## **3.2** SVD算子低位宽实现

### 3.2.1 SVD算法

通过对现有的SVD算法进行调研，带位移QR迭代算法[25]比较成熟，而且可以通过控制迭代次数等方式来达到计算复杂度和精度之间的平衡，适合在低位宽计算中采用；除带位移的QR迭代方法之外还将对现在广泛使用的单边Jacobi SVD方案进行介绍和仿真。

带位移QR迭代算法需要先将复数矩阵进行实双对角化，由于实双对角化方法的不同，本节对较为成熟的Householder双对角化和Lanczos双对角化进行介绍。

3.2.1.1 Householder SVD

假设维度为的复矩阵，且有，运用Householder变换双对角化和隐式QR迭代方法对A进行SVD分解的具体过程如下：

首先利用复数Householder变换，将原矩阵转化为一个实数双对角矩阵，这样在第二部隐式QR迭代时就只需要考虑实数矩阵。

将分块为

 (3.1)

先计算阶复Householder变换矩阵

 (3.2)

使得

 (3.3)

其中

 (3.4)

 (3.5)

并且形成

 (3.6)

得到行向量，接着计算阶复Householder变换矩阵

 (3.7)

使得

 (3.8)

其中

 (3.9)

 (3.10)

并且形成

 (3.11)

然后对依次进行

(a)计算阶Householder变换

 (3.12)

使得

 (3.13)

并且形成

 (3.14)

(b)计算n-k阶Householder变换

 (3.15)

使得

 (3.16)

并形成：

 (3.17)

最后，计算阶Householder变换矩阵

 (3.18)

使得

 (3.19)

现令

 (3.20)

 (3.21)

 (3.22)

 (3.23)

则有

 (3.24)

即实现对复矩阵的二对角化，得到实二对角矩阵B。

3.2.1.2 Lanczos SVD

将复数矩阵转化为实双对角阵以及酉矩阵的过程还可以通过Lanczos双对角化实现，传统的Lanczos算法受机器计算精度舍入误差的影响，其Lanczos向量之间会随着迭代逐渐失去正交性，想要获得正交的Lanczos向量，就需要对Lanczos双对角化过程进行再正交化[1]。其算法如下：

|  |
| --- |
| 算法3.1 Lanczos正交双对角化 |
| 1. 初始化QianJianTec1711355068280，以及QianJianTec1711355938877，QianJianTec1711355979730，QianJianTec1711355228115（算法此处设置QianJianTec1711355427815） 2. QianJianTec1711356232856 3. QianJianTec1711356297745 4. QianJianTec1711356339720 5. QianJianTec1711356392756 6. QianJianTec1711356495823 7. QianJianTec1711356545727 8. QianJianTec1711356569536 9. QianJianTec1711356611980 10. QianJianTec1711356650730 11. QianJianTec1711356682507 12. QianJianTec1711356714280 13. QianJianTec1711356725232 |

执行以上迭代后可获得式(3.24)，以及其中

, (3.25)

从而和Householder变换一样完成了复数矩阵的实双对角化。接下来需要对对实矩阵进行带位移QR迭代。

得到后，要对进行带位移的隐式QR迭代，但如果先计算再分解，会极大提高计算复杂度，因此实际操作中直接对矩阵进行迭代。

第一次QR迭代的过程如下

取矩阵的右下角2×2主子阵

 (3.26)

靠近最近的特征值作为位移

 (3.27)

其中

 (3.28)

确定Givens变换，其中，满足

 (3.29)

矩阵右乘后，在第二行第一列出现了非零元素，可以交替左乘和右乘一系列Givens变换矩阵，每次都会将非零元素移动一个位置，直到矩阵又重新恢复为双对角阵，即使

 (3.40)

为双对角阵，这样第一次QR迭代结束，该过程相当于

 (3.41)

其中

 (3.42)

 (3.43)

经过若干次QR迭代之后，矩阵右下角的次对角元逐渐减小。如果满足下面关系

 (3.44)

式中与计算精度有关。则认为已经足够小，直接将其设为0。此时，是矩阵的一个奇异值，下次迭代时，就可以不对这一行元素进行计算，即把原矩阵的阶数降低一阶。随着迭代的进行，矩阵阶数逐渐减小，最终收敛到实对角阵，其对角线上的元素即为所有的奇异值

 (3.45)

即

 (3.46)

其中

 (3.47)

 (3.48)

获得以上结果后就能通过以下方法求得SVD分解的左奇异向量。

结合复矩阵的二对角化过程：

 (3.49)

和实矩阵的QR迭代分解

 (3.50)

可得

 (3.51)

令

 (3.52)

 (3.53)

则有

 (3.54)

那么S即为SVD分解的左奇异向量。至此，使用两种实双对角化方法的复矩阵SVD完成。

3.2.1.3 单边Jacobi SVD

在本次项目中对单边Jacobi SVD方法也进行了计算上的仿真与讨论。传统Jacobi SVD的方法是双边Jacobi的方法，而单边Jacobi是对传统方法的改进，接下来介绍其算法原理。

对于一个矩阵，其SVD为：。若将看做一个整体，由于酉矩阵列向量之间的正交性，的列向量之间也是正交的，故考虑将，即原矩阵通过酉变换使得列向量之间正交，此时将相互正交的列向量取其模单独提出获得酉矩阵U，最后再根据提出的模进行大小排序并同步到U和V的排列，此时排序好的模所组成的对角阵即为奇异值矩阵，U和V分别是其左右奇异向量矩阵。至此完成单边Jacobi SVD的分解。

但对于大规模的非方阵来说，其秩不超过矩阵长宽较小的维度，此时每次旋转都是对原矩阵的操作，会增加额外的计算，并在低位宽计算环境时产生额外的误差传播。故可以考虑将其进行QR窄分解：，此时对R进行分解既没有损失信息也降低了计算的维度，故项目方案将单边Jacobi增加了QR分解进行计算上面的降维。其原理为：对共轭转置与自身相乘则有，将QR分解代入则，即可以通过对R进行SVD获得原矩阵A的奇异值矩阵和右奇异向量矩阵，再通过获得其左奇异向量矩阵。

具体算法见下算法3.2。

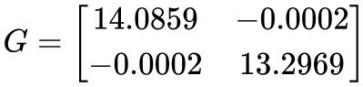
|  |
| --- |
| 算法3.2 单边Jacobi SVD |
| 输入：矩阵，收敛精度   1. 对A进行窄QR分解：，， 2. 对R进行Jacobi旋转直到其每列向量相互正交（收敛条件：）： 3. 记为A的右奇异向量 4. 提取B中每列向量的模：，并构成对角阵为奇异值矩阵： 5. 计算左奇异向量矩阵   输出： |

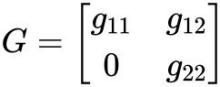
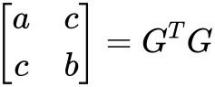
### 3.2.2 SVD算法的低位宽实现与改进

通过上述带位移的QR迭代SVD算法可以实现对任意维度复数矩阵的SVD分解，通过将过程的各个计算算子使用本项目的低位宽算子替代，即可在低位宽（16bit）完成复数矩阵的SVD分解。

但是由于16bit位宽的限制，运算过程最值表示无法超出最大整数表示范围。在SVD计算过程中，平方、除法都极易面临着溢出的情况，并且在Givens变换中，若副对角元素接近0，旋转因子就会出现大数除以接近0的小数的情况，从而产生溢出，故在本次项目低位宽SVD的设计中采用局部放缩的方式，用于避免溢出的情况，具体实现如下例子所示：

