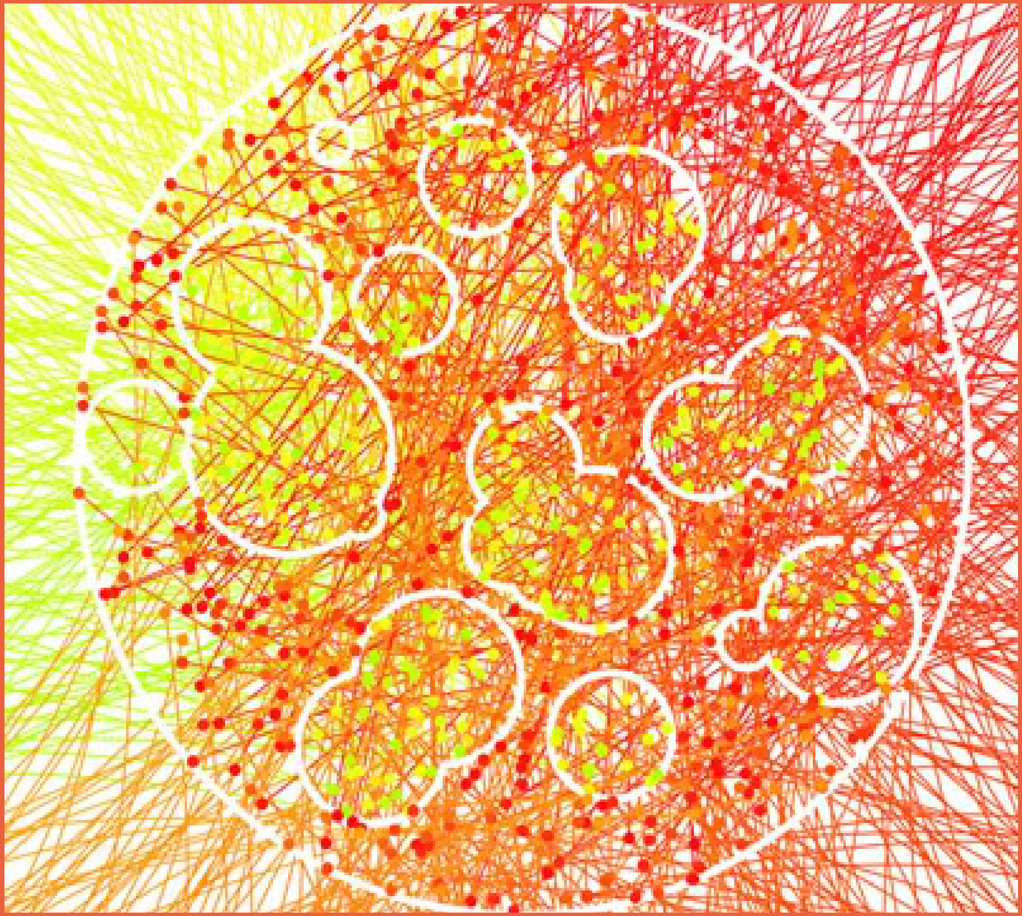
MONTE CARLO METHODS



Daniel Guirado Rodríguez
CCD 09/12/2009

STOCHASTIC ALGORITHMS

PROBLEM = GET OUTPUT FROM INPUT THROUGH AN ALGORITHM

Deterministic algorithm:

$\text{Input}_1 \rightarrow \text{Output}_1$

Stochastic algorithm:

Input_1 + random decision $\rightarrow$ Output_1, Output_2, Output_3, ...

Example of probabilistic methods:

- ***Monte Carlo***: the given results has a probability p<1 to be the correct solution.
- ***Las Vegas***: gives the correct solution or informs that it could not be obtained.

STOCHASTIC METHODS WITH COMPUTERS? HOW?

Stochastic method = generating random number + performing operation

$$N \rightarrow \infty \Rightarrow t \rightarrow \infty$$

Computers make operations much faster.
Computer are deterministic, no random numbers.
Solution: *pseudo-random numbers*.

PSEUDO-RANDOM NUMBERS

Finite (but large) lists of numbers generated by an algorithm.
No correlations beteween the numbers.

Examples:

- *Numerical Recipes*.
- *MKL* (optimized for *iFort*)
- http://www.psychicscience.org/random.aspx

ran2 (Numerical Recipes)

- For any *Fortran* or *C* compiler.
- The authors offer $1000 to the first person who finds a statistical test that proves that *ran2* fails.

MONTE CARLO METHODS

Definition:

Iterative method including random decisions that gives the correct solution of a problem with a certain probability <1.

Named after the district of Monte Carlo (Monaco), European gambling capital.

Monte Carlo

Monte Carlo casino

Origin:

Study of the diffusion of neutrons in a fusion experiment (Los Álamos Laboratory).

TYPICAL MONTE CARLO METHOD

Input parameters + ξ_i $\Rightarrow$ x_i

$$X = \frac{1}{N}\sum_{i=1}^{N} x_i$$

CENTRAL LIMIT THEOREM

$\rho(x) =$ Any probability density function with average μ and variance σ^2 .
$x_1, x_2, \ldots, x_N =$ collection of N values sampled through ρ .

Then, the probability density distribution $\rho_N(X)$, which elements are
$X_N = \frac{1}{N}\sum_{i=1}^{N} x_i$, has a average of μ and a variance of $\frac{\sigma^2}{N}$.

