Laboratoire Kastler Brossel

Physique quantique et applications

Ion cloud dynamics simulations

Laurent Hilico

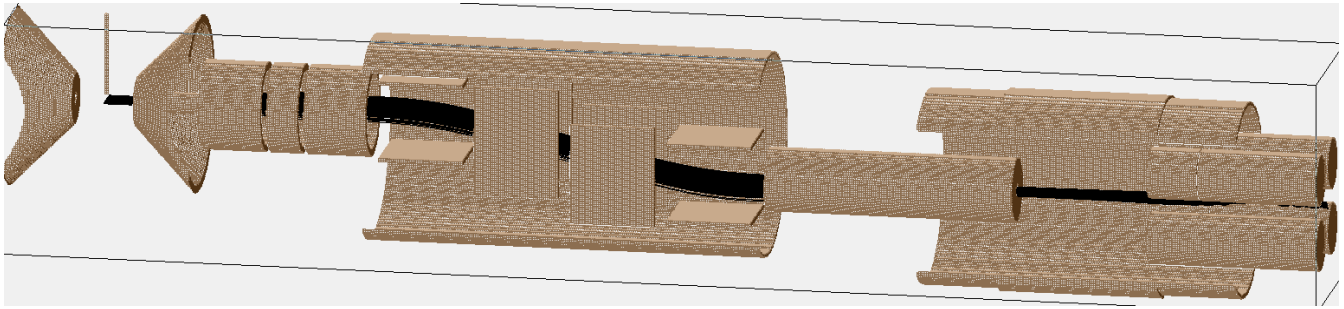
ÉCOLE DE PHYSIQUE
des HOUCHES

- ✓ Introduction and general scheme
- ✓ Orders of magnitude
- ✓ Computer requirements
- ✓ Integration algorithms
- ✓ Coulomb force parallelisation
- ✓ Laser interaction, Collisions, Chemical reactions

Introduction

- Ion beam optics

Complexe electrode and B field arrangement
Short calculation times time of flight < 0.1 ms



Commercial codes

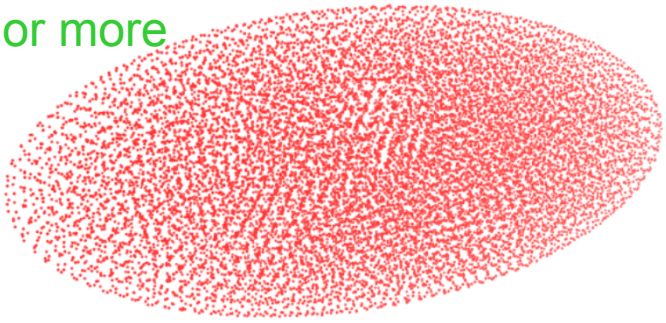
SIMION®

IB
IMU
Ion beam simulator

- Ion trapping / cooling dynamics

Simpler electrode and B field arrangement
Much longer calculation times ms to 100 ms or more
Interactions with lasers
Collisions, chem. reactions …

Home made codes



General scheme

- Electrode/ Magnet design $V(\vec{r}) \rightarrow \vec{E}(\vec{r})$ $\vec{B}(\vec{r})$ Analytical formula (ideal trap) Numerical representation

- Charged particles m_i, q_i, internal state $\vec{r}_i, \vec{v}_i, \vec{a}_i$ i=1 .. N

- Forces $q_i\vec{E}(\vec{r}_i,t) + q\vec{v}_i \times \vec{B}(\vec{r}_i,t)$ Lorentz

$$\vec{F}_i = \sum_{j \neq i} \frac{q_i q_j}{4\pi\varepsilon_0} \frac{\vec{r}_i - \vec{r}_j}{r_{ij}^3}$$ Coulomb

  - ✓ Laser interaction
  - ✓ Collisions, chemical reactions

- Newton's equations $m_i \ddot{\vec{r}} = \sum \vec{F}orces_i$ + initial conditions

Integrate the evolution ?

- Integration algorithm
- Time step
- Plateform ? CPU / GPU

- Langage

Orders of magnitude

Orders of magnitude

- Ion number from 1 to 10's of thousand

- Characteristic times

  Paul trap RF period 0.01 to 1 µs
  secular motions 1 .. 100 µs

  Penning trap cyclotron period 0.01 to 0.1 µs
  axial period : 0.1 .. 1 µs
  magnetron period : 10 … 200 µs

  Collision with neutral gas ~ instantaneous events
  rate 1000 to 10 000 /s

  Laser interaction : photon diffusion rates ~ 10^6 … 10^8/s

  Relaxation times : ms or longer

Short time scale
~10^{-9} s

Long time scale
> 10^{-3} s

Orders of magnitude

- Characteristic times

Coulomb collision :

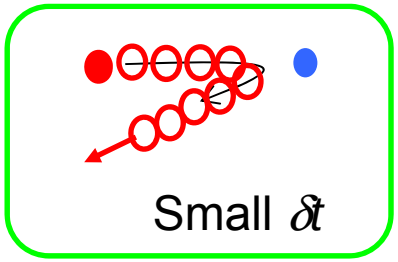
$$E_{coll} \approx \frac{1}{2} m v^2 \qquad E_{coll} \approx \frac{q_1 q_2}{4\pi\varepsilon_0 b}$$

