

导入库

```
1 import numpy as np
2 import sympy as sp
3 from IPython.display import Math
4 import matplotlib.pyplot as plt
```

用以解决一个类似如下的问题：

$$\begin{bmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_5 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_5 \end{bmatrix}$$

下面给出一组测试数据并写入 `ThomasTestData`：

```
1 N = 10
2 a = np.random.randint(20, size=N)
3 c = np.random.randint(20, size=N)
4 b = np.random.randint(20, size=N)
5 d = np.random.randint(20, size=N)
6 a[0] = 0.0
7 c[N-1] = 0.0
8
9 data = np.array(
10     [a, b, c, d],
11     dtype=float
12 )
13 np.savetxt("ThomasTestData.txt", data)
14 print("a = ", a)
15 print("b = ", b)
16 print("c = ", c)
17 print("d = ", d)
```

```
1 a = [ 0  2  2  1  5 11  0  3  2 16]
2 b = [11  0  5 14 17 10 17 19 11  6]
3 c = [ 6  1 17  0  7 18  2 11 10  0]
4 d = [17  6 15 16  7  2 16  1  4  4]
```

为方便观赏写成矩阵形式：

```
1 mat = sp.zeros(N, N)
2 for i in range(N):
3     mat[i, i] = b[i]
4     pass
5 for i in range(1, N):
6     mat[i, i-1] = a[i]
7     pass
8 for i in range(0, N-1):
9     mat[i, i+1] = c[i]
10    pass
```

```

11 d_mat = sp.Matrix([
12     d
13 ]).transpose()
14 x = sp.Matrix(sp.symbols("x_1:"+str(N+1)))
15 display(
16     Math(
17         sp.latex(mat)+
18         sp.latex(x)+
19         "="+
20         sp.latex(d_mat)
21     )
22 )

```

$$\begin{bmatrix} 11 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 5 & 17 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 14 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 17 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 11 & 10 & 18 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 17 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 19 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 11 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} 17 \\ 6 \\ 15 \\ 16 \\ 7 \\ 2 \\ 16 \\ 1 \\ 4 \\ 4 \end{bmatrix}$$

Python解

```

1 sol_python = mat.solve(d_mat)
2 print("分数解: ")
3 sol_python

```

1 分数解:

$$\begin{bmatrix} \frac{689}{236} \\ -\frac{1189}{472} \\ \frac{19}{118} \\ \frac{267}{236} \\ \frac{228280361}{175814454} \\ -\frac{1041316739}{351628908} \\ \frac{15426}{16021} \\ -\frac{2953}{16021} \\ \frac{2350}{16021} \\ \frac{4414}{16021} \end{bmatrix}$$

```

1 print("python数值解: ")
2 py_res = np.float64(np.array(sol_python.evalf()).T[0])
3 py_res

```

1 | python数值解:

```
1 | array([ 2.91949153, -2.5190678 ,  0.16101695,  1.13135593,  1.29841635,  
2 |        -2.96140822,  0.96286124, -0.18432058,  0.14668248,  0.27551339])
```

cpp 解

代码在 `ThomasCpp.cpp` 文件中，核心程序如下：

```
1 | std::vector<double> Thomas(std::vector<double> b, std::vector<double> a,  
  | std::vector<double> c, std::vector<double> d){  
2 |     int N = b.size();  
3 |     std::vector<double> beta(N);  
4 |     std::vector<double> y(N);  
5 |     std::vector<double> l(N);  
6 |     std::vector<double> x(N);  
7 |     beta[0] = b[0];  
8 |     y[0] = d[0];  
9 |     for(int j=1; j<N; j++){  
10 |         l[j] = a[j] / beta[j-1];  
11 |         beta[j] = b[j] - l[j] * c[j-1];  
12 |         y[j] = d[j] - l[j] * y[j-1];  
13 |     }  
14 |     x[N-1] = y[N-1] / beta[N-1];  
15 |     for(int j=N-2; j>=0; j--){  
16 |         x[j] = (y[j] - c[j] * x[j+1]) / beta[j];  
17 |     }  
18 |     return x;  
19 | }
```

用 `python` 编译一下并且读取文件：

```
1 | import os  
2 | cpp_result_file = "CppResult.txt"  
3 | os.system("g++ -o ThomasCpp ThomasCpp.cpp ")  
4 | os.system("./ThomasCpp > "+cpp_result_file)
```

