Spectra denoising: using graphic cards ?

G. LAURENT[1], W. WOELFFEL[2], V. BARRET-VIVIN[1], E. GOUILLART[2], C. BONHOMME[1]

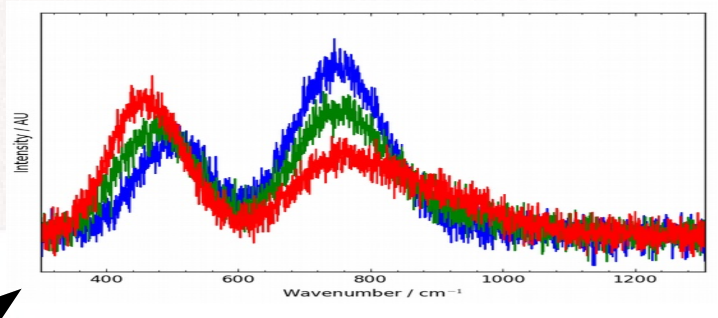
CNRS Studies Engineer & PhD student
guillaume.laurent@upmc.fr

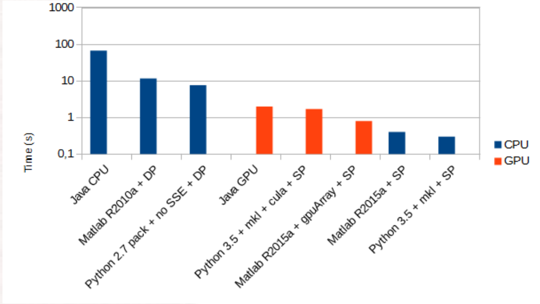
1 Sorbonne Universités, Collège de France, CNRS, Laboratoire de Chimie de la Matière Condensée de Paris, 11 place Marcelin Berthelot, 75005 Paris, France

2 Saint-Gobain, CNRS, Surface du Verre et Interfaces, 39 quai Lucien Lefranc, 93300 Aubervilliers, France

Outline

- Context
- Spectra denoising
- Java CPU vs GPU
- Software improvements
- Conclusion and future work



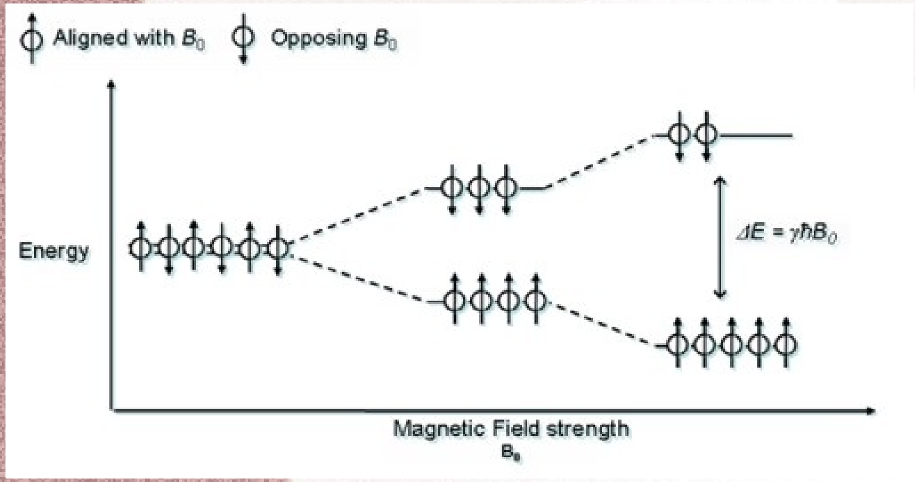


Context 1/2 - Sensitivity

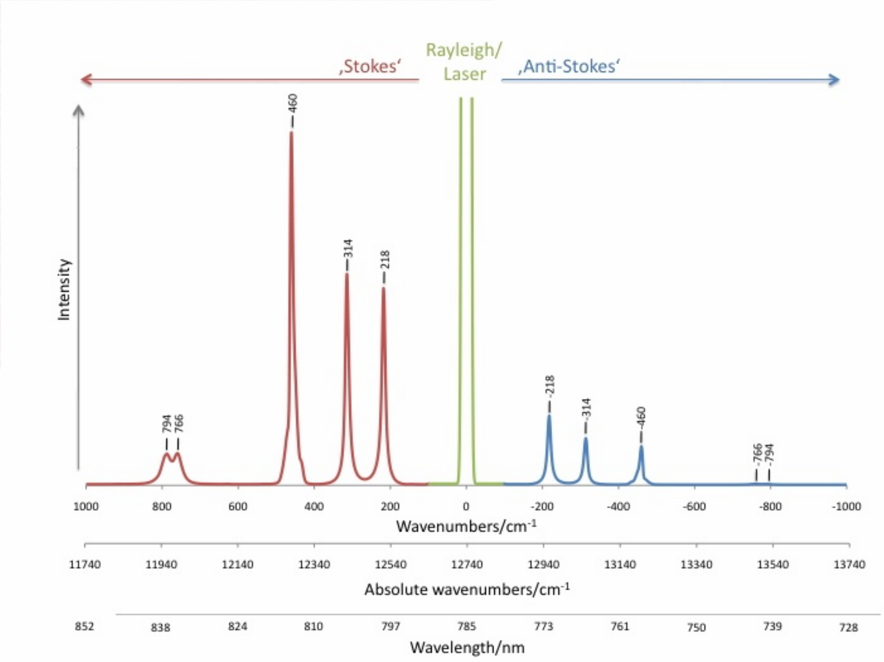
NMR spectroscopy

Raman spectroscopy

hackaday.io



SBD group - York University



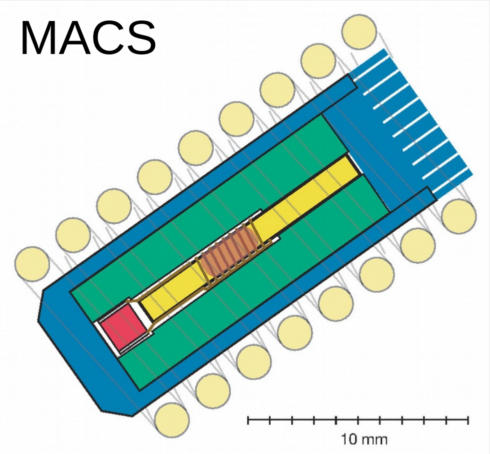
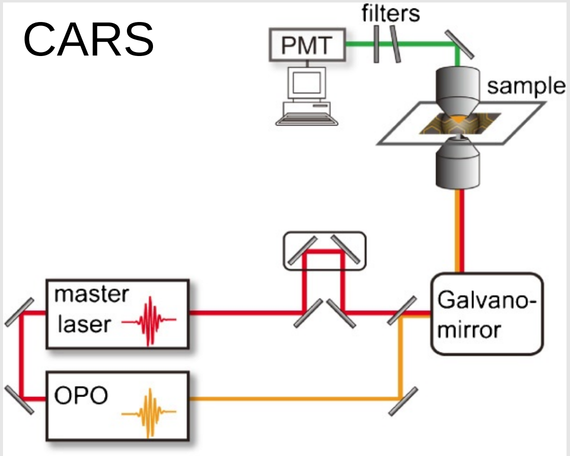
raman.de

One nucleus over 10^5 is visible

One photon over 10^6 is visible

[1] M. H. Levitt, Spin Dynamics: Basics of Nuclear Magnetic Resonance, Second edition, p. 268. John Wiley & Sons Ltd, 2008.
[2] R. Gautam *et al*, Curr. Sci., vol. 108, no. 3, pp. 341–356, Feb. 2015.

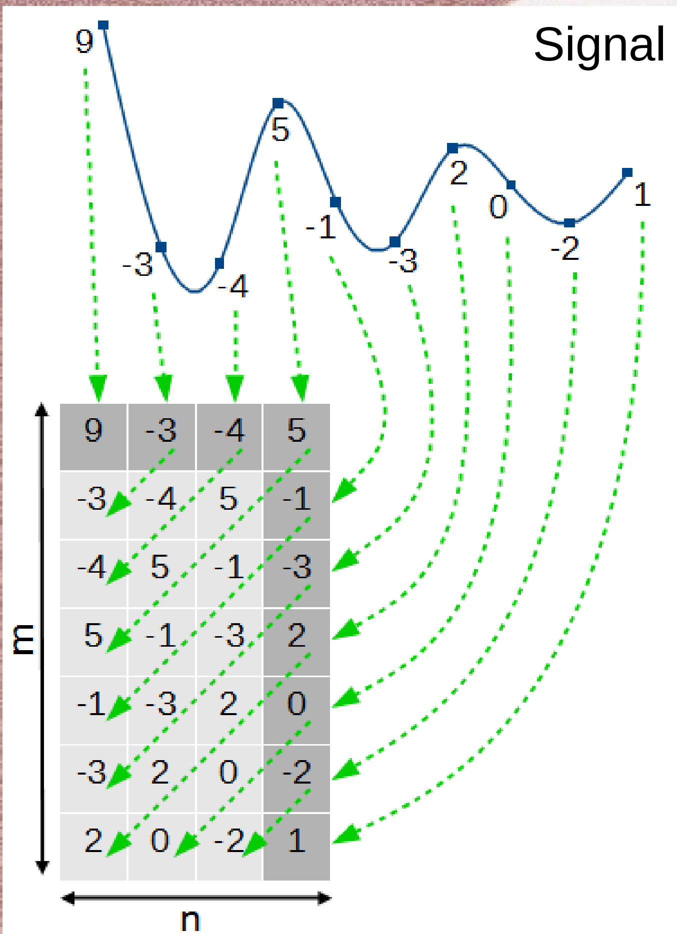
Context 2/2 - Improvements

	NMR spectroscopy	Raman spectroscopy
Instrumentation	Magic Angle Spinning High power decoupling Hyper-polarisation (DNP,…) Micro-coils  Sakellariou *et al*, 2007	Tip-Enhanced Raman Spectroscopy Picoseconds lasers Non-linear light sources  Kano *et al*, 2014
Acquisition	CPMG Non-uniform sampling	Ultra-fast Raman spectral imaging
Processing	Smoothing Spectra denoising	

