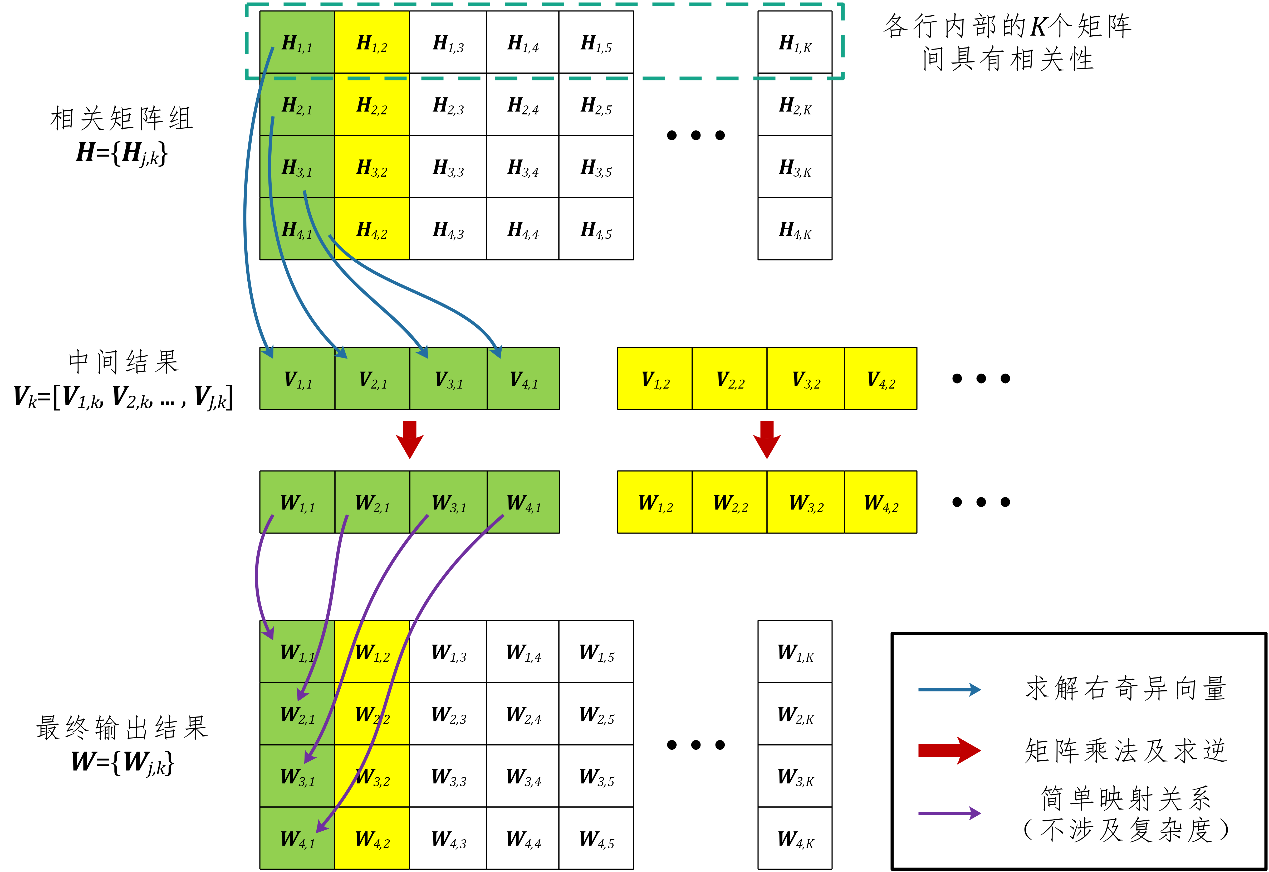
# 附录一：矩阵的奇异值分解简介

矩阵的奇异值分解，其含义在于，对于任意一个复数矩阵，都可以找到两个酉矩阵和（当为实数矩阵时，则对应地可以找到两个正交矩阵和）使得，其中

为对角矩阵，为矩阵的秩。此时，对角线上的各个元素为矩阵的由大到小顺序的奇异值，为矩阵的左奇异向量（列向量）构成的矩阵，为矩阵的右奇异向量（列向量）构成的矩阵。

在多种信号处理场景中，经常可以发现矩阵的部分奇异值较大，而另一部分奇异值较小。较小的奇异值对矩阵具体数值的贡献也较低，因此在矩阵数据处理中通常会截断到前个奇异值，利用该前个奇异值以及对应的左奇异向量、右奇异向量计算得到的矩阵与原矩阵的差异性可以很低。

# 附录二：相关矩阵组的计算流程图

****

注意，纵向排列的（如绿色列块、黄色列块），在求解右奇异向量得到后是将它们横向地拼接为。在常规的计算流程中，矩阵组各列块是独立计算的。当考虑各行块内部矩阵间的相关性时，也可以关注将不同列块结合在一起进行联合计算的可能性，这潜在地可以降低总体计算复杂度。

