

READ ME

Question1

算法

`Gossein` 是高斯消去法, `Column` 是列主元消去法
其定义如下

```
1. function [timeSpan, result] = Gossein(A, b, n)
2.
3. function [timeSpan, result] = Column(A, b, n)
```

输入为(即方程 $Ax=b$)

- `A`, `b`: 即 $Ax=b$ 的系数
- `n`: A 矩阵的大小

输出为

- `timeSpan`: 算法主体部分耗时 (除去用无穷范数计算相对误差的部分)
- `result`: 计算结果

Runner

`Runner.m` 无需参数可以直接运行, 随机生成一个矩阵并使用两种直接法计算并给出结果以及图像分析。

Question2

算法

`CG` 即共轭梯度法

`Jacobi` 即雅克比迭代法

`Gauss_Seidel` 即高斯塞德尔迭代法

`SOR` 即逐次超松弛迭代法

对于CG, Jacobi, Gauss_Seidel

输入为矩阵A和b

输出为:

`result` : 即结果

`errors` : 即每一步迭代的相对误差

`timeUse` : 即迭代使用的步数

例如

```
1. function [ result, errors, timeUse ] = Gauss_Seidel( A, b )
```

对于SOR

输入为矩阵A和b, 以及将要使用的松弛因子w。

输出与上述迭代法算法相同。

```
1. function [ result, errors, timeUse ] = SOR( A, b, w )
```

Runner

有三个不同的Runner。

1. `Runner.m`

用雅克比、高斯塞德尔、共轭梯度法来计算同一矩阵(生成四次, 维度分别为10,50,100,200)。绘制三种方法的迭代步数差异。

2. `SORRunner.m`

对于维度为10,50,100,200的随机生成的矩阵, 分别用 $w = 0.5, 0.6, \dots 1.5$ 来计算SOR方法, 分析其收敛速度并绘图。

3. `EachFunctionRunner.m`

可以将其中所使用的的函数替换为 `Jacobi, Gauss-Seidel, CG` 中的任意一种, 分析其迭代步数与相对误差的变化。

PageRank

源数据存放在source.txt中, 其中去除了开头部分的注释, 只保留数据部分。

结果生成在test.xlsx中, 第一列为PageRank值, 第二列为该节点原本对应的序号(要比源文件中的节点大一号, 因为源文件中有0而矩阵的下标不能为0)。