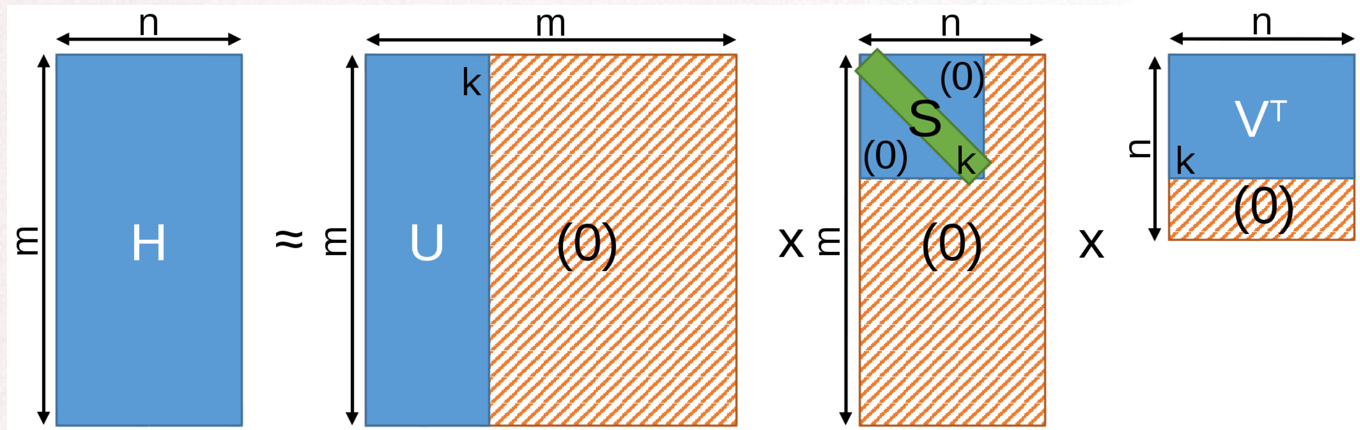
Sensitivity increased but still low S/N
Broad peaks for distributed environments

[1] D. Sakellariou and J.-F. Jacquinot, WO/2007/020537, 23-Feb-2007.
[2] H. Kano *et al*, OPT REV, vol. 21, no. 6, pp. 752–761, Nov. 2014.

Spectra denoising 1/4
Singular Value Decomposition



Hankel Matrix



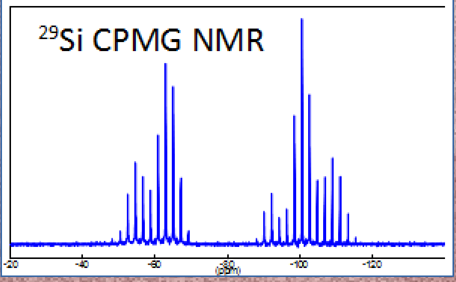
Low rank matrix approximation

3 parameters:

- Number of rows m
- Number of columns n
- Number of singular values k

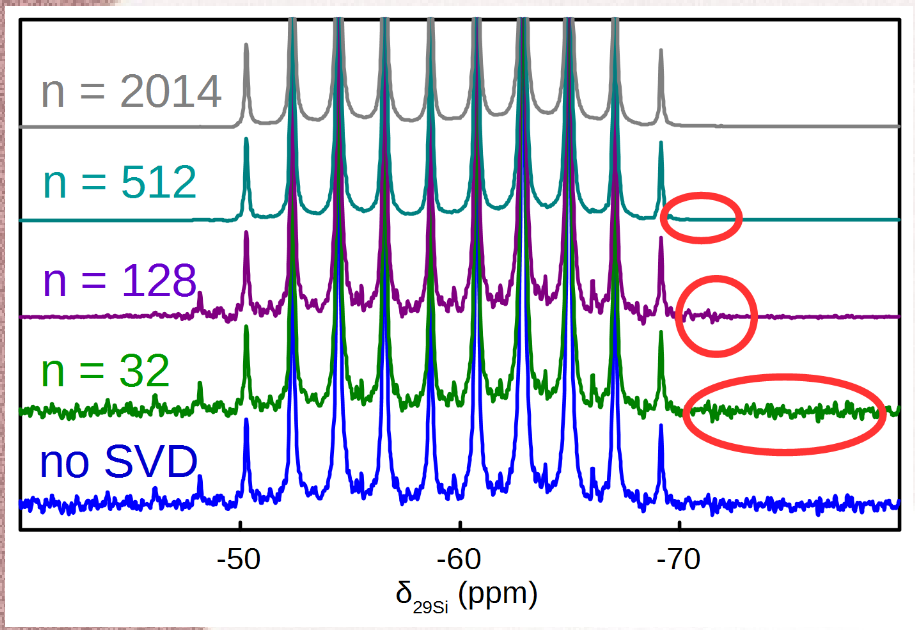
SVD is also used for data mining (PCA) or image compression

[1] M. A. Arbib and E. G. Manes, Journal of Computer and System Sciences, vol. 20, no. 3, pp. 330–378, Jun. 1980.
[2] D. W. Tufts *et al*, Proceedings of the IEEE, vol. 70, no. 6, pp. 684–685, Jun. 1982.
[3] J. A. Cadzow, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 36, no. 1, pp. 49–62, Jan. 1988.