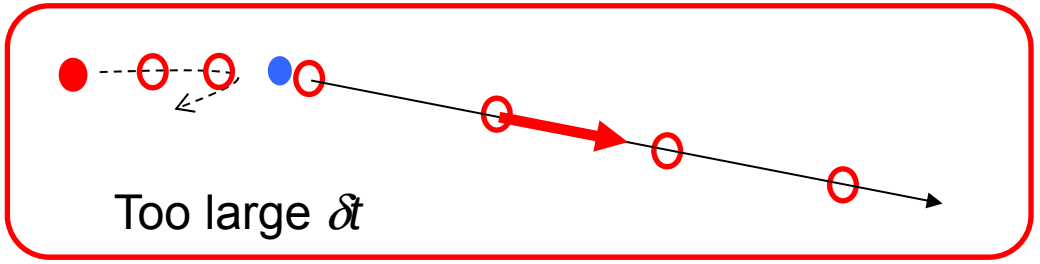
criterion $v_{max}\,\delta t << b$

$$\delta t << \frac{q_1 q_2}{4\pi\varepsilon_0} \frac{2}{m v_{max}^3}$$

Example M = 20, v_{max}=1000 m/s $\delta t << 1.3 \times 10^{-11}$ s

E_{coll} ~100 meV





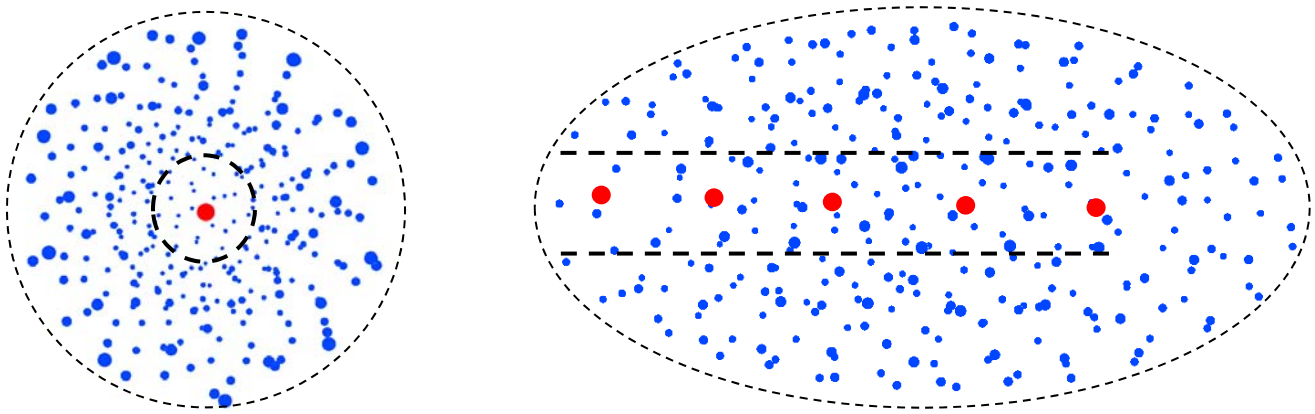
Example Chemical reaction $Be^+ + H_2 \rightarrow BeH^+ + H$ + kinetic energy

Photodissociation $HD^+ + h\nu \rightarrow D^+ + H$ v_{D+}= 6400 m/s $\delta t << 5 \times 10^{-13}$ s

Orders of magnitude

- The Coulomb interaction

$$\vec{F}_i = \sum_{j \neq i} \frac{q_i q_j}{4\pi\varepsilon_0} \frac{\vec{r}_i - \vec{r}_j}{r_{ij}^3}$$



Sympathetic cooling of 5 $^{40}Ar^{13+}$ with 300 Be^{+}

Mesoscopic medium : density representation is not relevant

Boundary conditions plays an essential role

Coulomb interaction
Time dependant trapping potential

} have to be represented exactly
N^2 complexity unavoidable

Orders of magnitude

- Conclusion

✓ Short integration time steps $\delta t < 10^{-9}$ s

✓ Very short time steps for energetic events

✓ Long simulation times $\Delta t > 1 \ldots 100$ ms

✓ Number of integration steps $N_{steps} = \Delta t/\delta t > 10^{6} \ldots 10^{8}$

✓ Adaptative time step for energetic events

For large N → Intensive simulations

Compiled language + double precision + parallelized CPU C, C++, FORTRAN
+ GPU openCL, Cuda

Computer requirements

Computer requirements

Memory (RAM)

1 integer 32 bits or 4 bytes
1 double precision 64 bit or 8 bytes

- Ion representation

9 double for position, velocity, acc.
2 double for mass, charge
0 .. few integers : internal state

~ 100 byte per ion

For 10^5 ions → 10 Mbyte negligible for CPU

- E or B field representation

Analytical formula 3 double per field negligible

Numerical representation on a 3 D grid $n_x n_y n_z$ double per field

$1000^3 = 10^9$ double 8 Gb per field

Memory (hard drive)

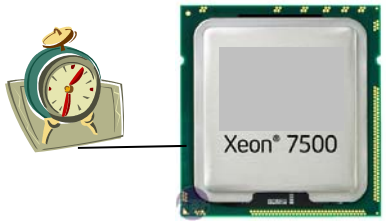
- Full trajectories : 9 double /time step/ion 72 bytes /step/ion

10^6 time steps 72 Mb/ion

1000 ions 72 Gb x 10 more steps → Tb

Computer requirements

Flops — Floating point operation / second



operations : +, -, x ~ 1 clock period
÷ ~ 10
√ ~ 10

- Linear contributions with respect to ion number N
  - E and B field evaluations
  - Trap force evaluation
  - Laser interaction, collisions
  - savings

