Ministero Istruzione Università Ricerca

Anno 2012
Protocollo: RBFR12JF2Z_002

FIRB Fondo per gli Investimenti della Ricerca di Base

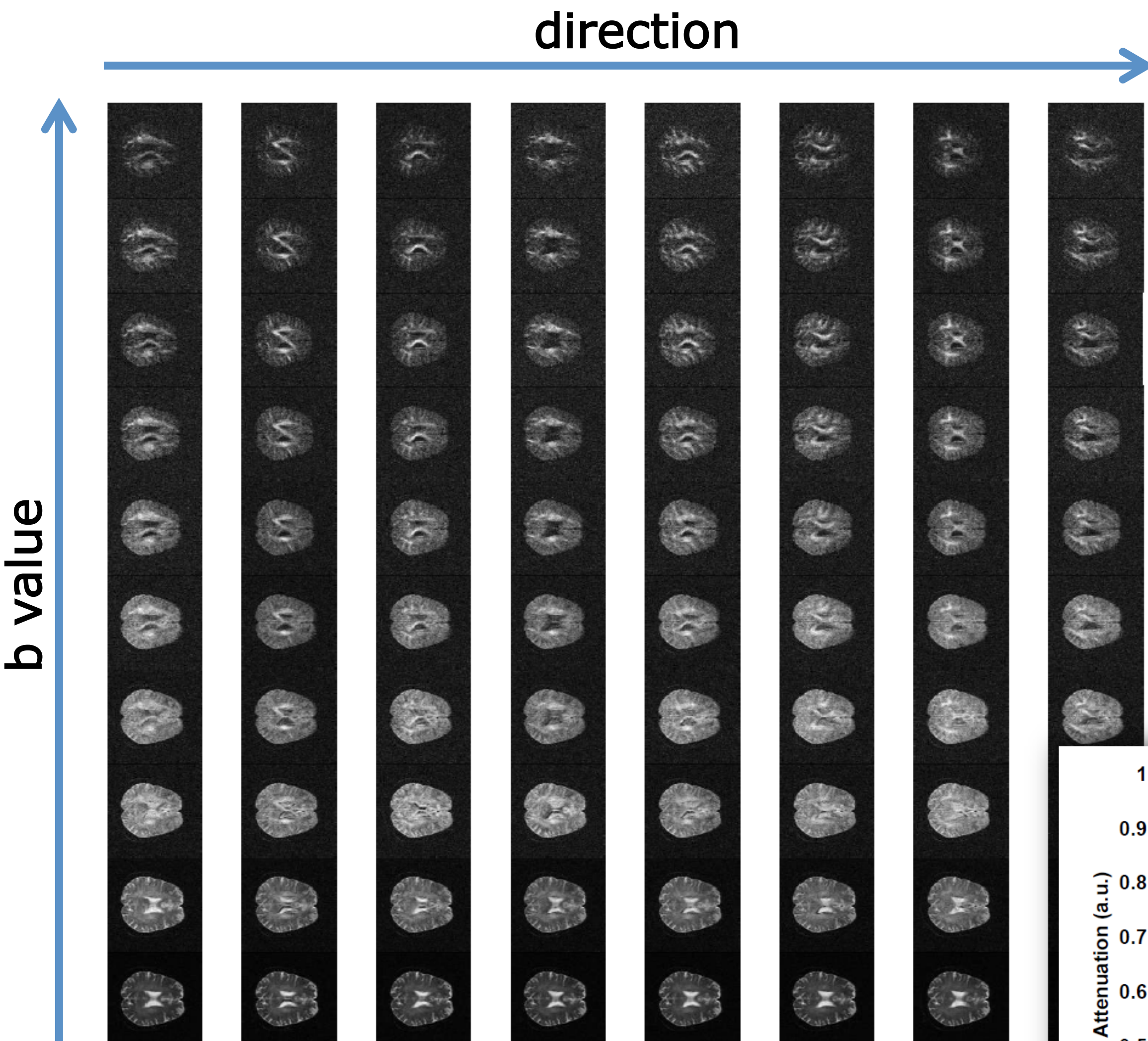
FUTURO IN RICERCA

Unità A2: Imaging NMR, Roma

Accelerazione della ricostruzione di immagini NMR ottenute con il metodo del contrasto in diffusione

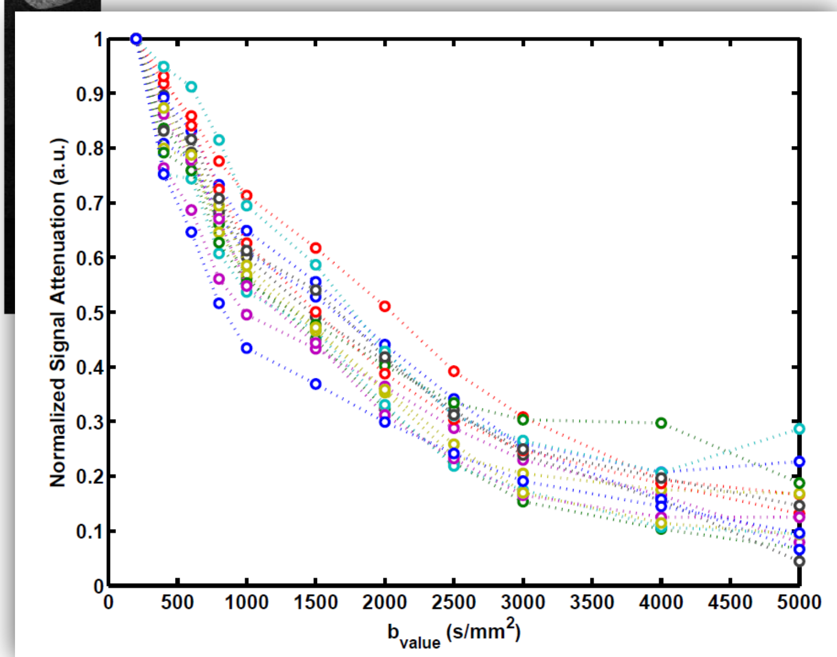
Dr.ssa **Silvia Capuani**[*$], Dr. **Marco Palombo**[*$]