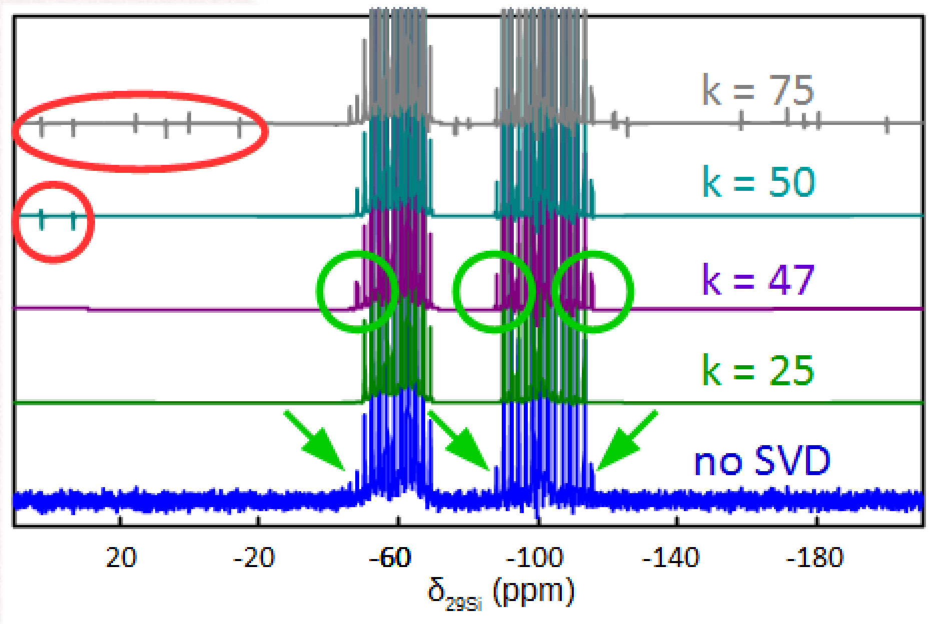


Spectra denoising 2/4
Matrix parameters

Number of columns n



Number of singular values k



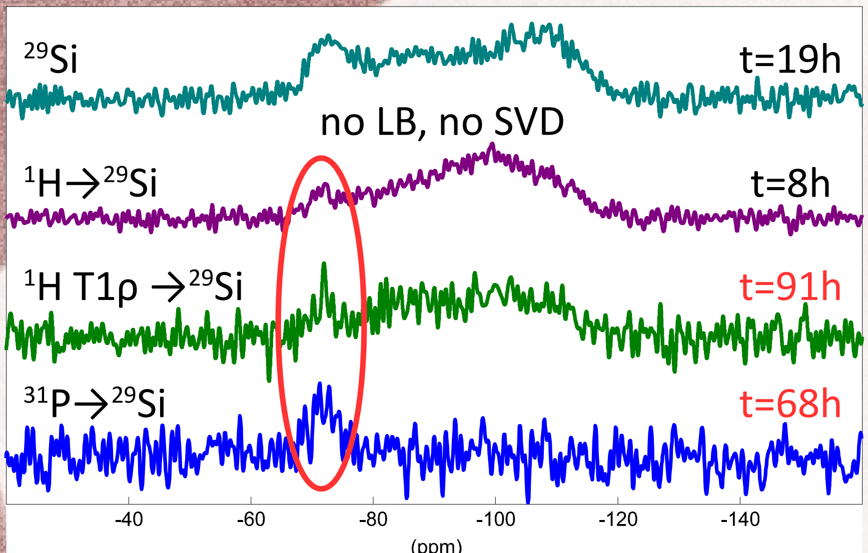
TEOS:MTEOS 1:1 – 4028 points

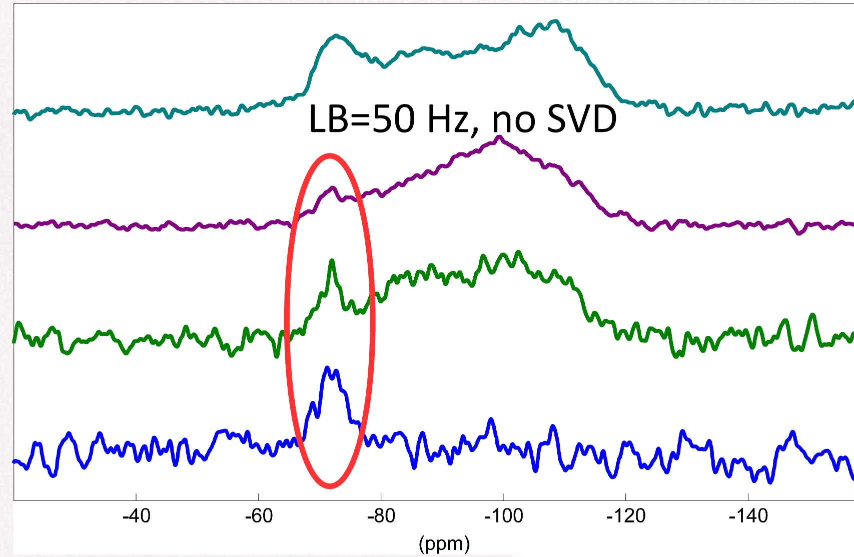
Best results with square matrix
Number of values needs to be finely adjusted

[1] H. Y. Carr and E. M. Purcell, Phys. Rev., vol. 94, no. 3, pp. 630–638, May 1954.
[2] S. Meiboom and D. Gill, Review of Scientific Instruments, vol. 29, no. 8, pp. 688–691, Jun. 1958.

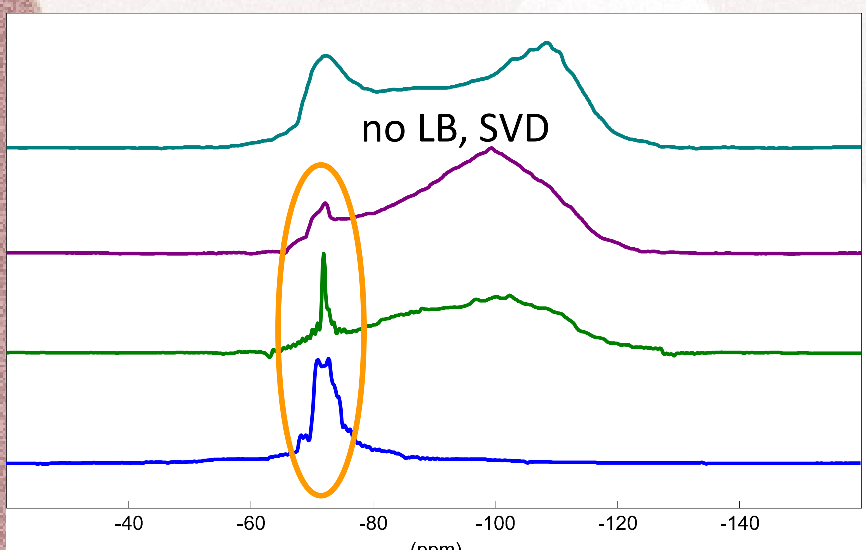
Spectra denoising 3/4
Signal / noise

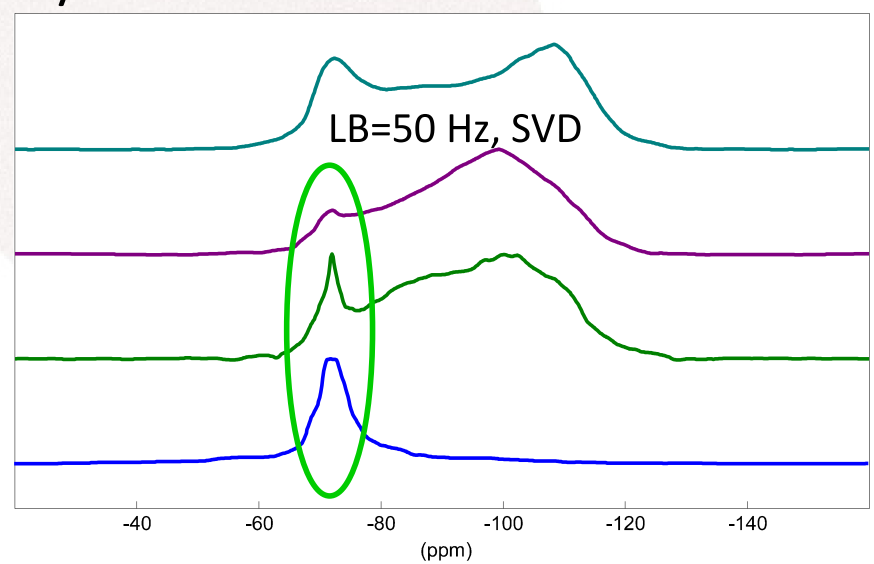
Morocco ^{29}Si Tanger Shale,
^{31}P Bengurir natural phosphate,
1:1 wt, 800°C





