HPC for analysis of atomistic simulations

Eduardo M. Bringa

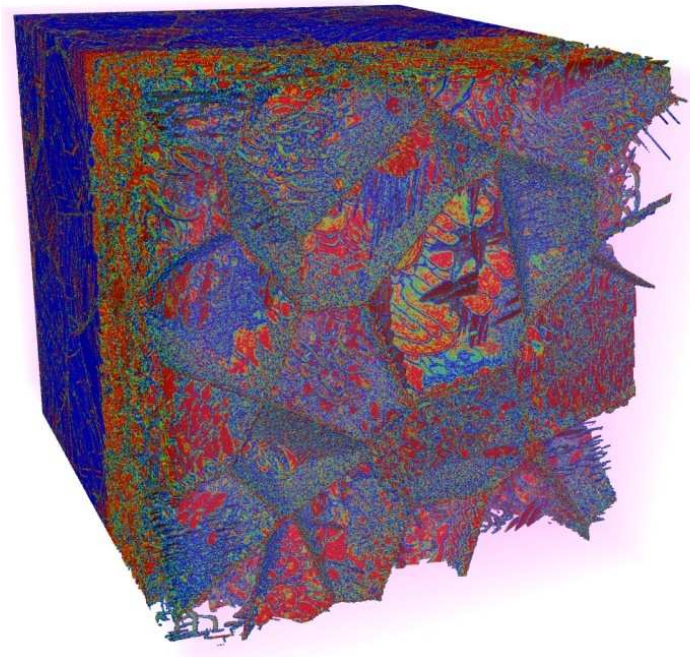
CONICET / Instituto de Ciencias Básicas, Universidad Nacional de Cuyo, Mendoza

(ebringa@yahoo.com)

40 JAIIO, HPC 2011, Cordoba, September 2011

Collaborators:

C. Ruestes, D. Bertoldi, C. Garcia Garino, E. Millán, S. Goirán, J. Aranibar (U.N. Cuyo), A. Stukowsky (LLNL), M.A. Meyers, Y. Tang (UCSD), J. Wark, A. Higginbotham (Oxford) , H. Urbassek, C. Ringl (TU Kaiserslautern)



Una herramienta muy útil para estudiar materiales:
Dinámica Molecular clásica =Molecular Dynamics=MD

• N partículas clásicas. Partícula *i* con posición $\boldsymbol{r}_i$, tiene velocidad $\mathbf{v}_i$ y aceleración $\boldsymbol{a}_i$.

• Partículas interactúan a través de un potencial empírico, $V(\boldsymbol{r}_1,.., \boldsymbol{r}_i,.., \boldsymbol{r}_N)$, que generalmente incluye interacciones de muchos cuerpos.

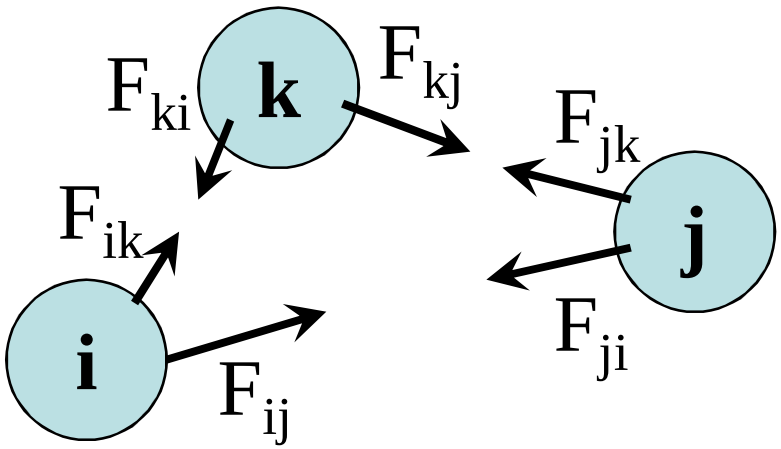
• Partículas obedecen las ecuaciones de movimiento de Newton. Partícula *i*, masa m_i: $\boldsymbol{F}_i = -\nabla_i V(\boldsymbol{r}_1,.., \boldsymbol{r}_i,.., \boldsymbol{r}_N) = m_i \boldsymbol{a}_i = m_i (d^2\boldsymbol{r}_i / dt^2)$

