BLAS 总结

数据类型

标量/矩阵元素类型

S:Single Real
D:Double Real
C:Single Complex
Z:Double Complex
有些标识类型为ZD等，表示输出为Z，输入为D（表中SDS疑为逗号缺失）
I:Integer

矩阵类型

GE - GEnearl 稠密矩阵
GB - General Band 带状矩阵

SY - SYmmetric 对称矩阵
SB - Symmetric Band 对称带状矩阵
SP - Symmetric Packed 压缩存储对称矩阵

HE - HEmmitian Hemmitian矩阵，自共轭矩阵
HB - Hemmitian Band 带状Hemmitian矩阵
HP - Hemmitian Packed 压缩存储Hemmitian矩阵

TR - TRiangular 三角矩阵
TB - Triangular Band 三角带状矩阵
TP - Triangular Packed 压缩存储三角矩阵

函数命名

character + name + mod
character为上面的数据类型
name在Level 1中表示操作，包括rot向量旋转、swap向量交换、scal数乘、dot点积等；在Level 23中表示矩阵类型，具体见上
mod中表示操作的细节。包括mv矩阵乘向量、mm矩阵乘矩阵、sv解有一个未知向量的线性方程组、sm解有多个未知向量（也就是矩阵）的线性方程组

函数参数

RowMajor、ColMajor表示矩阵元素读取顺序
NoTrans、Trans、ConjTrans表示A、A^T、A^H
Upper、Lower指定使用矩阵的上三角部分还是下三角部分

Unit、NonUnit矩阵是否为对角阵
Left、Right矩阵左乘（AB）还是右乘（BA）

函数实现

下面忽略步长的参数INCX和INCY
基础操作（即认为可以用芯片的汇编直接实现）：
赋值操作iCOPY(N, x, y)和sCOPY(N, x, y)实现$y = x$
矩阵乘向量iGEMV(M, N, A, x, y)和sGEMV(M, N, A, x, y)，实现$y = Ax$
向量加法iXPY(N, x, y)和sXPY(N, x, y)，（姑且叫这个名字）实现$y = x + y$
下面尝试函数扩展，先不考虑复数（x表示i/s）

Level 1

xSWAP(N, x, y)
功能：交换xy向量
实现：$t = x, x = y, y = t$

xSCAL(N, ALPHA, x)
功能：$x = \alpha x$
实现：$xGEMV(N, 1, x, \alpha, y), x = y$
原理：$y_{N*1} = x_{N*1} * \alpha_{1*1}$

xAXPY(N, ALPHA, x, y)
功能：$y = \alpha x + y$
实现：$y = xXPY(N, xSCAL(N, \alpha, x), y)$

xDOT(N, x, y)
功能：返回值为$x^T y$
实现：$xGEMV(1, N, x, y, res)$
原理：注意到列向量和行向量在线性存储中是等价的

xNRM2(N, x)
功能：返回值为$||x||^2$
实现：$res = \sqrt{xDOT(N, x, x)}$，需要PIM支持开根号

xASUM(N, x)
功能：返回值为$\Sigma|x[i]|$
实现：难以从三个基本函数扩展

IxAMAX(N, x)
功能：返回值为|x[i]|最大的下标
实现：难以从三个基本函数扩展

xROTG(A,B,C,S)
功能：生成吉文斯旋转矩阵，即计算Cos、Sin使下式成立，r的值在a中返回

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}$$

实现：$r = \sqrt{a^2 + b^2}$需要PIM支持开根号
$c = 1, s = 0 (r = 0); c = a/r, s = b/r (r != 0)$

xROT(N, x, y, C, S)
功能：对【向量】xy进行吉文斯旋转

$$\begin{pmatrix} x[i] \\ y[i] \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x[i] \\ y[i] \end{pmatrix}$$

实现：

$$t = [x, y], r = [c, s, -s, c], xGEMV(2, 2, r, t, res), x = res[0], y = res[1]$$

xROTMG(D1, D2, A, B, PARAM)
其中PARAM为包含5个数的向量，第一个为flag，后面为H11,H12,H21,H22
约定：
SFLAG=-1.E0 SFLAG=0.E0 SFLAG=1.E0 SFLAG=-2.E0

```
  (SH11  SH12)    (1.E0  SH12)    (SH11  1.E0)    (1.E0  0.E0)
H=(          )    (          )    (          )    (          )
  (SH21  SH22),   (SH21  1.E0),   (-1.E0 SH22),   (0.E0  1.E0).
```