LAW OF LARGE NUMBERS

$$N \rightarrow \infty \Rightarrow \rho_N \rightarrow \text{gaussian.}$$

HOW TO CALCULATE THE ERROR IN A MONTE CARLO METHOD

$$(N \to \infty)$$

- Fit all calculated x_i to a gaussian distribution and obtain σ. Let us call the correct solution of the problem μ. Then:

- $X \in \left[\mu - \frac{\sigma}{\sqrt{N}}, \mu + \frac{\sigma}{\sqrt{N}}\right]$ with $p = 0.682$.
- $X \in \left[\mu - 2\frac{\sigma}{\sqrt{N}}, \mu + 2\frac{\sigma}{\sqrt{N}}\right]$ with $p =$ ٠.٩٥٤.
- $X \in \left[\mu - 3\frac{\sigma}{\sqrt{N}}, \mu + 3\frac{\sigma}{\sqrt{N}}\right]$ with $p = 0.998$.
- If the accuracy is not enough, increase N (the error decreases as $\sim \frac{1}{\sqrt{N}}$).

RESULTS FROM MONTE CARLO SIMULATIONS HAVE ERRORS!

MONTE CARLO EXAMPLE I: CALCULATING THE TV SHARE

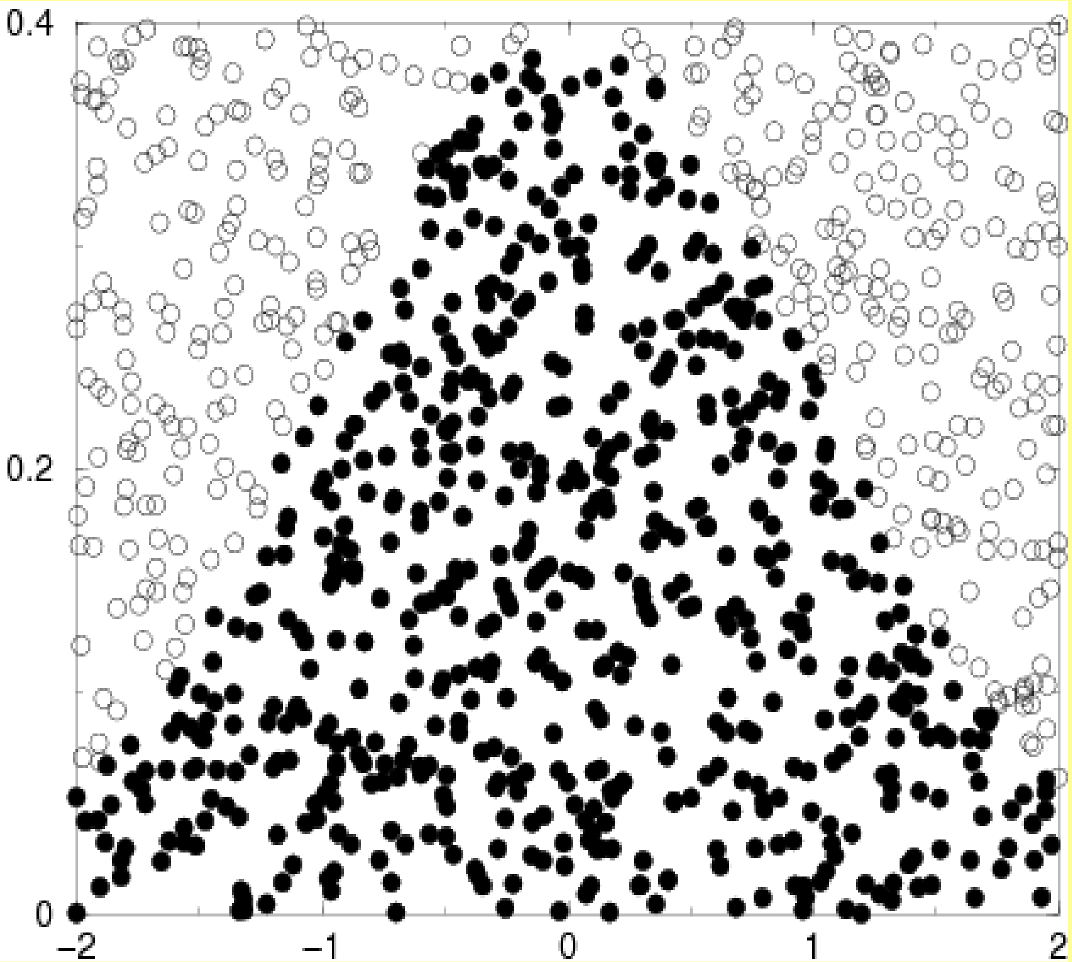
Survey among a small group of people.
x_i = percentage of time that a certain TV channed is watched by a person.

$$X = \frac{1}{N}\sum_{i=1}^{N} x_i =$$ Monte Carlo estimation of the Share.

The number of persons to participate in the survey is deduced by calculating σ and assuming a tolerable error $\frac{\sigma}{\sqrt{N}}$.

MONTE CARLO EXAMPLE II: PLAYING GUESS WHO

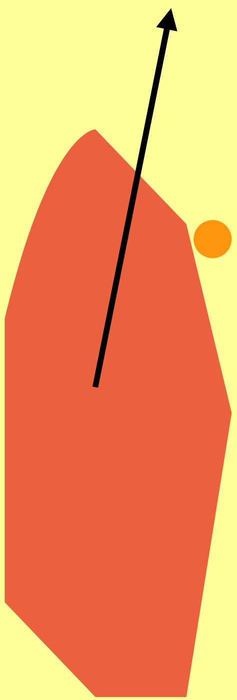
MONTE CARLO EXAMPLE III: INTEGRATION



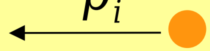
S

A number of points are uniformly randomly distributed over an area *S*. Then, the area under the curve is obtained by multiplying *S* by the number of points under the curve.