• Volumen<0.5 $\mu m^3 \sim 10^9$ átomos)

• Tiempos t<1 ns, $\Delta t \sim 1$ fs)

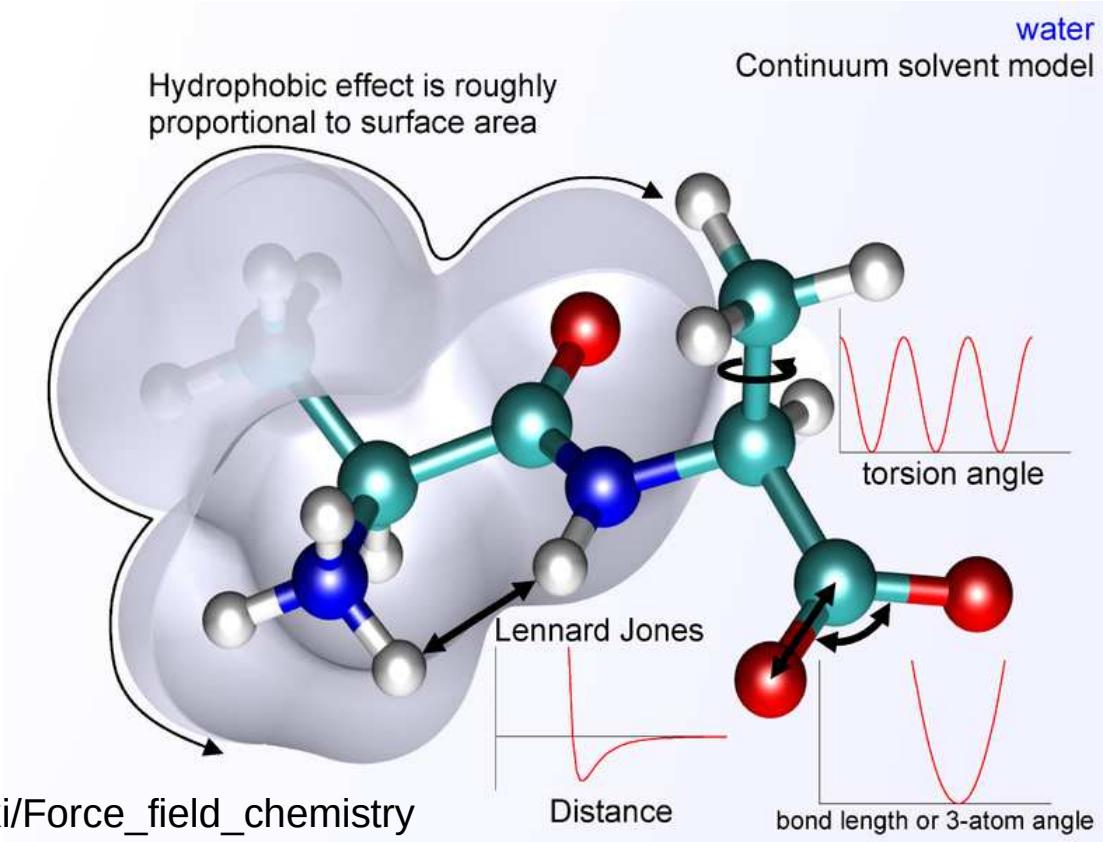
• Varios integradores disponibles

• Pueden incorporarse efectos electrónicos (Koci *et al*, PRB 2006).



