**数值分析第三次上机练习实验报告**

**——矩阵特征值问题求解**

**力5 黄云帆**

**一、问题描述**

设A为以下形式的三对角矩阵(取适当阶数计算)

A = ．

1. 用Jacobi方法计算其特征值；
2. 用QR算法计算其特征值．

要求精度达到．

**二、方法描述——Jacobi方法和QR算法**

**1.矩阵特征值问题求解策略概述**

幂法和反幂法适用于最大(小)模分离矩阵的特征值的求解．值得注意的是：Aitken加速思想仍可适用；反幂法可以用于精确“打靶”，即求解某个值p附近的特征值，也可以用于求解给定特征值的特征向量．对于Hermite矩阵，由于各特征子空间正交，利用Rayleigh商加速可以更快地收敛到模最大且分离的特征值(三阶收敛)．

为了求得矩阵的全部特征值，考虑到特征值为相似变换下的不变量，我们可以利用涉及酉(正交)阵的矩阵分解以及某些简易的相似变换，这种方法可以同时获得对应的特征向量．其中比较具有代表性的正是题干所述的两者：适用于Hermite阵、应用Givens变换的Jacobi方法及其变式，适用于特征值之模分离的矩阵、应用Householder变换和QR分解的QR算法及其延伸．

**2.Jacobi方法**

Jacobi方法适用于求解Hermite矩阵的特征值问题，这类矩阵可实对角化的性质为 Jacobi方法提供了理论基础．题干中矩阵显然满足要求．概括地说，Jacobi方法的主要思想是通过一系列Givens变换，利用变换的旋转性质每次都精准地将“冒尖”的非对角元消成零，使得矩阵非对角元逐步趋于零，从而实现实对角化，并求得特征值和特征向量．需要说明的是，将所有非对角元同时消为零的情况几乎不可能做到，因此实际上只要各非对角元绝对值足够小即可．

经典Jacobi方法每次全矩阵搜索只对最大非对角元做一次Givens变换，效率太低；经过改进的Jacobi过关法为每次搜索更新阈值，只要某非对角元绝对值超过阈值，便对其做一次Givens变换，从而很大地提高了效率．后者在实际应用中，有至少两点需要注意：

(1) Givens矩阵的确定，即变换系数的确定 ．由于对于给定的 ，可求出不止一个，从而 等等．为了避免正负号的取舍麻烦，这里我们采取先求出 ，再利用万能公式求出其他三角函数值的策略．对于上面 的解算( 类似)，通过半角公式易反解出

选择这一解是为了充分利用 的值．但是这同时又带来了一个问题，即当时，，出现了奇异现象．因此在算法中，考虑到计算机存储与计算的有效位数，设置了的阈值，即当且仅当时，应用上述公式；否则应用一阶近似公式

应用该近似公式相对误差不超过，完全符合计算的精度要求．

(2) 计算结果精度的阈值设定，即非对角元上界以及对角元收敛精度．这种双重阈值的设定对结果带来的影响，由于属于后验的经验，因此将在结果分析一段中进行稍详细的分析．

**3.QR算法**

QR算法的核心思想是利用一般矩阵的QR分解结果对矩阵不断做正交变换，使其收敛于Schur分解上三角形式，从而求得特征值和特征向量．为了达到上述目的，也为了节约计算量，常常在QR算法之前利用Householder变换将一般矩阵化为Hessenberg矩阵，这类矩阵能够关于该算法保持形式不变．

QR算法的适用范围受到其收敛性定理的限制．在最简单的特征值按模分离的实矩阵情形，且特征向量构成的矩阵逆成立时，上述算法产生的矩阵序列基本收敛于上三角阵．对于特征值按模分离的复矩阵情形，还需要加上一定的子空间正交条件；对于等模特征值情况，仍是一个重要的充分条件，只是由于特征值并不按模相互分离，只能基本收敛于分块对角阵形式，相当于矩阵成为可约形式，可以进一步求解规模较小的子问题．