MONTE CARLO EXAMPLE III: DUST GRAIN ALIGNMENT BY THE SOLAR WIND



Completely
inelastic
collision



$\vec{p}_i$

electrons
protons,He+

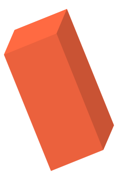
We assume that the crystalline structure of the grain absorbs the energy of the collision, so that the collision is completely inelastic.

$$\vec{L}_i = m\,\vec{r}_i \times \vec{p}_i$$

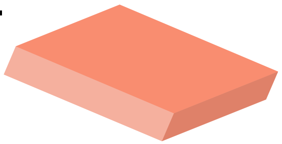
$$\vec{L} = \frac{1}{N}\sum_{i=1}^{N} \vec{L}_i$$

Grains for testing:

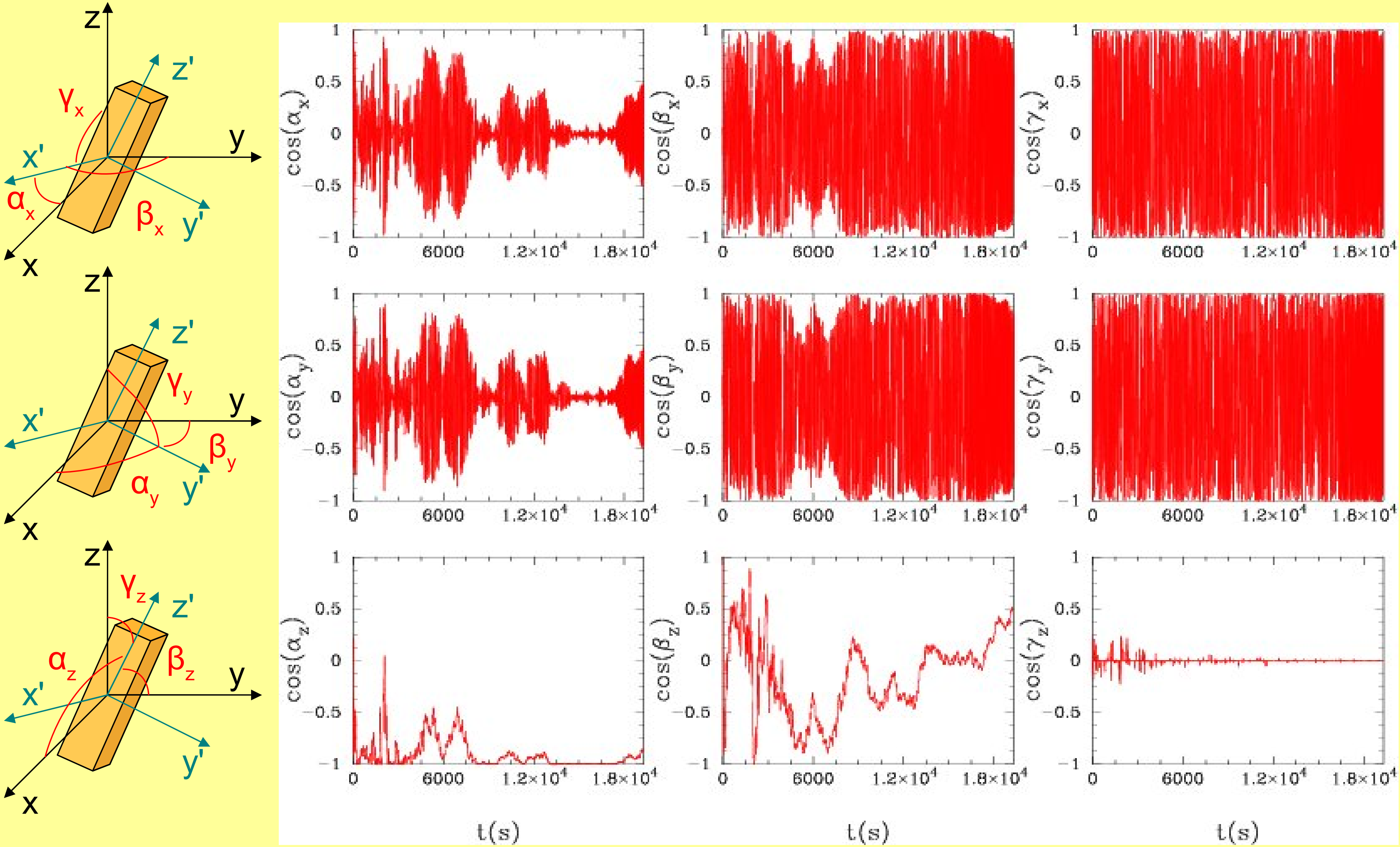
- Rectangular prisms
- Density=3 $g\ cm^{-3}$
- $0.1\,\mu m \times 0.1\,\mu m \times 0.2\,\mu m$
- (A) Prolate 1:1:2



