

GPU Computing with MATLAB

Research Computing Day 2017

Thomas Anthony

Scientist | Director – Big Data Research and Analytics Lab

Research Computing, UAB IT | Dept. Of Electrical and Computer Engineering

Agenda

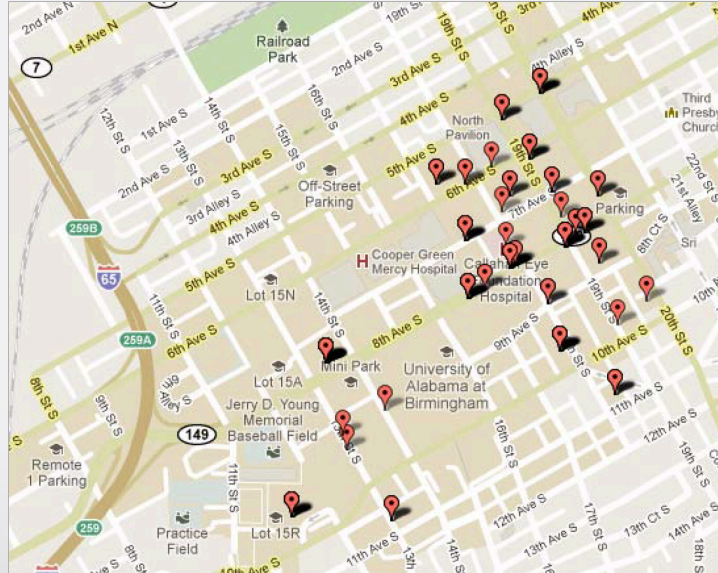
- MATLAB on The CPU (PCT & DCS)
- Why compute on a GPU?
- MATLAB and GPU Computing
- Some Benchmarks
- Demo (If time permits)

Introduction

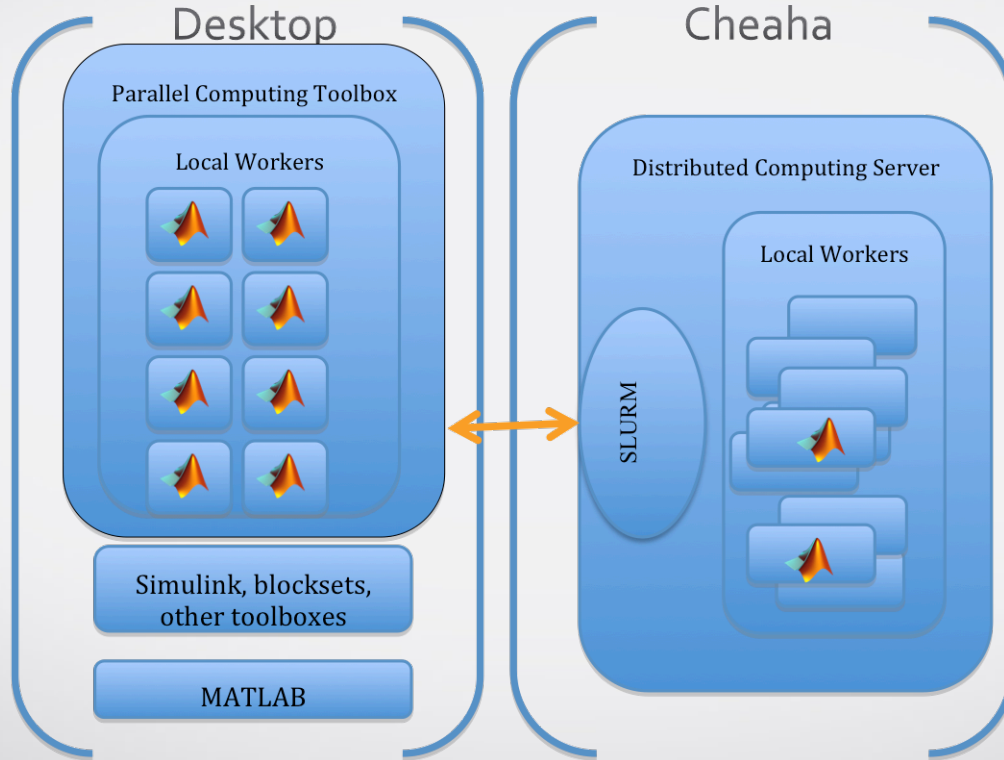
- **MATLAB (matrix laboratory)** is a high-level technical computing language with an interactive numerical computing environment. Developed by Mathworks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, and Fortran.
- Built around the MATLAB scripting language
- Command window used as an interactive shell for executing MATLAB code
- Over 2 million users worldwide (2017)
- Widely used in a number of fields

