Final Project Report

PB19071501 李平治

1

当$\alpha = 0$时，记

$$\min_{\{u_1,u_2\dots u_{n-1}\}} \sum_{i=1}^{n} (\frac{1}{2}(\frac{u_i - u_{i-1}}{h})^2 h - f_i u_i h) = \min_{\{u_1,u_2\dots u_{n-1}\}} G_n(u_1, u_2 \dots, u_{n-1})$$

求G对每个变量的偏微分，并令它们为0，得到

$$\frac{\partial G}{\partial u_1} = u_1 - u_0 - f_1 h^2 + u_1 - u_2 = 0$$
$$\frac{\partial G}{\partial u_2} = u_2 - u_1 - f_2 h^2 + u_2 - u_3 = 0$$
$$\vdots$$
$$\frac{\partial G}{\partial u_{n-1}} = u_{n-1} - u_{n-2} - f_{n-1} h^2 + u_{n-1} - u_n = 0$$

其中$u_0 = u_n = 0$，因此化简得

$$A_h u_h = f_h$$
$$\begin{pmatrix} \frac{2}{h^2} & -\frac{1}{h^2} & 0 & 0 & \cdots & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & 0 & \cdots & 0 \\ 0 & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & \cdots & 0 \\ 0 & 0 & -\frac{1}{h^2} & \frac{2}{h^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & -\frac{1}{h^2} \\ 0 & 0 & 0 & 0 & -\frac{1}{h^2} & \frac{2}{h^2} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ \vdots \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ \vdots \\ f_{n-1} \end{pmatrix}$$

2

> 为节省空间，求解结果放在附录中

取$e_h = ||u_h - u_e||_2$，$\epsilon = 10^{-10}$，$n = \{10, 20, 40, 80, 160\}$，Jacobi和Gauss-Seidel迭代法求解上述线性方程组误差列表为

	Jacobi iteration error	Gauss-Seidel iteration error
n=10	0.018482034928874364	0.018482032197992628
n=20	0.006510206868033681	0.006510189798674751
n=40	0.002299599922536855	0.002299502237862865
n=80	0.000812978994829882	0.000812425160801029
n=160	0.000288183348886839	0.000285044509734070

3

作log变换，则有$\log(e_h) = \beta \log(h) + b$。

同时，**2**中相同**n**下Jacobi和Gauss-Seidel产生的不同e_h值取平均值，产生n和$\overline{(}e_h)$的一一对应关系

h	$\log(h)$	e_h	$\log(e_h)$
0.1	-2.30258509299404	0.018482034928874364	-3.99095610363200
0.1	-2.30258509299404	0.018482032197992628	-3.99095625139072
0.05	-2.99573227355399	0.006510206868033681	-5.03438404630283
0.05	-2.99573227355399	0.006510189798674751	-5.03438666824429
0.025	-3.68887945411393	0.002299599922536855	-6.07502011790062
0.025	-3.68887945411393	0.002299502237862865	-6.07506259778935
0.0125	-4.38202663467388	0.000812978994829882	-7.11480528536784
0.0125	-4.38202663467388	0.000812425160801029	-7.11548675778609
0.00625	-5.07517381523382	0.000288183348886839	-8.15191365231071
0.00625	-5.07517381523382	0.000285044509734070	-8.16286521538408

记$x = \log(h), y = \log(e_h)$，作最小二乘法：

$$\sum xy = 238.52042154548081$$
$$\overline{x} = -3.688879454113932$$
$$\overline{y} = -6.0745836696108535$$
$$\sum x_i^2 = 145.68737654820305$$
$$\hat{\beta} = \frac{\sum xy - 10\overline{x}\,\overline{y}}{\sum x_i^2 - 10\overline{x}^2} = 1.50237$$

