2019–12–12

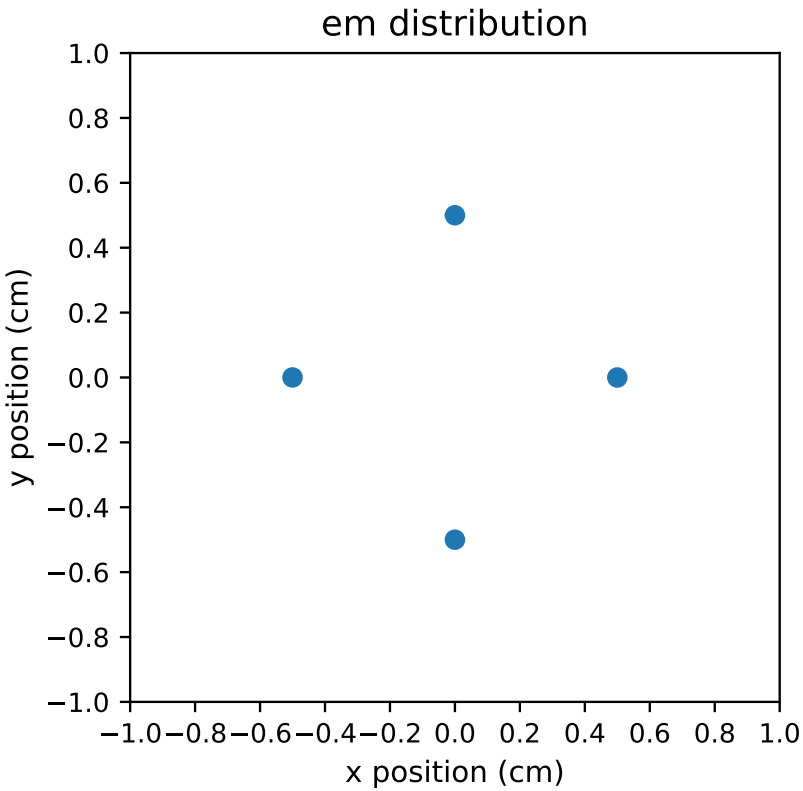
- I wrote a script to show that we can simulate the ems in parallel and do the reconstruction afterwards.
- I tested the execution time of the 2DFT simulation.
- I tested the execution time of numpy's random number generator.
- I kept the divorce between the pulse sequence sampling period and the simulation time step but now apply rotation at every simulation time step.
- I implemented metabolic conversion.