*Physics Department, "Sapienza" University of Rome,
$ CNR-IPCF UOS Roma, Rome



Typical NMR imaging post-processing

128x128 pixel image
x
number of slice (~10-40)
x
number of diffusion gradient directions (≥15)



Cumulant expansion approach: kurtosis tensor measurement

$$S(b) \propto e^{-D_{app}\, b + \frac{1}{6} K_{app} \left[D_{app}\, b \right]^2 + \ldots}$$

$$S(b) = S(0) \exp\left[-b \sum_{i,j} n_i n_j D_{ij} + \frac{1}{6} b^2 \left(\frac{1}{3} \sum_i D_{ii} \right)^2 \sum_{i,j,k,l} n_i n_j n_k n_l W_{ijkl} + O(b^3) \right]$$

N ≥ 15 independent measures to get W_{ijkl}

$$K(\mathbf{n}) = \frac{\overline{D}^2}{[D(\mathbf{n})]^2} \sum_{i,j,k,l=1}^{3} n_i n_j n_k n_l W_{ijkl}.$$

$$\boxed{MK = \frac{1}{N} \sum_{i=1}^{N} K(\mathbf{n}_i)}$$

$$\boxed{\begin{aligned}\overline{K} = {} & F_1(\lambda_1,\lambda_2,\lambda_3)\tilde{W}_{1111} + F_1(\lambda_2,\lambda_1,\lambda_3)\tilde{W}_{2222} \\ & + F_1(\lambda_3,\lambda_2,\lambda_1)\tilde{W}_{3333} + F_2(\lambda_1,\lambda_2,\lambda_3)\tilde{W}_{2233} \\ & + F_2(\lambda_2,\lambda_1,\lambda_3)\tilde{W}_{1133} + F_2(\lambda_3,\lambda_2,\lambda_1)\tilde{W}_{1122},\end{aligned}}$$

$$\boxed{K_{||} = \frac{(\lambda_1+\lambda_2+\lambda_3)^2}{9\lambda_1^2}\tilde{W}_{1111},}$$

$$\boxed{\begin{aligned}K_{\perp} = {} & G_1(\lambda_1,\lambda_2,\lambda_3)\tilde{W}_{2222} + G_1(\lambda_1,\lambda_3,\lambda_2)\tilde{W}_{3333} \\ & + G_2(\lambda_1,\lambda_2,\lambda_3)\tilde{W}_{2233},\end{aligned}}$$

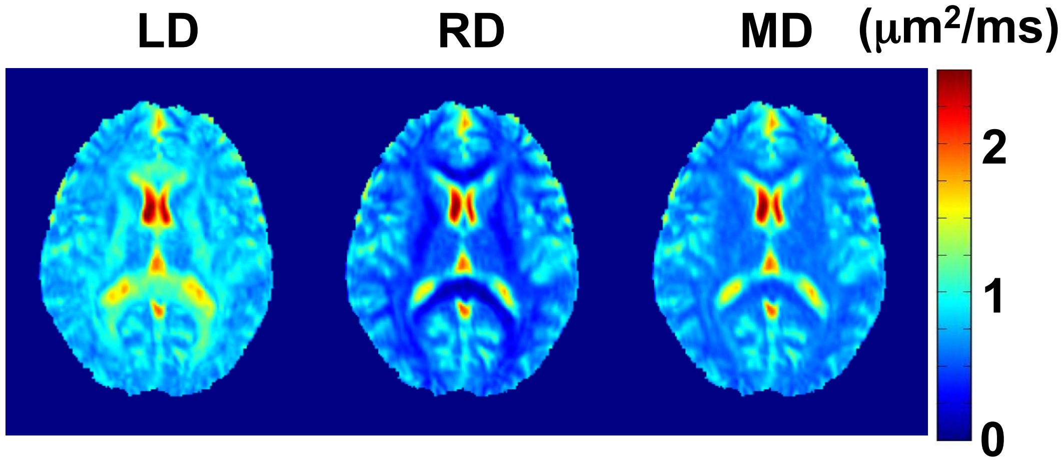
$$G_1(\lambda_1,\lambda_2,\lambda_3) = \frac{(\lambda_1+\lambda_2+\lambda_3)^2}{18\lambda_2(\lambda_2-\lambda_3)^2}\left(2\lambda_2 + \frac{\lambda_3^2 - 3\lambda_2\lambda_3}{\sqrt{\lambda_2\lambda_3}}\right),$$

$$G_2(\lambda_1,\lambda_2,\lambda_3) = \frac{(\lambda_1+\lambda_2+\lambda_3)^2}{3(\lambda_2-\lambda_3)^2}\left(\frac{\lambda_2+\lambda_3}{\sqrt{\lambda_2\lambda_3}} - 2\right).$$

$$F_1(\lambda_1,\lambda_2,\lambda_3) \equiv \frac{(\lambda_1+\lambda_2+\lambda_3)^2}{18(\lambda_1-\lambda_2)(\lambda_1-\lambda_3)}\left[\frac{\sqrt{\lambda_2\lambda_3}}{\lambda_1}R_F\left(\frac{\lambda_1}{\lambda_2},\frac{\lambda_1}{\lambda_3},1\right) + \frac{3\lambda_1^2-\lambda_1\lambda_2-\lambda_2\lambda_3-\lambda_1\lambda_3}{3\lambda_1\sqrt{\lambda_2\lambda_3}}R_D\left(\frac{\lambda_1}{\lambda_2},\frac{\lambda_1}{\lambda_3},1\right) - 1\right],$$

and

$$F_2(\lambda_1,\lambda_2,\lambda_3) \equiv \frac{(\lambda_1+\lambda_2+\lambda_3)^2}{3(\lambda_2-\lambda_3)^2}\left[\frac{\lambda_2+\lambda_3}{\sqrt{\lambda_2\lambda_3}}R_F\left(\frac{\lambda_1}{\lambda_2},\frac{\lambda_1}{\lambda_3},1\right) + \frac{2\lambda_1-\lambda_2-\lambda_3}{3\sqrt{\lambda_2\lambda_3}}R_D\left(\frac{\lambda_1}{\lambda_2},\frac{\lambda_1}{\lambda_3},1\right) - 2\right].$$