MATLAB @ UAB

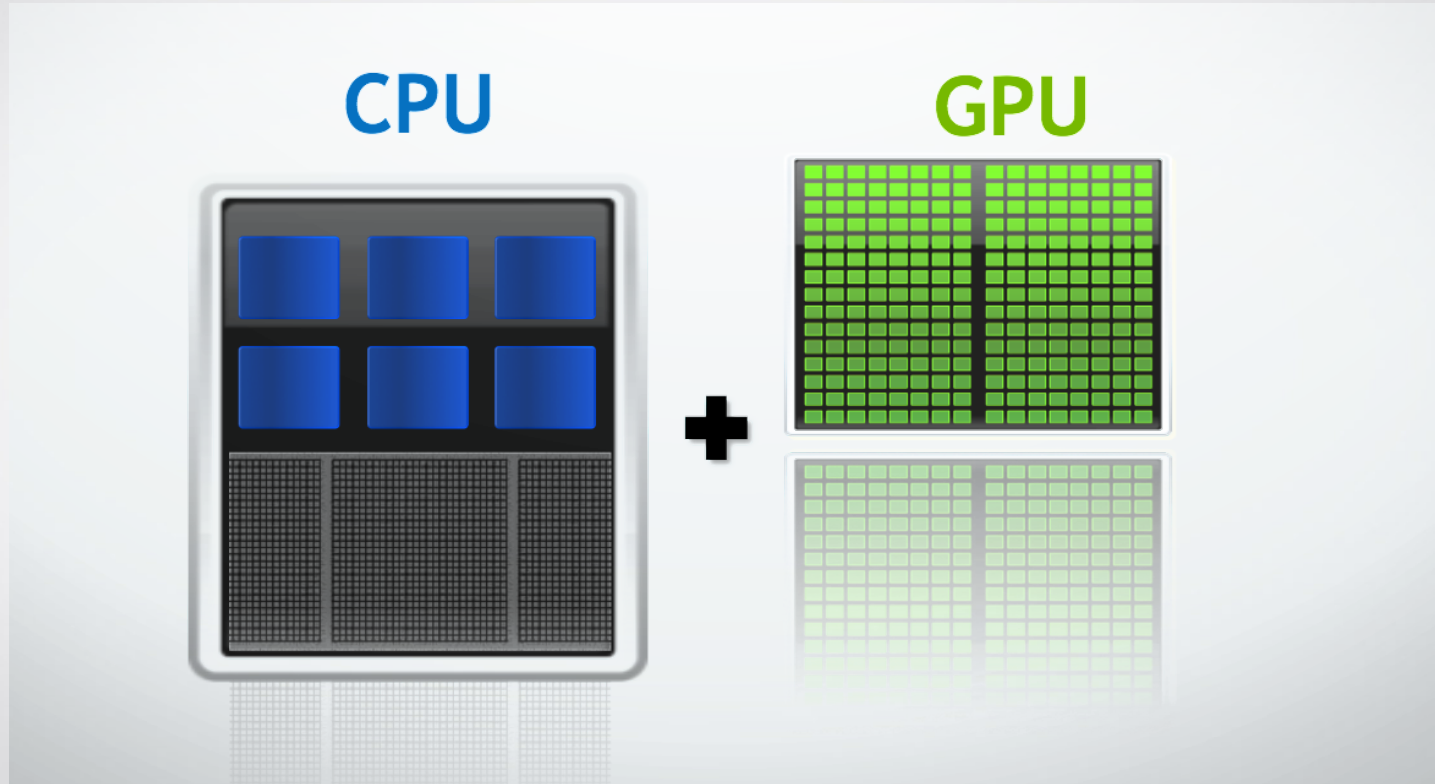
- Over 2000 individual users on the UAB TAH license.



