实验报告

主要思路

1.在相同的基规模下，通过计算比较考虑对偶问题运算时间问题-->time 2.在相同的基规模下，比较两种算法在0处误差大小，考虑duality是否在0处误差更小-->errorat0 3.进一步通过比较两者平均误差，考虑duality是否以在其他地方误差更大为代价-->average_error

主体代码

对偶问题

```
% Duality
t1=clock;
uk = Duality_approx_simple_PDE(BASE_SIZE);
t2=clock;
time_dual(k)=etime(t2,t1);
errorat0_dual(k)=double(subs(uk,x,0)-subs(u,x,0));
average_error_dual(k)=int(uk-u,x,0,1);
```

原问题

```
% original
t1=clock;
uk = Approx_simple_PDE(BASE_SIZE);
t2=clock;
time_ori(k)=etime(t2,t1);
errorat0_ori(k)=double(subs(uk,x,0)-subs(u,x,0));
average_error_ori(k)=int(uk-u,x,0,1);
```

实例

BASE_SIZE = 4; k = 1

```
time_dual=28.8642
time_ori=56.0236

errorat0_dual=0.0166
errorat0_ori=9.8790

average_error_dual=1/1688849860263936
average_error_ori=1/1688849860263936
```

对偶问题求解在此基规模下用时短、效率高

BASE_SIZE = 6; k = 2

```
time_dual=58.7925
time_ori=153.6298

errorat0_dual=0.0008
errorat0_ori=9.8723
```

两者平均误差均很小，对偶方法有较为明显优势

BASE_SIZE = 10; k = 3

```
time_dual=130.9959
time_ori=577.8754

errorat0_dual=2.0012e-05
errorat0_ori=9.8697

average_error_dual=-109/105553116266496000
average_error_ori=3113/211106232532992000
```

BASE_SIZE = 14; k = 4

```
time_dual=274.3599

errorat0_dual=1.0253e-05
```

此基规模下原算法计算时间仍过长。