~ 100-200 Flop per ion/step

- Quadratic contribution
  - Coulomb force evaluation

$$\vec{F}_i = \sum_{j \neq i} \frac{q_i q_j}{4\pi\varepsilon_0} \frac{\vec{r}_i - \vec{r}_j}{r_{ij}^3}$$

3 -, 3 x, 2 +, 1 ÷, 1 √, 5 x

~ 37 Flop per ion^2/step

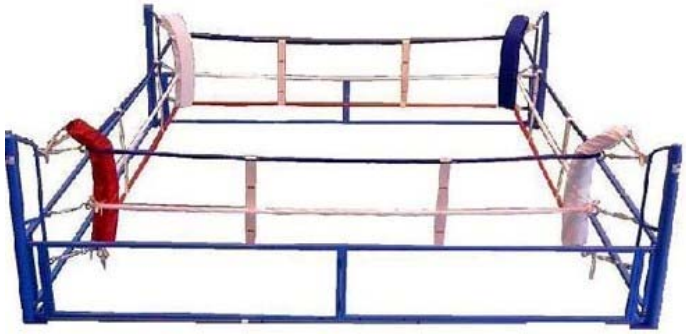
Amount of calculations : 37 N^2 + 200 N / step on a CPU

dominates if $N \geq 6$

Computer requirements

Flops

GPU graphic process unit / CPU



CPU 20 parallelized cores ~ 100 GFlops

GPU Titan black 15360 threads 1.3 TFlops

Obtained performances

N	N_{op}/step CPU	Time/step CPU	Time for 10^8 steps CPU	Time/step GPU	Time/step 6 x GPU
100	370 000	1.3 10^{-6} s	6 min		
1000	3.7 10^7	1.3 10^{-4} s	10 h 20		
10000	3.7 10^9	1.3 10^{-2} s	43 days		
15360	8.7 10^9	3 10^{-2} s	101 days	**8 days**	
86016	2.7 10^{11}	1 s	~ 9 years	243 days	**41 days**

do the experiment !

Integration algorithms

Numerical integration

Newton's equation $\ddot{\vec{r}} = \vec{f}(\vec{r}, \dot{\vec{r}}, t)$ 3D 2[d] order differential equation

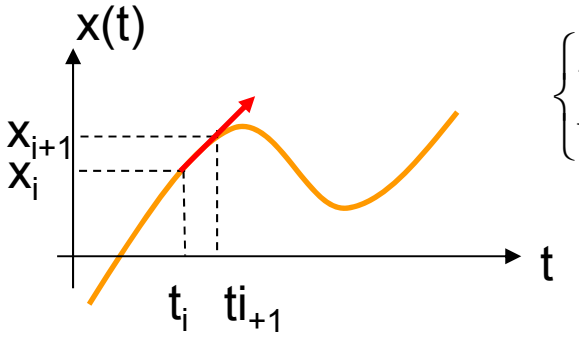
$$\begin{cases} \dot{\vec{r}} = \vec{v} \\ \dot{\vec{v}} = f(\vec{r}, \vec{v}, t) \end{cases}$$ 2 3D 1[st] order differential equations

$$\frac{d}{dt}\begin{pmatrix} \vec{r} \\ \vec{v} \end{pmatrix} = \vec{F}\left(\begin{pmatrix} \vec{r} \\ \vec{v} \end{pmatrix}, t\right)$$ 6D 1[st] order differential equation

Numerical integration

Basic idea : Euler's method

$$\ddot{x} = a(x, v, t)$$



$$\begin{cases} x_i \\ v_i \end{cases} \mapsto \begin{cases} x_{i+1} = x_i + v_i \delta t \\ v_{i+1} = v_i + a_i \delta t \end{cases}$$

Does it work ? example : 1 D harmonic oscillator $a(t) = -\omega^2 x(t) \quad \rightarrow a_i = -\omega^2 x_i$

$$\begin{cases} x_{i+1} = x_i + v_i \delta t \\ v_{i+1} = v_i - \omega^2 x_i \delta t \end{cases}$$

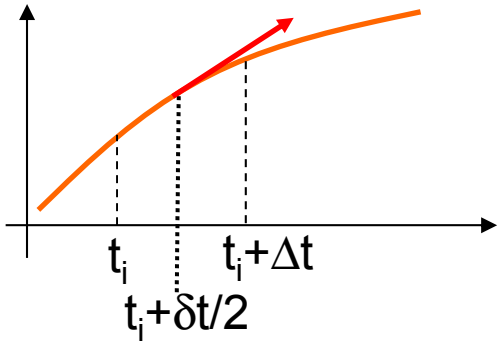
Total energy $E = \frac{1}{2} m v^2(t) + \frac{m\omega^2 x^2(t)}{2} \quad \rightarrow \quad E_i = \frac{m v_i^2}{2} + \frac{m\omega^2 x_i^2}{2}$

One integration step $E_{i+1} = \frac{m v_{i+1}^2}{2} + \frac{m\omega^2 x_{i+1}^2}{2} = \underbrace{(1 + \omega^2 \delta t^2)}_{> 1} E_i$

Energy is not preserved !