当迭代至最后左上2×2子矩阵时，子矩阵已收敛为

 (3.55)

其中假定（在高精度下左下角元素在Householder变换时逼近于0，在16bit位宽下会可见地接近0而不为0），接下来计算，有

QianJianTec1711434741883 (3.56)

接下来计算雅各比旋转因子

QianJianTec1711434914936 (3.57)

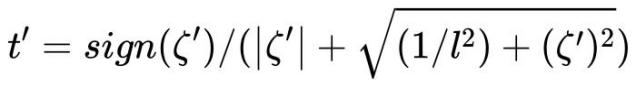
QianJianTec1711435122002 (3.58)

QianJianTec1711435181263 (3.59)

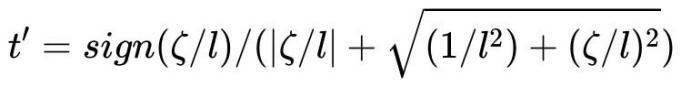
QianJianTec1711435190442 (3.60)

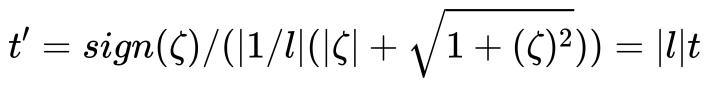
根据已有结果，则QianJianTec1711435361422，QianJianTec1711435475102超出了16bit的最大表示范围，为了运算能继续进行，故采取局部放缩，如下：

QianJianTec1711436891091 (3.61)

 (3.62)

其中*l*表示放缩因子，在本次例子中可选为200，此时t的计算避免了计算数值超出最大表示范围，然后需要根据原式进行恢复，将式带入，有

 (3.63)

 (3.64)

故需要还原成未放缩的结果，仅需赋值QianJianTec1711438176854，由于放缩因子取值常设为正数，故以上绝对值可去。至此，完成低位宽超出运算范围时进行的放缩设计。

但是并不是每一个矩阵都会出现超出表示范围的运算，而且低位宽的放缩可能会对最终的计算结果产生一定的误差，导致误差传播，故需在进行放缩前需对其进行判别，在此例子中，若QianJianTec1711442591886则其平方会出现溢出，故在放缩前增加判别条件，若QianJianTec1711442662706则进行放缩，否则正常运算。

在整个QR迭代过程当中，由于低位宽的限制，对于左上角的2×2矩阵，式3.35的收敛精度终止条件可能无法达到，会一直执行最后的迭代，受低位宽精度影响，其误差传播将导致计算错误，所以需要对收敛精度进行调整。而且对于不同规模的矩阵其收敛精度的设置也会对其SVD分解结果产生影响，小规模矩阵在设置较大的收敛精度时，会导致其SVD分解在副对角线还未收敛逼近至0时就会停止计算，但对于大规模的矩阵而言，较小的收敛精度设置将会导致SVD分解无法让副对角线元素达到收敛范围从而进入死循环并且导致计算错误，所以需要在矩阵规模和收敛精度设置之间需要一个权衡的设置。

### 3.2.3 不同SVD算法低位宽下的分析与比较

3.2.3.1 Householder 与Lanczos的比较

根据3.2.1中对Householder算法的描述，不难发现，Householder算法对复数矩阵进行实双对角化计算时，每一次Householder变换产生的Householder矩阵都需要对左右矩阵U、V和中间矩阵B进行乘积才能实现变换。

对于一个维度为QianJianTec1712471573869的矩阵进行式双对角化时，需要对矩阵进行n次左变换和n-2次右变换，每次左变换产生的Householder矩阵与U和B同时相乘，需要QianJianTec1712471959823次浮点运算；每次右变换产生的Householder矩阵与B和V同时相乘，需要QianJianTec1712472091794次浮点运算。一次完整的实双对角化不包括计算Householder矩阵就已经需要QianJianTec1712472236552次浮点运算，并且左/右变换在计算时是串联式计算，前一次左/右变换在低位宽下产生的精度误差将会随着Householder变换的迭代不断传播，从而导致实双对角化误差变大。

与之相比，Lanczos算法在实现实双对角化时，U和V的计算过程并没有像Householder变换那样需要对整个矩阵进行相乘，而是根据原矩阵的范数性质进行逐个向量的更新，其双对角线的元素也是根据范数进行更新最后直接赋值，其中间涉及的浮点运算包括矩阵与向量相乘及向量的数乘，其浮点运算次数远小于Householder实双对角化中矩阵与矩阵相乘的运算数，并且逐列更新向量且逐列计算正交降低了U和V计算的误差传播，并且避免了传统Lanczos算法由于机器精度限制导致的U和V向量之间随着维度增大而产生的不正交现象及“幽灵”特征值问题。

除浮点运算上的差别外，对于复数矩阵SVD而言，Householder变换本质上是一种镜像变换，在无限精度下可以实现完美的复数向量到单一实数方向的映射，但在低位宽的限制下，复数经过镜像变换时会造成辐角方向和幅度大小上的“泄露”（见图3.1），使得经过变换产生的双对角矩阵不是完全的实数矩阵。与之相比，Lanczos算法直接计算并赋值双对角元素的方法避免了这种低位宽导致的计算错误。

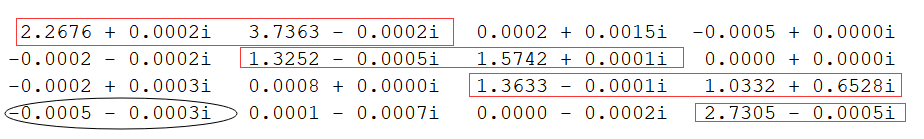


图 3.1 16bit Householder变换双对角化产生的“泄露”

结合浮点运算次数多而导致的误差传播上的区别以及Householder的“泄露”问题，在不同位宽的计算条件下Lanczos在双对角化过程中相对误差都比Householder更低（见图3.2），并且在高维度矩阵计算双对角化时Householder算法低位宽计算的相对误差会大幅度上升。

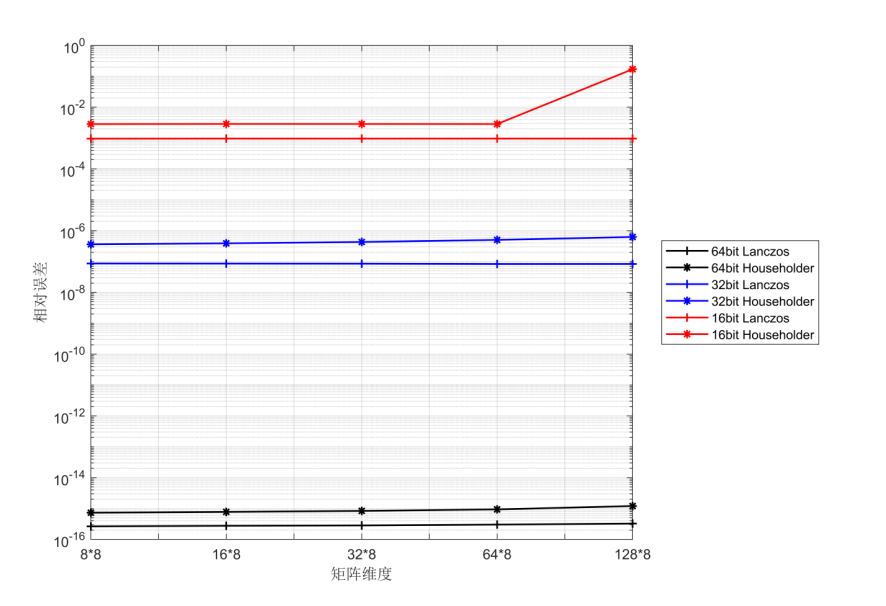


图 3.2 Lanczos和Householder在不同精度下双对角化的相对误差

下表是Lanczos和Householder两种算法对QianJianTec1712483425641的矩阵进行SVD的浮点运算数，由于两种算法在SVD过程中仅双对角化部分不同，QR迭代部分算法一致，故QR迭代求SVD部分算法浮点运算数统一记为QianJianTec1712480757476[2]。