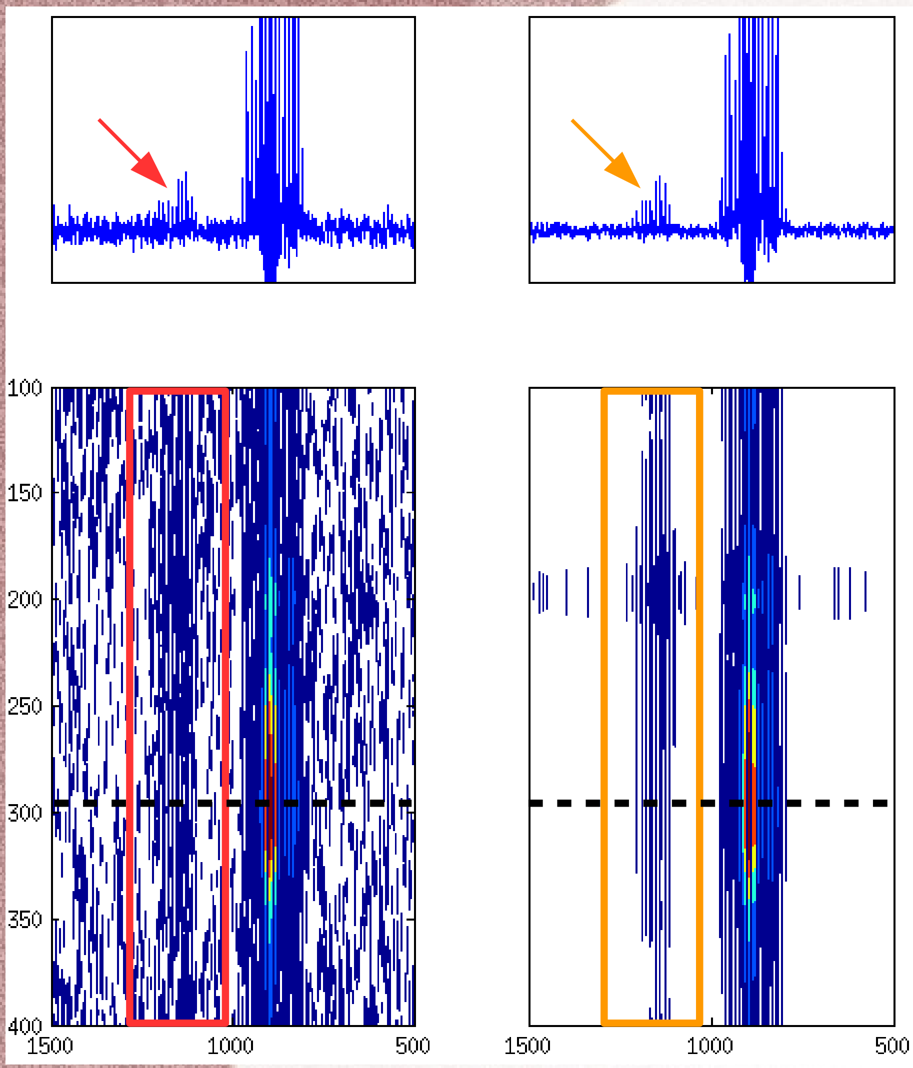
Minimum signal / noise = 2



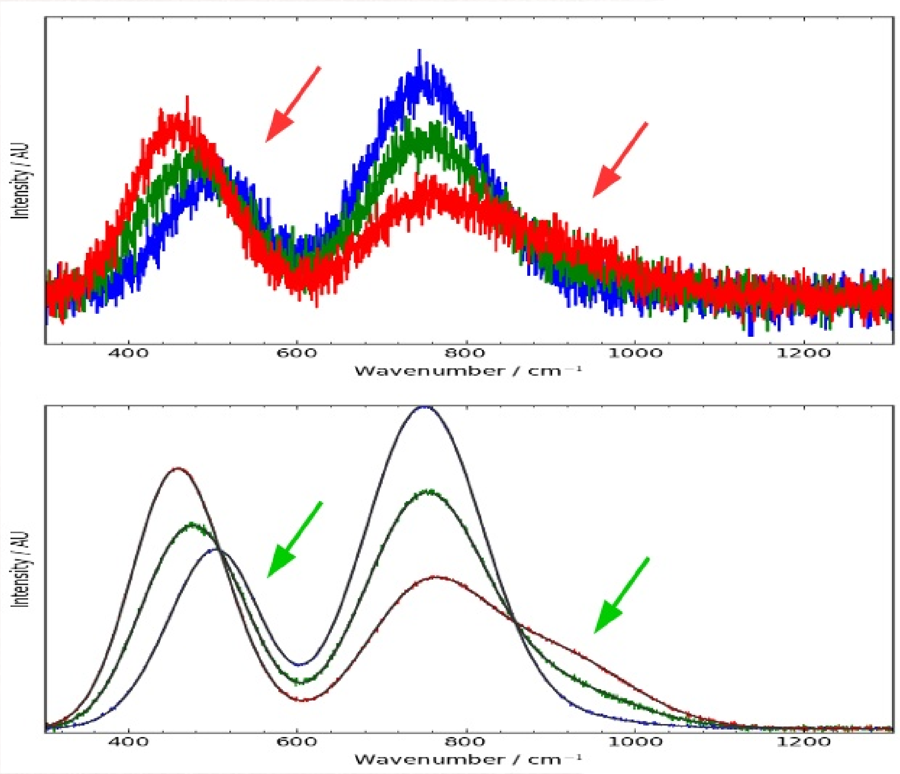


[1] H. Rhaiti *et al*, Materials Chemistry and Physics, vol. 136, no. 2–3, pp. 1022–1026, Oct. 2012.
[2] C. Coelho *et al*, J Sol-Gel Sci Technol, vol. 40, no. 2–3, pp. 181–189, Dec. 2006.
[3] G. Gasquères *et al*, Magn. Reson. Chem., vol. 46, no. 4, pp. 342–346, Apr. 2008.

Spectra denoising 4/4
2D datasets



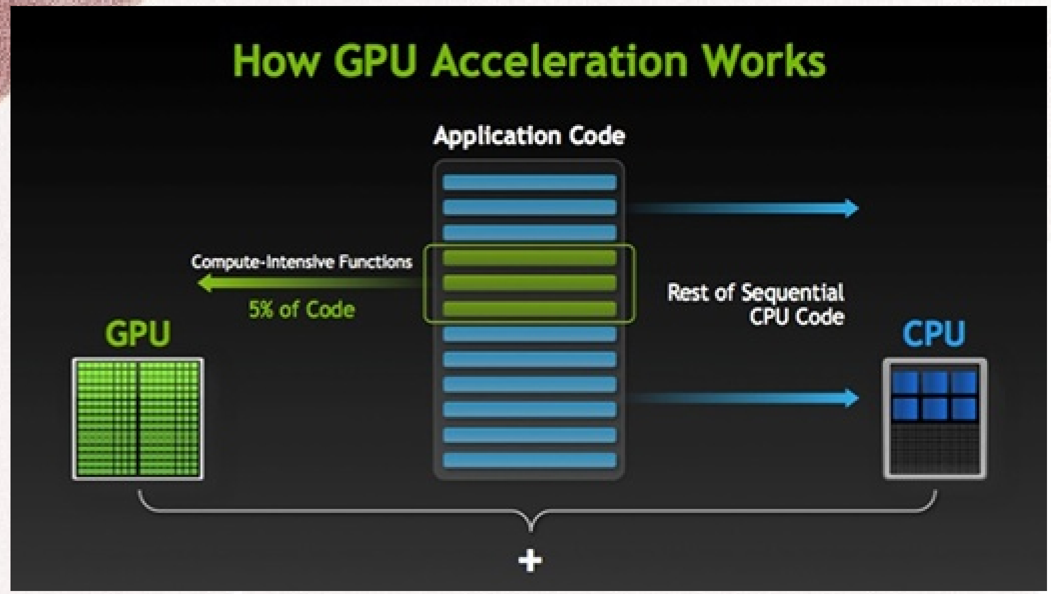
^{1}H-^{29}Si CPMG 2D spectrum
TEOS:MTEOS 95:5 %m
24 sclices, 512x2048 points, 2 values



Raman simulated spectra
2000x2000 points, 2 values

Both 1D and 2D spectra are usable, same time

Java CPU vs GPU 1/4
CUDA



nvidia.com

8400 GS, GTX 260, GTX 660

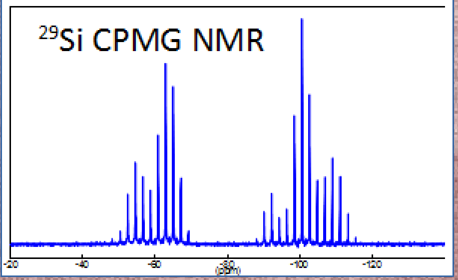
Nvidia CUDA (Compute Unified Device Architecture)
All cards since 2006
Massively parallel

[1] P. P. Man *et al*, Solid State Nucl. Mag., vol. 61–62, pp. 28–34, Jul. 2014.
[2] D. Kirk, ‘NVIDIA CUDA Software and GPU Parallel Computing Architecture’, presented at the ISMM, Oct-2007.
[3] J.-M. Richer, ‘Cuda - Introduction et Historique’. [Online]. Available: http://www.info.univ-angers.fr/~richer/cuda_crs1.php.

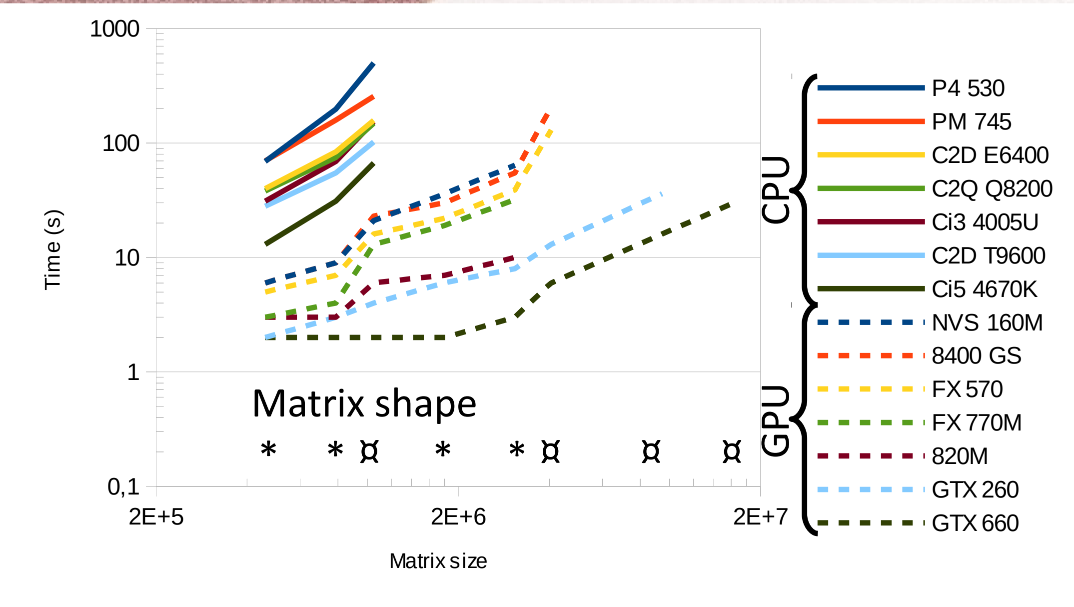
Java CPU vs GPU 2/4
CPU and GPU list

Central Processing Unit (CPU)	Type	Year	Techno-logy (nm)	Number of cores	Cache (MB)	Core frequency (MHz)	Memory frequency (MHz)	Memory size (MB)	CPU Mark	Monothread (Mops/s)	Matrices (millions/s)
Intel Pentium M 745	laptop	2004	90	1	2	1800	133	1024	444,8	577	1,41
Intel Pentium 4 530	desktop	2004	90	2	1	3000	200	1024	335,2	726	0,29
Intel Core 2 Duo E6400	desktop	2006	65	2	2	2130	333	2048	1451	849	3,72
Intel Core 2 Quad Q8200	desktop	2008	45	4	4	2330	400	4096	2001	1004	5,8
Intel Core 2 Duo T9600	laptop	2008	45	2	6	2800	400	4096	2190	1161	4,67
Intel Core i3 4005U	laptop	2013	22	4	3	1700	800	4096	2551	1010	11,4
Intel Core i5 4670K	desktop	2013	22	4	6	4200	1000	8192	8824	2519	31,6