Numerical integration

Leap frog algorithm



Velocity independant force
Not suitable for Penning traps

$$x(t_i) \to x_i$$
$$v(t_i + \delta t/2) \to v_i$$
$$a(t_i) \to a_i$$

$$\begin{cases} x_i \\ a_i \mapsto \\ v_i \end{cases} \begin{cases} x_{i+1} = x_i + v_i \delta t + \dfrac{a_i \delta t^2}{2} \\ a_{i+1} = F(x_{i+1}) \\ v_{i+1} = v_i + \dfrac{a_i + a_{i+1}}{2} \delta t \end{cases}$$

Does it work ?

1D harmonic oscillator

$$\to a_i = -\omega^2 x_i \qquad \beta = \omega \delta t$$

$$\begin{pmatrix} \omega x_{i+1} \\ v_{i+1} \end{pmatrix} = \begin{pmatrix} 1 - \dfrac{\beta^2}{2} & \beta \\ -\dfrac{\beta}{2}\left(1 - \dfrac{\beta^2}{2}\right) & 1 - \dfrac{\beta^2}{2} \end{pmatrix} \begin{pmatrix} \omega x_i \\ v_i \end{pmatrix}$$

Energy conservation

$$E_i = \frac{m}{2}(v_i^2 + (\omega x_i)^2) \qquad E_{i+1} = E_i \ ?$$

$$-1 \le 1 - \frac{\beta^2}{2} \le 1$$

$$\omega \delta t \le 1$$

$$M(\beta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

Time reversal algorithm

$$\det(M) = 1 \to M(\beta)^{-1} = M(-\beta)$$

Numerical integration

4th order Runge Kutta

$$k_1 = h\ f(x_i, y_i)$$
$$k_2 = h\ f(x_i + \tfrac{h}{2}, v_i + \frac{k_1}{2})$$
$$k_3 = h\ f(x_i + \tfrac{h}{2}, v_i + \frac{k_2}{2})$$
$$k_4 = h\ f(x_i + h, v_i + k_3)$$
$$y_{i+1} = y_i + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)$$

assets

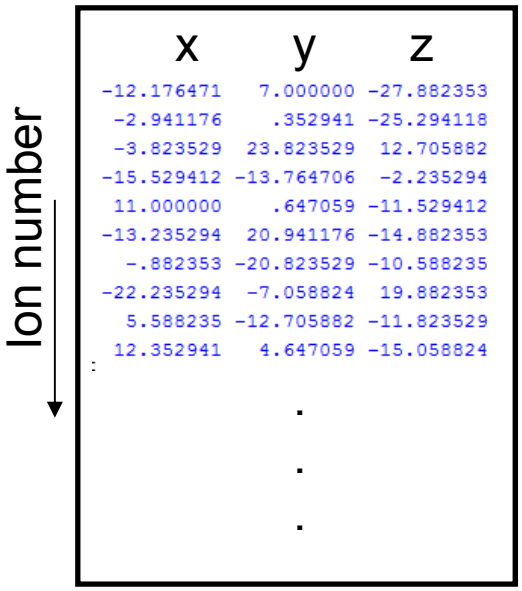
Easy to implement
Available in libraries
More precise : order 4 in h (2 for Leap Frog, and 1 for Euler)
Allows for larger time steps
Has a variable time step version RK 45

drawbacks

4 calls to the force (the longest part of the code)
For energetic events, too long time steps are not allowed

Coulomb force parallelisation

Coulomb force implementation



Position data set

compute independently

$$\vec{F}_1 = \frac{q_1}{4\pi\varepsilon_0}\sum_{j\neq 1} q_j \frac{\vec{r}_1 - \vec{r}_j}{r_{1j}^3}$$

$$\vec{F}_N = \frac{q_N}{4\pi\varepsilon_0}\sum_{j\neq N} q_j \frac{\vec{r}_N - \vec{r}_j}{r_{Nj}^3}$$

- Single CPU

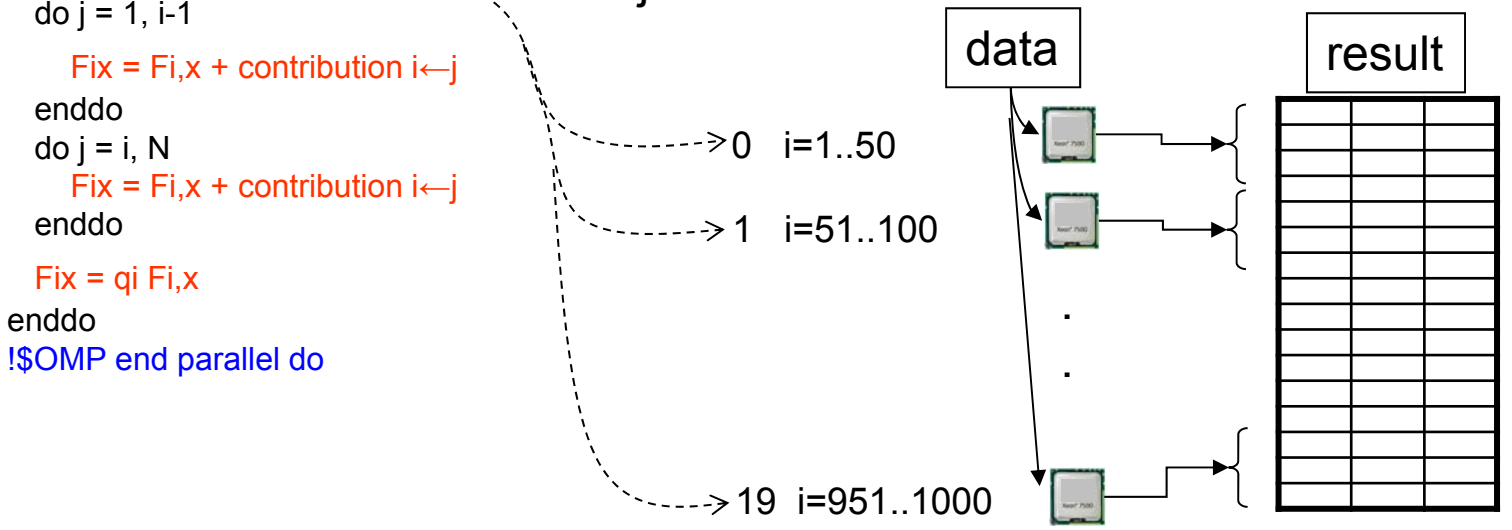
```
do i=1,N
  Fix = 0   initialisation
  do j=1,i-1
     Fix = Fix + contribution i←j
  enddo
  do j=i+1
     Fix = Fix + contribution i←j
  enddo
  Fix = qi Fix
enddo
```