# 附录三：基于双对角化和QR分解的SVD分解简介

对于任意一个复数矩阵，其中，可以通过如下步骤来实现较低计算复杂度的SVD分解过程。

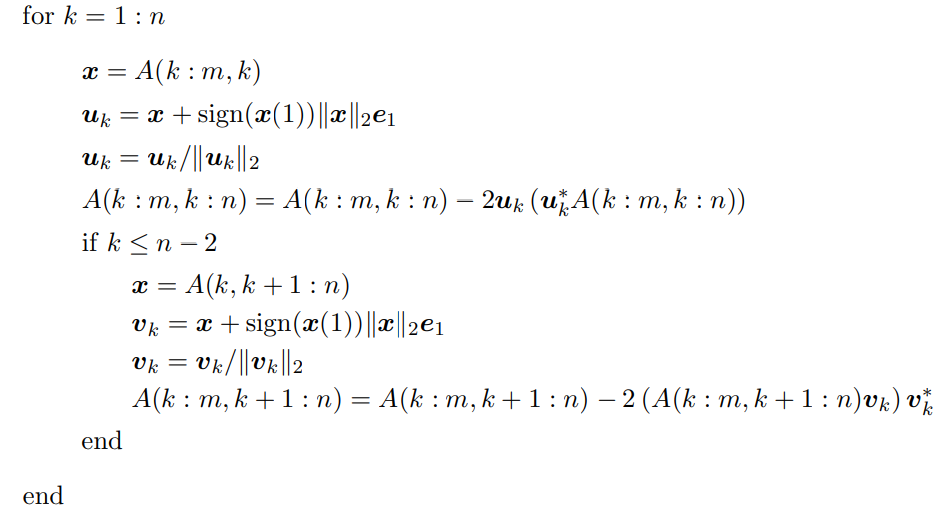
① 对矩阵进行QR分解，得到：

其中，为酉矩阵，维度为；为上三角矩阵，维度为。由于矩阵中非零元素仅在于对角线及对角线以上的部分，在矩阵上进行的后续计算的复杂度将主要受到取值的影响，对于的取值不再敏感。

② 对矩阵进行双对角化分解，得到：

其中，和均为酉矩阵，维度分别为和；为双对角矩阵（bidiagonal matrix），即在两个对角线上存在非零元素的矩阵，维度为。

如下为Golub-Kahan双对角化方法的具体算法流程：



③ 利用一系列的酉矩阵，通过QR迭代的方式将双对角矩阵转换为对角矩阵：

其中，和均为酉矩阵，维度分别为和；基于迭代的QR分解过程，实际上是若干个酉矩阵的乘积，同样的，也实际上是若干个酉矩阵的乘积；为对角矩阵，维度为。

将以上步骤结合在一起，即实现了对矩阵的SVD分解过程：

注意，在本问题中，也可以改写为，即可以简单地转化为针对行数大于列数的矩阵的SVD分解过程。或者，将前述QR分解更换为LQ分解，再针对下三角矩阵进行双对角化及QR迭代操作。

# 附录四：降低矩阵乘法的计算复杂度的思想

历史文献已经证明，矩阵求逆的计算复杂度与矩阵乘法的计算复杂度在均使用的方式进行度量时是相同的，因此，降低矩阵乘法的计算复杂度可以有效地降低矩阵求逆的计算复杂度。

一种降低矩阵乘法的计算复杂度的思想是通过合理的构造（probabilistic constructions）和转化来减少运算数目的需求。这里通过如下例子进行简要说明：在正文中，一种直观的复数乘法过程使用了4次实数乘法和2次实数加(减)法。下面换用另一种计算方法，我们令

，使用2次实数加(减)法和1次实数乘法；

，使用1次实数乘法；

，使用1次实数乘法。

则有。经统计可知，该复数乘法过程共使用3次实数乘法和5次实数加(减)法。虽然根据表格，原方法与上述计算方法具有相同的计算复杂度14，但将标准计算进行构造（probabilistic constructions）和转化来降低计算复杂度的思想仍具有较强的启发性。

基于这种思想，文献[3]中的Strassen's方法采用了分治法（divide-and-conquer）的思路，对于阶数为的矩阵乘法可以方便地使用，其算法流程如下：

输入，

其中，

。

此时，两个阶矩阵间的矩阵乘法将使用次复数乘法，以及次复数加法，当结合了正文提供的表格后，可知Strassen's方法下阶矩阵乘法的计算复杂度为：

文献[4]针对矩阵乘法的计算复杂度进行了进一步的理论分析，并证明两个阶矩阵的矩阵乘法渐进性地具有的计算复杂度。2010年后出现了新的文章将上述计算复杂度的极限进一步降低，然而，真正地将上述方法用于实践仍然十分困难，一般无法获取到通用的算法步骤。

# 附录五：基于随机SVD的SVD分解简介

本附录将网页[6]中的内容进行部分摘录和总结。

对于任意一个复数矩阵，其中，尝试寻找维度为的矩阵、维度为的对角矩阵、维度为的酉矩阵，使得，且要求。对此，文献[5]给出了一种基于随机（randomness）方法，包括如下步骤：

① 利用随机方法计算矩阵的值域（range，由矩阵列向量张成的线性空间）的近似基（basis），构造由个标准正交（orthonormal）列矢量组成的矩阵，满足。

② 计算维度为的矩阵的SVD分解：

由于矩阵的行数小于矩阵，因此对矩阵执行SVD分解可以消耗较少的计算资源。

将以上步骤结合在一起，即实现了对矩阵的近似SVD分解过程：

注意，此公式中的不必呈现为酉矩阵。

在使用了随机方法的步骤①中，可以使用如下算法获取矩阵：

* 抽取个维度为的高斯随机矢量，构成维度为的高斯随机矩阵；
* 令矩阵；
* 利用如QR分解的方法，获取标准正交矩阵。

进一步地，考虑到的取值可能涉及到近似过程中所能接受的误差，可以采用增量的方式来构建矩阵。令表示维度为（即，包含列）的迭代标准正交矩阵，为空矩阵，则可以使用如下迭代过程产生可接受的矩阵：

* 抽取个维度为的高斯随机矢量，令；
* 构造；
* 令；
* 构造**。**

持续以上迭代过程，直到与的近似程度满足需求。