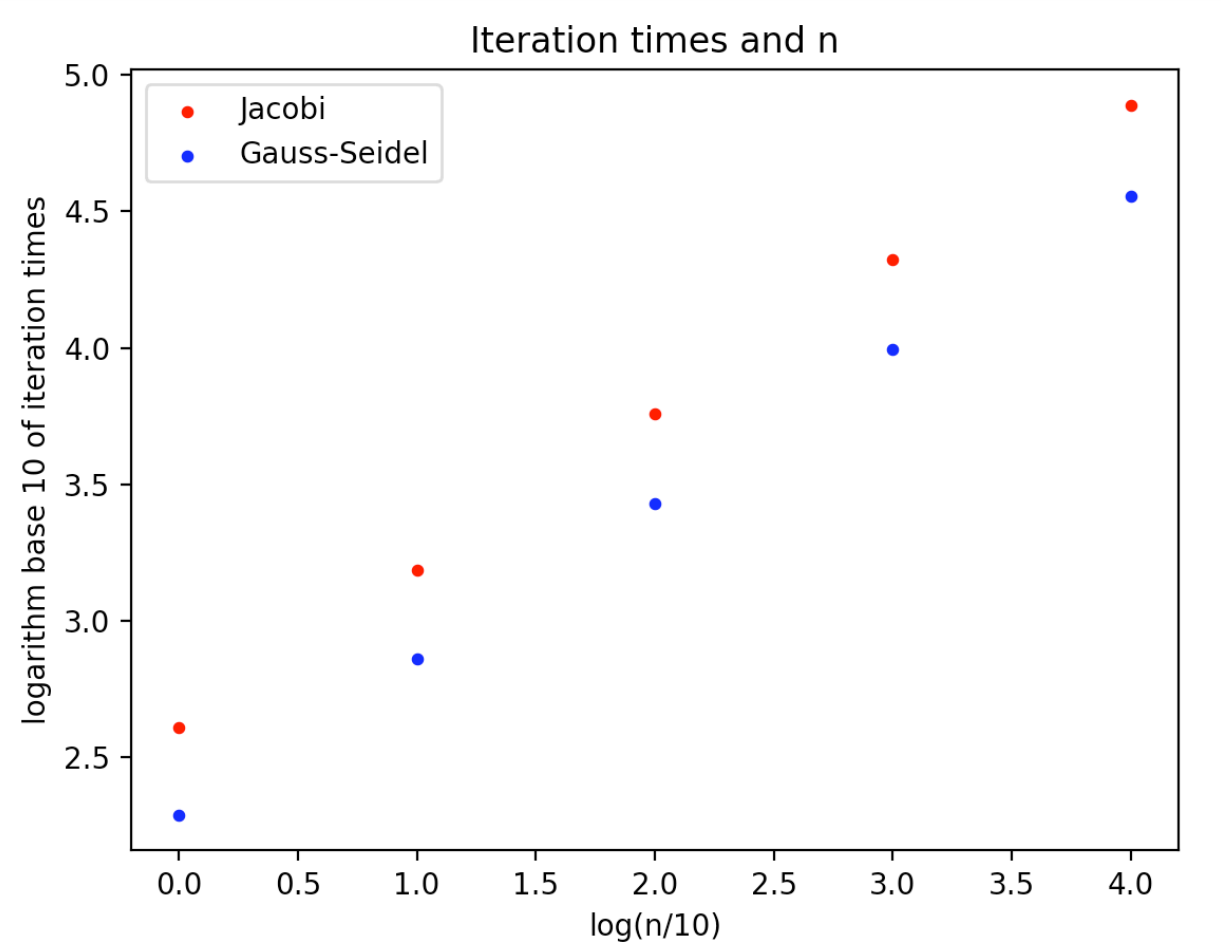
4

2中Jacobi和Gauss-Seidel迭代法求解上述线性方程组到收敛的迭代次数分别为

	Jacobi iteration times	**Gauss-Seidel iteration times**
n=10	408	195
n=20	1539	729
n=40	5727	2697
n=80	21127	9891
n=160	77331	35970