Coulomb force implementation

- Multi core CPU Openmp library www.openmp.org

```
!$OMP parallel do private(j,Fix)
do i = 1, N
  Fix = 0
  do j = 1, i-1
    Fix = Fi,x + contribution i←j
  enddo
  do j = i, N
    Fix = Fi,x + contribution i←j
  enddo
  Fix = qi Fi,x
enddo
!$OMP end parallel do
```

j and Fix are local variables of each thread



Compile with -openmp

ifort –o prog –openmp prog.f90
Gcc –o prog –fopenmp prog.c

Set environement variables before execution

>OMP_NUM_THREADS=20
>OPM_SCHEDULE=‘’guided,50’’
>prog&

Coulomb force implementation

- Multi processor CPU MPI interface

  Up to thousand's of cores in computing centers

  Needs shared memory
  or huge data flows $\vec{r}_i, \vec{v}_i, \vec{a}_i$ at each time steps

- Reminder

  Reasonable amount of memory

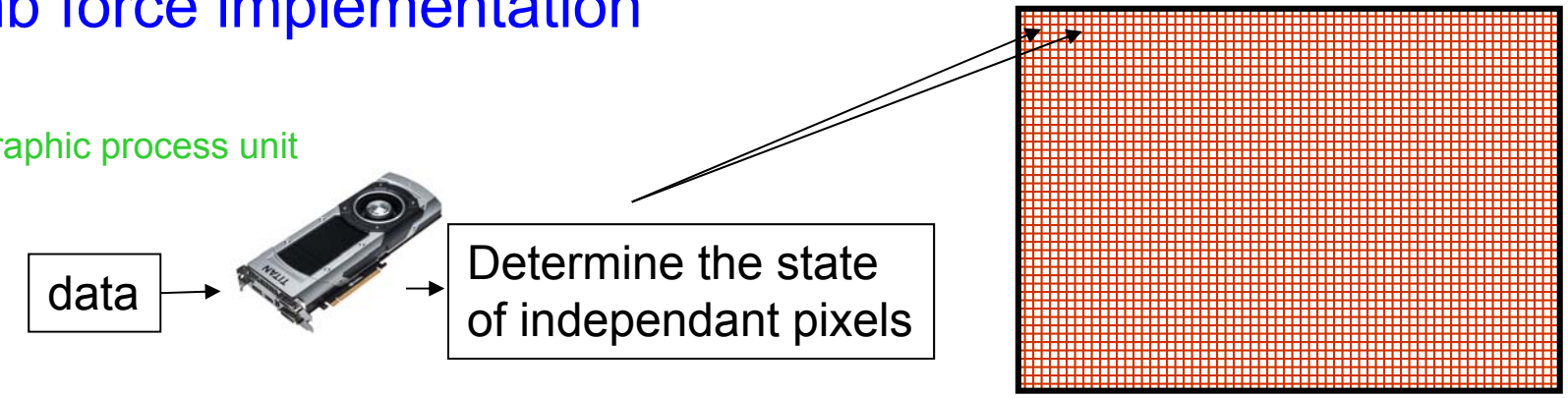
  For each ion, independant force calculation