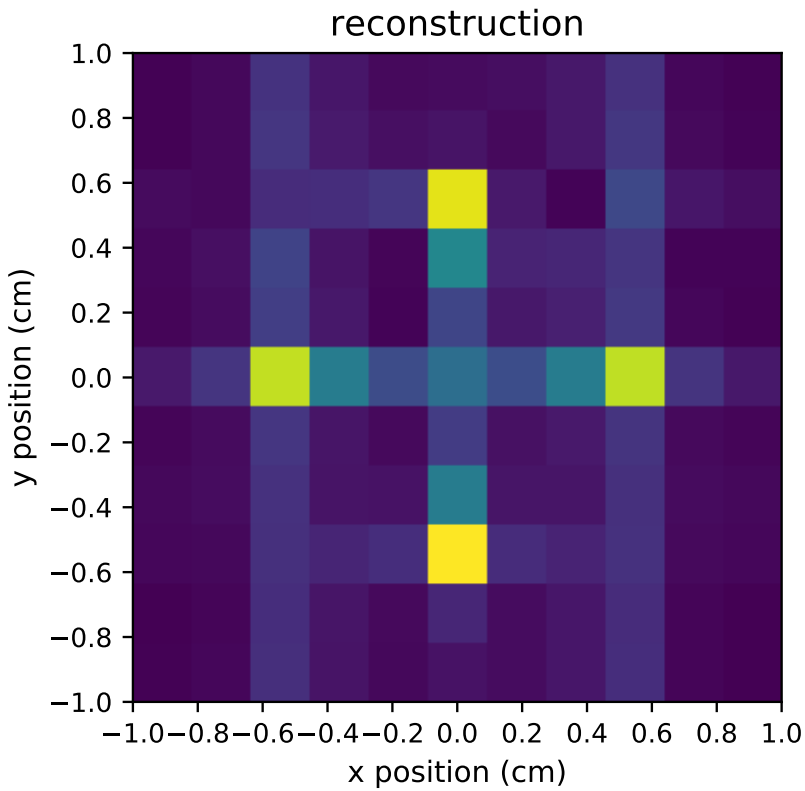
Simulating ems in parallel

Summary

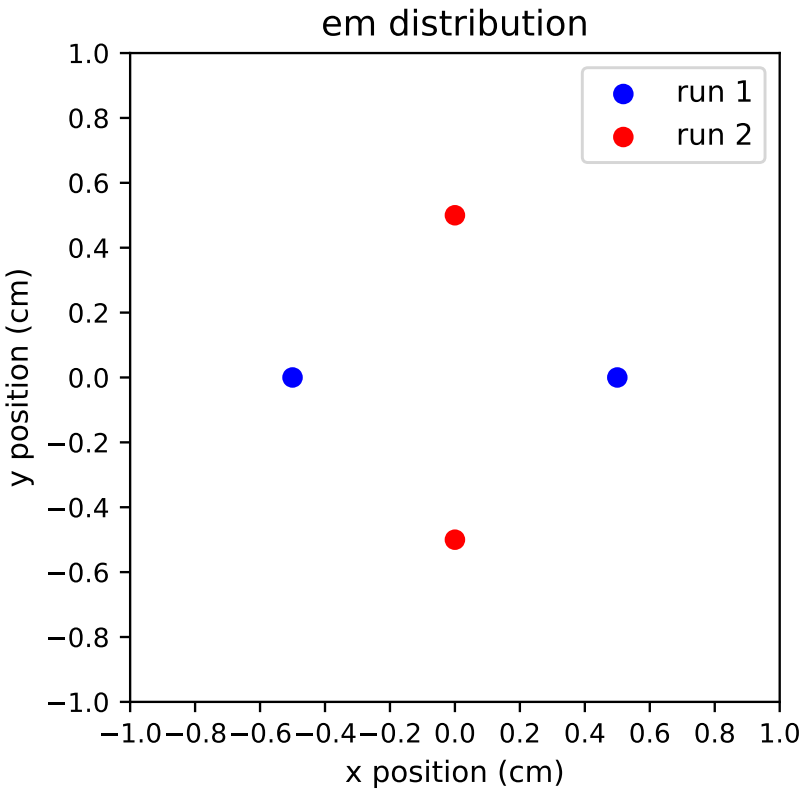
- I simulated 2DFT+reconstruction with four ems using the script of 2019-12-11.
- I ran the same simulation with two of the four ems and stored the MR signals. I repeated the simulation with the other two ems and stored the MR signals. I summed the MR signals and did the reconstruction.

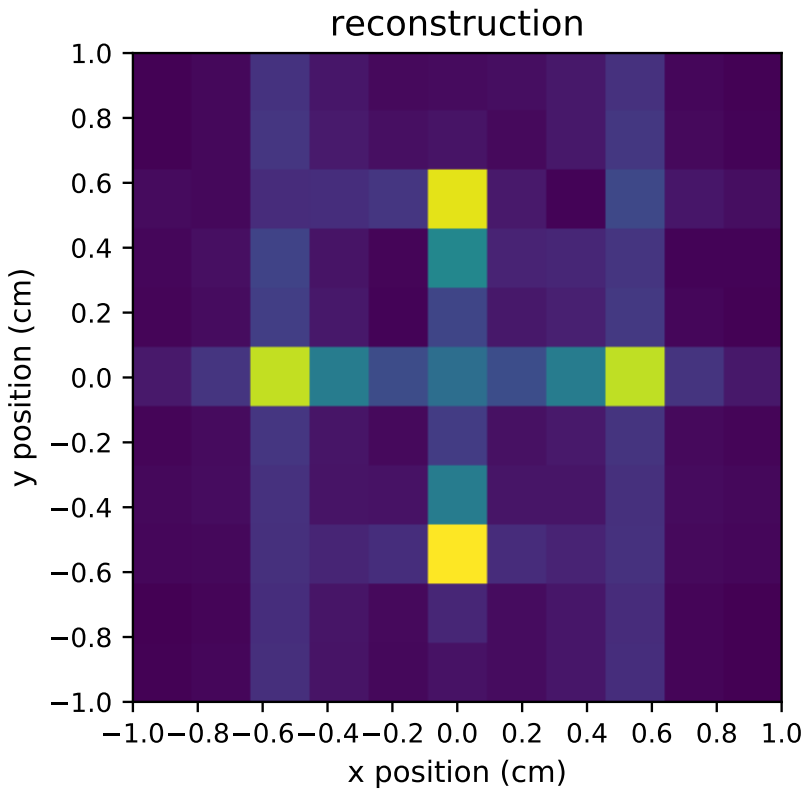
4 ems in one run





2 ems in run 1, 2 ems in run 2





Comments

- The reconstructions are identical.
- We can parallelize the simulation easily. A central computer partitions the ems across the cluster. Each computer simulates its own set of ems and stores the MR signals. Once the simulation is complete, each computer sends the MR signals to the central computer. The central computer does the reconstruction.