观察结论：

- 相同n下，Jacobi迭代次数高于Gauss- Seidel
- 迭代次数随n增大而增大，且和n呈超线性关系：



5

> 为节省空间，求解结果放在附录中

当$\alpha=1$时，记

$$\min_{\{u_1,u_2\dots u_{n-1}\}}\sum_{i=1}^{n}(\frac{1}{2}(\frac{u_i-u_{i-1}}{h})^2h+\frac{\alpha}{4}u_i^4h-f_iu_ih)=\min_{\{u_1,u_2\dots u_{n-1}\}}G_n(u_1,u_2\dots,u_{n-1})$$

求G对每个变量的偏微分，并令它们为0，得到

$$\begin{aligned}\frac{\partial G}{\partial u_1}&=u_1-u_0+u_1^3h^2-f_1h^2+u_1-u_2=0\\ \frac{\partial G}{\partial u_2}&=u_2-u_1+u_2^3h^2-f_2h^2+u_2-u_3=0\\ &\vdots\\ \frac{\partial G}{\partial u_{n-1}}&=u_{n-1}-u_{n-2}+u_{n-1}^3h^2-f_{n-1}h^2+u_{n-1}-u_n=0\end{aligned}$$

使用牛顿迭代法求解结果与精确解$u_e(x)=\sin(\pi x)$的误差和n的关系见下表：

n	e_h
10	0.01501803282490173
20	0.00530088945911111
40	0.00187336860374472
80	0.00066226693305820
160	0.00023414062963731

同理3进行最小二乘法拟合：

h	$\log(h)$	e_h	$\log(e_h)$
0.1	-2.30258509299404	0.01501803282490173	-4.198503611602896
0.05	-2.99573227355399	0.00530088945911111	-5.239880650030883
0.025	-3.68887945411393	0.00187336860374472	-6.280017076300147
0.0125	-4.38202663467388	0.00066226693305820	-7.319841861157172
0.00625	-5.07517381523382	0.00023414062963731	-8.359588641767521

记$x=\log(h),y=\log(e_h)$，作最小二乘法：

$$\begin{aligned}\sum xy&=123.03302474409142\\ \overline{x}&=-3.6888794541139327\\ \overline{y}&=-6.279566368171724\\ \sum x_i^2&=72.84368827410151\end{aligned}$$

$$\hat{\beta} = \frac{\sum xy - 5\overline{x}\,\overline{y}}{\sum x_i^2 - 5\overline{x}^2} = 1.50237$$