Graphics Processing Unit (GPU)	Type	Year	Techno-logy (nm)	Number of cores	Bandwidth (GB/s)	Core frequency (MHz)	Memory frequency (MHz)	Memory size (MB)	Single precision float (Gflop/s)	Double precision float (Gflop/s)	CUDA compute capability
Nvidia Quadro FX 570	desktop	2007	80	16	12,8	460	400	256	29	#N/D	1.1
Nvidia GeForce 8400 GS	desktop	2008	65	8	6,4	567	400	512	21	#N/D	1.1
Nvidia Quadro NVS 160M	laptop	2008	65	8	11,2	580	700	256	23	#N/D	1.1
Nvidia Quadro FX 770M	laptop	2008	65	32	25,6	500	800	512	79	#N/D	1.1
Nvidia GeForce GTX 260	desktop	2008	65	216	111,9	576	1000	896	533	67	1.3
Nvidia GeForce 820M	laptop	2012	28	96	14,4	625	900	2048	315	31	2.1
Nvidia GeForce GTX 660	desktop	2012	28	960	144,2	1100	1500	2048	1707	88	3.0



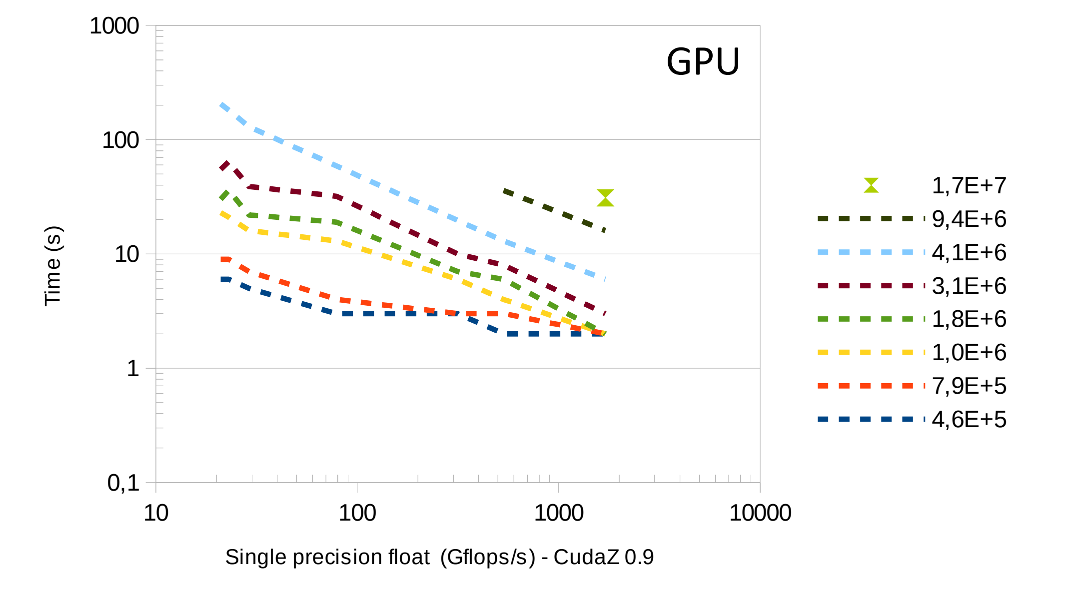
Java CPU vs GPU 3/4 Benchmarks



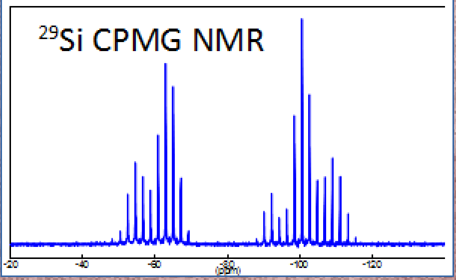
CPU: 67-499 s
GPU: 2-23 s
> 20 times faster

CPU: time limited + app limited
GPU: memory limited

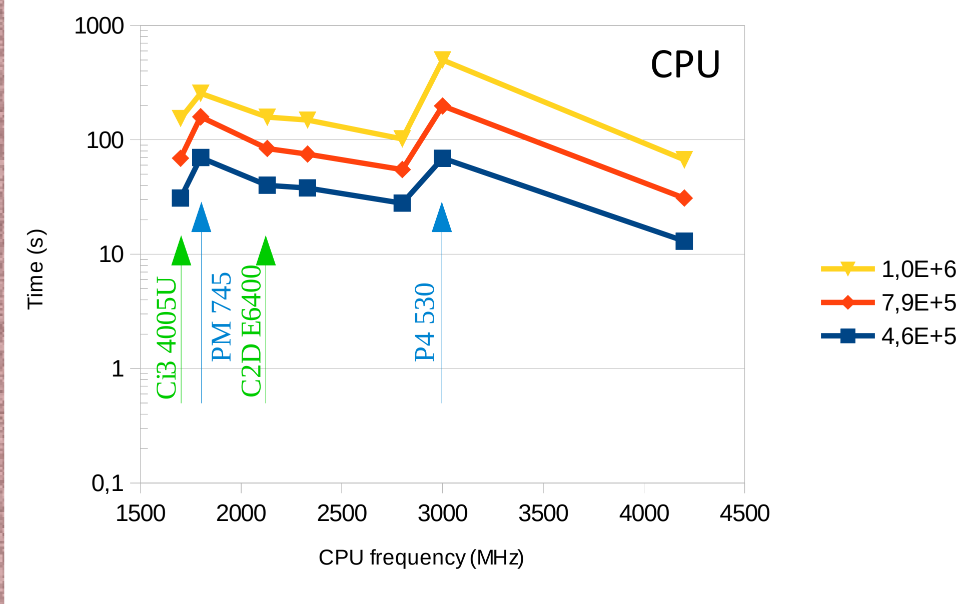
Time jump for square matrices



SP float = good GPU indicator



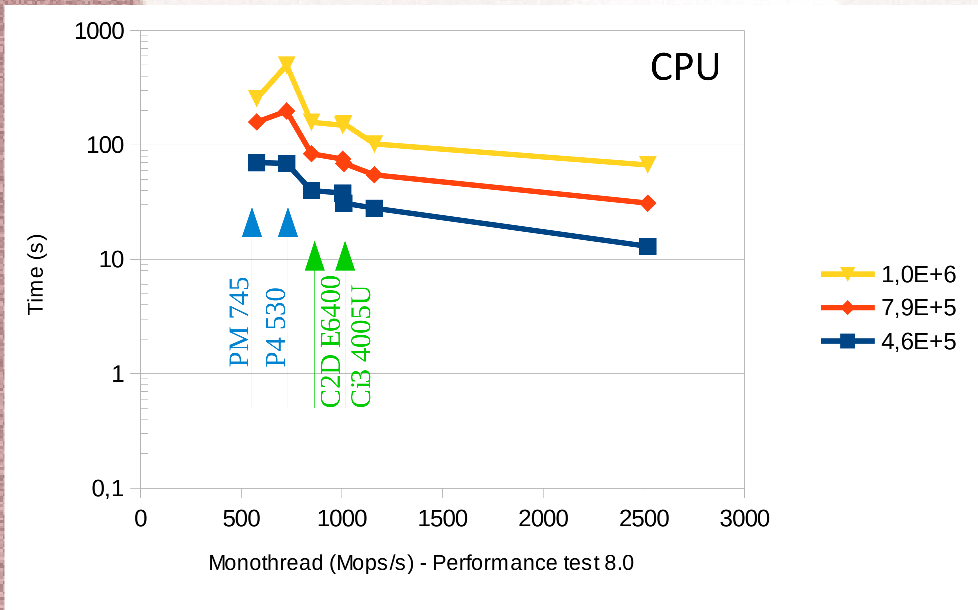
Java CPU vs GPU 4/4
CPU indicator



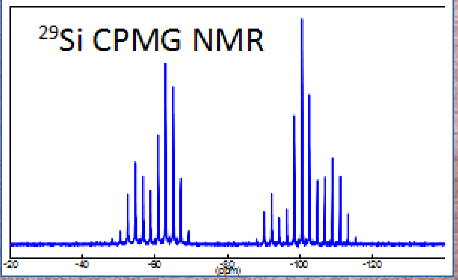
CPU frequency = bad indicator
Monothread = good indicator