Execution time of 2DFT simulation

Summary

- I tested the execution time of the 2DFT simulation using one em with 9 TRs of length 111 ms (1 s pulse sequence length) and a time step of 1 us (implying 10^6 time steps).
- The total execution time was 5.41 seconds.
- The execution time is 5.41 us per em per time step.

Execution time of random number generation with Numpy

Summary

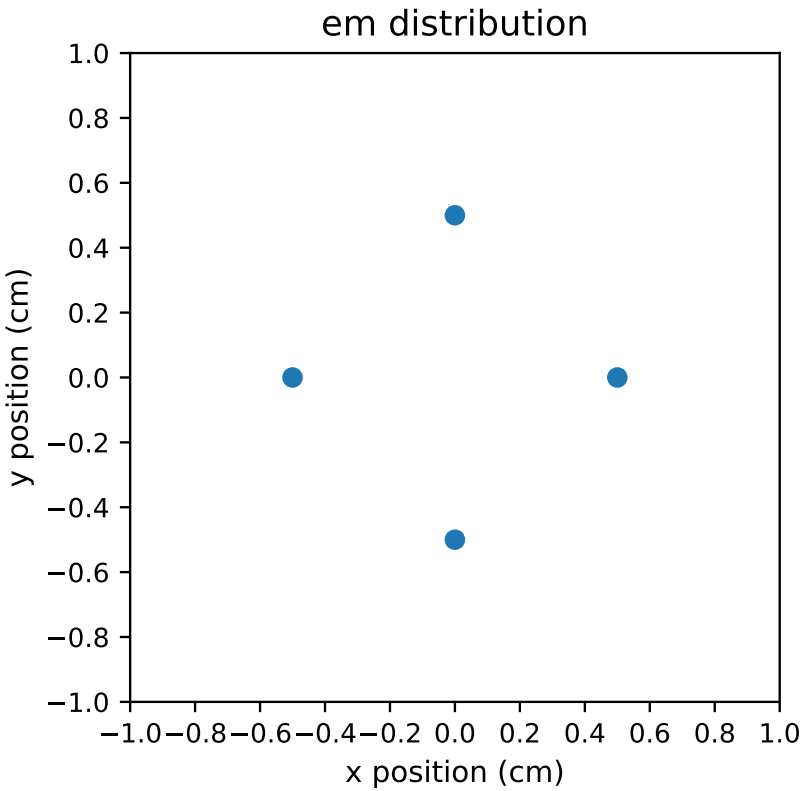
- Using the command `numpy.random.random()` to generate a random number between 0.0 and 1.0, I found it took 0.346 seconds to generate 1 million random numbers.
- This means 0.346 microseconds per random number.
- This would increase the execution time from 5.41 us to 5.76 us per em per time step (6.5% increase in execution time).