因此收敛阶仍约为1.5

附录

1. *Jacobi和Gauss- Seidel迭代Python代码*

```python
from math import pi, sin, sqrt
import numpy as np

def gause_seidel_iteration(A: np.array, b: np.array, epsilon: float):
    iter_times = 0
    n = len(A)
    x_1 = np.zeros(n)
    x_2 = np.ones(n)
    while True:
        iter_times += 1
        x_1 = x_2.__copy__()
        for i in range(n):
            s = 0
            for j in range(n):
                s += A[i][j] * x_2[j]
            x_2[i] = (b[i] - s + A[i][i] * x_2[i]) / A[i][i]
        if np.max(np.abs(x_2 - x_1)) < epsilon:
            break
    return x_2, iter_times


def Jacobi_iteration(A:np.array, b:np.array, epsilon: float):
    iter_times = 0
    x_1 = np.zeros(len(A))
    x_2 = np.ones(len(A))
    while True:
        iter_times += 1
        x_1 = x_2.__copy__()
        for i in range(len(A)):
            s = 0
            for j in range(len(A)):
                s += A[i][j] * x_1[j]
            x_2[i] = (b[i] - s + A[i][i] * x_1[i]) / A[i][i]
        if np.max(np.abs(x_2 - x_1)) < epsilon:
            break
    return x_2, iter_times

def run_question_2(n: int, print_result: bool = False):
```

```python
    def f(x):
        return pi * pi * sin(pi*x)
    def u_e(x):
        return sin(pi*x)
    A = 2 * np.eye(n-1)
    for i in range(n-2):
        A[i][i+1] = -1
        A[i+1][i] = -1
    b = [f(i/n)/n/n for i in range(1, n)]
    u_h_Jacobi, iter_times_Jacobi = Jacobi_iteration(A, b, 1e-10)
    e_h_Jacobi = sqrt(sum([(u_h_Jacobi[i] - u_e((i+1)/n))**2 for i in range(n-1)]))
    u_h_Gauss, iter_times_Gauss = gause_seidel_iteration(A, b, 1e-10)
    e_h_Gauss = sqrt(sum([(u_h_Gauss[i] - u_e((i+1)/n))**2 for i in range(n-1)]))
    if print_result:
        print(f"n={n}, Jacobi: {u_h_Jacobi}")
        print(f"n={n}, Gauss Seidel: {u_h_Gauss}, Error: {e_h_Gauss}")
    print(f"n={n}, Jacobi Error: {e_h_Jacobi}")
    print(f"n={n}, Gauss Seidel Error: {e_h_Gauss}")
    return (e_h_Jacobi, e_h_Gauss), (iter_times_Jacobi, iter_times_Gauss)


question_2_n = [10, 20, 40, 80, 160]
question_2_errors = []
question_2_iters = []
for n in question_2_n:
    error, iters = run_question_2(n, True)
    question_2_errors.append(error)
    question_2_iters.append(iters)
print("Question 2 error table:", question_2_errors)
print("Question 2 iter times table:", question_2_iters)

