(x) = x^4 - 6x^2 + 3$$
$$H_5(x) = x^5 - 10x^3 + 15x$$

故它依赖于$H_5(x) = c(x^5 - 10x^3 + 15x)$

证毕

b.

```python
expectation, c = ns.get_c_hermite(5)
print("期望为: %f, 因此c = %f"%(expectation, c))
```

✓ 0.0s

```
期望为: 120.000000, 因此c = 0.057659
```

证明:

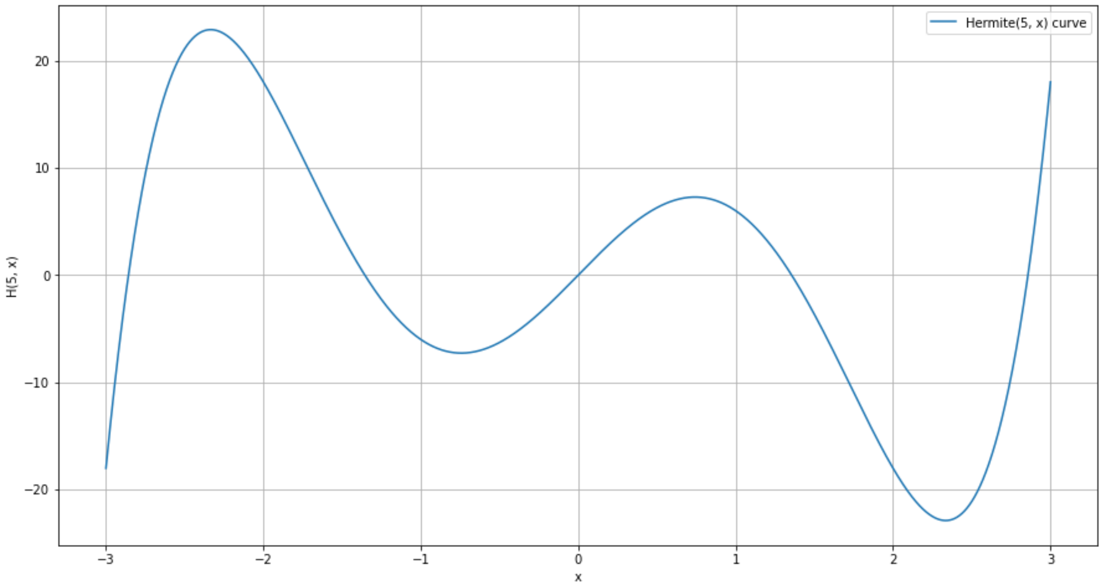
$$\langle H_5(x), H_5(x)\rangle$$
$$= \int_{-\infty}^{+\infty} e^{-\frac{x^2}{2}}\left(H_5(x) \cdot H_5^*(x)\right) \mathrm{d}x$$
$$= \int_{-\infty}^{+\infty} c^2 e^{-\frac{x^2}{2}}(x^{10} - 20x^8 + 130x^6 - 300x^4 + 225x^2)\ \mathrm{d}x$$
$$= \sqrt{2\pi}c^2 E[x^{10} - 20x^8 + 130x^6 - 300x^4 + 225x^2]$$
$$= 1$$

解得

$$c$$
$$= \sqrt{\frac{1}{\sqrt{2\pi}E[x^{10} - 20x^8 + 130x^6 - 300x^4 + 225x^2]}}$$
$$= \frac{1}{\sqrt{120\sqrt{2\pi}}} \approx 0.05765864205256028$$

证毕

c. 图像为




```
roots = ns.solveHermite(5)
N_roots = np.array(sp.N(sp.Matrix(roots))).astype(np.float64).reshape(-1)
print(ns.symHermite(5)[1], "的解为:")
for root in roots:
    print(root, end=", ")
print()
print("具体数值为:")
for root in N_roots:
    print(root, end=", ")
```

✓ 0.0s

```
x**5 - 10*x**3 + 15*x 的解为:
0, -sqrt(5 - sqrt(10)), sqrt(5 - sqrt(10)), -sqrt(sqrt(10) + 5), sqrt(sqrt(10) + 5),
具体数值为:
0.0, -1.355626179974266, 1.355626179974266, -2.8569700138728056, 2.8569700138728056,
```

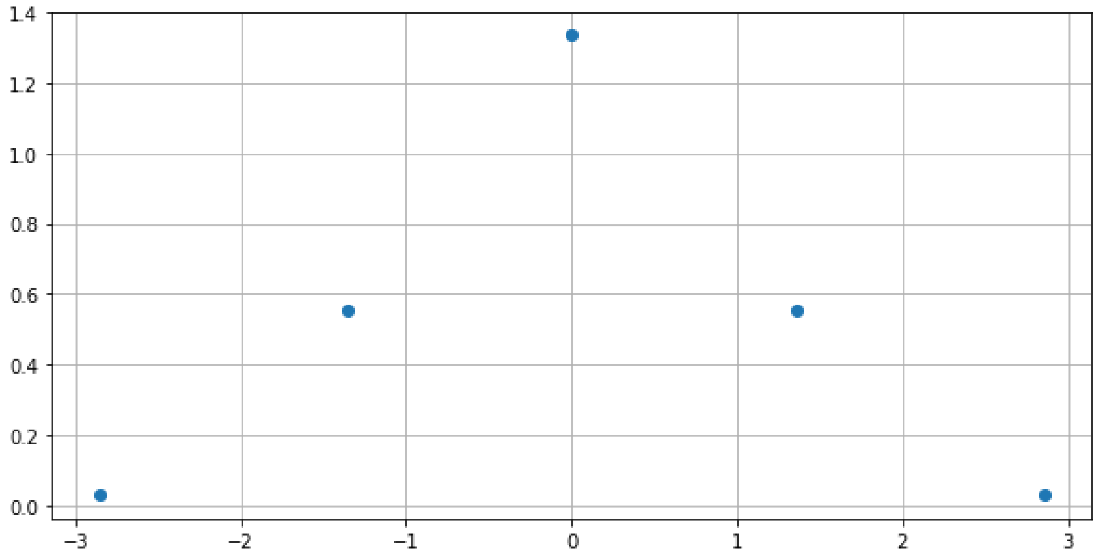
d.

```
weights = ns.NodeWeight(5, N_roots)
print(weights)

fig = plt.figure(figsize=(10,5))
plt.plot(N_roots, weights, "o")
plt.grid(True)
fig.show()
```

✓ 0.1s

```
[1.33686841 0.55666179 0.55666179 0.02821815 0.02821815]
```



e.

```
k = 1/ns.RiemannIntegrate(-1e4, 1e4, n=1e
posterior_var = ns.RiemannIntegrate(-1e4,
print("归一化常数为%f, mu的期望方差为%f"%(k
```

✓ 0.9s

归一化常数为3.715349, mu的期望方差为4.532931