Conventional DTI

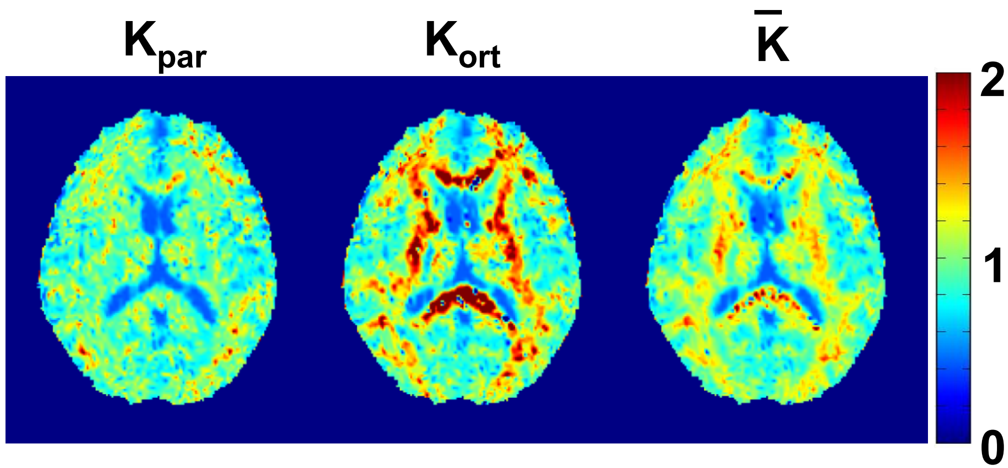
128x128x32x15

=

7.864320 10^6

linear systems to solve

~ 15 s on CPU*



Kurtosis tensor imaging

128x128x32x15

=

7.864320 10^6

non-linear fit to perform

~7200 s on CPU*

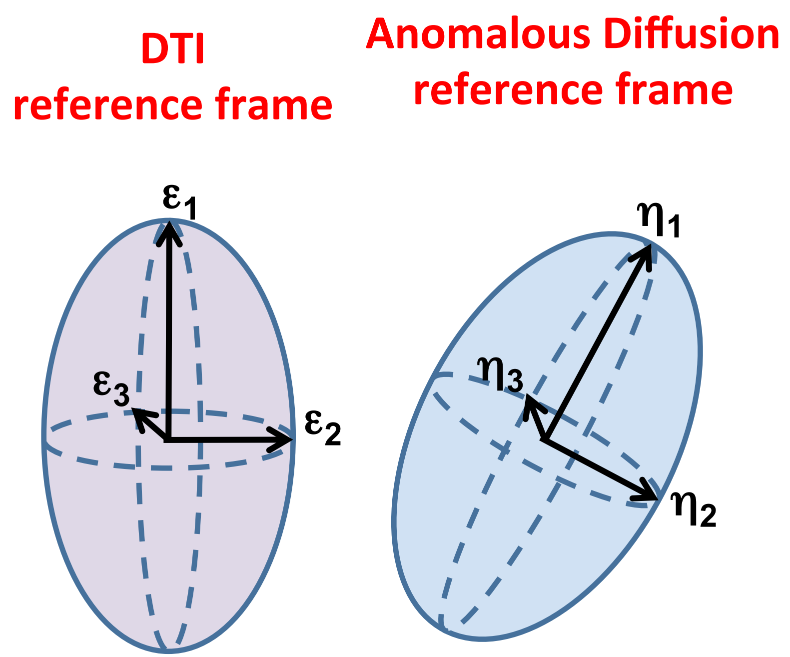
* 8 threads on an Intel Xeon E5-2609 CPU at 2.4 GHz