表 3.2 Lanczos和Householder两种算法进行双对角化和SVD所需运算数

|  |  |  |
| --- | --- | --- |
| 过程 | 所需浮点运算数 | 64×8复数矩阵所需运算数 |
| Lanczos正交双对角化 | QianJianTec1712480975712 | 27456 |
| Householder双对角化 | QianJianTec1712481086385  QianJianTec1712481170159 | 111384 |
| Lanczos SVD | QianJianTec1712483218629 | 195904 |
| Householder SVD | QianJianTec1712483269803  QianJianTec1712483305346 | 279832 |

3.2.3.2 单边Jacobi算法的分析与仿真

单边Jacobi算法由于其使用不同的扫描旋转方式会导致不同的计算顺序，同时受收敛精度的影响，因此没有严格的理论可以预测对指定的收敛精度所需要的扫描数，Brent和Luk直观认为扫描次数正比于[3]，故本小节并无对单边Jacobi SVD计算所需浮点数的量化，仅在算法流程及仿真的角度做简要的分析。

在仿真中发现，两种以QR迭代为主要过程的算法在莱斯信道的计算场景下会出现较大的偏差，其原因在于QR迭代是带位移的迭代。位移与主对角线上的元素有关，当相邻的两个主对角元素大小差距过大，计算位移时就会使得在最大奇异值的计算当中产生偏差。而单边Jacobi SVD算法是构建向量之间的正交化，因此对于相邻奇异值差距的鲁棒性较强，但在瑞利信道、CDL-B、CDL-D信道中并没有出现相邻奇异值过大的情形，因此Lanczos SVD仍适用于三种信道的奇异值分解；后续第五章节中出现的莱斯信道的仿真，完整的SVD计算将使用单边Jacobi SVD计算。

在64bit位宽下Lanczos SVD和单边Jacobi SVD算法的分解性能与信道条件数的关系，以及瑞利信道、CDL-B、CDL-D信道的条件数分布见下图所示，此处由于CDL-B、CDL-D信道可能为非满秩信道，因此此处条件数定义为最大奇异值与最小非零奇异值的比值。

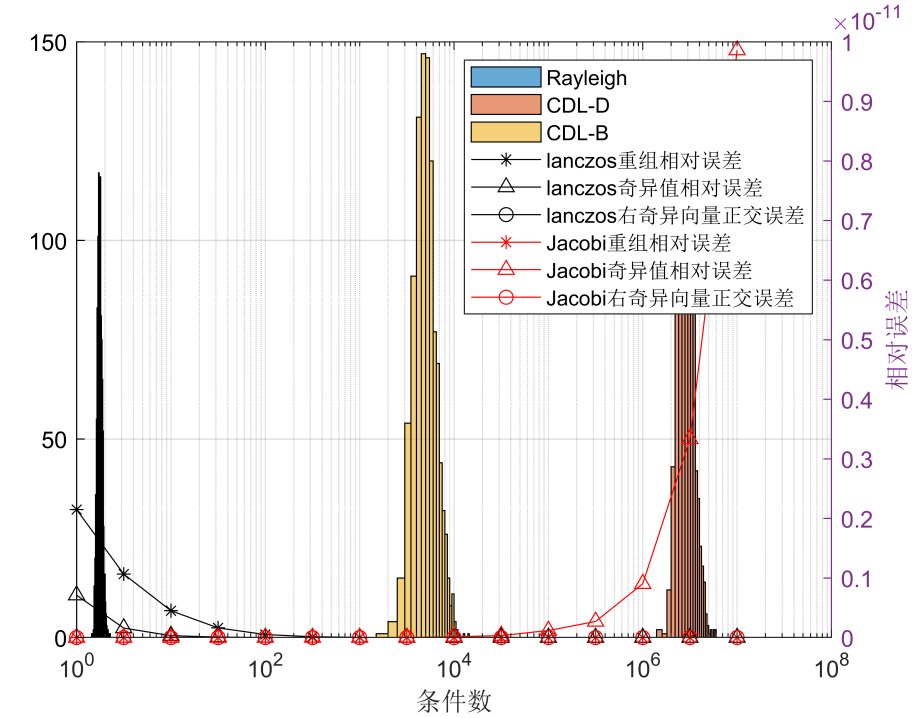


图 3.3 三种信道的条件数分布及两种算法的性能随条件数的变化

可以看出三种信道分布的条件数区间内单边Jacobi SVD和Lanczos SVD都具有很好的性能，当条件数非常大时单边Jacobi SVD的奇异值误差会有所升高。

其高位宽下与Lanczos和Householder SVD的仿真如下。

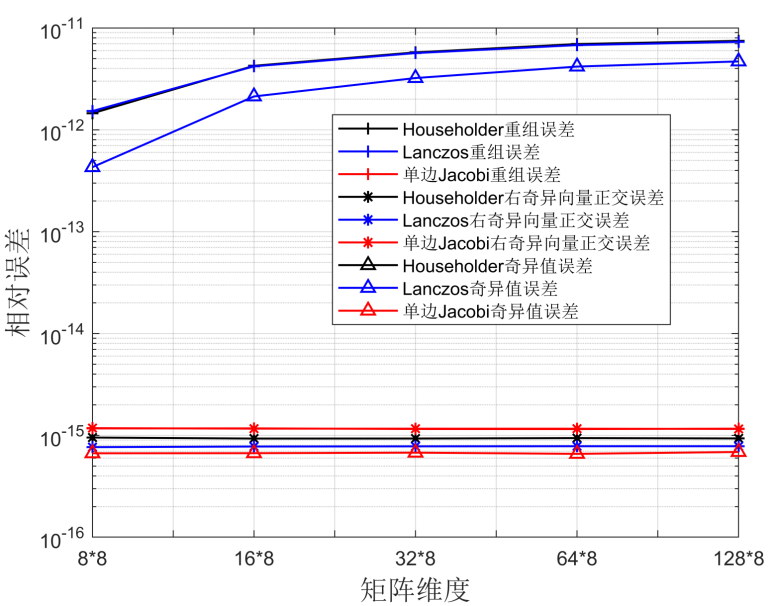


图 3.4 不同矩阵维度下三种SVD算法的性能比较

图中重组误差三种算法几乎一致，受收敛精度的影响。根据仿真结果可以看出，在右奇异向量正交性、重组误差上三种算法性能相近，但单边Jacobi SVD的奇异值相对误差明显优于另外两个算法。这是在计算过程中单边Jacobi一直进行着正交变换，在高位宽的计算环境中几乎不会对矩阵的F范数有改变；但是另外两个算法在双对角化的过程和QR迭代的过程中都一定程度上对矩阵的元素进行了舍入，从而导致其在奇异值上有一定的误差。