  independant evolution for a time steps

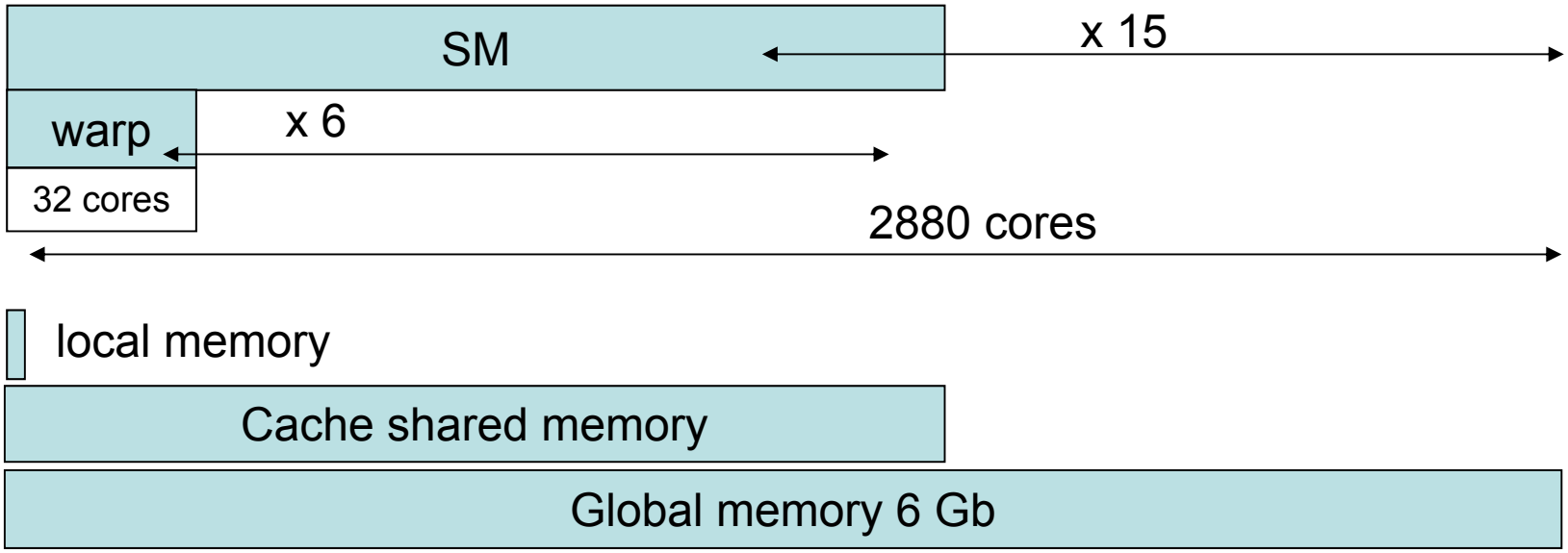
  identical algorithm with almost the same data

Coulomb force implementation

- GPU graphic process unit



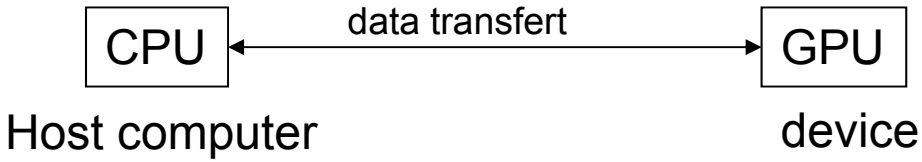
Example Titan Black



15360 threads for 15360 ions 35 10^9 Coulomb forces /s
x 37 Flops ~ **1.3 TFlops**

Coulomb force implementation

- GPU graphic process unit



warp

asynchrone transfert

stream

threadIdx.x

constant memory

memory

occupancy

streaming multiprocessors

kernel

cudaMemcpyAsync

cudaMallocHost

blocks

blockIdx.z

cudaMemcpyDeviceToHost

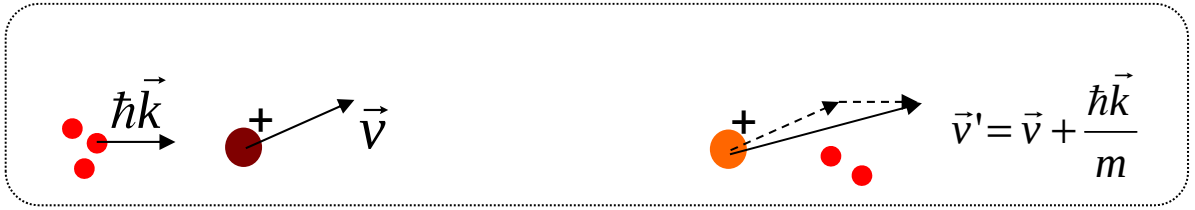
Laser interaction, Collisions, Chemical reactions