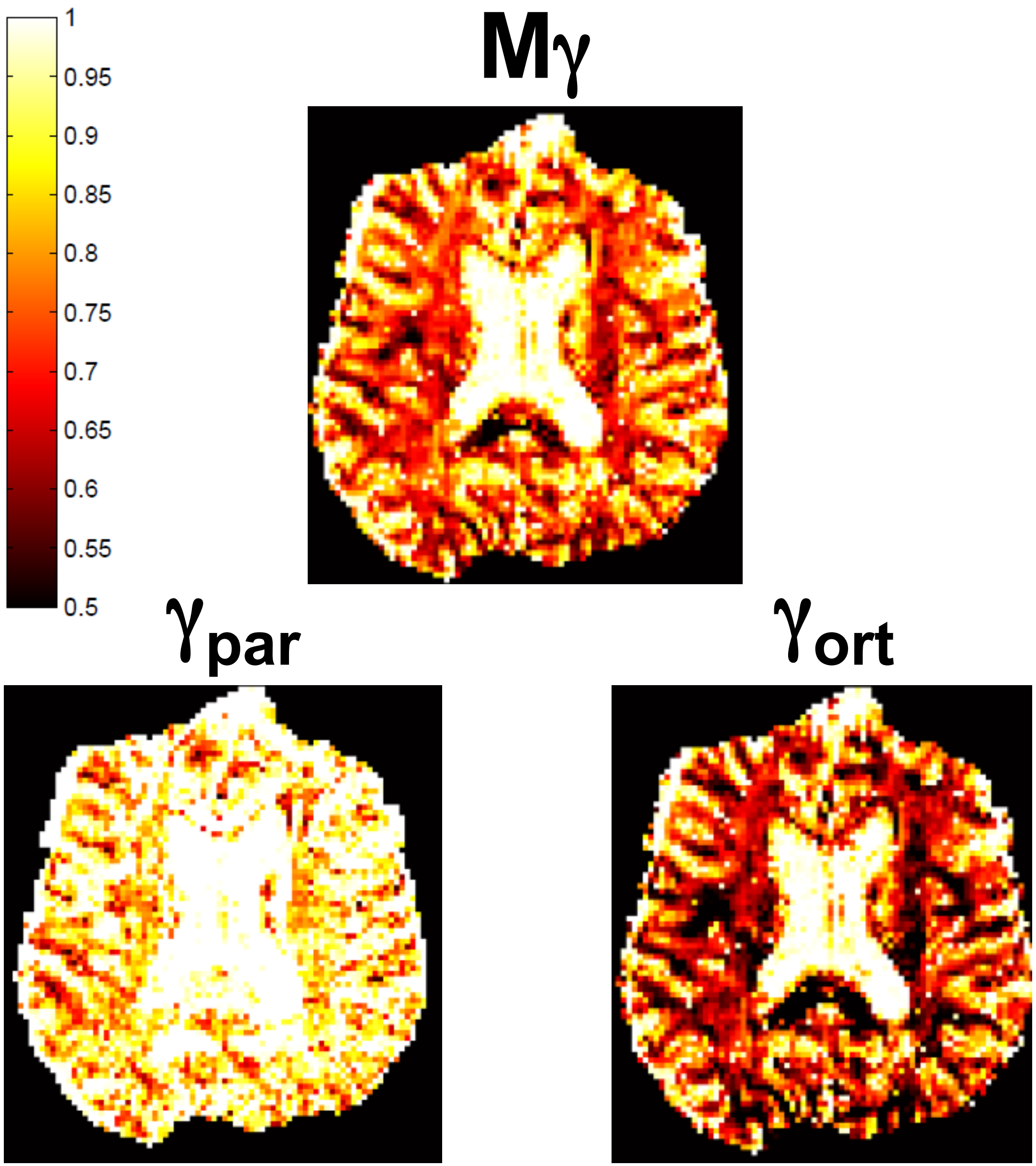
Stretched exponential model

$$S(b) = S(0)e^{-A_1(b_1^*)^{\gamma 1} - A_2(b_2^*)^{\gamma 2} - A_3(b_3^*)^{\gamma 3}}$$

$$b_i^* = \vec{b} \cdot \vec{\eta}_i$$

Anomalous diffusion reference frame





Conventional DTI

128x128x32x15

=

7.864320 10^6

linear systems to solve

~ 15 s on CPU*

Stretched exponential imaging

128x128x32x15

=

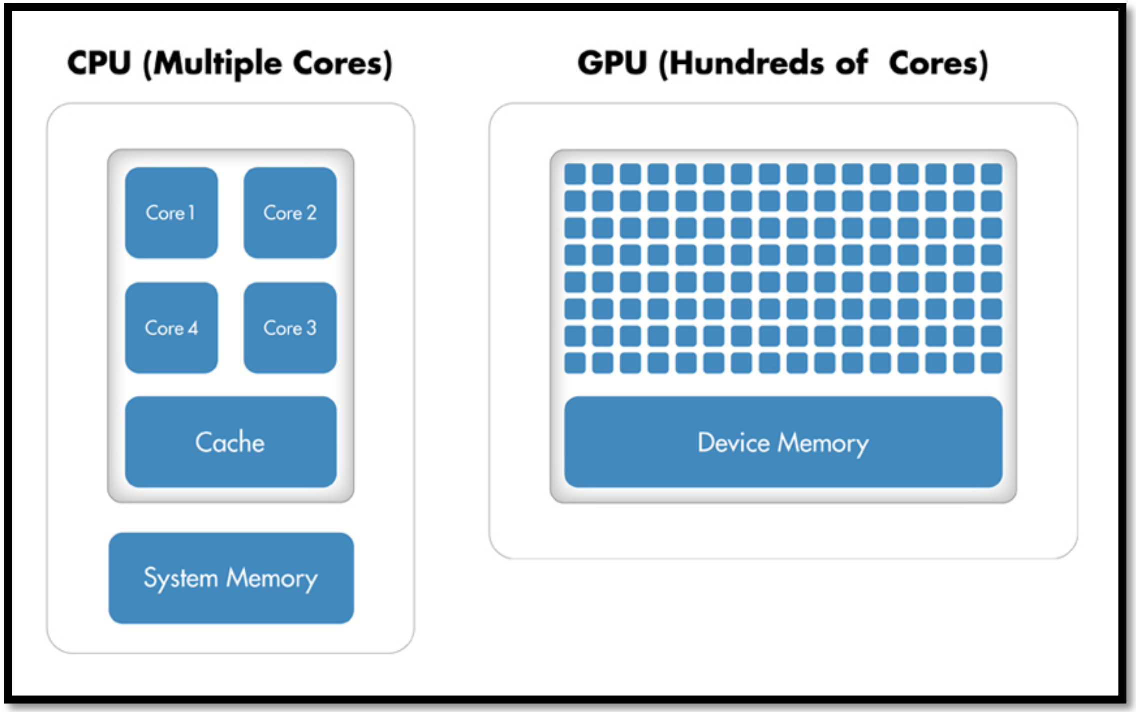
7.864320 10^6

non-linear fit to perform

~6600 s on CPU*

* 8 threads on an Intel Xeon E5-2609 CPU at 2.4 GHz

When GPU can accelerate an application



- **Computationally intensive** — The time spent on computation significantly exceeds the time spent on transferring data to and from GPU memory.

- **Massively parallel** — The computations can be broken down into hundreds or thousands of independent units of work.