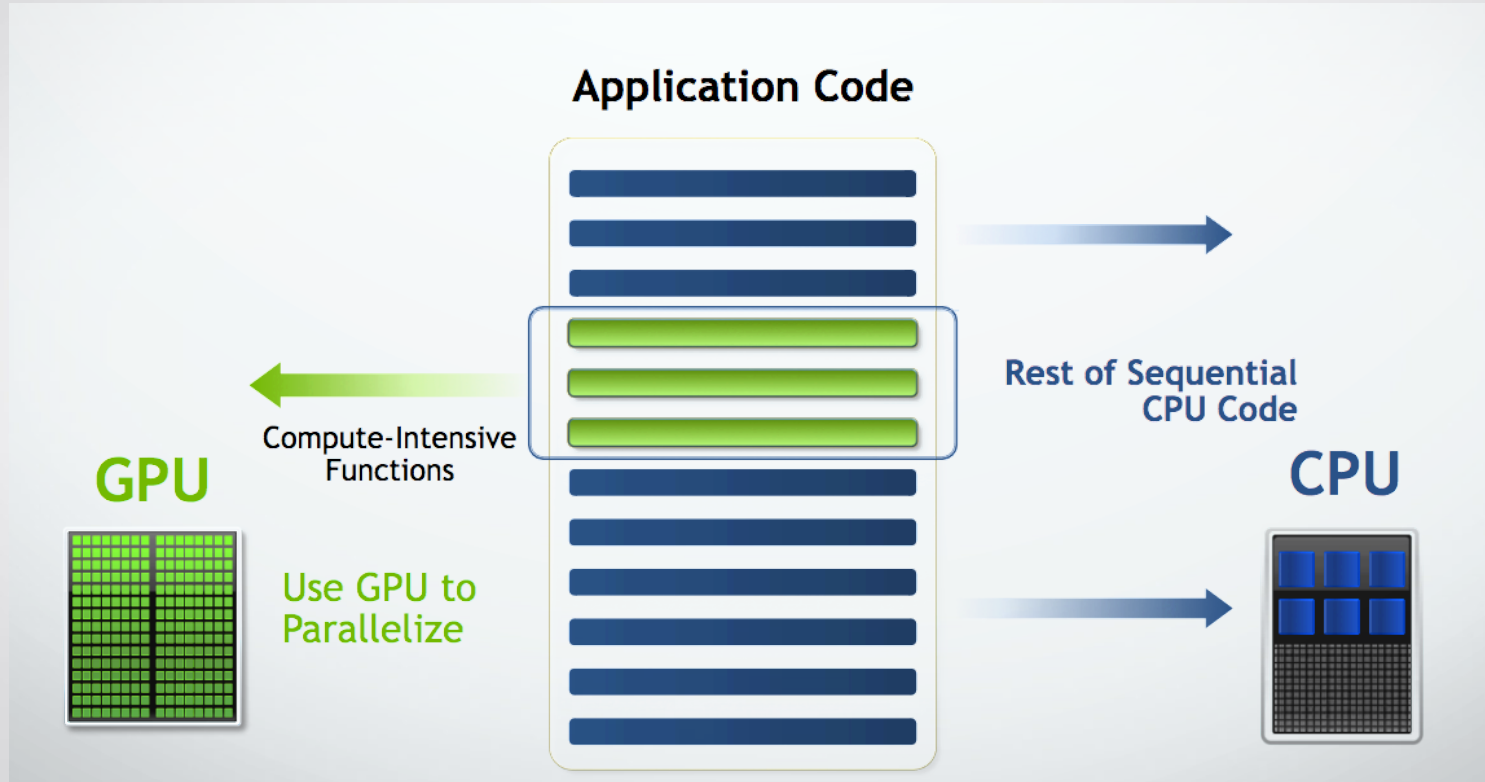
MATLAB on the CPU (PCT & DCS)



Why compute on a GPU?