# **第五章** 低位宽SVD算子优化设计

## 5.1低秩近似SVD算法

在深度学习领域，许多工作为了降低矩阵的计算维度同时尽可能保留矩阵的主要特征，都对高维矩阵使用主成分分析(PCA)等降维方法。其中随机奇异值分解(Randomize Singular Value Decomposition, RSVD)[4]是使用利用随机方法在尽可能保留最大若干个奇异值的角度降维来保留矩阵的主要信息，这种随机化的方法确保其对所有矩阵都具有效果，但其具体性能会受被分解的矩阵和采样矩阵之间隐含的关系影响。

在通信领域的大规模MIMO技术中，SVD预编码被广泛应用。在MIMO系统SVD预编码的低流传输中，我们仅需较大的前几个奇异值与其对应的奇异向量进行预编码，对于比较小的奇异值及对应的奇异向量可以进行舍弃。

现在业界使用的SVD预编码是进行完整SVD计算后再取用需要使用的前几个奇异值及其右奇异向量，但完整SVD的计算复杂度随着大规模MIMO信道的矩阵维度变大而极大的增加，在非全流的通信场景中计算后几个不需要的奇异值对计算的性能是一种计算资源上的浪费。尤其是在低位宽的计算环境中，计算多余不需要的奇异值及奇异向量所增加的计算复杂度还会使得需要的奇异值产生偏差。因此，在低位宽计算环境下利用降维算法降低SVD的复杂度，同时尽可能地减少对所需奇异值和奇异向量近似的误差，是提升低位宽通信性能的一个重要的思想，故考虑将如RSVD的低秩近似方法引入通信系统，并考虑将其进一步改进。

本小节将介绍被普遍使用的RSVD算法及其数学意义，并根据其数学意义在通信领域进行联动，提出基于角域思想的进一步的改进方法，并讨论这两种算法的计算复杂度，并进行高位宽计算环境下的仿真。

### 5.1.1 随机奇异值分解

随机奇异值分解，即RSVD，其数学意义是通过将高维矩阵与构建低维的高斯随机矩阵相乘，在低维高斯矩阵的随机方向上采样进行投影，获取其正交基，此正交基包含了高维矩阵的低维投影信息，再将其乘回原高维矩阵就可以实现降维，最后对降维后的矩阵进行奇异值分解，就可以获得高维矩阵的低秩近似。算法流程见算法5.1。

|  |
| --- |
| 算法5.1 RSVD |
| 输入：矩阵，采样数   1. 构建高斯随机矩阵。 2. ， 3. 对Y进行窄QR分解：，， 4. 将正交基Q乘回高维矩阵A实现降维：， 5. 对B进行低维SVD分解：，，， 6. 将正交基乘回进行还原：，   输出： |

但RSVD由于其随机采样的特性，无法根据矩阵的特性自适应地向更优的性能进行调整，所以其不论是矩阵重组误差，还是前若干个奇异值的相对误差都会较大，在面对特征明显的矩阵时其损失的信息就会相当大，因此不适合直接套用进通信系统，需要对其进一步的调整，使其能针对矩阵的特性进行处理。

### 5.1.2 角域采样奇异值分解

RSVD的降维方法是在采样方向和长度上的随机，这种广泛的采样对于瑞利信道这样奇异值差距不大的矩阵中有着比较好的优势，但在一些条件数大的矩阵中，RSVD可能会在奇异值较小的方向上有比较大的采样和投影，这样一种平衡的采样方法会保留过多不需要的信息，为了能在奇异值大的方向尽可能地多些采样，在奇异值小的方向尽量舍弃，提出了基于角域的采样方法：角域采样奇异值分解(Angular Domain Sampling Singular Value Decomposition, ADSSVD)。

信道矩阵可以通过下式转换至角域：

(5.1)

其中，是角域矩阵，和是酉化的DFT矩阵，角域矩阵的元素能粗略反映出信道传输的发射角和接收角以及每个角度上的增益和相移，并且由于经过的变换是酉变换，角域转换并不改变信道的能量和矩阵奇异值大小。经过角域变换后的信道矩阵具有非常明显的特征，根据本次项目仿真使用的信道，将接收天线数为8发射天线数为64的瑞利信道、莱斯信道、CDL-B信道及CDL-D信道转换至角域的模值可视化，见图5.1。

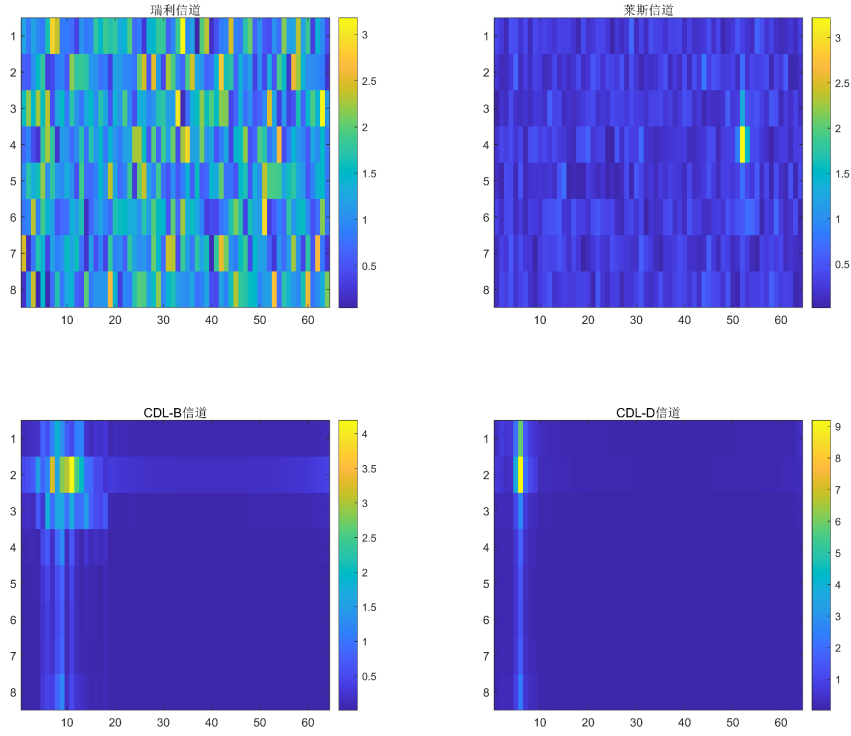


图 5.1 不同种类信道矩阵的角域幅值可视化

其中瑞利信道是非视距路径传输，其信道矩阵满秩，条件数较小，在角域中体现为每个元素的幅值相近，没有特别突出的元素；莱斯信道则是视距路径传输和非视距路径传输按照莱斯因子的比例结合而成，是同时具有视距路径和非视距路径传输特性的的信道，其信道矩阵满秩，条件数随莱斯因子变化；CDL-B信道是以非视距路径簇的传输为主，信道矩阵不满秩，其角域体现为在一定的角度范围会比较突出，在其他范围模值几乎为0；与CDL-B信道类似，CDL-D信道以视距路径簇的传输为主，在角域表示中仅存在个别列有非零值，且存在特别突出的单一值。