Laser interaction

The three process

absorption

proba B_{12} δt

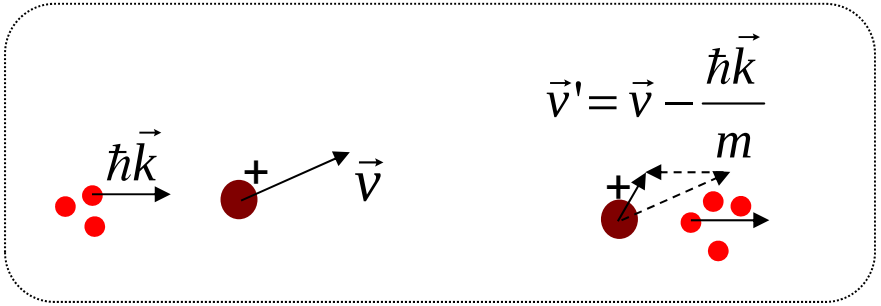


spontaneous emission

proba A_{21} δt

stimulated emission

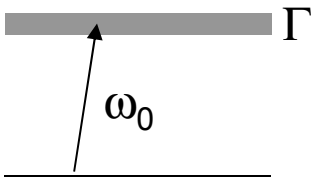
proba B_{21} δt

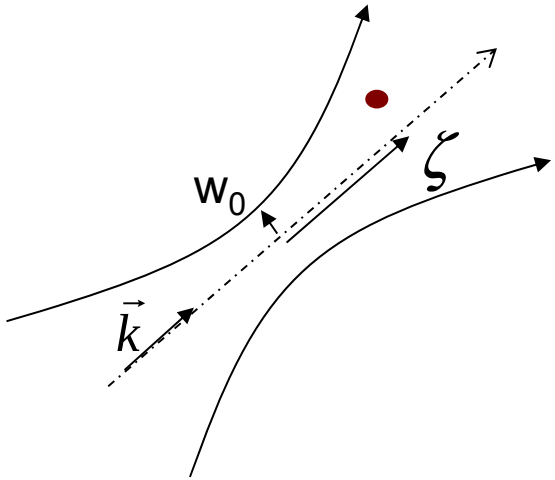


Results in a velocity kick

Laser interaction

Transition probabilities





$$A_{21}\delta t = \Gamma \delta t$$

$$B_{12}\delta t = \underbrace{\left(\frac{w_0}{w(\zeta)}\right)^2 e^{-2\frac{r^2}{w^2(\zeta)}}}_{\text{Gaussian profile}} \quad \frac{I}{I_{sat}} \quad \frac{1/2}{1+4\left(\frac{\omega_L + \vec{k}.\vec{v} - \omega_0}{\Gamma}\right)^2}\delta t$$

Laser intensity

Lorentz line shape

$$I_{sat} = \frac{2\pi^2 \hbar \Gamma}{3\lambda^3}$$

$$B_{21}\delta t = B_{12}\delta t$$

Laser interaction

Implementation Monte Carlo approach

Absorption, emission : instantaneous process

So at each time step, for each ion

- Evaluate the laser field intensity at the ion location $I = \left|\vec{E}_L(\vec{r}_i)\right|$

- Ground state ion evaluate $B_{12}\,\delta t$ (<< 1)
  if random[0,1] < $B_{12}\,\delta t$ then velocity kick $\hbar\vec{k}$, ion → excited
  else velocity unchanged

- Excited state ion evaluate $A_{21}\,\delta t$ and $B_{21}\,\delta t$ (<< 1)
  if random[0,1] < $A_{21}\,\delta t$ then velocity kick by random $\hbar\vec{k}'$
  else if $A_{21}\,\delta t$ < random[0,1] < $(A_{21}+B_{21})\,\delta t$ then v-kick $-\hbar\vec{k}$ } Ion → ground
  else v unchanged

Need : randomization of a direction
the proba of an event

Laser interaction

Other approaches

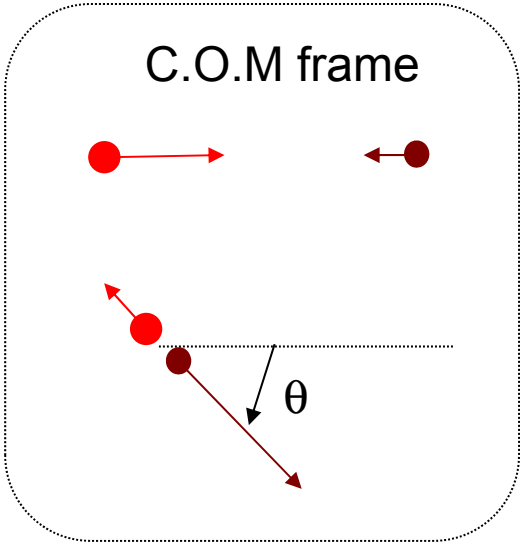
Effective drag force including saturation + Langevin force

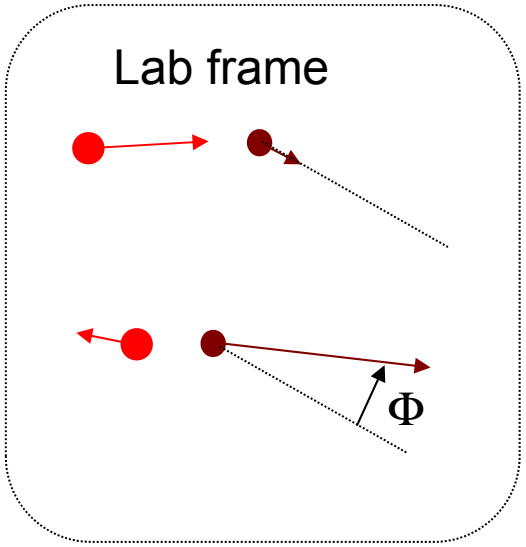
random process

Collisions, chemical reactions, Photodissociation

Charge – induced dipole interaction : short range → very fast process

→ described in terms of rates and velocity kicks





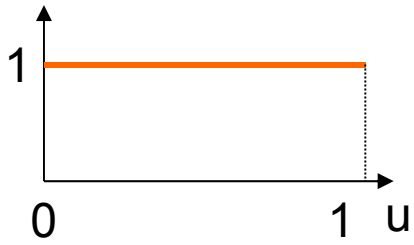
Ex : $Tryp+ + He \rightarrow Tryp^{+} + He$

$Ca^{+} + H_2 \rightarrow CaH^{+} + H + energy$

$HD^{+} + photon \rightarrow D^{+} + H$

Need : randomization of a direction

Random process

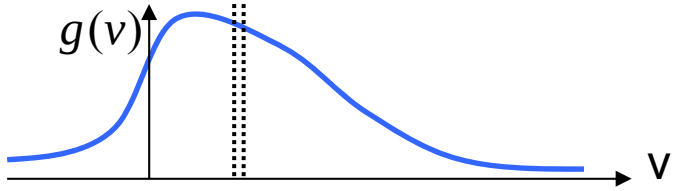


• Pseudo random number generator → uniform density in [0,1]

Choose a seed
Call random_number

0.569854 , 0.24865 , 0.15487 , 068754 , 0.125896 , 0.90110 , ….