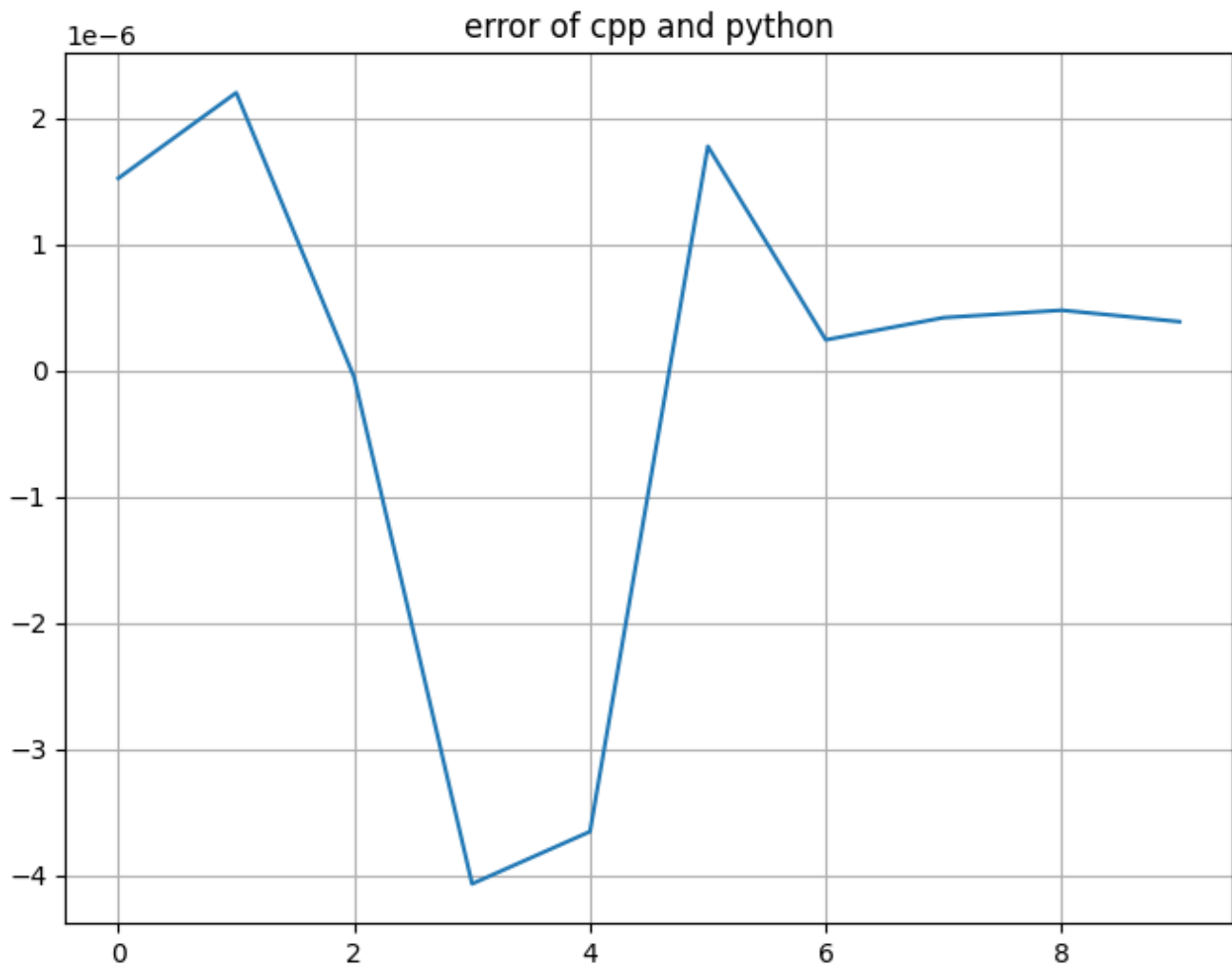
1 | 0

```
1 | cpp_res = np.loadtxt(cpp_result_file).T  
2 | print("cpp数值解: ")  
3 | print(cpp_res)
```

```
1 | cpp数值解:  
2 | [ 2.91949 -2.51907  0.161017  1.13136  1.29842 -2.96141  0.962861  
3 | -0.184321  0.146682  0.275513]
```

比较

```
1 | plt.figure(figsize=(8, 6), facecolor="white")  
2 | plt.plot(py_res-cpp_res)  
3 | plt.title("error of cpp and python")  
4 | plt.grid(True)  
5 | plt.show()
```



```
1 |
```