角域采样的思想是将信道矩阵的能量在角域中集中起来，将信道从以奇异向量为基底构成的高维空间转换至以DFT向量为基底构成的高维空间，此时矩阵的能量集中于特定的矩阵元素。于是此时在最大元素所在的行列对应的DFT向量方向进行采样能获得最大程度的采样效果。于是在RSVD原型中，将角域矩阵列向量最大对应的DFT向量组成的矩阵作为采样矩阵，即在角域能量最大的方向进行采样降维。具体算法如下算法5.2所示。

|  |
| --- |
| 算法5.2 ADSSVD |
| 输入：信道矩阵，采样数   1. 构建角域矩阵 2. 计算角域矩阵的列范数，并排序取前个最大的索引 3. 构建采样矩阵 4. ， 5. 对Y进行窄QR分解：，， 6. 将正交基Q乘回高维矩阵H实现降维：， 7. 对B进行低维SVD分解：，，， 8. 将正交基乘回进行还原：，   输出： |

可以看出ADSSVD与RSVD的区别仅在采样矩阵的不同，其采样后的算法步骤两者一致。

接下来将讨论两者的计算复杂度及其对误差传播的影响。由于这两种算法都对降维后的低维矩阵进行SVD分解，故此处SVD的计算复杂度参考表3.2中Lanczos SVD的计算复杂度进行计算。对于RSVD而言，与高斯采样矩阵的乘法使用次浮点计算，QR分解使用次浮点运算，正交基乘回原始矩阵使用了次浮点计算；对于ADSSVD而言，除了具备RSVD全流程的计算复杂度外，还有构建角域矩阵的，以及计算列范数的复杂度和排序查找复杂度。整合其复杂度由下表5.1所示。

表 5.1 RSVD和ADSSVD两种算法所需运算数

|  |  |  |
| --- | --- | --- |
| 算法 | 所需浮点运算数 | 64×8复数矩阵k=4所需运算数 |
| RSVD |  | 17206 |
| ADSSVD | 获取角域信息过程：  排序查找复杂度  计算过程：  同RSVD | 获取角域信息的复杂度+17206 |

从算法过程看，RSVD和ADSSVD都只是寻找矩阵在低维空间的正交基，ADSSVD和RSVD相比，仅是在寻找正交基的过程中增加了复杂度，查找的过程会增加时间上的复杂度，其计算时间受不同的排序算法影响；同时寻找的正交基是DFT向量，不同矩阵查找选择的正交基仅会存在DFT向量选取的区别，而不会引入额外的计算误差，即ADSSVD在误差传播的角度能保持和RSVD一样的性能，在提升降维性能同时也不会导致计算误差的传播。与此同时，固定的DFT矩阵也相应减少了存储的开支。

### 5.1.3高位宽下的低秩近似SVD算法仿真

本部分对RSVD和ADSSVD算法进行64bit高位宽计算环境下的仿真，以考究两种算法不同采样阶数下对不同类型信道矩阵、矩阵维度的矩阵重组相对误差、奇异值相对误差、右奇异向量正交性的三个维度的SVD性能效果，其计算比较方式参照以下式5.2至5.5。其中SVD分解：，k表示采样数，当RSVD和ADSSVD降维近似时，仅输出前k个奇异值及其对应的奇异向量，故考虑其性能指标时都只计算前k个奇异值及其对应的奇异向量。参考值选取为Matlab svd()函数计算输出的前k个奇异值及其对应奇异向量的性能。

重组相对误差：

(5.2)

奇异值相对误差：

(5.3)

右奇异向量正交性：

(5.4)

接下来将从不同采样数对算法指标的影响以及过采样对最大奇异值的影响进行高位宽下的仿真，参数设置上采用8×64的信道矩阵，莱斯信道莱斯因子设置为10，CDL-B和CDL-D信道采用Matlab自带函数生成，参数为64发射天线8接收天线，其余设置保持默认。

值得注意的是，在[4]中作者估算出RSVD的重组误差界：

(5.5)

(5.6)

可见当采样数k大于等于矩阵的秩时，将是一个零矩阵，此时式5.6右边为0，即此时采样为全采样，即RSVD、ADSSVD和完整计算的SVD的结果几乎没有差别，在对数坐标下RSVD与ADSSVD在低采样数下几乎重合，为了体现RSVD和ADSSVD之间的差别，故以下仿真仅从k=1到k=7。

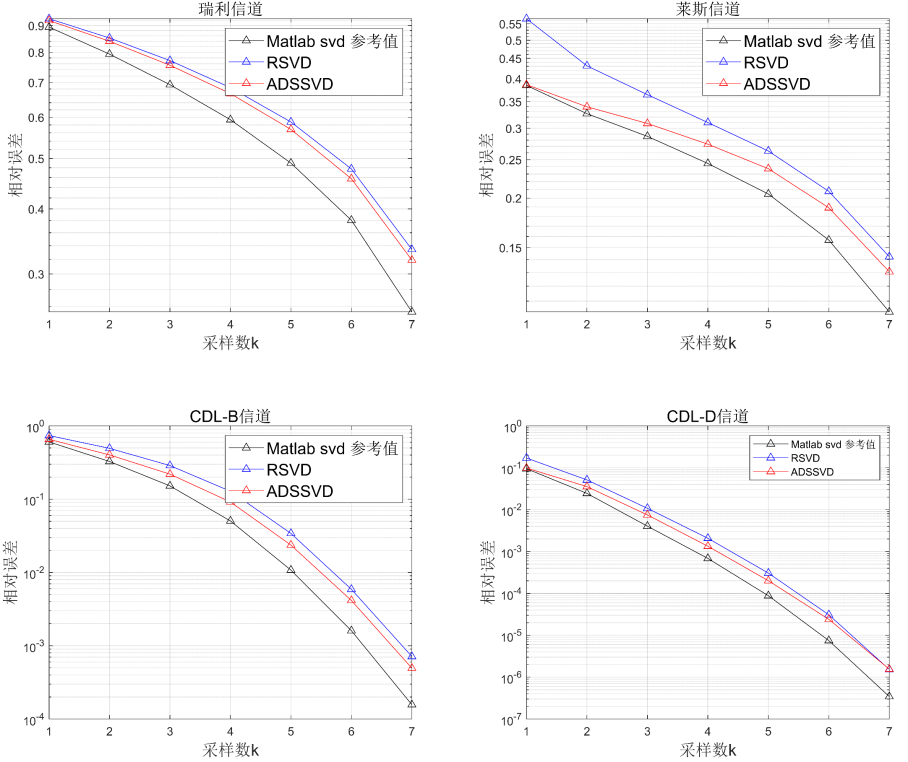


图 5.2 高位宽近似SVD矩阵重组误差与采样数的关系

从上四图来看，两种低秩近似SVD的重组误差都能随着采样数的增加而减少。对于非视距路径传输的瑞利信道和CDL-B信道，ADSSVD相较于RSVD有一定的提升；对于视距路径传输的莱斯信道和CDL-D信道，ADSSVD在采样数k=1时，其误差逼近于参考值。