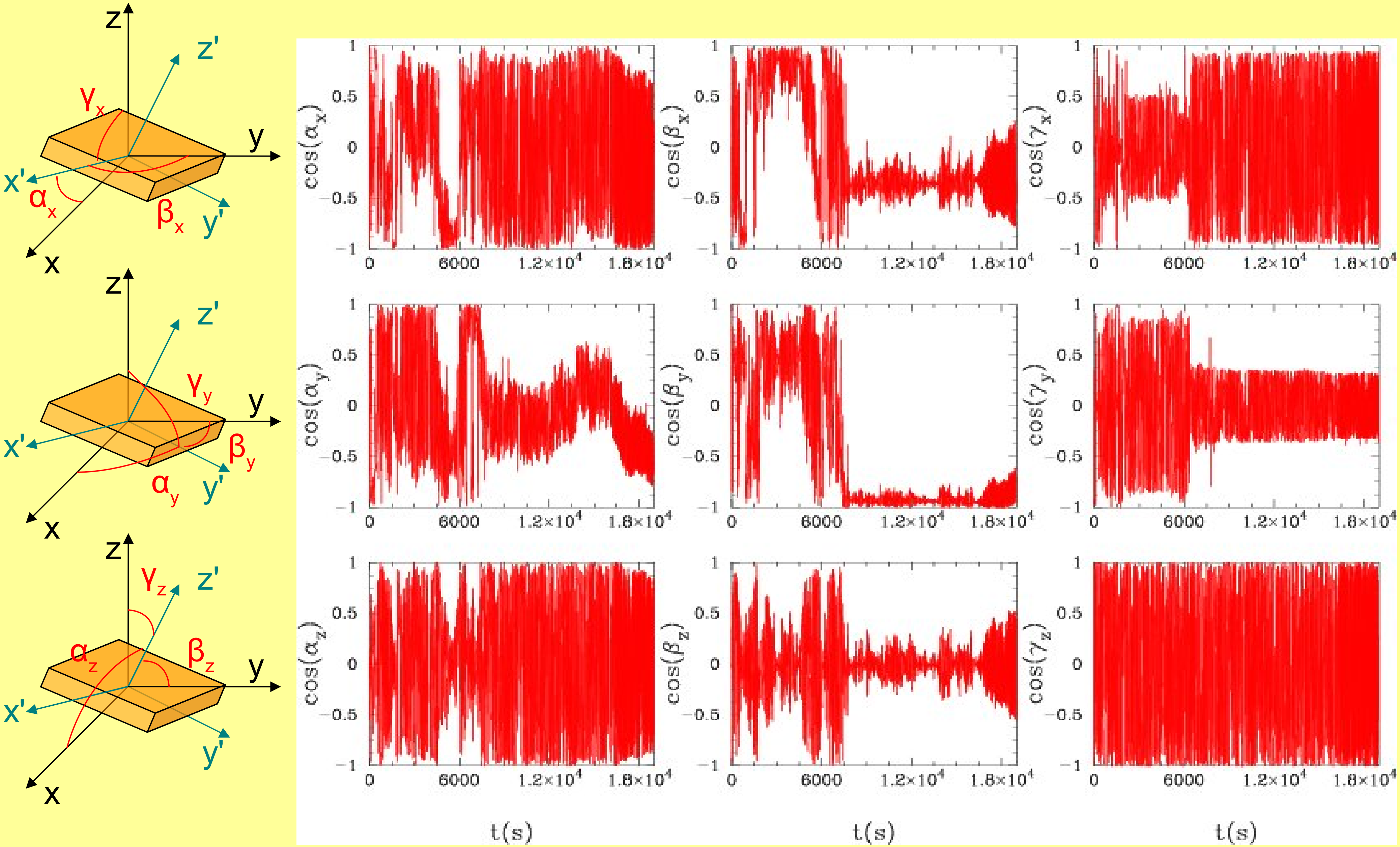
- (B) Oblate 2:2:1
- Same volume as the prolate.