CPU: Cache memory limited

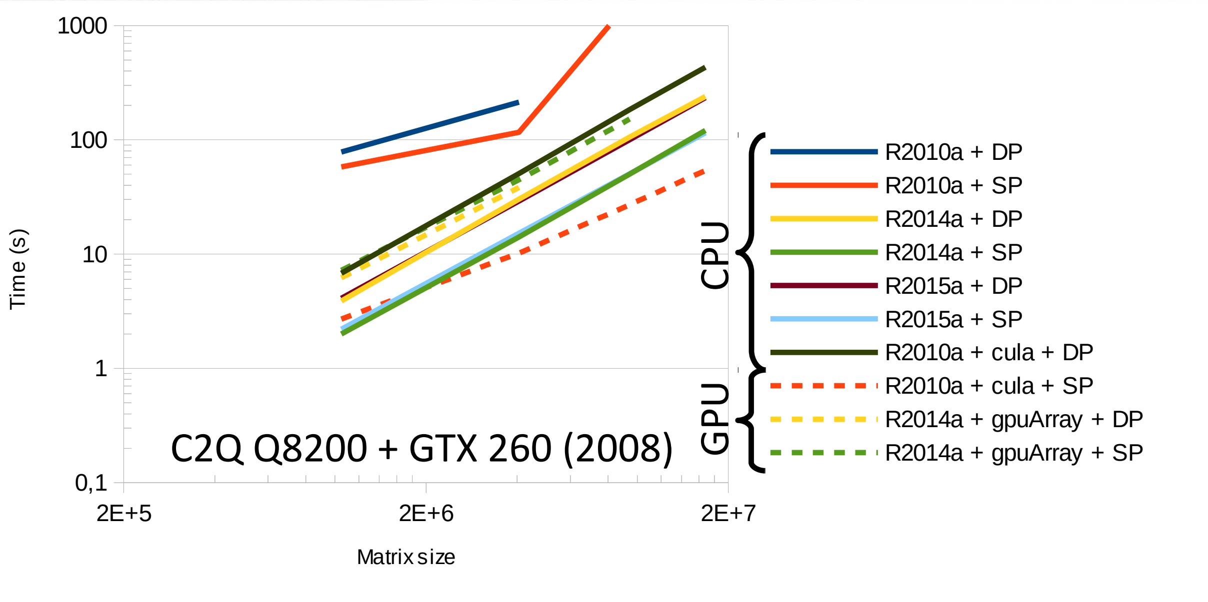
Ci3 (2013) = C2D (2008)
No technology improvement

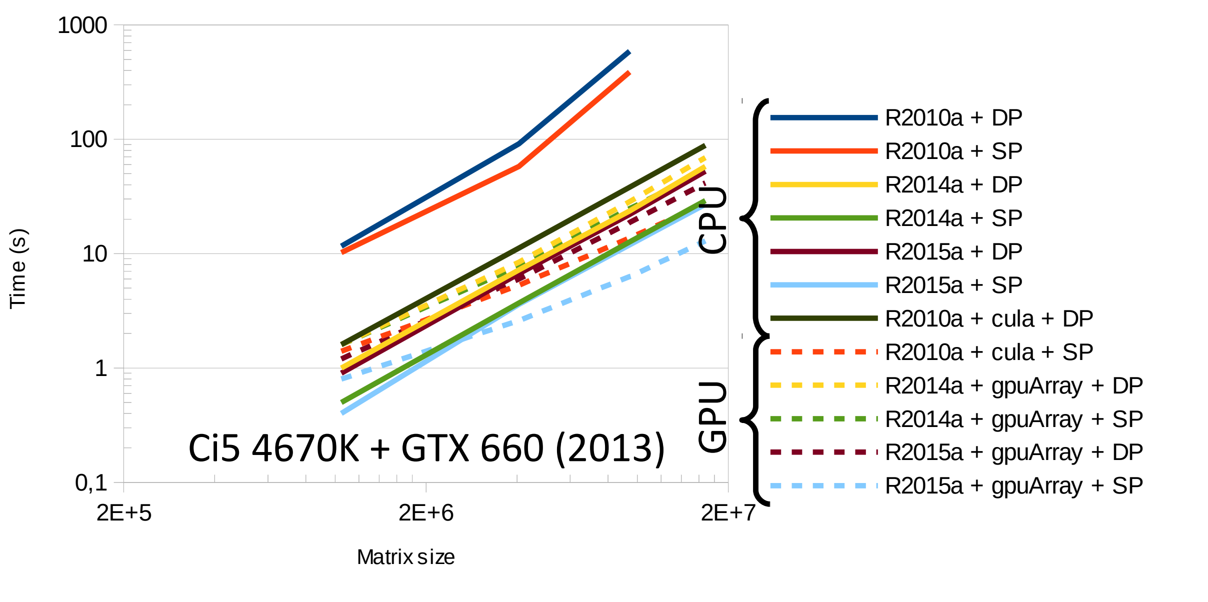


Is improvement
really due to GPU ?



Software improvements 1/3
Matlab versions





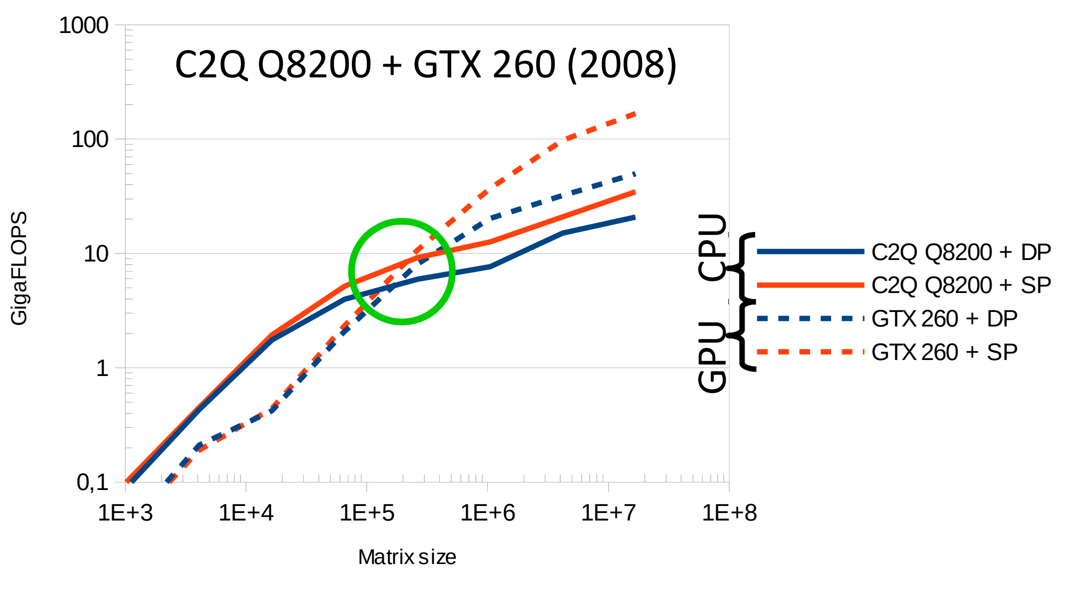
Cula free: GPU SP only
Cula free: Matlab R2010a
GTX 260: Matlab R2014a
GTX 660: Matlab R2015a

CPU: divide and conquer
→ 7-32 times faster
CPU: SP ≈ DP/2
GPU: avoid R2014a
CPU < GPU for small matrix

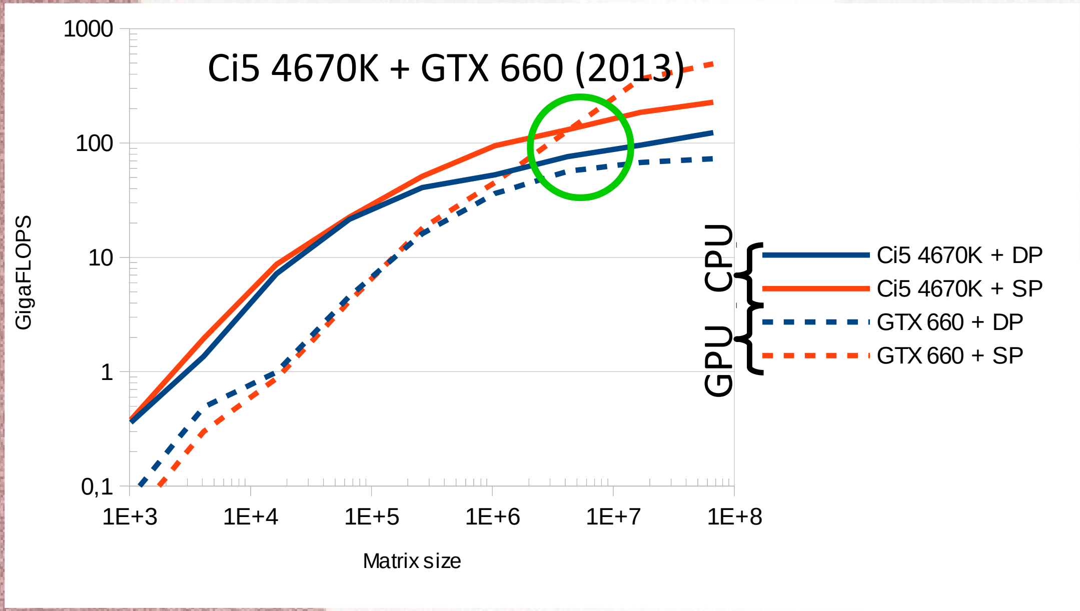
[1] J. R. Humphrey *et al*, in Proc. SPIE 7705, Orlando, USA, 2010, vol. 7705, pp. 770502–770502–7.
[2] G. Laurent, ‘svd under Matlab with cula link’, CULA tools. 24-Jun-2015.
[3] M. Gu and S. Eisenstat, SIAM. J. Matrix Anal. & Appl., vol. 16, no. 1, pp. 79–92, Jan. 1995.