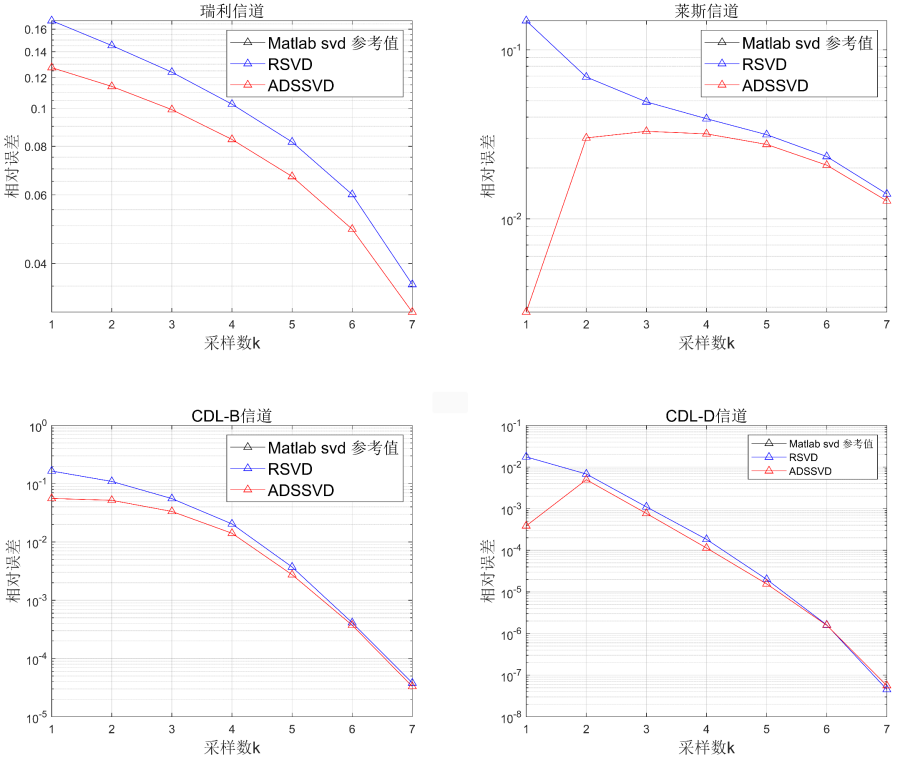


图 5.3 高位宽近似SVD奇异值误差与采样数的关系

在奇异值的相对误差中，瑞利信道和CDL-B信道下ADSSVD较RSVD都有较明显的性能上的提升，对于莱斯信道和CDL-D信道，当k=1时其第一个奇异值误差远低于RSVD，k>1之后的奇异值整体相对误差随着采样数的增加两者趋同。

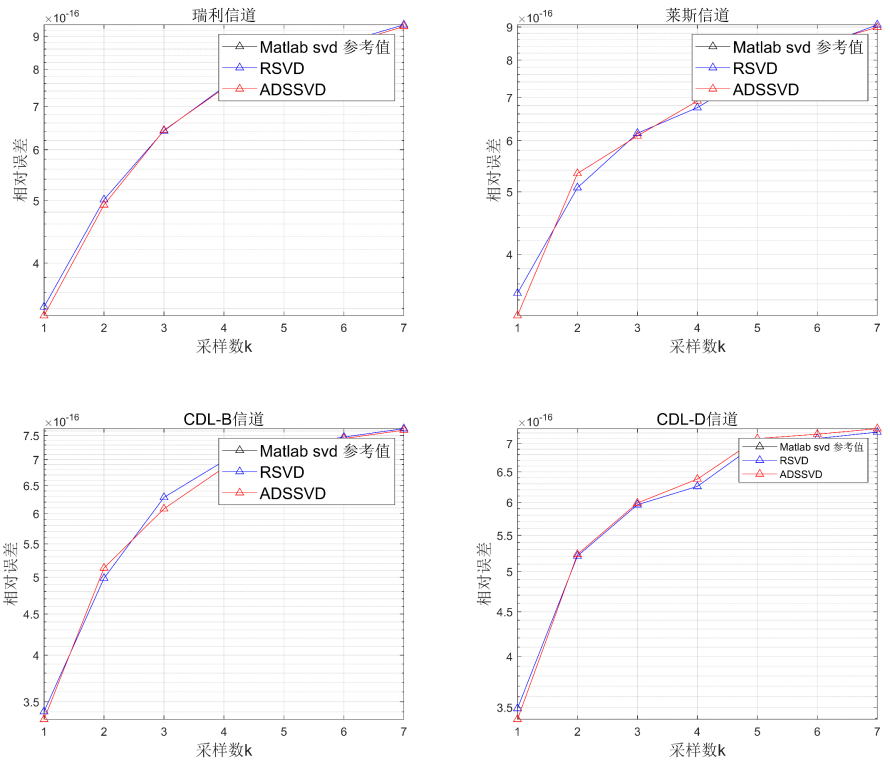


图 5.4 高位宽近似SVD右奇异向量正交误差与采样数的关系

由于两种算法的右奇异向量的正交误差都是基于低秩分解时使用的完整SVD算法，在高位宽计算环境下其右奇异向量正交性和信道种类无明显相关，其误差量级在，仅随采样数的增加导致计算复杂度上升而有略微的影响，在此不做更深的讨论。

## 5.2 近似SVD算法的子空间迭代

### 5.2.1 算法原理

RSVD和ADSSVD都可以通过子空间迭代的方法来改进其近似分解性能，其本质是通过使用下式5.7幂迭代来放大较大的奇异值：

(5.7)

由于酉矩阵的正交性，乘上采样矩阵时会在放缩后较大的奇异值方向进行投影采样，此时投影后QR分解获得的正交基张成的子空间随着迭代次数的增大愈加接近前k个奇异向量张成的子空间，使用此迭代后的正交基对原矩阵进行降维和SVD即可完成子空间迭代SVD。其显式算法见算法5.3，这种算法仅和采样矩阵相乘一次，不会影响采样矩阵的选取，且获得正交基之后的降维和SVD过程与RSVD原型算法一致，故算法描述将省去固定的步骤。

|  |
| --- |
| 算法5.3 显式子空间迭代SVD |
| 输入：信道矩阵，采样数，迭代次数   1. 构建采样矩阵（高斯矩阵/选取DFT向量组成的矩阵） 2. ， 3. 对Y进行窄QR分解：，，   输出：正交基Q |

由于显式算法中不断地矩阵相乘迭代，当矩阵规模较大时随着迭代次数的增加会出现数值溢出的现象，因此显式的算法尤其不适用于低位宽工程上的应用，接下来将给出隐式的算法5.4。隐式的算法从数学上和显式的算法是等价的，但每一次迭代过程都是酉变换，不会产生数值溢出的现象，因此在实际工程应用中建议使用隐式的迭代算法。但由于隐式迭代算法较多使用QR分解，QR分解低位宽的计算误差将使得隐式的子空间迭代算法的误差相较显式算法更高。

|  |
| --- |
| 算法5.4 隐式子空间迭代SVD |
| 输入：信道矩阵，采样数，迭代次数   1. 构建采样矩阵（高斯矩阵/选取DFT向量组成的矩阵） 2. ， 3. 对进行窄QR分解：，， 4. For 5. ，计算的QR分解： 6. ，计算的QR分解： 7. End   输出：正交基 |

子空间迭代算法能通过迭代来提高对矩阵较大奇异值及奇异向量的近似，但会将较小的奇异值的采样不断地降低，因此可以考虑低流场景下的应用。

由于子空间迭代是对采样后的矩阵进行迭代，其迭代过程会对分解产生误差传播，因此需要考虑迭代所带来的计算复杂度，下表5.2将介绍两种隐式子空间迭代过程所需浮点计算数。