由于题干中的矩阵本身可以实对角化，其Schur形式即为对角阵，因此可以应用QR算法．

**三、方案设计**

本次实验针对题干中的两小问各编写了一个Matlab程序，分别保存为eig\_Jacobi.m，eig\_QR.m，其中第一个文件中包含主函数eig\_Jacobi和子函数Givens．需要说明的有三点：

(1)阈值的设置：上述两个程序中均根据Cauchy收敛准则的形式设置了特征值的精度阈值eps作为形参之一．另外，Jacobi方法的程序还设置了非对角元绝对值的上界阈值ubound作为形参，并在其子程序Givens中设置了可调参数 作为一阶近似计算的阈值．

(2)误差表征量的设置：在Jacobi方法中，为了对比起见，还设置了表征对角元矩阵误差2-范数的lamda变量，以及表征计算结果的非对角元平方和的sum变量．

(3)扫描(分解)次数跟踪：上述两个程序均设置变量tic用来跟踪次数．

在实验中将重点探求上面设置的各种参数、阈值之间的制约关系，以求计算出有效位数较多、精确度较高的特征值．

**四、计算结果及其分析**

通过上述程序的运行以及过程中参数、阈值的不断调整，我们获得了如下结果：

表格 1：Jacobi方法参数选取对计算结果的影响

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 维数n | ubound | eps | lambda | sum | 等价tic |
| 10 | 1.0E-03 | 1.0E-03 | 3.2058E-07 | 1.9660E-06 | 3\*1=3 |
| 1.0E-07 | 1.0E-14 | 1.0848E-08 | 6.1162E-15 | 7\*1=7 |
| 20 | 1.0E-03 | 1.0E-03 | 1.0267E-04 | 4.2212E-05 | 3\*4=12 |
| 1.0E-07 | 1.0E-11 | 1.5186E-07 | 1.5672E-13 | 5\*4=20 |

(1)表格1给出了Jacobi方法调整参数过程中比较有代表性的例子，其中对角元矩阵的误差是同相对准确的特征值矩阵(表格2中QR算法求得)而言的；第2、4行给出的是该维数下Jacobi方法程序所能给出的最优参数．从中可以清楚地看出，随着所设非对角元上界、特征值精度阈值的缩小，结果的精确程度不断提高，这是预料之中的．根据表格1以及上机实践的经验，需要指出以下几点：

a.上述过程的顺利进行有赖于一阶近似计算的阈值 的适当设置．若无此设置，经验指出，最终将会出现近似于”0/0”的情况，这将带来NaN的错误，该参数设置过小情况类似；若该参数设置过大，将产生较大的截断误差，致使结果误差较大．

