THE CUBLAS LIBRARY

Will Landau, Prof. Jarad Niemi

WHAT IS CULA

CULA is another set of gpu-accelerated linear algebra libraries. Essentially, it's the CUDA equivalent of LAPACK.

Its routines include:

- Utilities for initializing matrices, copying them, scaling them, etc.
- Matrix multiplication
- Factorizations: LU, QR, RQ, QL, SVD, and Cholesky
- Solving systems of linear equations (which gives you matrix inversion)
- Solving least squares problems
- Eigenproblem solvers

CULA INTERFACES

- **Standard**
  - Each CULA function micromanages GPU resources so the user doesn't have to.
  - Matrix arguments of CULA functions are pointers to CPU memory.
  - Activate by including the header, "cula_lapack.h".
- **Device**:
  - The user micromanages GPU resources as with CUBLAS.
  - Matrix arguments of CULA functions are pointers to GPU memory.
  - Activate by including the header, "cula_lapack_device.h"

Note: although convenient, the standard interface is often slow because every single function writes back and forth between the CPU and the GPU.

COMPILING WITH CULA

1. Include the header file in your source:
   "cula_lapack.h" for the standard interface,
   "cula_lapack_device.h" for the device interface.

2. Enter in the command line:

```
nvcc -lcula_core -lcula_lapack
   your_source.cu -o your_binary
```

WHAT'S IN CULA?

FIRST: NAMING CONVENTIONS

Prefix	Data Type	CULA Standard Interface Type	CULA Device Interface Type
`S`	Single Precision Real	`culaFloat`	`culaDeviceFloat`
`C`	Single Precision Complex	`culaFloatComplex`	`culaDeviceFloatComplex`
`D`	Double Precision Real	`culaDouble`	`culaDeviceDouble`
`Z`	Double Precision Complex	`culaDoubleComplex`	`culaDeviceDoubleComplex`

Abbreviation	Matrix Type
bd	Bidiagonal
ge	General
gg	General matrices, generalized problem
he	Hermitian Symmetric
or	(Real) Orthogonal
sb	Symmetric Band
sy	Symmetric
tr	Triangular
un	(Complex) Unitary

Abbreviation	Computation Routine
`trf`	Compute a triangular factorization
`sv`	Factor the matrix and solve a system of equations
`qrf`	Compute a QR factorization without pivoting
`svd`	Compute the singular value decomposition
`ls`	Solve over- or under-determined linear system

AUXILIARY FUNCTIONS

Matrix Type	Operation	S	C	D	Z
General	Copy from one Matrix into another	SLACPY	CLACPY	DLACPY	ZLACPY
	Convert a matrix's precision	SLAG2D	CLAG2Z	DLAG2S	ZLAG2D
	Apply a block reflector to a matrix	SLARFB	CLARFB	DLARFB	ZLARFB
	Generate an elementary reflector	SLARFG	CLARFG	DLARFG	ZLARFG
	Generate a vector of plane rotations	SLARGV	CLARGV	DLARGV	ZLARGV
	Apply a vector of plane rotations	SLARTV	CLARTV	DLARTV	ZLARTV
	Multiple a matrix by a scalar	SLASCL	CLASCL	DLASCL	ZLASCL
	Initialize a matrix	SLASET	CLASET	DLASET	ZLASET
	Apply a sequence of plane rotations	SLASR	CLASR	DLASR	ZLASR
Symmetric	Apply a vector of plane rotations	SLAR2V	CLAR2V	DLAR2V	ZLAR2V
Triangular	Triangular precision conversion	SLAT2D	CLAT2Z	DLAT2S	DLAT2Z

MATRIX MULTIPLICATIONS

Matrix Type	Operation	S	C	D	Z
General	Matrix-matrix multiply	SGEMM	CGEMM	DGEMM	ZGEMM
	Matrix-vector multiply	SGEMV	CGEMV	DGEMV	ZGEMV
Triangular	Triangular matrix-matrix multiply	STRMM	CTRMM	DTRMM	ZTRMM
	Triangular matrix solve	STRSM	CTRSM	DTRSM	ZTRSM
Symmetric	Symmetric matrix-matrix multiply	SSYMM	CSYMM	DSYMM	ZSYMM
	Symmetric rank 2k update	SSYR2K	CSYR2K	DSYR2K	ZSYR2K
	Symmetric rank k update	SSYRK	CSYRK	DSYRK	ZSYRK
Hermitian	Hermitian matrix-matrix multiply		CHEMM		ZHEMM
	Hermitian rank 2k update		CHER2K		ZHER2K
	Hermitian rank k update		CHERK		ZHERK