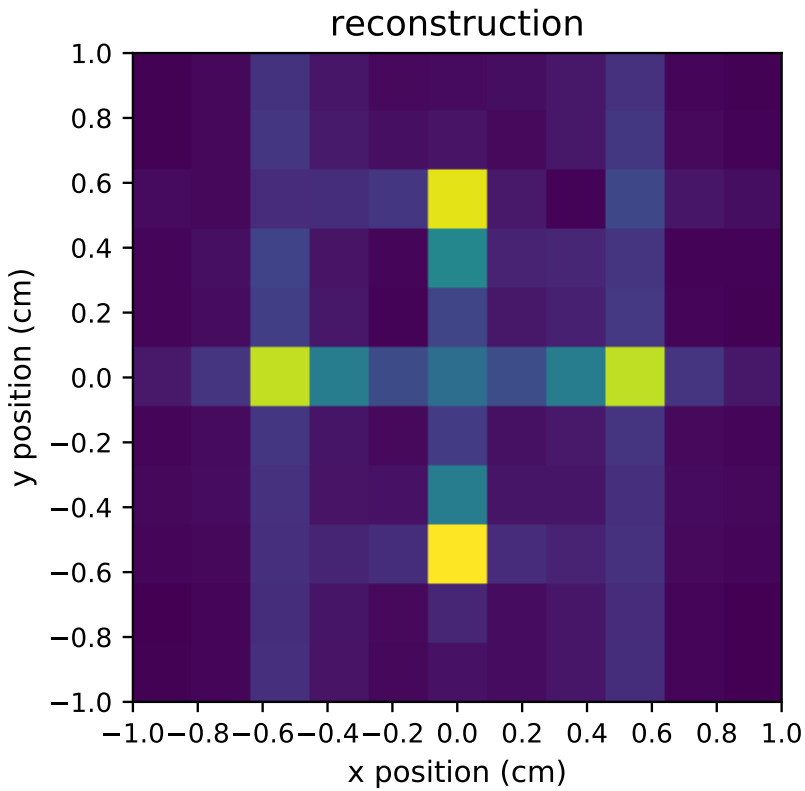
Metabolic conversion

Summary

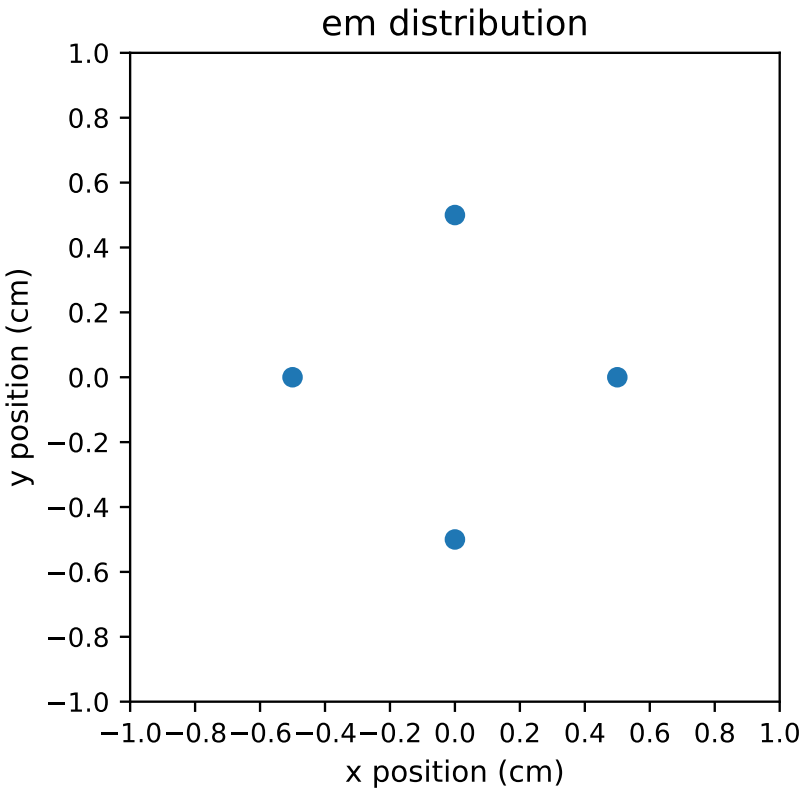
- I implemented metabolic conversion in the ems.
- At each simulation time step for each em, a random number in $[0.0, 1.0)$ is generated. If this number is less than the conversion probability for the duration of the time step (e.g. 5% per second $\Rightarrow 5 \times 10^{-6}$% per microsecond), then the shielding constant of the em is updated.

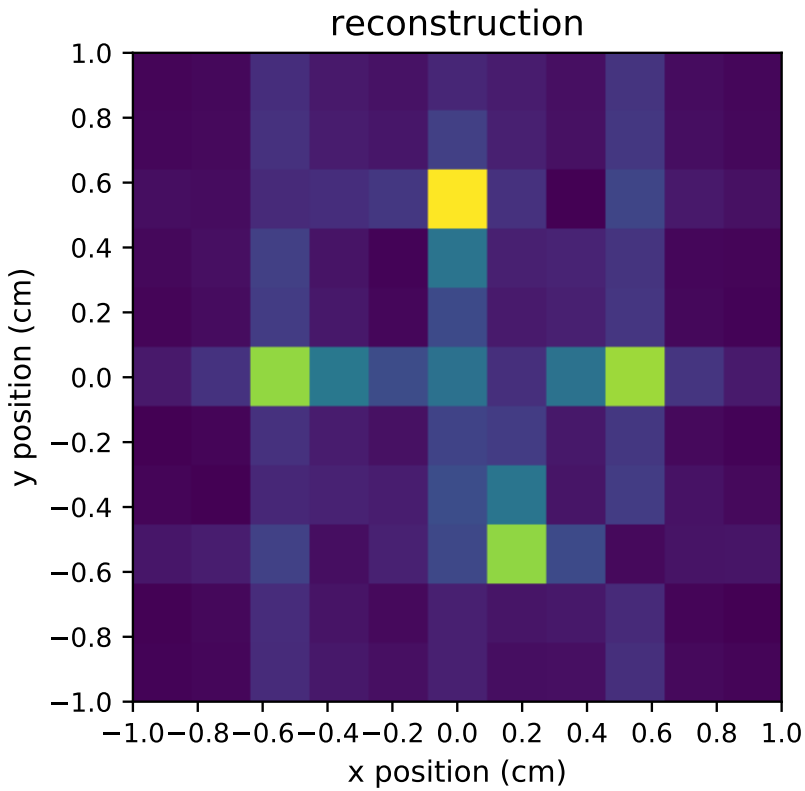
Metabolic conversion off





Metabolic conversion on, rate = 20% per second, chemical shift = 10 ppm





Comments

- The change in resonance frequency distorts the image.