表 5.2 RSVD和ADSSVD子空间迭代算法所需运算数

|  |  |  |
| --- | --- | --- |
| 算法 | 所需浮点运算数 | 64×8复数矩阵，q=50，k=1所需运算数 |
| RSVD |  | 67396 |
| ADSSVD | 获取角域信息过程：  排序查找复杂度  计算过程：  同RSVD | 获取角域信息的复杂度+67396 |

### 与先前的分析相同，ADSSVD的复杂度相较于RSVD仅差异在采样矩阵的选取，所引入的额外的复杂度仅影响其采样矩阵的构成，并不会导致额外的计算误差传播，即在计算误差受计算复杂度影响的传播角度，ADSSVD与RSVD是一致的，不会引入额外的传播误差。

### 5.2.2 高位宽仿真

此部分仿真参考指标及信道参数与5.1.3相同，采样数k设置为1，将对不同迭代次数下RSVD与ADSSVD的重组误差、右奇异向量误差、奇异值误差进行仿真。

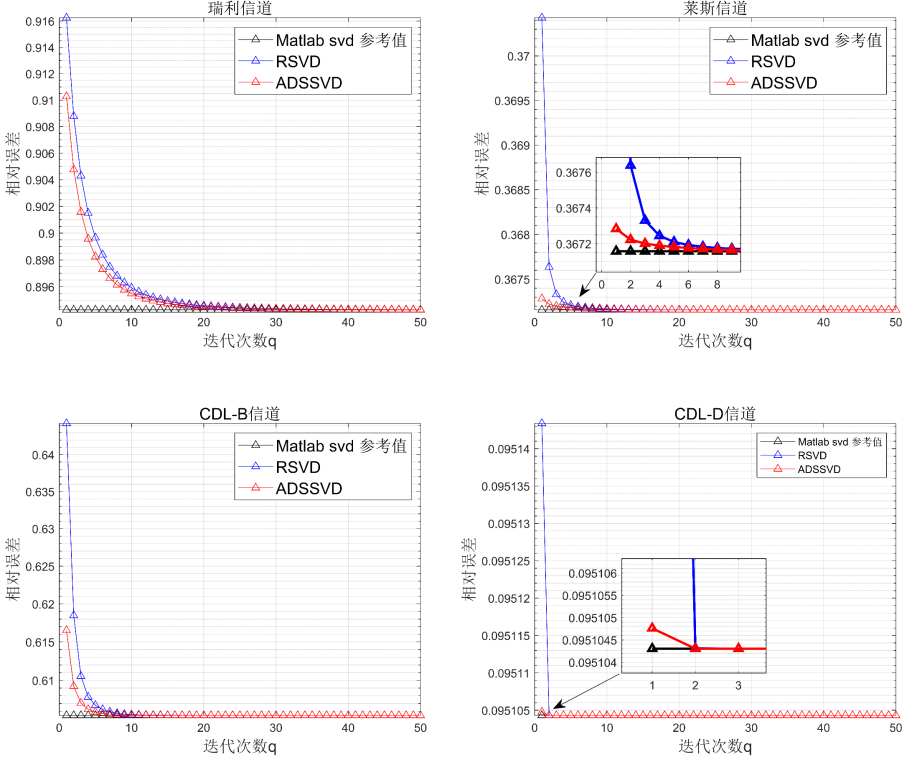


图 5.5 高位宽近似SVD子空间迭代k=1时矩阵重组误差与迭代次数的关系

从上述仿真中可以看出子空间迭代算法能通过增加迭代数来降低其重组误差，在视距路径传播的信道中，ADSSVD仅需特别少的迭代其重组误差即可与参考值相同，同时ADSSVD具备着同迭代数下其误差远低于RSVD的性能。

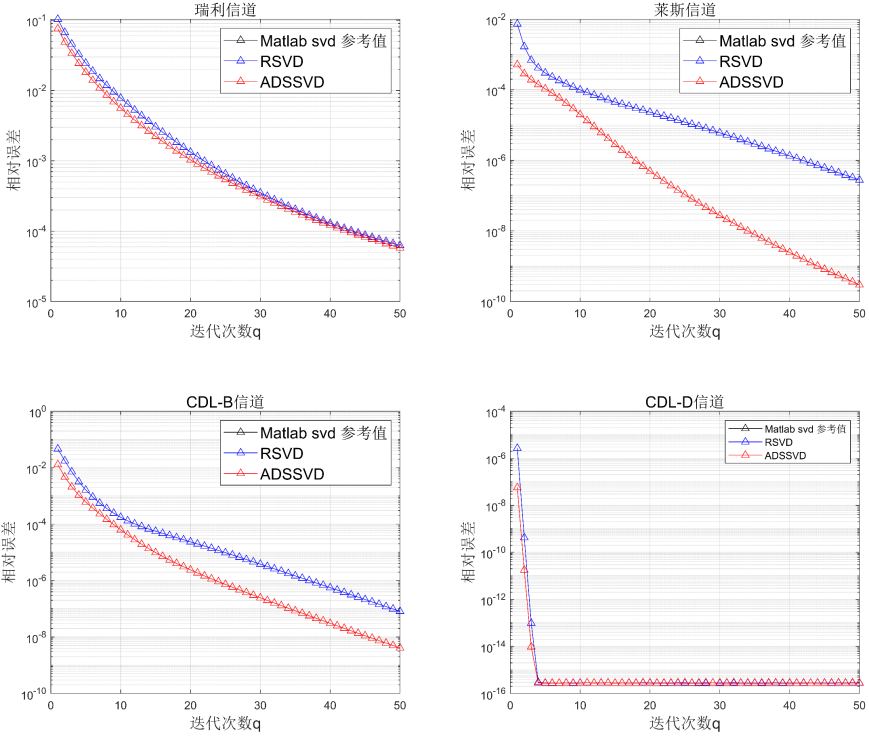


图 5.6 高位宽近似SVD子空间迭代k=1时奇异值误差与迭代次数的关系

在奇异值误差的仿真中ADSSVD子空间迭代仍具有在各信道矩阵特征的情况下优于RSVD子空间迭代的性能，两种子空间迭代能够在一定的迭代次数内将相对误差降低到一定的程度。