Small Changes -> Big Speed UP



Advantages

- Faster Performance on Scientific Applications
 - Applications have seen between 3x -20x performance improvement
- Cost of computing is reduced
 - Perf/Watt especially if you scale up computing is prohibitive with the CPU

MATLAB and GPU Computing

- MATLAB makes GPU computing easy
 - Many functions support gpuArrays
- A large number of built-in functions are already optimized for the NVIDIA GPU
 - ~ 275 (at last count)
- Single precision/Double precision support
- Works well with PCT and the DCS

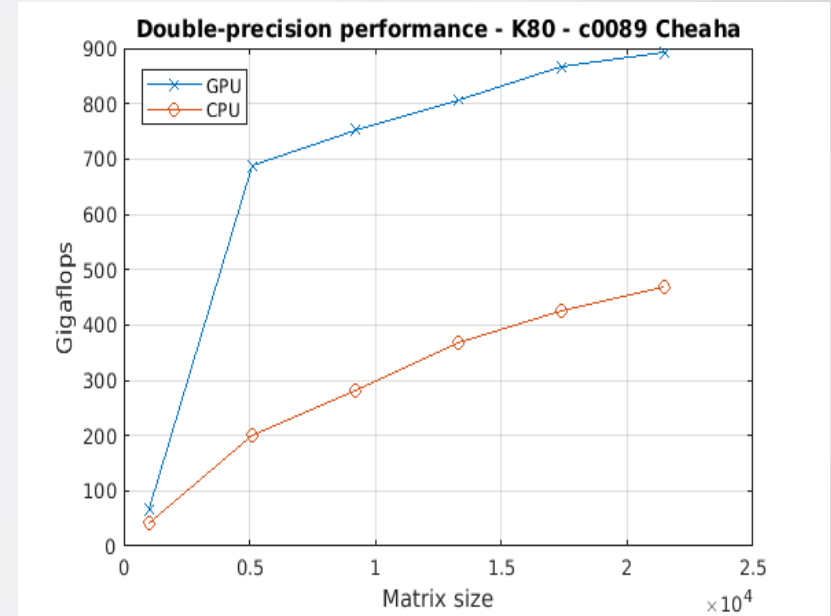
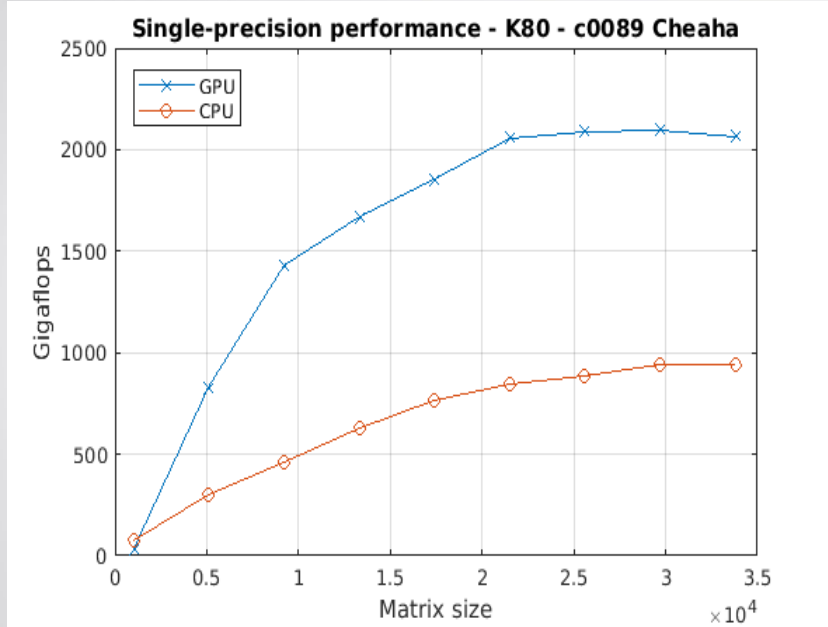
MATLAB GPU supported functions