Computationally intensive

Kurtosis tensor imaging

non-linear fit

~7 ms per pixel

vs

~ 10-80 ns to read data from memory

Stretched exponential imaging

non-linear fit

~6 ms per pixel

vs

~10-80 ns to read data from memory

Massively parallel

Kurtosis tensor imaging

~10^6-10^7 **independent** non-linear fit

Stretched exponential imaging

10^6-10^7 **independent** non-linear fit

- Non-conventional NMR image processing algorithms code optimization;

Algorithms porting on GPU

1 or 2 months | ***6 or 8 months***

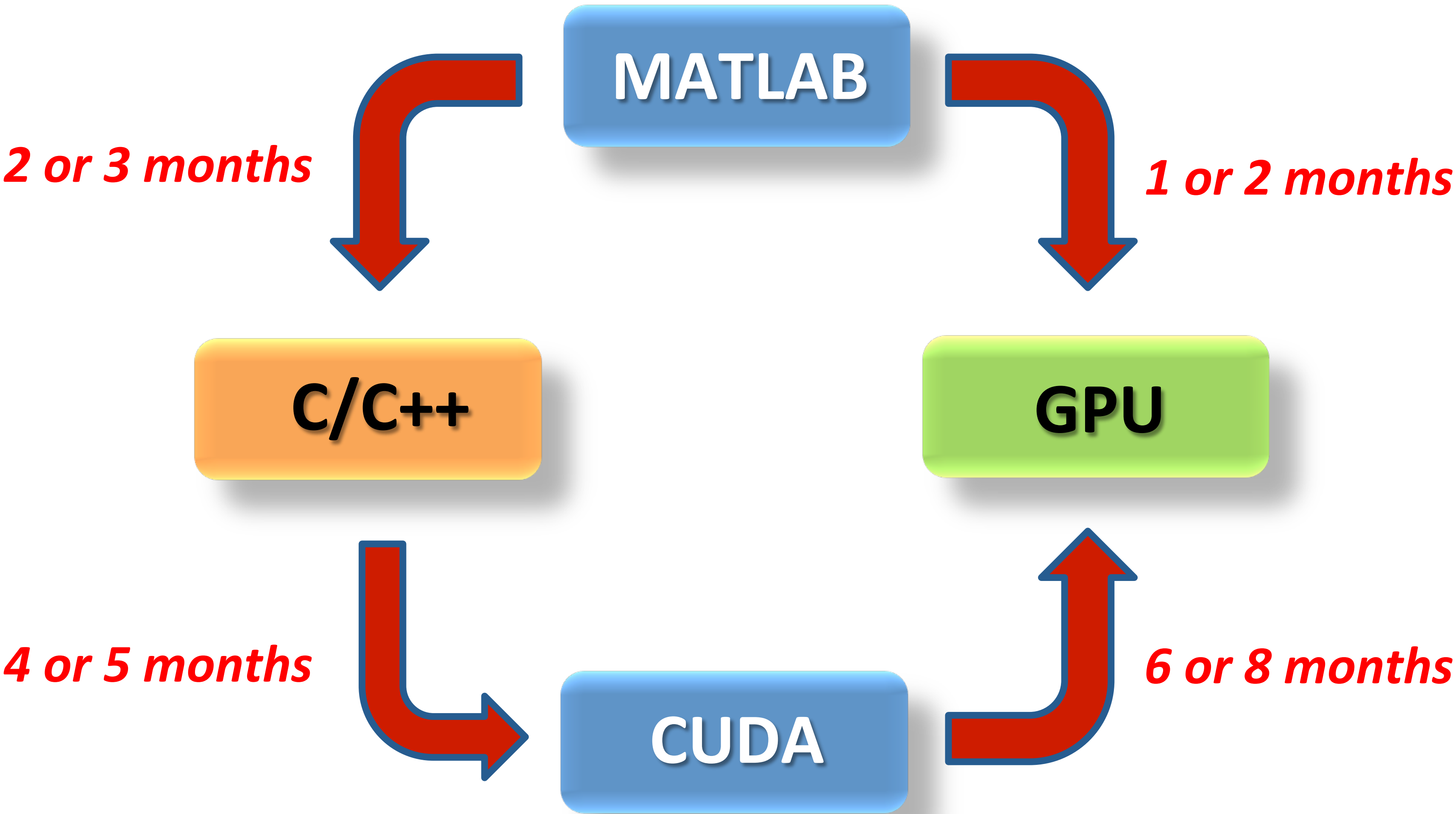
MATLAB | **CUDA**

MATLAB

- Easier and faster (all our codes are written for Matlab)
- Bypass the study of the specific hardware architecture
- Only a single GPU can be used

CUDA

- Slower (all our codes are written for Matlab)
- Require the study of the specific hardware architecture
- Multiple GPUs can be used


MATLAB
2 or 3 months
1 or 2 months
C/C++
GPU
4 or 5 months
6 or 8 months
CUDA

Completed Steps to achieve our goals

(0) Configuring gap01 server for NMR imaging applications:

1. New license for MATLAB R2013b server usage by multiple accounts (2 simultaneous users) bought;

2. Matlab Parallel Toolbox; Matlab Image Processing Toolbox and Matlab Optimization Toolbox bought;

(00) Recruitment and training of new people to involve in the Project

1. One new PhD Student and one new Graduating Student recruited and under training for DW-NMR image processing and related numerical simulations

(i) Fast-implementation: using of available Matlab custom script.

1. Optimization of available Matlab custom script for DKI and SEM analysis for usage on GPU by means of the Matlab Parallel Toolbox: feasibility study by means of numerical simulations.

(ii) Implementation of numerical simulations supporting experimental data analysis: Monte Carlo and Finite Elements Method (FEM) on GPU

1. Implementation of a parallelized version of available Monte Carlo and FEM algorithms to numerically simulate DW-MRI signal attenuation in complex restricting/hindering geometries for usage on GPU by means of the Matlab Parallel Toolbox;