```

2. *Jacobi和Gauss- Seidel迭代输出解*

```
n=10, Jacobi: [0.31157115 0.59264354 0.81570386 0.9589174  1.00826542 0.9589174
 0.81570386 0.59264354 0.31157115]
n=10, Gauss Seidel: [0.31157115 0.59264354 0.81570386 0.9589174  1.00826542 0.9589174
 0.81570386 0.59264354 0.31157115], Error: 0.018482036208807052

n=20, Jacobi: [0.15675652 0.30965317 0.45492513 0.58899533 0.70856251 0.81068252
 0.89284085 0.95301446 0.9897217  1.00205871 0.9897217  0.95301446
 0.89284085 0.81068252 0.70856251 0.58899533 0.45492513 0.30965317
 0.15675652]
n=20, Gauss Seidel: [0.15675652 0.30965317 0.45492514 0.58899533 0.70856251 0.81068253
 0.89284085 0.95301447 0.98972171 1.00205871 0.98972171 0.95301447
```

```
 0.89284085 0.81068253 0.70856251 0.58899533 0.45492513 0.30965317
 0.15675652], Error: 0.006510214971563716

n=40, Jacobi: [0.07849944 0.1565149  0.2335654  0.30917589 0.38288021 0.45422394
 0.52276724 0.5880875  0.649782   0.70747038 0.76079697 0.809433
 0.8530786  0.89146469 0.9243546  0.95154556 0.97286992 0.98819622
 0.99742995 1.00051421 0.99742995 0.98819622 0.97286992 0.95154556
 0.9243546  0.89146469 0.8530786  0.809433   0.76079697 0.70747038
 0.649782   0.5880875  0.52276724 0.45422394 0.38288021 0.30917589
 0.2335654  0.1565149  0.07849944]
n=40, Gauss Seidel: [0.07849944 0.15651491 0.23356541 0.3091759  0.38288021
0.45422395
 0.52276724 0.5880875  0.64978201 0.70747039 0.76079698 0.809433
 0.85307861 0.89146469 0.92435461 0.95154557 0.97286993 0.98819623
 0.99742997 1.00051422 0.99742997 0.98819623 0.97286993 0.95154557
 0.92435461 0.89146469 0.85307861 0.809433   0.76079698 0.70747039
 0.649782   0.5880875  0.52276724 0.45422395 0.38288021 0.3091759
 0.23356541 0.15651491 0.07849944], Error: 0.0022996463089598887

n=80, Jacobi: [0.03926486 0.07846918 0.11755251 0.15645457 0.1951154  0.23347537
 0.27147534 0.30905672 0.34616155 0.38273262 0.41871355 0.45404886
 0.48868405 0.52256573 0.55564165 0.58786081 0.61917353 0.64953153
 0.678888   0.70719768 0.7344169  0.76050371 0.78541788 0.80912099
 0.83157649 0.85274977 0.87260816 0.89112106 0.90825991 0.92399829
 0.93831193 0.95117877 0.96257895 0.97249491 0.98091135 0.9878153
 0.99319611 0.99704548 0.99935748 1.00012854 0.99935748 0.99704548
 0.99319611 0.9878153  0.98091135 0.97249491 0.96257895 0.95117877
 0.93831193 0.92399829 0.90825991 0.89112106 0.87260816 0.85274977
 0.83157649 0.80912099 0.78541788 0.76050371 0.7344169  0.70719768
 0.678888   0.64953153 0.61917353 0.58786081 0.55564165 0.52256573
 0.48868405 0.45404886 0.41871355 0.38273262 0.34616155 0.30905672
 0.27147534 0.23347537 0.1951154  0.15645457 0.11755251 0.07846918
 0.03926486]
n=80, Gauss Seidel: [0.03926486 0.07846918 0.11755251 0.15645458 0.19511541
0.23347538
 0.27147535 0.30905673 0.34616156 0.38273264 0.41871357 0.45404888
 0.48868407 0.52256575 0.55564167 0.58786083 0.61917356 0.64953156
 0.67888803 0.70719771 0.73441693 0.76050374 0.78541791 0.80912102
 0.83157653 0.8527498  0.8726082  0.89112109 0.90825995 0.92399833
 0.93831197 0.95117881 0.96257899 0.97249495 0.9809114  0.98781534
 0.99319615 0.99704552 0.99935752 1.00012859 0.99935752 0.99704552
 0.99319615 0.98781534 0.98091139 0.97249495 0.96257899 0.95117881
 0.93831197 0.92399833 0.90825995 0.89112109 0.8726082  0.8527498
 0.83157653 0.80912102 0.78541791 0.76050374 0.73441693 0.7071977
 0.67888803 0.64953156 0.61917355 0.58786083 0.55564167 0.52256575
 0.48868407 0.45404888 0.41871357 0.38273264 0.34616156 0.30905673
 0.27147535 0.23347538 0.19511541 0.15645458 0.11755251 0.07846918
 0.03926486], Error: 0.0008132435845628745

n=160, Jacobi: [0.01963433 0.03926108 0.0588727  0.07846162 0.0980203  0.11754118
 0.13701676 0.15643951 0.17580194 0.19509661 0.21431606 0.23345289
 0.25249971 0.2714492  0.29029403 0.30902695 0.32764074 0.34612821