• How to generate a random number set with a given density $g(v)$ on]a,b[?



Inverse transform sampling $\frac{dG(v)}{dv} = g(v)$ $G(a) = 0, G(b) = 1$

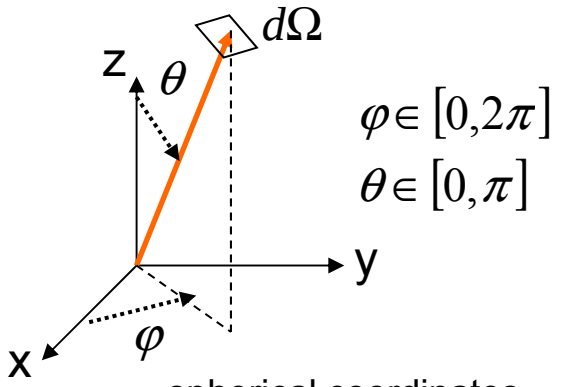
if $u = G(v)$ $du = d(G(v)) = g(v)dv$

Randomize *u* and compute $v = G^{-1}(u)$

Random process

- Generate a uniform random direction in 3D space

Uniform solid angle $d\Omega = \sin\theta\, d\theta\, d\varphi$



$\varphi \in [0, 2\pi]$

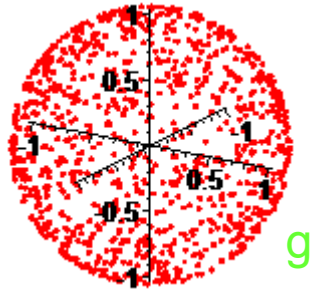
$\theta \in [0, \pi]$

spherical coordinates

Peters or Lambert projection
sphere → rectangle
preserving surface

$$\frac{d\Omega}{4\pi} = \frac{\sin\theta\, d\theta}{2} \quad \frac{d\varphi}{2\pi}$$

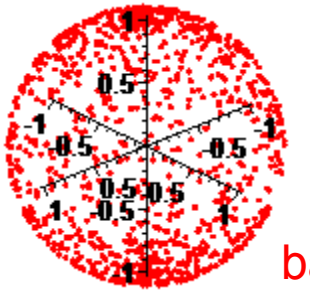
- For φ, randomize u and set $\varphi = 2\pi u$



good

- For θ, $g(\theta) = \frac{\sin(\theta)}{2}$ $G(\theta) = -\frac{\cos(\theta) - 1}{2}$

randomize u and set $\theta = \arccos(1 - 2u)$



bad

Random process

- Generate a Gaussian distribution

  - Inverse function method with *erf* function

  - Box-Muller method

    Take two random numbers *u* and *v* in [0,1]

    Compute $x = \sqrt{-2\ln u}\cos(2\pi v)$

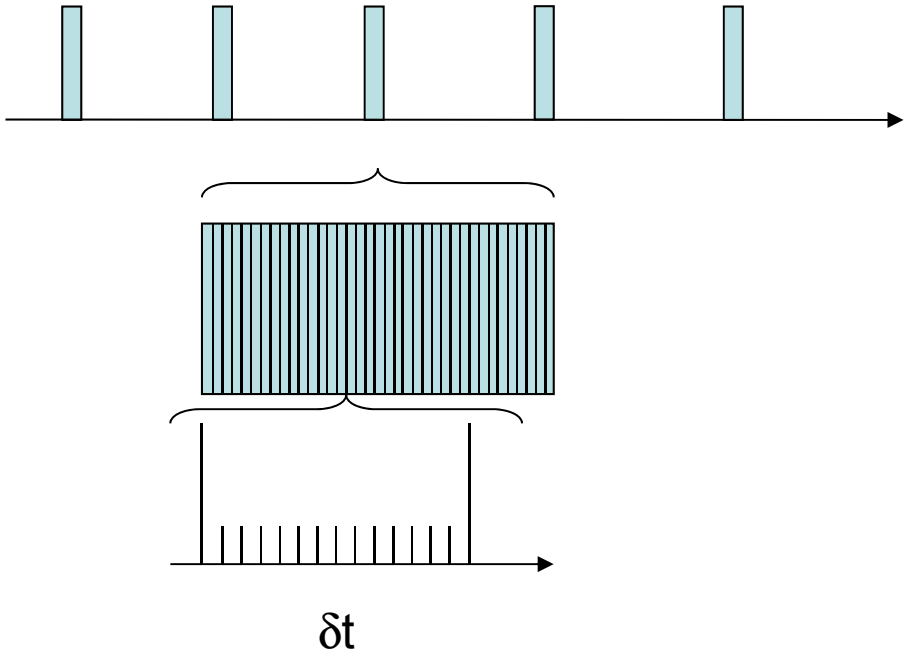
    $y = \sqrt{-2\ln u}\sin(2\pi v)$ → exercice

    x an *y* : two independant normalized gaussian centered variables

    Useful to initialize the temperature of an ion cloud

    Randomize the 3 velocity components $P(v_x) = \sqrt{\frac{2k_BT}{\pi m}} e^{-\frac{mv_x^2}{2k_BT}}$

A few examples



Saving times

Example 1

Capture and cooling of 2 $^{40}Ar^{13+}$ HCI by 300 Be^{+} ions

Example 2

Capture and cooling of 1 H^+ by 1800 Be^+ ions and 199 HD^+ ions

H^+ / Be^+ 1/9 mass ratio too large for efficient cooling