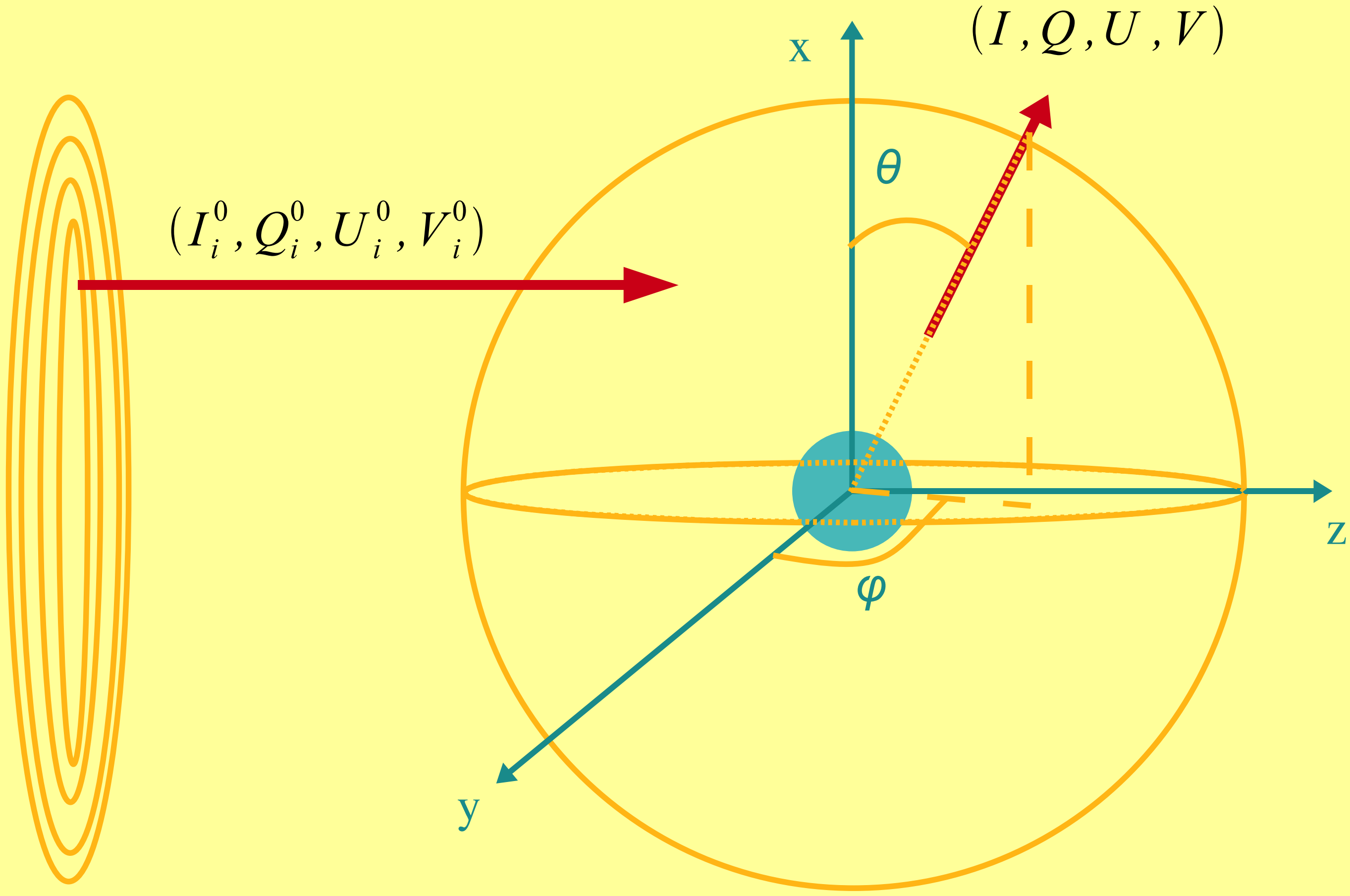
ALIGNMENT OF PROLATE GRAINS



ALIGNMENT OF OBLATE GRAINS

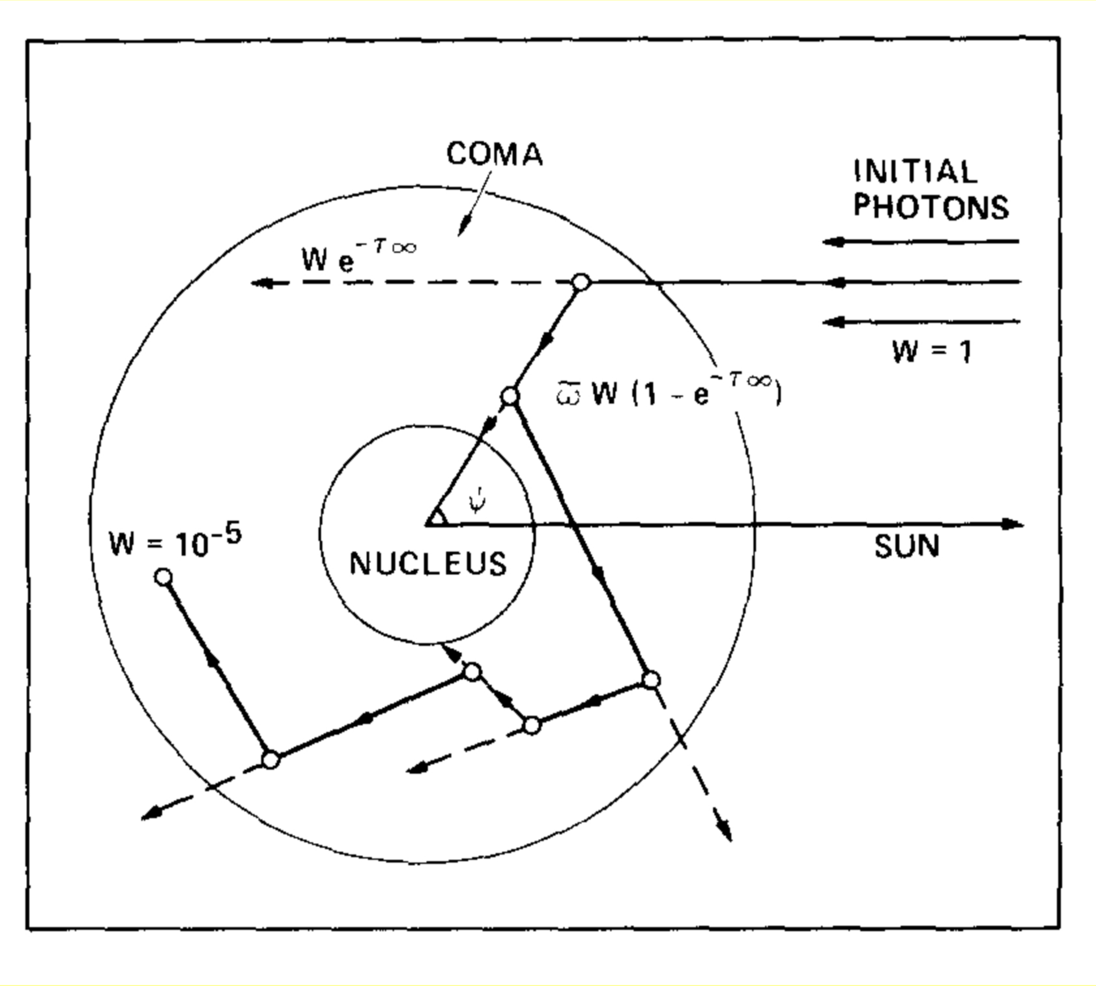


MONTE CARLO EXAMPLE IV: RADIATIVE TRANSFER IN A COMET



$$(I, Q, U, V) = \frac{1}{N} \sum_{i=1}^{N} (I_i^0, Q_i^{\cdot}, U_i^0, V_i^{\cdot})$$

DESCRIPTION OF THE RADIATIVE TRANSFER MODEL



Particles:
- Any size distribution.
- Any refractive index (even optically activity).
- Any geometry.
- Any orientation (aligned or not).

