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

测试程序

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

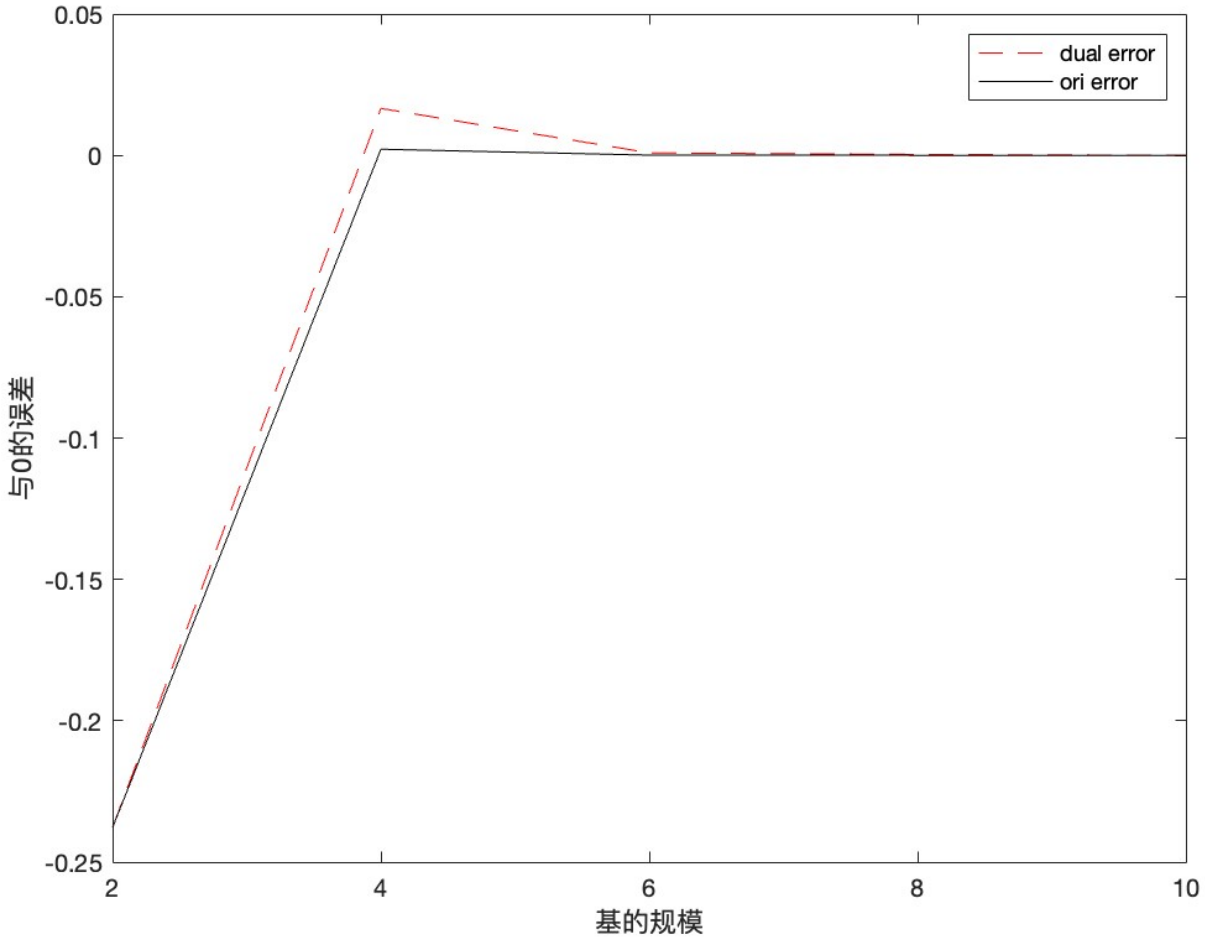
```
% original
t1=clock;
uk = Approx_simple_PDE(BASE_SIZE);
t2=clock;
time_ori(k)=etime(t2,t1);
errorat0_ori(k)=double(subs(uk,x,0)-subs(u,x,0));
average_error_ori(k)=int(uk-u,x,0,1);
```