```

```
 0.36448224 0.38269576 0.40076175 0.41867323 0.4364233  0.45400513
 0.47141193 0.48863698 0.50567367 0.5225154  0.53915569 0.55558813
 0.57180638 0.58780419 0.60357539 0.6191139  0.63441372 0.64946897
 0.66427384 0.67882262 0.69310969 0.70712956 0.72087682 0.73434617
 0.74753241 0.76043047 0.77303536 0.78534223 0.79734634 0.80904306
 0.82042788 0.8314964  0.84224437 0.85266764 0.86276218 0.87252412
 0.88194968 0.89103523 0.89977727 0.90817243 0.91621748 0.9239093
 0.93124494 0.93822156 0.94483649 0.95108716 0.95697117 0.96248625
 0.96763027 0.97240125 0.97679735 0.98081688 0.98445829 0.98772016
 0.99060126 0.99310045 0.99521679 0.99694945 0.99829777 0.99926123
 0.99983945 1.00003222 0.99983945 0.99926123 0.99829777 0.99694945
 0.99521679 0.99310045 0.99060126 0.98772016 0.98445829 0.98081688
 0.97679735 0.97240125 0.96763027 0.96248625 0.95697117 0.95108716
 0.94483649 0.93822156 0.93124494 0.9239093  0.91621748 0.90817243
 0.89977727 0.89103523 0.88194968 0.87252412 0.86276218 0.85266764
 0.84224437 0.8314964  0.82042788 0.80904306 0.79734634 0.78534223
 0.77303536 0.76043047 0.74753241 0.73434617 0.72087682 0.70712956
 0.69310969 0.67882262 0.66427384 0.64946897 0.63441372 0.6191139
 0.60357539 0.58780419 0.57180638 0.55558813 0.53915569 0.5225154
 0.50567367 0.48863698 0.47141193 0.45400513 0.4364233  0.41867323
 0.40076175 0.38269576 0.36448224 0.34612821 0.32764074 0.30902695
 0.29029403 0.2714492  0.25249971 0.23345289 0.21431606 0.19509661
 0.17580194 0.15643951 0.13701676 0.11754118 0.0980203  0.07846162
 0.0588727  0.03926108 0.01963433]
n=160, Gauss Seidel: [0.01963433 0.03926109 0.05887271 0.07846164 0.09802032
0.1175412
 0.13701678 0.15643953 0.17580197 0.19509664 0.21431609 0.23345293
 0.25249976 0.27144924 0.29029408 0.309027   0.32764079 0.34612827
 0.36448231 0.38269583 0.40076181 0.4186733  0.43642338 0.4540052
 0.47141201 0.48863707 0.50567375 0.52251549 0.53915579 0.55558823
 0.57180648 0.58780429 0.60357549 0.619114   0.63441383 0.64946908
 0.66427395 0.67882273 0.69310981 0.70712968 0.72087694 0.73434629
 0.74753254 0.76043059 0.77303549 0.78534237 0.79734648 0.8090432
 0.82042802 0.83149654 0.84224451 0.85266778 0.86276233 0.87252427
 0.88194983 0.89103538 0.89977743 0.90817259 0.91621763 0.92390946
 0.9312451  0.93822172 0.94483665 0.95108732 0.95697133 0.96248641
 0.96763043 0.97240141 0.97679752 0.98081705 0.98445845 0.98772033
 0.99060142 0.99310062 0.99521696 0.99694962 0.99829794 0.9992614
 0.99983962 1.00003239 0.99983962 0.9992614  0.99829794 0.99694962
 0.99521696 0.99310062 0.99060142 0.98772033 0.98445845 0.98081705
 0.97679752 0.97240141 0.96763043 0.96248641 0.95697133 0.95108732
 0.94483665 0.93822172 0.93124509 0.92390945 0.91621763 0.90817259
 0.89977742 0.89103538 0.88194983 0.87252426 0.86276233 0.85266778
 0.84224451 0.83149654 0.82042801 0.8090432  0.79734648 0.78534236
 0.77303549 0.76043059 0.74753254 0.73434629 0.72087694 0.70712968
 0.69310981 0.67882273 0.66427395 0.64946908 0.63441383 0.619114
 0.60357549 0.58780429 0.57180648 0.55558823 0.53915578 0.52251549
 0.50567375 0.48863707 0.471412   0.4540052  0.43642337 0.4186733
 0.40076181 0.38269583 0.3644823  0.34612827 0.32764079 0.309027
 0.29029408 0.27144924 0.25249975 0.23345292 0.21431609 0.19509664
 0.17580197 0.15643953 0.13701678 0.1175412  0.09802031 0.07846164
 0.05887271 0.03926109 0.01963433], Error: 0.00028968235456973407
```

```

Question 2 error table: [(0.018482034928874364, 0.018482036208807052), (0.006510206868033681, 0.006510214971563716), (0.0022995999225368553, 0.0022996463089598887), (0.0008129789948298825, 0.0008132435845628745), (0.0002881833488868389, 0.00028968235456973407)]
Question 2 iter times table: [(408, 195), (1539, 729), (5727, 2697), (21127, 9891), (77331, 35970)]
```