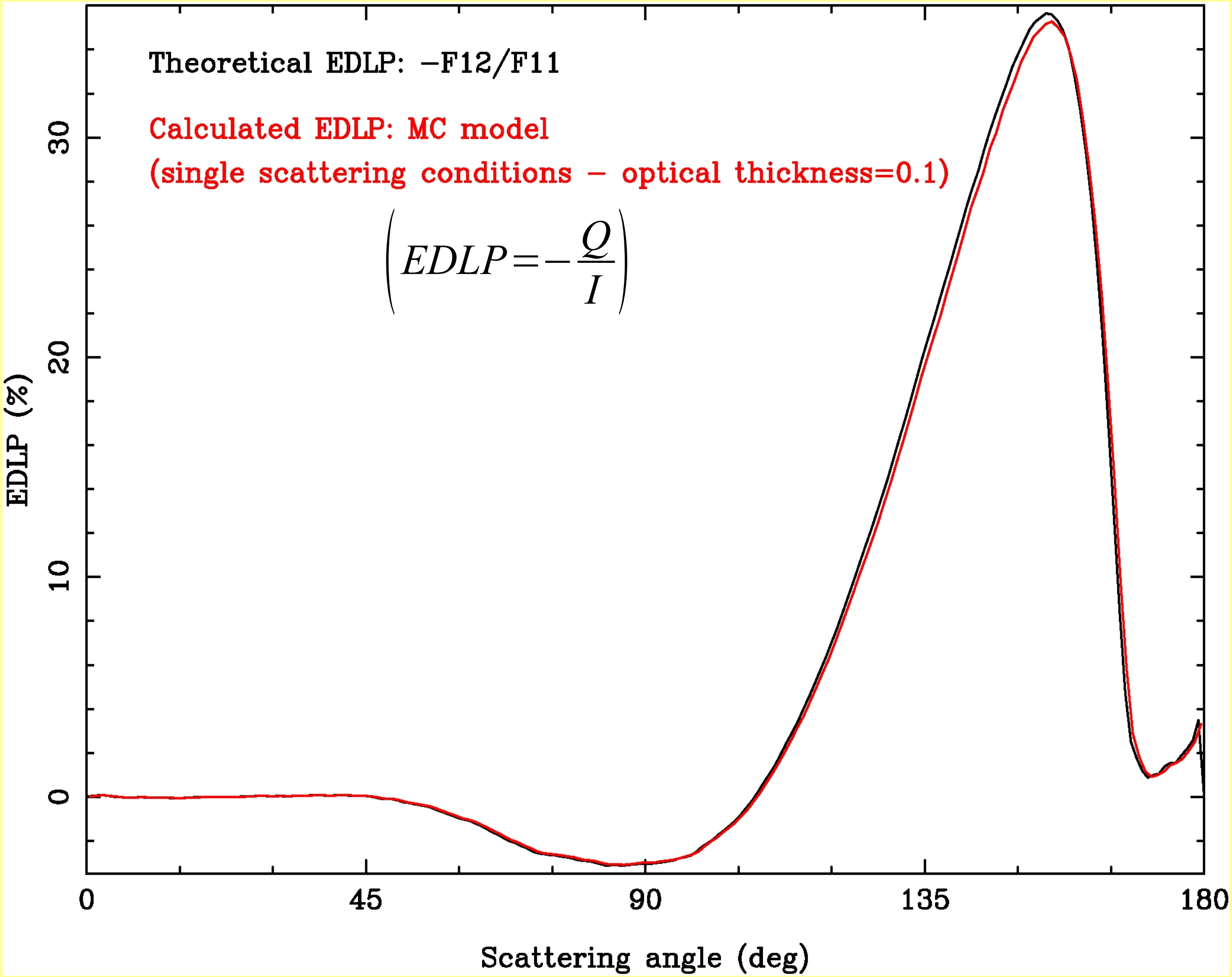
Any optical thickness of the coma (single or multiple scattering).

Inhomogeneous coma.

Packet of photons (W)
- Improves statistics.
- Reduces computation time.

W=1 at the beginning
Ends when W<Wmin

EXACTITUDE OF THE MODEL



For 10^8 packets of photons (~4 hours/core).

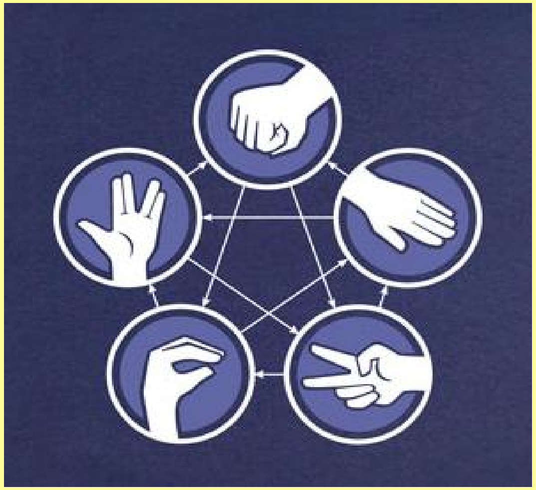
LAS VEGAS ALGORITHM

Definition:

Iterative method including random decisions that may either give the correct solution of a problem or inform that the solution has been not obtained.