运行结果

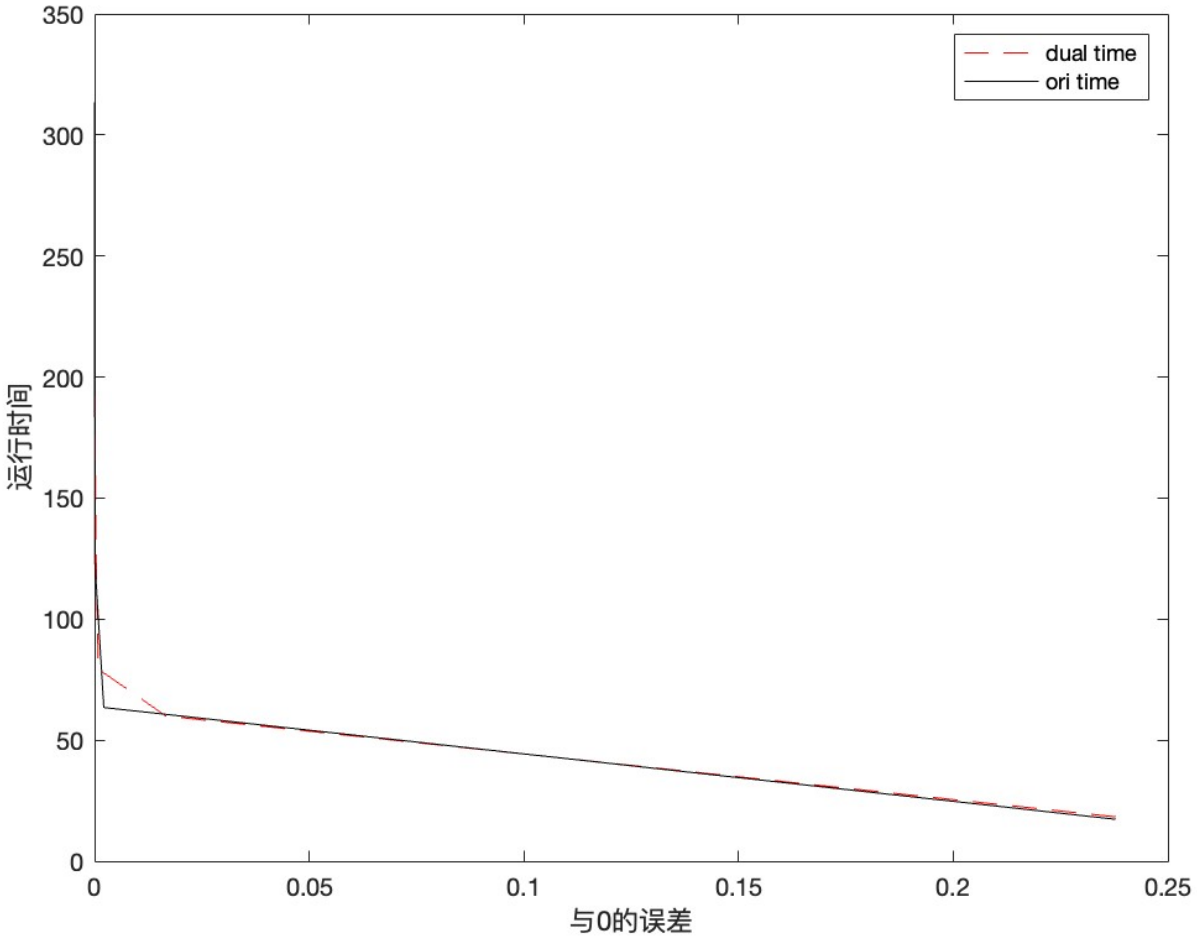
time_dual	time_ori	error at 0_dual	error at 0_ori	average error_dual	average error_ori
18.3203190	17.3211230	-0.23784	-0.237840011320181	-0.309945062063126	-0.309945062063126
59.9963550	63.4911460	0.01660	0.00215762969316735	5.92118946466750e-16	3.76587649952853e-16

time_dual	time_ori	error at 0_dual	error at 0_ori	average error_dual	average error_ori
79.3734140	123.723180	0.00087	6.98938710428365e-05	-2.58060239843871e-15	-4.04417240436790e-16
129.793720	217.263306	0.00032	1.70970360242226e-05	-7.13858601860314e-15	-4.18535576566607e-16
202.121174	313.499025	2.00123e-05	1.03498927499590e-05	-1.03265544263801e-15	-2.02800739164862e-17