FACTORIZATIONS

Matrix Type	Operation	S	C	D	Z
General	Factorize and solve	SGESV	CGESV	DGESV	ZGESV
	Factorize and solve with iterative refinement			DSGESV	ZCGESV
	LU factorization	SGETRF	CGETRF	DGETRF	ZGETRF
	Solve using LU factorization	SGETRS	CGETRS	DGETRS	ZGETRS
	Invert using LU factorization	SGETRI	CGETRI	DGETRI	ZGETRI
Positive Definite	Factorize and solve	SPOSV	CPOSV	DPOSV	ZPOSV
	Cholesky factorization	SPOTRF	CPOTRF	DPOTRF	ZPOTRF
Triangular	Invert triangular matrix	STRTRI	CTRTRI	DTRTRI	ZTRTRI
	Solve triangular system	STRTRS	CTRTRS	DTRTRS	ZTRTRS
Banded	LU factorization	SGBTRF	CGBTRF	DGBTRF	ZGBTRF
Pos Def Banded	Cholesky factorization	SPBTRF	CPBTRF	DPBTRF	ZPBTRF

ORTHOGONAL FACTORIZATIONS

Matrix Type	Operation	S	C	D	Z
General	QR factorization	SGEQRF	CGEQRF	DGEQRF	ZGEQRF
	QR factorize and solve	SGEQRS	CGEQRS	DGEQRS	ZGEQRS
	Generate Q from QR factorization	SORGQR	CUNGQR	DORGQR	ZUNGQR
	Multiply matrix by Q from QR factorization	SORMQR	CUNMQR	DORMQR	ZUNMQR
General	LQ factorization	SGELQF	CGELQF	DGELQF	ZGELQF
	Generate Q from LQ factorization	SORGLQ	CUNGLQ	DORGLQ	ZUNGLQ
	Multiply matrix by Q from LQ factorization	SORMLQ	CUNMLQ	DORMLQ	ZUNMLQ
General	RQ factorization	SGERQF	CGERQF	DGERQF	ZGERQF
	Multiply matrix by Q from RQ factorization	SORMRQ	CUNMRQ	DORMRQ	ZUNMRQ
General	QL factorization	SGEQLF	CGEQLF	DGEQLF	ZGEQLF
	Generate Q from QL factorization	SORGQL	CUNGQL	DORGQL	ZUNGQL
	Multiply matrix by Q from QL factorization	SORMQL	CUNMQL	DORMQL	ZUNMQL

SINGULAR VALUE DECOMPOSITIONS

Singular Value Decomposition

CULA contains the following LAPACK function equivalents from the Singular Value Decomposition family of computational routines:

Matrix Type	Operation	S	C	D	Z
General	General Singular Value Decomposition	SGESVD	CGESVD	DGESVD	ZGESVD
	Bidiagonal reduction	SGEBRD	CGEBRD	DGEBRD	ZGEBRD
	Generate Q from bidiagonal reduction	SORGBR	CUNGBR	DORGBR	ZUNGBR
Bidiagonal	Find singular values and vectors	SBDSQR	CBDSQR	DBDSQR	ZBDSQR

LEAST SQUARES

Matrix Type	Operation	S	C	D	Z
General	General least squares solve	SGELS	CGELS	DGELS	ZGELS
	Equality-constrained least squares	SGGLSE	CGGLSE	DGGLSE	ZGGLSE

SYMMETRIC EIGENPROBLEMS

Matrix Type	Operation	S	C	D	Z
Symmetric	Symmetric Eigenproblem solver	SSYEV		DSYEV	
	Symmetric Eigenproblem solver (expert)	SSYEVX		DSYEVX	
	Symmetric band reduction	SSYRDB		DSYRDB	
Tridiagonal	Find eigenvalues using bisection	SSTEBZ		DSTEBZ	
	Find eigenvalues using QR/QL iteration	SSTEQR	CSTEQR	DSYRDB	ZSTEQR

NON-SYMMETRIC EIGENPROBLEMS

Matrix Type	Operation	S	C	D	Z
General	General Eigenproblem solver	SGEEV	CGEEV	DGEEV	ZGEEV
	Hessenberg reduction	SGEHRD	CGEHRD	DGEHRD	ZGEHRD
	Generate Q from Hessenberg reduction	SORGHR	CUNGHR	DORGHR	ZUNGHR

GPU SERIES MATERIALS

These slides, a tentative syllabus for the whole series, and code are available at:

https://github.com/wlandau/gpu.

After logging into you home directory on impact1, type:

```
git clone https://github.com/wlandau/gpu
```

into the command line to download all the materials.

REFERENCES

CUDA Toolkit 4.2 CUBLAS Library. February 2012.
http://developer.download.nvidia.com/compute/DevZone/docs/html/CUDALibraries/doc/CUBLAS_Library.pdf