3. Newton迭代Python代码

```python
from math import pi, sin, sqrt, nan
import numpy as np

def newton_iteration(n:int , epsilon: float, Jacobi_function, F_function):
    X = np.zeros(n-1)
    iter_times = 0
    while True:
        iter_times += 1
        Jacobi_matrix = Jacobi_function(X)
        delta = np.linalg.inv(Jacobi_matrix).dot(F_function(X).reshape(n-1, 1))
        X = X - delta.reshape(n-1)
        if np.max(np.abs(delta)) < epsilon:
            break
    return X, iter_times

def run_question_6(n: int, print_result: bool = False):
    h = 1/n
    def f(x):
        return pi * pi * sin(pi*x) + sin(pi*x) ** 3

    def u_e(x):
        return sin(pi*x)

    def Jacobi(input:np.array):
        res = np.zeros((n-1, n-1))
        for i in range(n-1):
            res[i][i] = 2 + 3 * input[i] * input[i] * h * h
            if i > 0:
                res[i][i-1] = -1
            if i < n-2:
                res[i][i+1] = -1
        return res

    def F(X:np.array):
        res = np.zeros(n-1)
        res[0] = 2 * X[0] - X[1] + X[0] ** 3 * h * h - f(h) * (h ** 2)
        res[n-2] = 2 * X[n-2] - X[n-3] + X[n-2] ** 3 * h * h - f(1 - h) * (h ** 2)
        for i in range(1, n-2):
```

```python
            res[i] = 2 * X[i] - X[i-1] - X[i+1] + X[i] ** 3 * h * h - f((i+1) * h) * (h ** 2)
        return res

    X, iter_times = newton_iteration(n, 1e-8, Jacobi, F)
    if print_result:
        print(f"n={n}, Newton: {X}")
    print(f"n={n}, Newton iter times: {iter_times}")
    error = sqrt(sum([(X[i] - u_e((i+1)/n))**2 for i in range(n-1)]))
    return error, iter_times

question_6_n = [10, 20, 40, 80, 160]
question_6_errors = []
question_6_iters = []
for n in question_6_n:
    error, iters = run_question_6(n, True)
    question_6_errors.append(error)
    question_6_iters.append(iters)
print("Question 6 error table:", question_6_errors)
print("Question 6 iter times table:", question_6_iters)
```

