

```
#pragma omp parallel for num_threads(NT)
for (i=imin;i<imax;i++) coll[i-imin] = dt*(coll_term_f (i,I, F,g));
if (mpi_rank > 0) {
  MPI_Send(coll,N1,MPI_DOUBLE,0,0,MPI_COMM_WORLD)
} else {
  while (count < mpi_size) {
    MPI_Recv(tmp,N1,MPI_DOUBLE,MPI_ANY_SOURCE,0,MPI_COMM_WORLD,&mpi_status);
    sender = mpi_status.MPI_SOURCE;
    count++;
  }
}
```

Introduction to parallel programming (for physicists)

FRANÇOIS GÉLIS & GRÉGOIRE MISGUICH, IPhT courses, June 2019.



université
PARIS-SACLAY



IPhT, CEA-Saclay
Itzykson room
courses.ipht.fr

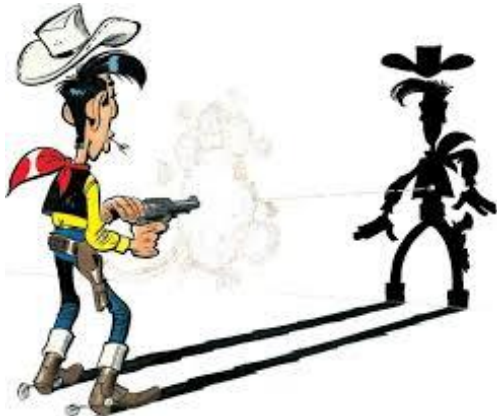
1. Introduction & hardware aspects (FG)
2. A few words about Maple & Mathematica
3. Linear algebra libraries
4. Fast Fourier transform
5. Python Multiprocessing
6. OpenMP
7. MPI (FG)
8. MPI+OpenMP (FG)

These slides (GM)

(discrete) Fourier transform

- FFTW: a high performance implementation (in C)
Fastest Fourier transform *in the West*
- <http://www.fftw.org/>
- Multi-threaded
- portable
- Open source (GPL license)
- Can be used from Python (pyFFTW)

FFTW



(discrete) Fourier transform example using multi-threaded FFTW v3

```
#include <complex.h>
#include <fftw3.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

#define L (1<<25)
#define NT 2

void main(void) {

    short unsigned int seed[3];
    fftw_complex * A;
    seed[0] = 123; seed[1] = 456;
    seed[2] = 789; seed48(seed);

    A =
(fftw_complex*)fftw_malloc(L*sizeof(fftw_complex));
    int i;
    for (i=0;i<L;i++) A[i] =
(fftw_complex) (drand48()+I*drand48());

    fftw_init_threads();

    fftw_complex *in =
(fftw_complex*)fftw_malloc(L*sizeof(fftw_complex));
    fftw_complex *out =
(fftw_complex*)fftw_malloc(L*sizeof(fftw_complex));

    fftw_plan_with_nthreads(NT);
    fftw_plan FP =
fftw_plan_dft_1d(L,in,out,1,FFTW_ESTIMATE);
    fftw_free(in);
    fftw_free(out);

    fftw_complex *fft = (fftw_complex
*)fftw_malloc(L*sizeof(fftw_complex));

    fftw_execute_dft(FP,A,fft);
}
```

Num. of points

Num of threads

Allocate the input array

Random input

Actual FFT

Output

Workspaces, and fft « plan »

Fourier transform

Speed-up with threads using FFTW3