表格 2：两种方法求出的特定阶数矩阵特征值及若干指标

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | 5 | 10 | 20 | 40 | |
| sum | 9.4152E-17 | 6.1162E-15 | 1.5672E-13 | 1.4945E-06 | |
| Jacobi方法 特征值列阵 | 0.2679491924 | 0.0810140528 | 0.0223383 | 0.0059 | 3.5430 |
| 3.7320508076 | 3.9189859472 | 3.9776617 | 3.9474 | 3.4410 |
| 3.0000000000 | 3.6825070657 | 3.9111456 | 3.9941 | 3.3307 |
| 1.0000000000 | 0.6902785321 | 1.2693180 | 1.3240 | 0.3641 |
| 2.0000000000 | 0.3174929343 | 0.1980623 | 0.0234 | 0.4570 |
| ubound取  eps取 | 3.3097214679 | 2.7306820 | 2.8181 | 0.5590 |
| 2.8308300260 | 3.2469796 | 3.0871 | 2.3808 |
| 2.2846296765 | 3.8019377 | 3.7191 | 3.2125 |
| 1.1691699740 | 0.0888544 | 3.9766 | 2.9554 |
| 1.7153703235 | 0.3475225 | 0.0932 | 1.9234 |
| \ | 1.5549581 | 1.0446 | 0.6693 |
| 0.5338963 | 0.9129 | 0.7875 |
| 3.4661037 | 0.0526 | 1.4700 |
| 3.6524775 | 3.8550 | 2.5300 |
| 0.7530204 | 3.7923 | 2.6760 |
| 1.0000000 | 3.6359 | 2.2294 |
| 1.8505398 | 3.9068 | 1.1819 |
| 3.0000000 | 0.1450 | 1.6192 |
| 2.4450419 | 0.2809 | 1.7706 |
| 2.1494602 | 0.2077 | 2.0766 |
| tic | 83 | 273 | 929 | 3250 | |
| QR算法 特征值列阵 | 3.7320508076 | 3.9189859472 | 3.9776616525 | 3.994132 | 1.923395 |
| 3.0000000000 | 3.6825070657 | 3.9111456116 | 3.976561 | 1.770633 |
| 2.0000000000 | 3.3097214679 | 3.8019377358 | 3.947391 | 1.619218 |
| 1.0000000000 | 2.8308300260 | 3.6524775486 | 3.906793 | 1.470037 |
| 0.2679491924 | 2.2846296765 | 3.4661037437 | 3.855005 | 1.323966 |
| eps取 | 1.7153703235 | 3.2469796037 | 3.792331 | 1.181863 |
| 1.1691699740 | 3.0000000000 | 3.719139 | 1.044560 |
| 0.6902785321 | 2.7306820487 | 3.635859 | 0.912865 |
| 0.3174929343 | 2.4450418679 | 3.542978 | 0.787549 |
| 0.0810140528 | 2.1494601872 | 3.441043 | 0.669349 |
| \ | 1.8505398128 | 3.330651 | 0.558957 |
| 1.5549581321 | 3.212451 | 0.457022 |
| 1.2693179513 | 3.087135 | 0.364141 |
| 1.0000000000 | 2.955440 | 0.280861 |
| 0.7530203963 | 2.818137 | 0.207669 |
| 0.5338962563 | 2.676034 | 0.144995 |
| 0.3475224514 | 2.529963 | 0.093207 |
| 0.1980622642 | 2.380782 | 0.052609 |
| 0.0888543884 | 2.229367 | 0.023439 |
| 0.0223383475 | 2.076605 | 0.005868 |

b. 非对角元绝对值的上界阈值ubound设置不能过小，而是应该与特征值的精度阈值eps以及一阶近似计算的阈值 相适应．否则，在对角元已经明显收敛的情况下，非对角元仍未达到要求，上述Jacobi程序便会继续循环，使得对角元偏离特征值．在实际计算中，当ubound参数设置得很大时，对角元会出现异常现象，即最后的对角元矩阵恰巧是初始矩阵的对角元，猜测可能有截断误差积累、Givens矩阵最终接近单位阵等方面的原因．

(2)表格2给出了两种方法下的维数为5,10,20,40的题干三对角矩阵的特征值计算结果，40阶矩阵的精度至少可达．首先，我们看到随着维数的增大，非对角元平方和最终值也随之增大，这也使得求得的对角元相对的有效位数减少．其次，我们可以看到QR算法对于eps的下界要求相对比较宽松，因此可以获得更加准确的特征值．但是，随着矩阵阶数的增大，QR分解所需次数不断增多，所耗费的时间也近似以比值为3.5的速度几何增长，这是其比较大的局限性．

**五、结论**

本次实验采用Jacobi和QR两种算法求解题给对称正定三对角矩阵的特征值，从中我们可以获得如下经验：

Jacobi算法由于其利用的Givens变换简单，因此速度较快．但在具体操作中，受到计算机舍入误差的制约，并且随着矩阵阶数的不断增加，各类阈值的细致优化是一个比较大的工程．相比之下，QR算法操作简便，Hessenberg矩阵的使用也大大减少了计算量，其计算精度也更高．其不足之处在于QR分解相较于Givens变换复杂了许多，随着矩阵阶数增高其求解时间大幅增加．

如果要考虑高阶矩阵的特征值求解，如果追求比较快的收敛速度且精度要求不高，Jacobi算法可以满足要求；如果追求比较高的收敛精度，则可以选择QR分解并考虑其进一步优化以缩短时间．