NMR simulation of diffusion weighted NMR images reconstruction

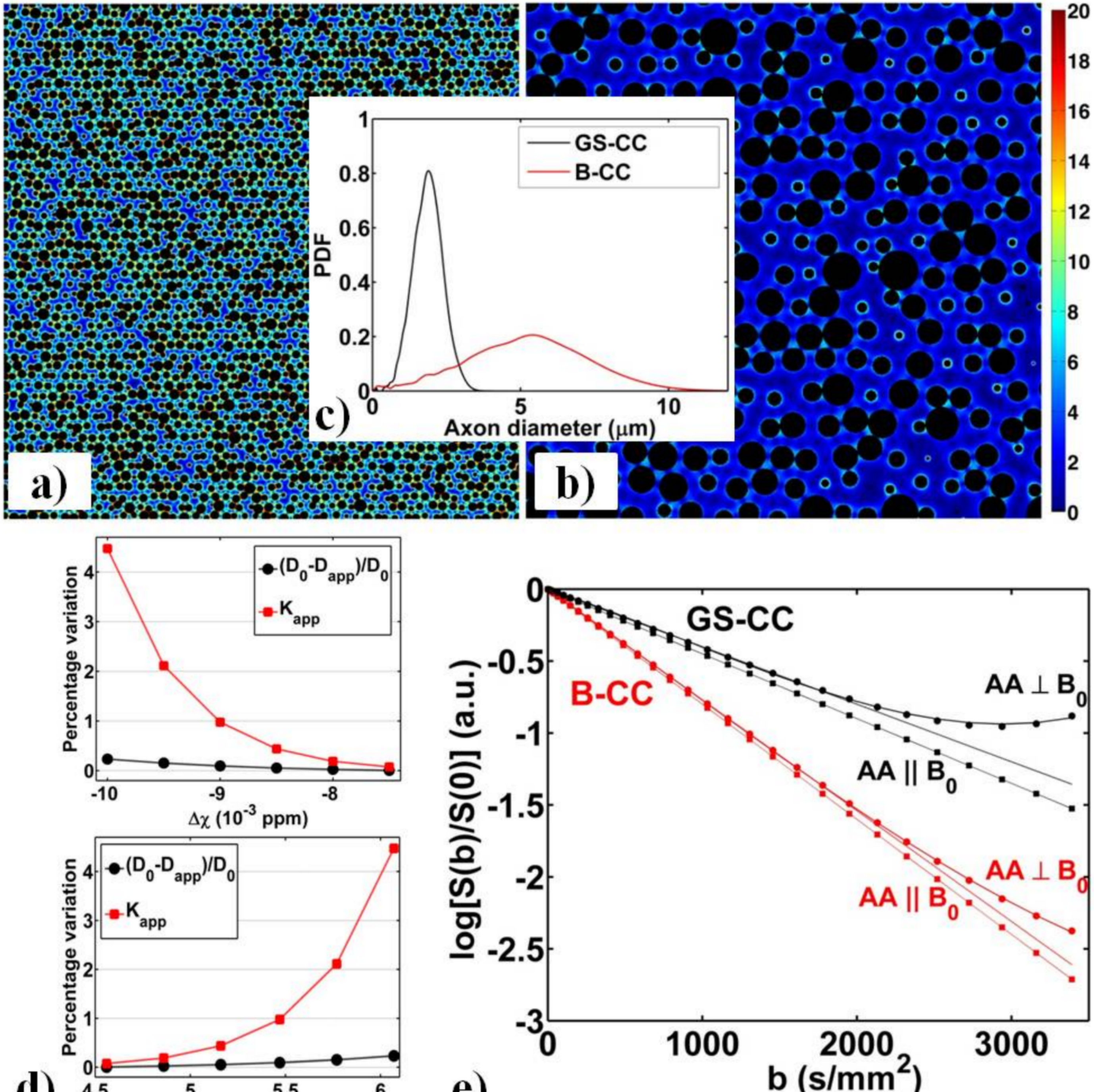
Simulated system: **500μm²** in plane resolution images of different human brain White Matter regions (Genu, Splenium and Body of Corpus Callosum), comprised of about **$2x10^3$ axons**;

Matrix dimension: from **32x32** to **2048x2048**;

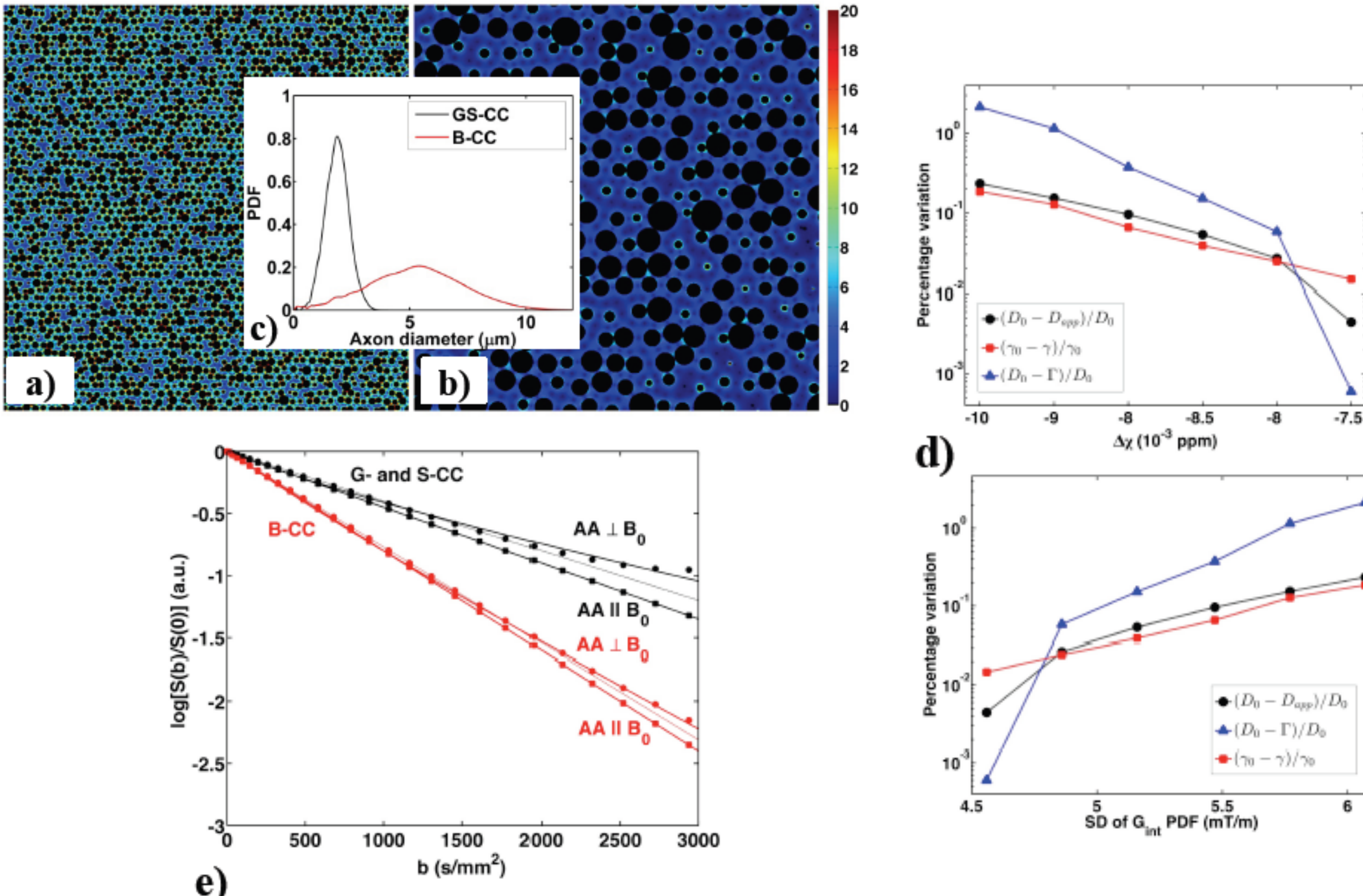
30 values of b: from **0** to **5000 s/mm²**;