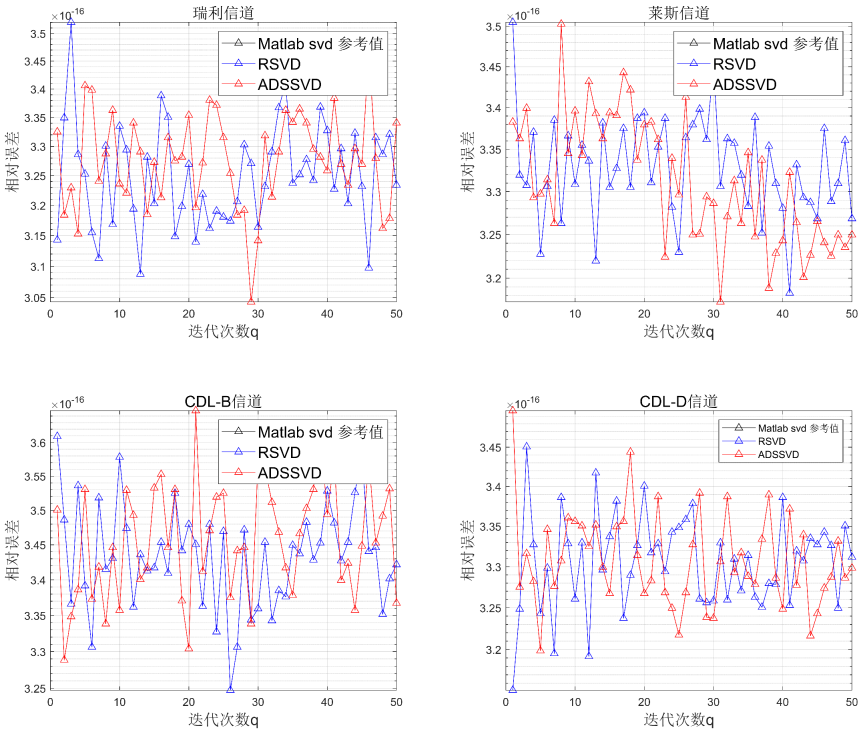


图 5.7 高位宽近似SVD子空间迭代k=1时右奇异向量正交误差与迭代次数的关系

与RSVD和ADSSVD中的分析相似，子空间迭代的右奇异向量正交误差与信道种类无明显关系，在同一采样数的情况下随着迭代的进行，其右奇异向量正交误差不会出现剧烈波动。

## 5.3 低位宽近似SVD算子的EFP实现

本节将介绍在EFP计算环境下低位宽近似SVD算子及其子空间迭代算法的仿真，包括有两种近似SVD算子及其子空间迭代算法对5.1.3节中三个指标的仿真，以及在通信系统的链路仿真。其中对矩阵的分解都采用EFP 8bit输入和EFP 8bit输出，计算部分采用EFP 16bit计算。

5.3.1 RSVD与ADSSVD的EFP实现

由于同一类路径传输的矩阵从高位宽的仿真中其结论相近，故本节中仅针对更具普遍性的瑞利信道和莱斯信道进行仿真。其中莱斯信道由于3.2.3.2节中分析中所示，将不使用Lanczos SVD的EFP算子，改用单边Jacobi SVD的EFP来实现。

|  |  |
| --- | --- |
| 图 5.8 瑞利信道近似SVD算法EFP重组误差 | 图 5.9 莱斯信道近似SVD算法EFP重组误差 |
| 图 5.10 瑞利信道近似SVD算法EFP奇异值误差 | 图 5.11 莱斯信道近似SVD算法EFP奇异值误差 |
| 图 5.12 瑞利信道近似SVD算法EFP右奇异向量正交误差 | 图 5.13 莱斯信道近似SVD算法EFP右奇异向量正交误差 |

从上图中可见在EFP16的计算环境下，RSVD和ADSSVD仍具有在高位宽计算环境下的近似特性，但可以注意到，如果仅是两种近似算法而没有进一步的优化，其无论是重组误差、奇异值误差还是右奇异向量正交误差都较完整计算的SVD性能较差，因此仅是RSVD和ADSSVD在低位宽的计算环境中是没有办法有效地将性能提升。

5.3.2近似SVD算法子空间迭代的EFP实现

在此仿真中将两种近似SVD算子的子空间迭代方法在EFP中进行仿真，计算矩阵为8×64的瑞利信道矩阵，并同时与EFP 8bit输入输出，FP 16bit计算环境的结果进行比较。由于低位宽下高采样数会导致较小奇异值采样方向上的淹没以及高迭代次数带来的计算复杂度增加导致误差传播，故此部分将展示低流场景(k=1)和较高流场景(k=4)的仿真。

|  |  |
| --- | --- |
| 图 5.14 k=1时子空间迭代重组误差 | 图 5.15 k=4时子空间迭代重组误差 |
| 图 5.16 k=1时子空间迭代奇异值误差 | 图 5.17 k=4时子空间迭代奇异值误差 |
| 图 5.18 k=1时子空间迭代右奇异向量误差 | 图 5.19 k=4时子空间迭代右奇异向量误差 |

在上图中，Lanczos SVD的重组相对误差、奇异值相对误差FP与EFP重合。可见低位宽仿真中采样数为1的角域子空间迭代算法其奇异值误差能随着迭代低于完整的Lanczos SVD计算，这是因为其能在迭代中寻找到最优的方向，同时由于其浮点运算数较低，误差传播影响较小，使得其能在低位宽的情形下优于完整的Lanczos SVD。但与之相应的，当采样数k较大时，较小的奇异值会被幂迭代淹没，导致其分解近似性能会有较大的损失。

5.3.3 低位宽近似SVD算子链路仿真的EFP实现

由于在低位宽下子空间迭代随着采样数的增加计算复杂度快速的上升会带来非常严重的误差传播，同时其较小的奇异值随着迭代幂的增大指数级缩小导致其方向会被淹没，故在低位宽的仿真中仅考虑采样数k=1的单流情况。其中链路仿真参数与基线相同，角域子空间迭代次数选取为30，为控制无关变量，莱斯信道及CDL-D信道SVD EFP实现的所有算法中的SVD分解均采用单边Jacobi SVD。此部分仿真仅做两种近似SVD算法及角域子空间迭代算法对低位宽下与未做优化的通信链路性能的对比，不涉及单边Jacobi SVD与Lanczos SVD的横向比较。

|  |  |
| --- | --- |
| 图 5.20 瑞利信道 | 图 5.21 莱斯信道 |
| 图 5.22 CDL-B信道 | 图 5.23 CDL-D信道 |

在低位宽EFP16单流情况下的链路仿真中，ADSSVD都具有良好的性能，对于不同的信道矩阵，其BLER性能指标中都能一定程度上优于完整的Lanczos计算。在瑞利信道的链路仿真中，SNR=-2的情况下ADSSVD的子空间迭代算法相较于完整的Lanczos SVD约有2dB的BLER下降。

在具有视距路径的信道环境中，ADSSVD及其子空间迭代都具有非常良好的性能。莱斯信道链路仿真中，ADSSVD以较低的复杂度逼近完整的单边Jacobi SVD的计算性能，在CDL-D信道中，SNR在-4以上时受限于仿真信道实现数，ADSSVD的BLER已经为0，子空间迭代在大于-3时BLER也为0，即在同等信噪比条件下，ADSSVD及其子空间迭代在一定程度上都优于完整计算的EFP SVD分解。

因此在实际工程应用中，低流场景的低位宽通信可以通过使用子空间迭代算法来减少计算复杂度的同时降低BLER，获得更优的性能；而在具备视距路径的通信场景中，低位宽的单流场景仍然可以仅使用ADSSVD来获得与完整SVD计算相近的性能。

1. R. M. Larsen. “Lanczos bidiagonalization with partial reorthogonalization,” *DAIMI Report Series*, vol. 27, no. 527, pp. 1-90, Oct. 1998.
2. C. H. Wu, P. Y. Tsai. “An SVD processor based on Golub–Reinsch algorithm for MIMO precoding with adjustable precision,” *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 66, no. 77, pp. 2572-2583, Mar. 2019.
3. R. P. Brent and F. T. Luk, “The solution of singular-value and symmetric eigenvalue problems on multiprocessor arrays,”*SIAM J.Sci. Statist. Comput.*, vol. 6, no. 1, pp. 69-84, Jan. 1985.
4. N. Halko, P. G. Martinsson, and J. A. Tropp, “Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions,” *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, Jan. 2011.