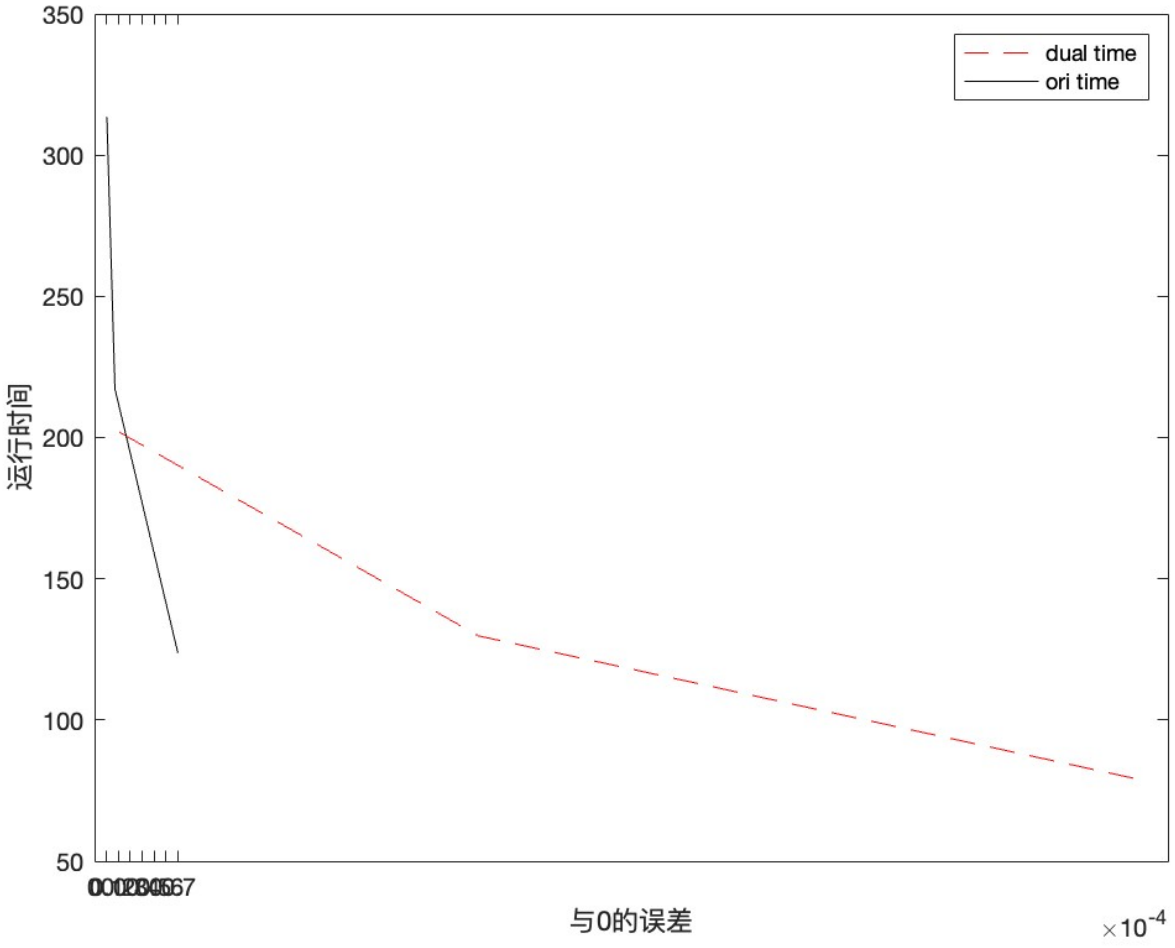
以基的规模为x轴，以0处误差为y轴，作图如下：



以0处误差为x轴，运行时间为y轴，作图如下：



取后三个误差在e-4之内的值作图：



总结

考虑了对偶问题后，虽然相同的基规模下，0处误差仍是原问题更小，但运行时间更短了。

我们考虑收敛速率，即第二、第三张图，可以发现红线(dual)与黑线（ori）不相上下，并且第三张图红线甚至有高于黑线的趋势。

由于误差随着基的规模增大收敛很快，第三张图必然是非常陡峭的；如果我们想要探索在基的规模很大的情况下两者的收敛速度，我们将需要更强的机器，因为得到此规模的数据开销很大。