Bloch-Torrey equation:

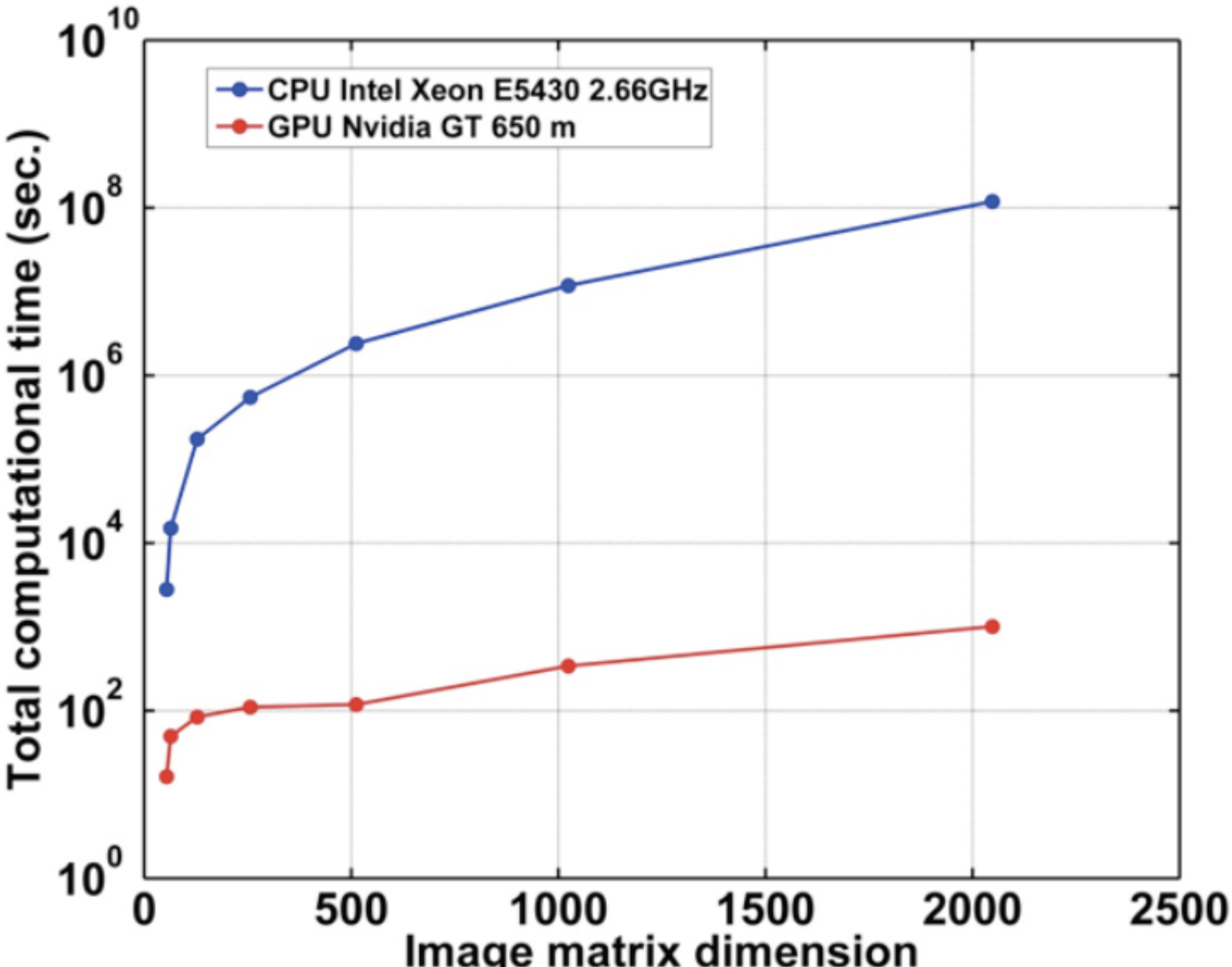
$$\frac{\partial M_{xy}(\overrightarrow{r},t)}{\partial t} = -i\gamma(\overrightarrow{r}\cdot\overrightarrow{G})M_{xy}(\overrightarrow{r},t) + D\nabla^2 M_{xy}(\overrightarrow{r},t)$$



a)
b)
c)
PDF
GS-CC
B-CC
Axon diameter (μm)
d)
Percentage variation
$(D_0-D_{app})/D_0$
K_{app}
$\Delta\chi$ (10^{-3} ppm)
SD of G_{int} PDF (mT/m)
e)
log[S(b)/S(0)] (a.u.)
b (s/mm^2)
GS-CC
B-CC
AA ⊥ B_0
AA || B_0
AA ⊥ B_0
AA || B_0



a)
b)
c)
GS-CC
B-CC
PDF
Axon diameter (μm)
d)
Percentage variation
$(D_0 - D_{app})/D_0$
$(\gamma_0 - \gamma)/\gamma_0$
$(D_0 - \Gamma)/D_0$
Δχ (10^-3 ppm)
SD of G_int PDF (mT/m)
e)
G- and S-CC
B-CC
AA ⊥ B0
AA ll B0
AA ⊥ B0
AA ll B0
log[S(b)/S(0)] (a.u.)
b (s/mm^2)