RESULTS COMING FORM LAS VEGAS ALGORITHM DO NOT HAVE ERRORS

LAS VEGAS EXAMPLE I: ROCK, PAPER, SCISSORS,LIZARD, SPOCK

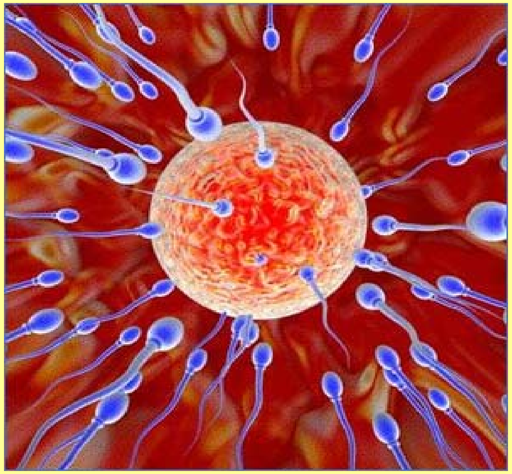


LAS VEGAS EXAMPLE II: LOTTERY

LAS VEGAS EXAMPLE IV: SOLVING EQUATIONS

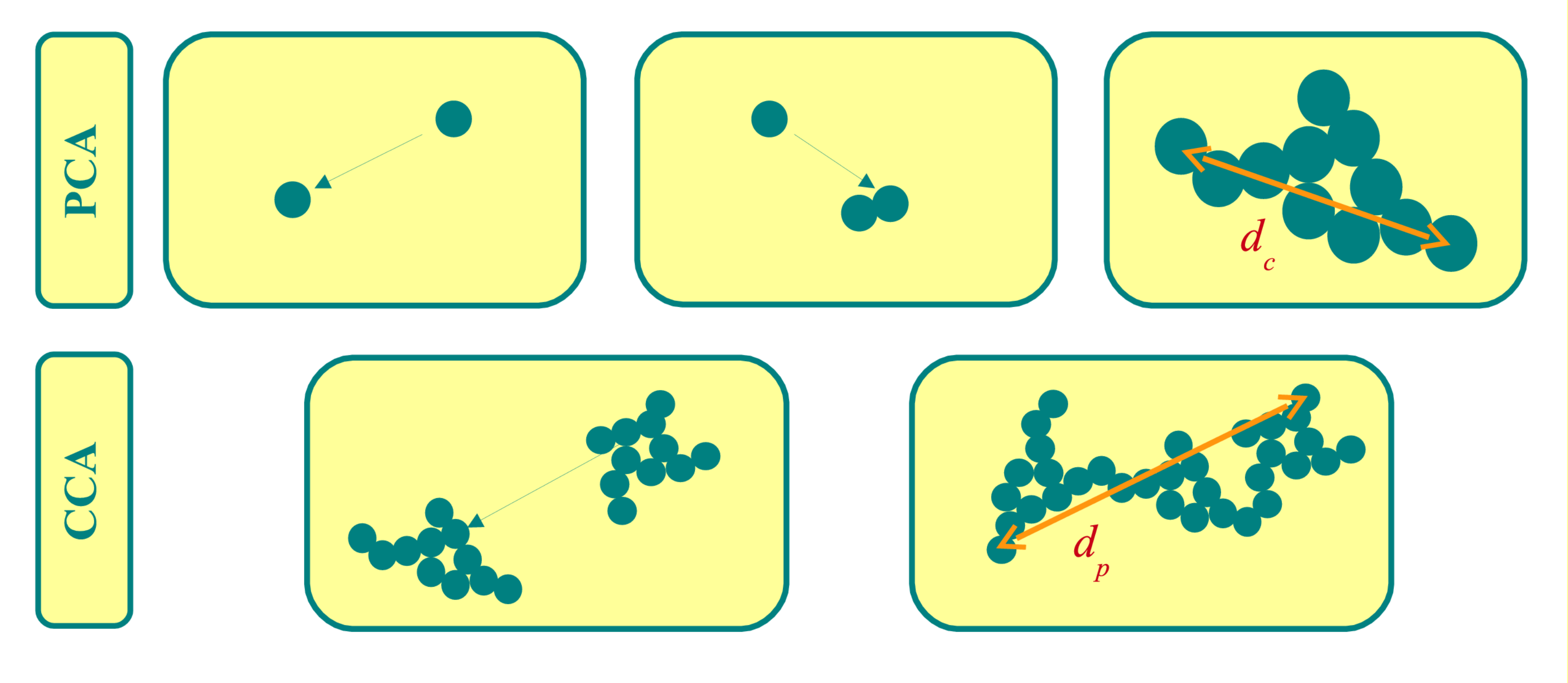
By randomly exploring around the zone of parameters where we know that the solution is.