Software improvements 2/3
CPU vs GPU operations

GPU Bench
Equation resolution
Matlab R2014a



Crossing is coherent with previous results
GPU at its memory limit

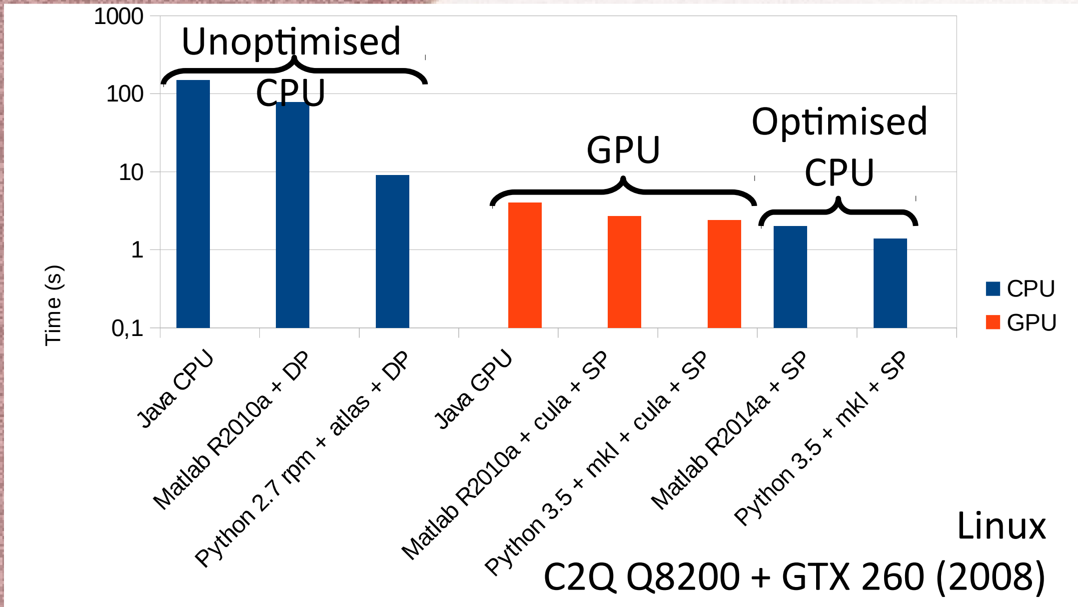


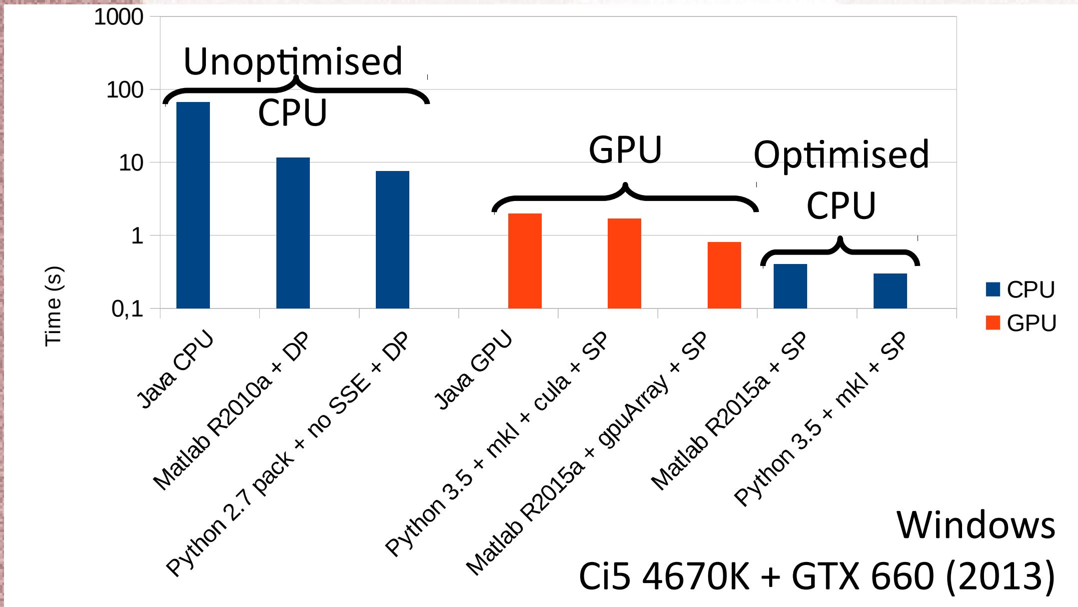
GPU transfers are limiting

[1] S. Lahabar and P. J. Narayanan, in IEEE International Symposium on Parallel Distributed Processing, 2009. IPDPS 2009, 2009, pp. 1–10.

Software improvements 3/3
Python optimised libraries

Matrix size
1025x1024
Max Java CPU



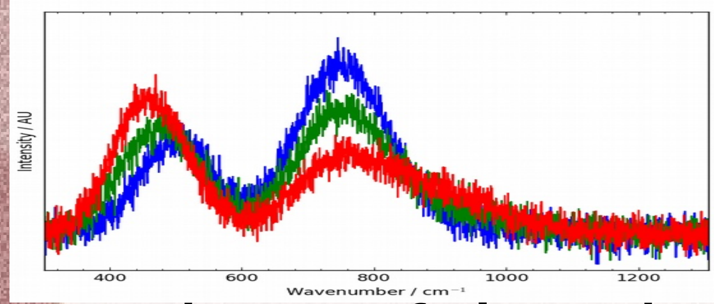


SSE3: optimised CPU instructions
MKL: Intel optimised libraries

Divide and conquer: /8
no SSE → SSE3: /3
DP → SP: /2
MKL: /2
Python MKL ≈ Java CPU / 100

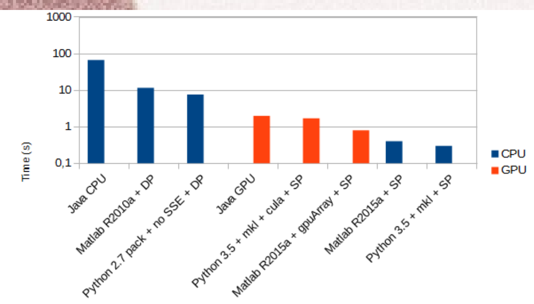
Java CPU / GPU difference is due to algorithm improvements

[1] J.-P. Gehrcke, ‘Building numpy and scipy with Intel compilers and Intel MKL on a 64 bit machine’, Jan-Philip Gehrcke, 18-Feb-2014.
[2] V. Nguyen, ‘Optimized R and Python: standard BLAS vs. ATLAS vs. OpenBLAS vs. MKL’, Super Nerdy Cool, 10-Nov-2014.



