数值分析 实验4

2019011265 计93 丁韶峰

实验内容

将微分方程离散化，得到线性方程组。用雅可比，G-S，SOR迭代法分别求解问题，并计算与精确解的误差。

实验过程

首先引入必要的包。

```python
import numpy as np
```

定义相关常数。

```python
n = 100
a = 0.5
h = 1 / n
```

生成数据。需要特别考虑边界情况。

```python
def gen_data(eps):
  A = np.zeros((n - 1, n - 1))
  for i in range(n - 1):
    if i != 0:
      A[i][i - 1] = eps
    A[i][i] = -(2 * eps + h)
    if i != n - 2:
      A[i][i + 1] = eps + h
  b = np.zeros(n - 1)
  for i in range(n - 1):
    b[i] = a * h * h
    if i == n - 2:
      b[i] -= eps + h
  return A, b
```

根据表达式求出某点对应的精确解。

```python
def accurate_sol(x, eps):
  return (1 - a) / (1 - np.exp(-1 / eps)) * (1 - np.exp(-x / eps)) + a * x
```

Jacobi迭代法。由于矩阵稀疏，只需对一行中两到三个非零元素进行计算即可。之后的两种方法同理。根据课程群中讨论，相邻解误差小于1e-4时停止计算，而不是1e-3。

```python
def jacobi(A, b, n):
```

```python
  x = np.ones_like(b)
  cnt = 0
  while True:
    y = np.copy(x)
    for i in range(n):
      x[i] = b[i]
      if i != 0:
        x[i] -= A[i][i - 1] * y [i - 1]
      if i != n - 1:
        x[i] -= A[i][i + 1] * y[i + 1]
      x[i] /= A[i][i]
    cnt += 1
    if np.max(np.abs(x - y)) < 1e-4:
      print("Jacobi stops after {} iterations".format(cnt))
      return x
```

GS迭代法。

```python
def gs(A, b, n):
  x = np.ones_like(b)
  cnt = 0
  while True:
    y = np.copy(x)
    for i in range(n):
      x[i] = b[i]
      if i != 0:
        x[i] -= A[i][i - 1] * x[i - 1]
      if i != n - 1:
        x[i] -= A[i][i + 1] * x[i + 1]
      x[i] /= A[i][i]
    cnt += 1
    if np.max(np.abs(x - y)) < 1e-4:
      print("GS stops after {} iterations".format(cnt))
      return x
```

SOR迭代法。

```python
def sor(A, b, omega, n):
  x = np.ones_like(b)
  cnt = 0
  while True:
    y = np.copy(x)
    for i in range(n):
      x[i] = b[i]
      if i != 0:
        x[i] -= A[i][i - 1] * x[i - 1]
      if i != n - 1:
        x[i] -= A[i][i + 1] * x[i + 1]
```

```
            x[i] /= A[i][i]
            x[i] = (1 - omega) * y[i] + omega * x[i]
        cnt += 1
        if np.max(np.abs(x - y)) < 1e-4:
            print("SOR stops after {} iterations".format(cnt))
            return x
```

计算无穷范数和二范数下迭代解和精确解的误差。

```
def compute_arr(appro_sol, acc_sol):
  appro_sol = appro_sol.reshape(np.shape(acc_sol))
  inv_norm = np.max(np.abs(appro_sol - acc_sol))
  two_norm = np.linalg.norm(appro_sol - acc_sol)
  return inv_norm, two_norm
```

外层过程。用三种方法进行求解，并计算误差。

```
def compute(eps):
  A, b = gen_data(eps)
  acc_sol = [accurate_sol(x, eps) for x in np.arange(h, 1, h)]
  jacobi_sol = jacobi(A, b, n - 1)
  gs_sol = gs(A, b, n - 1)
  sor_sol = sor(A, b, 0.9, n - 1)
  jacobi_inv, jacobi_two = compute_arr(jacobi_sol, acc_sol)
  gs_inv, gs_two = compute_arr(gs_sol, acc_sol)
  sor_inv, sor_two = compute_arr(sor_sol, acc_sol)
  print("jacobi error: 2 norm {}, inv norm {}".format(jacobi_inv, jacobi_two))
  print("gs error: 2 norm {}, inv norm {}".format(gs_inv, gs_two))
  print("sor error: 2 norm {}, inv norm {}".format(sor_inv, sor_two))
```

对题目要求的不同ϵ重复实验。

```
compute(1)
compute(0.1)
compute(0.01)
compute(0.0001)
```

```
Jacobi stops after 3301 iterations
GS stops after 1690 iterations
SOR stops after 1831 iterations
jacobi error: 2 norm 0.1040968874517717, inv norm 0.7325072021540333
gs error: 2 norm 0.09812757919867121, inv norm 0.6899053650346129
sor error: 2 norm 0.1197339261741952, inv norm 0.8417088491023286
Jacobi stops after 1536 iterations
GS stops after 999 iterations
SOR stops after 1134 iterations
jacobi error: 2 norm 0.05417529293513368, inv norm 0.309366940903285
```

```
gs error: 2 norm 0.025413701013942358, inv norm 0.14366877632279315
sor error: 2 norm 0.032773982651351674, inv norm 0.18560636065813976
Jacobi stops after 365 iterations
GS stops after 237 iterations
SOR stops after 277 iterations
jacobi error: 2 norm 0.06439168025723319, inv norm 0.09424968882833301
gs error: 2 norm 0.06518557051249729, inv norm 0.09651434971616092
sor error: 2 norm 0.06497258681895757, inv norm 0.095926404587772
Jacobi stops after 106 iterations
GS stops after 103 iterations
SOR stops after 120 iterations
jacobi error: 2 norm 0.004808154489188476, inv norm 0.004808235684323808
gs error: 2 norm 0.004926534567720187, inv norm 0.004926738187729387
sor error: 2 norm 0.004836294473050451, inv norm 0.0048363923124258865
```

可得结果如上。

实验结论

ϵ 较小时，原微分方程的解更接近线性，因此用差分的方式得到的解更为精确，且收敛速度更快。

对于本问题，Jacobi，GS和SOR迭代法有不同的收敛速度和精确度。ϵ 较大时，Jacobi法收敛最慢，误差最大；SOR法次之（取$\omega = 0.9$)，GS法最好。ϵ较小时，三种方法的效果比较接近。