abs	complan	flip	isnan	pcg	sph2cart
acos	complex	fliplr	isnumeric	perms	sprand
acosh	cond	flipud	isreal	permute	sprandn
acosh	conj	floor	isrow	pinv	sprandsym
acot	conv	fprinf	issorted	planerot	spconvert
acoth	conv2	full	issparse	plot (and related)	sph2cart
acsc	convn	gamma	issymeric	plus	sprand
acscd	corrcoef	gammaln	istrl	pol2cart	sprandn
acsch	cos	gammaln	istriu	poly	sprandsym
accumarray	cosd	gamma	isvector	polyarea	sprintf
all	cosh	gather	kron	polyder	sqrt
and	cot	ge	ldivide	polyfit	squeeze
angle	cotd	gmes	le	polyint	std
any	coth	gradient	legendre	polyval	sub2ind
arrayfun	cov	gt	length	polyvalm	subsasgn
asec	cross	hankel	log	pow2	subsindex
asecd	csc	head	log10	power	subspace
asech	cscd	histcounts	loglp	prod	subsubref
asin	csch	horzcat	log2	psi	sum
asind	ctranspose	hsv2rgb	logical	qmr	superiortfloat
asinh	cummax	hypot	lsqr	qr	svd
assert	cummin	idivide	lt	rad2deg	svds
atan	cumprod	ifft	lu	rand	swapbytes
atan2	cumsun	ifft2	mat2str	randi	tail
atan2d	deg2rad	ifftn	max	randn	tan
atand	del2	ifftshift	median	randperm	tand
atanh	det	imag	mean	rank	tanh
bandwidth	detrend	ind2sub	meshgrid	rdivide	times
besselj	diag	Inf	min	real	toeplitz
bessely	diff	inpolygon	minus	reallog	trace
beta	discretize	int16	mdivide	realpow	transpose
betainc	disp	int2str	mod	realsqrt	trapz
betaincinv	display	int32	mode	rectint	tril
betaIn	dot	int64	movmean	rem	triu
bicg	double	int8	movstd	repelem	true
bicgstab	eig	interp1	movsum	repmat	typecast
bitand	eps	interp2	movvar	reshape	uint16
bitcmp	eq	interp3	mpower	rgb2hsv	uint32
bitget	erf	interp4	mdivide	roots	uint64
bitor	erfc	intersect	mtimes	rot90	uint8
bitset	erfcinv	inv	NaN	round	uminus
bitshift	erfcx	ipermute	ndgrid	sec	union
bitxor	erfinv	isaUnderlying	ndims	secd	unique
blkdiag	exp	isbanded	ne	sech	unique_tol
bounds	expint	iscolumn	nextpow2	setdiff	unwrap
bsxfun	expm	isdiag	nnz	setxor	uplus
cart2pol	expml	isempty	nonzeros	shiftdim	vander
cart2sph	eye	isequal	norm	sign	var
cast	factorial	isequaln	normest	sin	vertcat
cat	false	isfinite	not	sind	xor
cd92rdf	fft	isfloat	nthroot	sinh	zeros
ceil	fft2	ishermitian	null	size	
cgs	fftn	isinf	num2str	sort	
chol	fftshift	isinteger	numel	sortrows	
circshift	filter	islogical	ones	svds	
classUnderlying	filter2	ismatrix	or	spconvert	
colon	find	ismember	orth	spdiags	
	fix	ismembertol	pagefun		

MATLAB and GPU Computing

```
>> A = rand (1000,1000);  
>> G = gpuArray(A);    % push function to GPU  
...  
>> F_cpu = fft (A);    % run fft on CPU  
>> F_gpu = fft(G);     % run fft on GPU  
...  
>> z = gather(F_gpu); % Bring output back from GPU
```

Some Benchmarks



CPU – INTEL E5-2680 v3 @ 2.5 GZ – Total 24 cores

GPU – NVIDIA – K80

Demo

- Run a simple piece of code on CPU & time it
- Convert it to run on the GPU & time it
- Discuss Speed-up, pitfalls, limitations
- PCT/DCS+GPU
- MATLAB GPU coder
- Conclusion

Special Thanks!

- Mathworks
- Dell
- Nvidia
- UAB IT
- Research Computing Team