CPU Intel Xeon E5430 2.66GHz
GPU Nvidia GT 650 m
Total computational time (sec.)
Image matrix dimension
10^{0}
10^{2}
10^{4}
10^{6}
10^{8}
10^{10}
0
500
1000
1500
2000
2500

Preliminary estimation of time gain by using GPU in NMR imaging

- **Typical dimensions** of NMR images are maximum **512x512 pixels**.

- The **estimated gain** in terms of computing time ranges from **10^2** (for images at low resolution) to **10^4** (for 512x512 pixels resolution images).

- In **practical applications** it is necessary to iteratively employ **optimization algorithm based on the minimization of non-linear functions** (from 40 to 400 iterations). The effective gain in terms of computing time is expected to be from **40 to 400 times lower**.

- The **real gain** in computing time for DW-NMR image analysis with non-Gaussian diffusion models is expected to be **20 -100 for 128x128 pixels resolution images**.

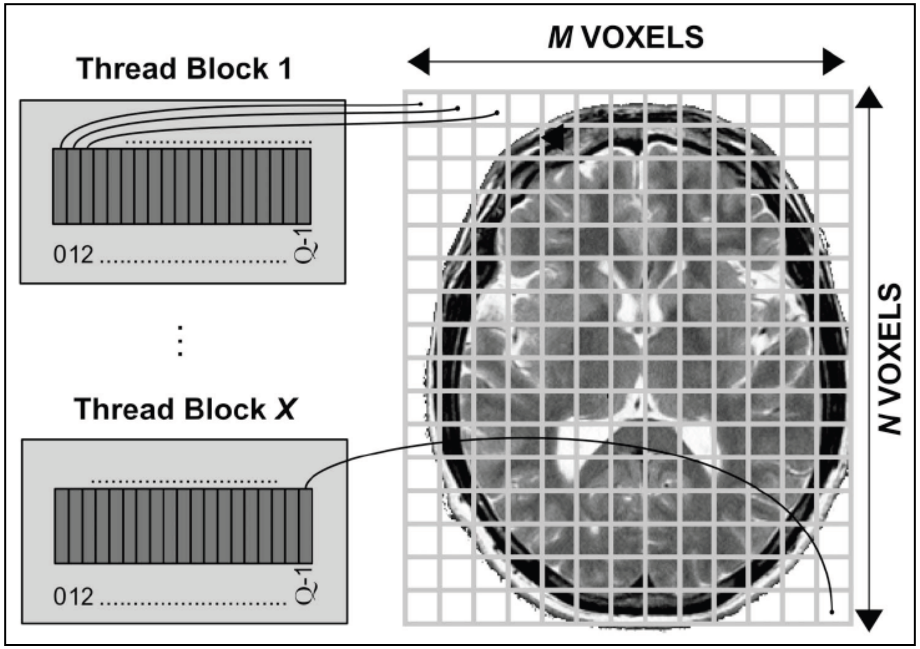
Non-Gaussian diffusion model based DW-MRI reconstruction by GPU

- The Levenberg-Marquardt algorithm is based on an iterative numerical optimization procedure that minimizes the sum of squared model residuals.

- The CUDA kernel performs the Levenberg-Marquardt algorithm. This kernel maps each CUDA thread to a voxel, and it launches as many threads as voxels contained in a particular slice.

- Because processing of different voxels is totally independent, the threads do not need to synchronize.

- It is noteworthy that each thread must compute all steps of the Levenberg-Marquardt algorithm using large intermediate structures (the size of the structures depends on the number of parameters to fit and the number of diffusion-sensitising gradient directions K of the input dataset).



Next Steps to achieve our goals

*(i) Fast-implementation: using of available Matlab custom script (**3 months**).*

1. Implementation of a parallelized version of the Levenberg-Marquadt algorithm for usage on GPU by means of the Matlab Parallel Toolbox;

2. Optimization of available Matlab custom script for DKI and SEM analysis for usage on GPU by means of the Matlab Parallel Toolbox.

*(ii) Slow-implementation: translating available Matlab custom script in CUDA language (**6 months**).*

1. Implementation of a parallelized version of the Levenberg-Marquadt algorithm for usage on GPU by means of CUDA platform;

2. Translation of available Matlab custom script for DKI and SEM analysis for usage on GPU by CUDA platform.

*(iii) Implementation of numerical simulations supporting experimental data analysis: Monte Carlo and Finite Elements Method (FEM) on GPU (**3 months**)*

1. Translation of parallelized version of Monte Carlo and FEM algorithms to numerically simulate DW-MRI signal attenuation for usage on GPU by CUDA platform.

People involved

- ❑ **Dr. Silvia Capuani** (Physics Dep. Sapienza University of Rome and CNR-IPCF UOS Rome): involved in **all topics**;

- ❑ **Dr. Marco Palombo** (Physics Dep. Sapienza University of Rome and CNR-IPCF UOS Rome): involved in **all topics**;

- ❑ **PhD Student in Biophysics Stella Caporale** (Physics Dep. Sapienza University of Rome): involved in **DW-NMR image reconstruction by means of non-Gaussian diffusion models using GPU**;

- ❑ **Graduating Student in Physics Ruggero Lo Sardo** (Physics Dep. Sapienza University of Rome): involved in **NMR Numerical Simulation on GPU**.