功能：计算PARAM，使得H = [H11, H12, H21, H22]满足下式

$$\begin{pmatrix} a \\ 0 \end{pmatrix} = \begin{pmatrix} H11 & H12 \\ H21 & H22 \end{pmatrix} \begin{pmatrix} a\sqrt{d1} \\ b\sqrt{d2} \end{pmatrix}$$

xROTM(N, x, y, PARAM)
功能：PARAM约定同上，将这个旋转操作应用在xy上
实现：

$$t = [x, y], xGEMV(2, 2, H, t, res), x = res[0], y = res[1]$$

暂：
xDOTU(N, x, y): $res = x^T y$
xDOTC(N, x, y): $res = x^H y$
xxDOT

Level 2

存储模式

B->带状存储，KL, KU分别表示下三角部分和上三角部分的"带数"，把每一"斜"作为一行存储，存储总空间为(KL + KU + 1) * n。储存地址的转换A[i, j]=Band[i - j + KU, j]

P->压缩存储，列优先。上三角则$a_{11}=A[1], a_{12}=A[2], a_{22}=A[3]$，依此类推；下三角则$a_{11}=A[1], a_{21}=A[2], a_{31}=A[3]$

UPLO为'u'/'l'表示为上三角还是下三角，另一半不会被引用。
DIAG为'u'(unity)表示A的对角线上的元素都是1，原来A的对角线元素则不会被引用。

以下为普通矩阵乘向量

xGEMV(TRANS, M, N, ALPHA, A, x, BETA, y)
功能：$y=\alpha Ax+\beta y$
实现：
$xGEMV(M,N,A,x,a), xSCAL(N,y,BETA), xAXPY(N,ALPHA,a,y)$
原理：$a=Ax, y=\beta y, y=\alpha a+y$

xGBMV(TRANS, M, N, KL, KU, ALPHA, A, x, BETA, y)
功能：$y=\alpha Ax+\beta y$
实现：同上，带状储存

以下为对称矩阵乘向量

xSYMV(UPLO, N, ALPHA, A, x, BETA, y)
功能：$y=\alpha Ax+\beta y$
实现：同上，UPLO=u/l表示存储的是上三角部分or下三角部分

xSBMV(UPLO, N, K, ALPHA, A, x, BETA, y)
功能：$y=\alpha Ax+\beta y$
实现：同上，在对称矩阵的基础上采用带状矩阵存储

xSPMV(UPLO, N, ALPHA, AP, x, BETA, y)
功能：$y=\alpha Ax+\beta y$
实现：同上，压缩存储。

以下为上/下三角矩阵乘向量

xTRMV(UPLO, TRANS, DIAG, N, A, x)
功能：$x=Ax/x=A^Tx$

xTBMV(UPLO, TRANS, DIAG, N, K, A, x)
功能：$x=Ax/x=A^Tx$
实现：同上，带状存储。

xTPMV(UPLO, TRANS, DIAG, N, AP, x)
功能：$x=Ax/x=A^Tx$
实现：同上，压缩存储。

以下为解上三角or下三角矩阵方程

xTRSV(UPLO, TRANS, DIAG, N, A, x)
功能：求x使Ax=b，b的值在x中传入
实现：
原理：$x = A^{-1}x$

xTBSV(UPLO, TRANS, DIAG, N, K, A, x)

xTPSV(UPLO, TRANS, DIAG, N, AP, x)

以下为xy^T

xGER(M, N, ALPHA, x, y, A)
功能：$A = \alpha xy^T + A$
实现：xAXPY(M, ALPHA * y[i], x, A + n * i)
原理：y是把x向量的倍数放到一个矩阵里，我们只需要让A的每一列加上对应的x的倍数即可

xSYR(UPLO, N, ALPHA, x, A)
功能：$A = \alpha xx^T + A$，A为对称矩阵

xSPR(UPLO, N, ALPHA, x, AP)
功能：同上，AP为压缩存储

xSYR2(UPLO, N, ALPHA, x, y, A)
功能：$A = \alpha(xy^T + yx^T) + A$
实现：xAXPY(N, ALPHA * y[i], x, A + n * i),xAXPY(N, ALPHA * x[i], y, A + n * i)

xSPR2(UPLO, N, ALPHA, x, y, AP)
功能：同上，AP为压缩存储

Level2 总结

三件事：矩阵乘向量、解非齐次线性方程组、算xy^T
其中每个都有不同的存储模式，如对称矩阵、上三角、下三角、带状存储、压缩存储

暂：
xHEMV,xHBMV,xHPMV

xGERU,xGERC
xHER,xHPR
xHER2,xHPR2

参考

https://www.netlib.org/lapack/explore-html/d4/d28/group__blas1__grp.html
https://blog.csdn.net/weixin_43800762/article/details/87811697
https://blog.csdn.net/dingding_tao/article/details/81087823