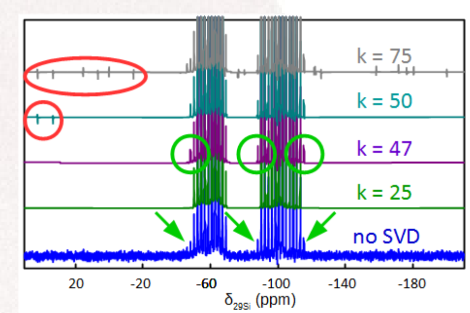
Conclusion

- SVD is useful to denoise 1D & 2D spectra
- Sensitivity gain, time gain
- Using graphic cards ?
  - Java GPU is fast → YES
  - Middle range GPU are needed → YES/NO
  - Use MKL & SSE3 if available (/100) → NO



Future work

- Enhance 2D denoising
- Automatic thresholding
- Sparse matrices



Acknowledgements

Laboratory management

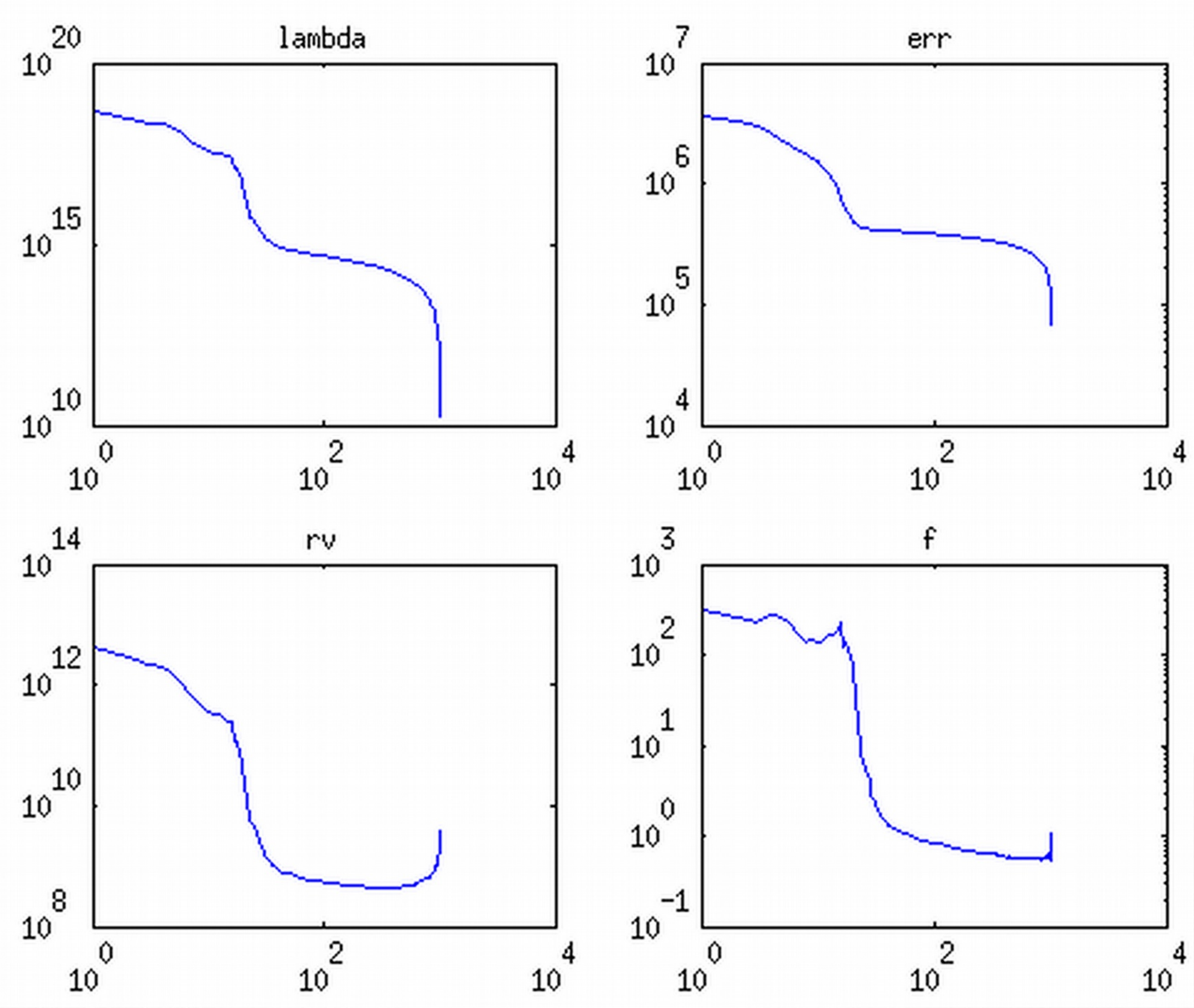
SMiLES group

Thank you for your attention

NMR colleagues

Logistic staff

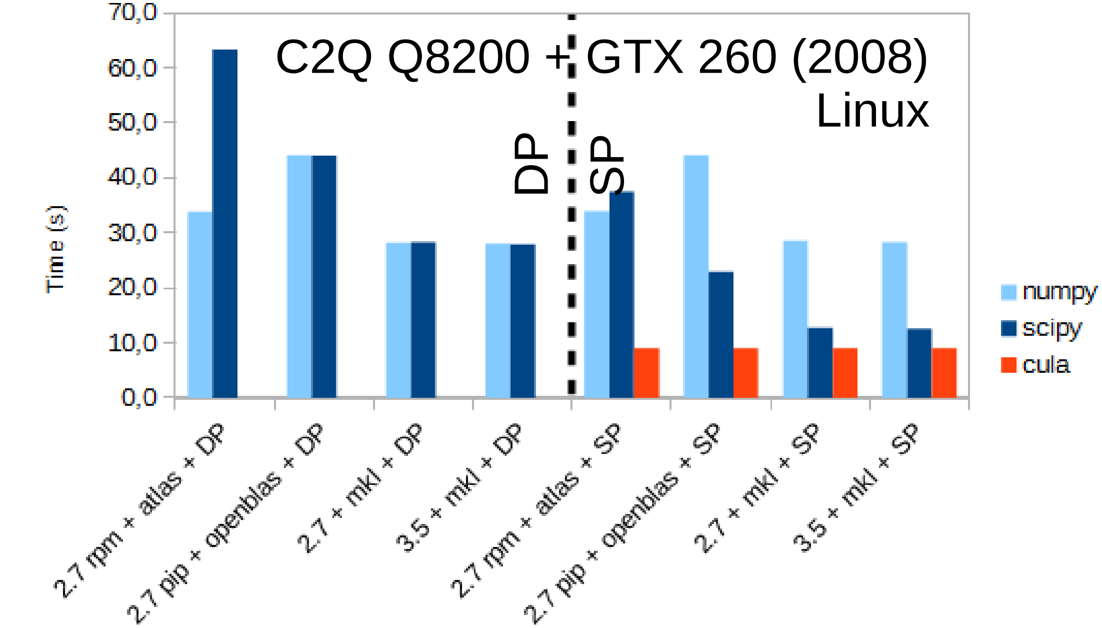
Automatic thresholding

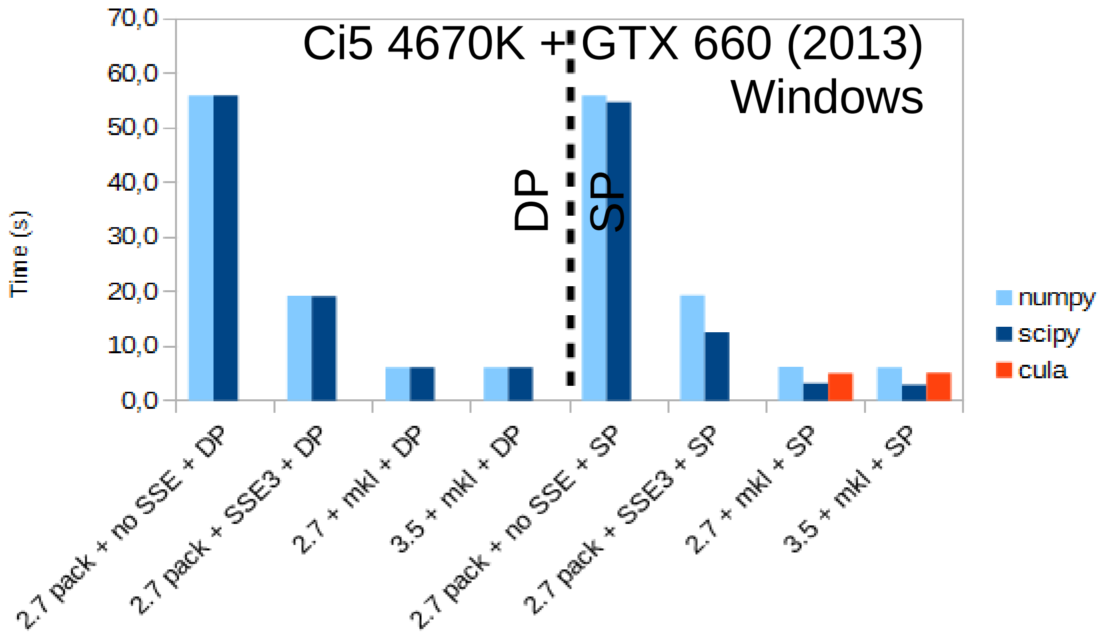


[1] E. R. Malinowski, J. Chemometrics, vol. 1, no. 1, pp. 33–40, Jan. 1987.
[2] P. Gemperline, Practical Guide To Chemometrics, Second edition, pp, 92-93. CRC Press, 2006.

Python libraries

Matrix size
2015x2014





Numpy & scipy: python functions
SSE: optimised CPU instructions

Atlas > openblas > mkl
Numpy > scipy for SP
SSE3 ≈ no SSE / 3
Python 2.7 = python 3.5

[1] J.-P. Gehrcke, ‘Building numpy and scipy with Intel compilers and Intel MKL on a 64 bit machine’, Jan-Philip Gehrcke, 18-Feb-2014.
[2] V. Nguyen, ‘Optimized R and Python: standard BLAS vs. ATLAS vs. OpenBLAS vs. MKL’, Super Nerdy Cool, 10-Nov-2014.