4. Newton迭代输出解

```
n=10, Newton: [0.31114144 0.59179026 0.81446879 0.95740831 1.00665583 0.95740831
 0.81446879 0.59179026 0.31114144]

n=20, Newton: [0.15670316 0.30954651 0.45476555 0.58878402 0.70830206 0.81037743
 0.8924978  0.95264236 0.98933131 1.00166208 0.98933131 0.95264236
 0.8924978  0.81037743 0.70830206 0.58878402 0.45476555 0.30954651
 0.15670316]

n=40, Newton: [0.07849278 0.15650159 0.23354543 0.30914928 0.38284697 0.45418413
 0.52272092 0.58803479 0.64972307 0.70740544 0.7607263  0.80935695
 0.85299759 0.8913792  0.92426519 0.95145284 0.97277458 0.98809896
 0.99733154 1.0004154  0.99733154 0.98809896 0.97277458 0.95145284
 0.92426519 0.8913792  0.85299759 0.80935695 0.7607263  0.70740544
 0.64972307 0.58803479 0.52272092 0.45418413 0.38284697 0.30914928
 0.23354543 0.15650159 0.07849278]

n=80, Newton: [0.03926403 0.07846752 0.11755001 0.15645124 0.19511124 0.23347038
 0.27146951 0.30905006 0.34615406 0.38272431 0.41870442 0.4540389
 0.48867328 0.52255414 0.55562926 0.58784763 0.61915956 0.64951679
 0.6788725  0.70718144 0.73439994 0.76048604 0.78539952 0.80910197
 0.83155684 0.85272951 0.87258733 0.89109968 0.90823803 0.92397594
 0.93828915 0.95115559 0.96255542 0.97247108 0.98088726 0.98779099
 0.99317163 0.99702088 0.9993328  1.00010384 0.9993328  0.99702088
 0.99317163 0.98779099 0.98088726 0.97247108 0.96255542 0.95115559
 0.93828915 0.92397594 0.90823803 0.89109968 0.87258733 0.85272951
```

```
 0.83155684 0.80910197 0.78539952 0.76048604 0.73439994 0.70718144
 0.6788725  0.64951679 0.61915956 0.58784763 0.55562926 0.52255414
 0.48867328 0.4540389  0.41870442 0.38272431 0.34615406 0.30905006
 0.27146951 0.23347038 0.19511124 0.15645124 0.11755001 0.07846752
 0.03926403]

n=160, Newton: [0.01963422 0.03926087 0.05887238 0.0784612  0.09801977 0.11754055
 0.13701602 0.15643866 0.17580099 0.19509555 0.21431489 0.23345162
 0.25249834 0.27144772 0.29029244 0.30902526 0.32763894 0.34612631
 0.36448024 0.38269365 0.40075953 0.41867091 0.43642088 0.4540026
 0.47140929 0.48863425 0.50567083 0.52251246 0.53915265 0.55558499
 0.57180314 0.58780085 0.60357194 0.61911035 0.63441008 0.64946523
 0.66427    0.67881868 0.69310567 0.70712544 0.72087261 0.73434187
 0.74752802 0.76042598 0.77303079 0.78533758 0.7973416  0.80903824
 0.82042297 0.83149142 0.84223931 0.8526625  0.86275697 0.87251884
 0.88194433 0.89102981 0.89977179 0.90816689 0.91621187 0.92390363
 0.93123922 0.93821579 0.94483066 0.95108128 0.95696525 0.96248028
 0.96762426 0.97239521 0.97679128 0.98081077 0.98445215 0.987714
 0.99059507 0.99309425 0.99521057 0.99694322 0.99829153 0.99925498
 0.9998332  1.00002596 0.9998332  0.99925498 0.99829153 0.99694322
 0.99521057 0.99309425 0.99059507 0.987714   0.98445215 0.98081077
 0.97679128 0.97239521 0.96762426 0.96248028 0.95696525 0.95108128
 0.94483066 0.93821579 0.93123922 0.92390363 0.91621187 0.90816689
 0.89977179 0.89102981 0.88194433 0.87251884 0.86275697 0.8526625
 0.84223931 0.83149142 0.82042297 0.80903824 0.7973416  0.78533758
 0.77303079 0.76042598 0.74752802 0.73434187 0.72087261 0.70712544
 0.69310567 0.67881868 0.66427    0.64946523 0.63441008 0.61911035
 0.60357194 0.58780085 0.57180314 0.55558499 0.53915265 0.52251246
 0.50567083 0.48863425 0.47140929 0.4540026  0.43642088 0.41867091
 0.40075953 0.38269365 0.36448024 0.34612631 0.32763894 0.30902526
 0.29029244 0.27144772 0.25249834 0.23345162 0.21431489 0.19509555
 0.17580099 0.15643866 0.13701602 0.11754055 0.09801977 0.0784612
 0.05887238 0.03926087 0.01963422]

Question 6 error table: [0.015018032824901736, 0.005300889459111111, 0.0018733686037447255, 0.0006622669330582057, 0.00023414062963731416]
Question 6 iter times table: [5, 5, 5, 5, 5]
```