LAS VEGAS EXAMPLE III: SINGLE FECUNDATION



LAS VEGAS EXAMPLE V: OBTAINING A PASSWORD

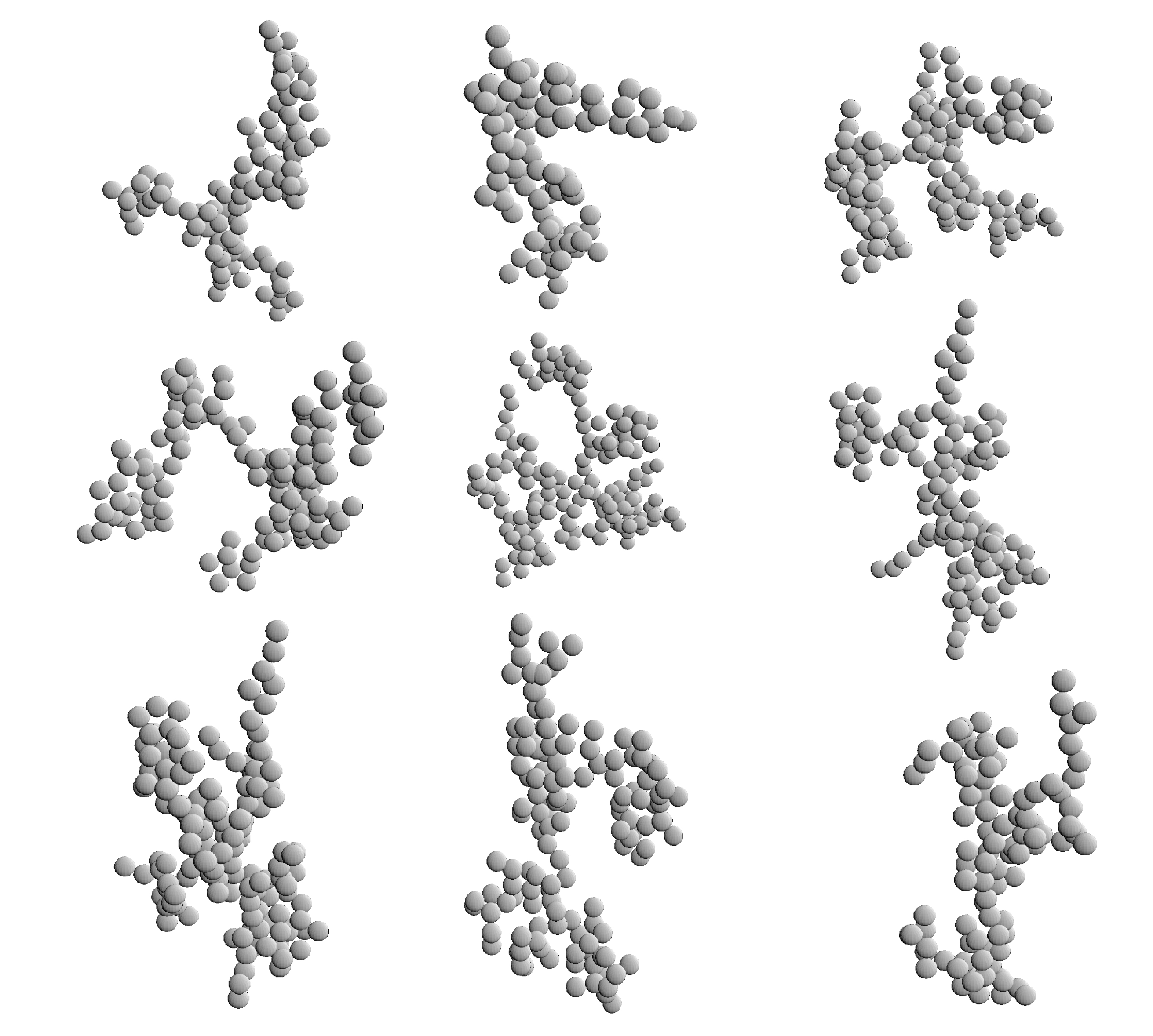
LAS VEGAS EXAMPLE Vi: BUILDING RANDOM AGGREGATES OF SPHERES



Arbitrary maximum size imposed to clusters (d_c) and the whole particle (d_p).

Maximum size ⟺ maximum distance between two monomers (realistic criterium).

AGGREGATES OF SPHERES



A solution to the problem is always achieved in this case.

MONTE CARLO COMPARED TO LAS VEGAS

- Monte Carlo methods always give a solution, along with a certain probability for this solution to be correct. Las Vegas either gives a correct solution to the problem or informs that no solution could be achieved.
- In problems solved by Monte Carlo methods, there is a unique solution, and the method converges to it while increasing the number *N* of iterations. Several correct solutions may exist for problems solved with Las Vegas algorithms.

SOME COMMENTS

Monte Carlo and Las Vegas codes requiere a large amount of memory and computational time to acomplish their tasks. However, Monte Carlo methods are able to exactly reproduce the microscopical elements of a complex system and reproduce their macroscopical properties. Personal computers have become fast enough for seriously considering this option when modelling a complex system.

PARALLELIZE YOUR CODE

1 sequential program runs in 1 only core.

As Monte Carlo and Las Vegas codes basicly consist of a large *do* loop with a number *N* of independent iterations, two solutions to accelerate the calculations by one of these codes in a multicore computer could be:

- Run the Monte Carlo code for *N/m* iterations, where *m* is the number of cores, and make a summation of the results.
- Parallelize the code.

Classical method: MPI. New easier alternative: ***OpenMP*** (Fortran and C).

How it works:

```
      PROGRAM OMP_SUM2
      INTEGER NMAX
      PARAMETER(NMAX=20000)
      INTEGER I
      REAL A(NMAX),C(NMAX)
      DO I = 1,NMAX
        A(I) = I * 1.0
        C(I)=1.
      ENDDO
C$OMP PARALLEL shared(A,B,C,NMAX) private(I,J)
C$OMP DO
      DO I = 1,NMAX
         DO J=1,I
            C(I)= C(I)+A(J)
         END DO
      ENDDO
C$OMP END DO
C$OMP END PARALLEL
      WRITE(*,*)C(NMAX)
      END
```

gfortran omp_sum2.f -o omp_sum2

↓

Sequential omp_sum2

gfortran -fopenmp omp_sum2.f -o omp_sum2

↓

Parallel omp_sum2

CONCLUSIONS

- Monte Carlo & Las Vegas are two stochastic methods for the solution of problems.

- Monte Carlo gives a solution along with a certain probability of the solution to be correct.

- Las Vegas either gives the correct solution or informes that no solution could be obtained.

- Monte Carlo methods deal with problems with one single solution, and it reduces its error while the number of iterations increases.

- Monte Carlo codes are usefull to simulate the microscopical elements of a complex system, and derive the macroscopical properties of the system.

- Monte Carlo and Las Vegas codes require a large ammount of computational resoruces: time and memory.

- Monte Carlo and Las Vegas codes may run faster in a multicore computer by adding the OpenMP directives to the programs and using the corresponding flags when compiling.