A typical Force Field

$$E_{total} = E_{bonded} + E_{nonbonded}$$
$$E_{bonded} = E_{bond} + E_{angle} + E_{dihedral}$$
$$E_{nonbonded} = E_{electrostatic} + E_{vanderWaals}$$



http://en.wikipedia.org/wiki/Force_field_chemistry

Time Evolution: Integrators

•Many integrators: need accuracy, stability, reversibility, phase space volume conservation (symplectic integrators) and low memory cost.

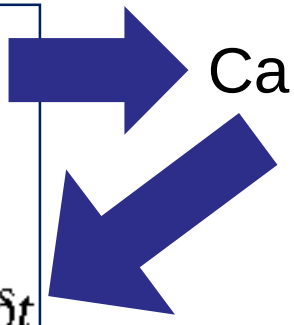
• Euler and Runge-Kutta integrators are not symplectic!

• Multiple-time step techniques, and also SHAKE integration (fixing fastest atoms).

• Most popular integrators: predictor-corrector (Gear, 3rd and 4th order), **Verlet** (symplectic: leap-frog & **velocity** versions, $O(\Delta t^2)$).

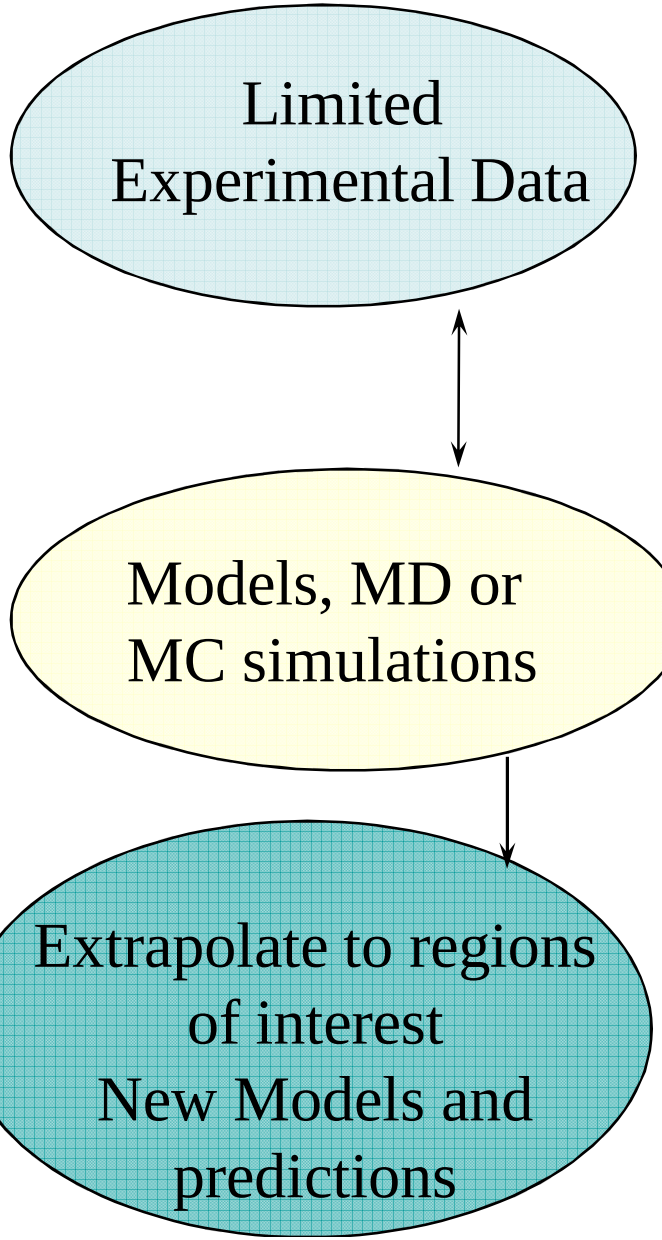
$$r(t+\delta t) = r(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2$$

$$v(t+\delta t) = v(t) + \frac{1}{2}\left[a(t) + a(t+\delta t)\right]\delta t$$

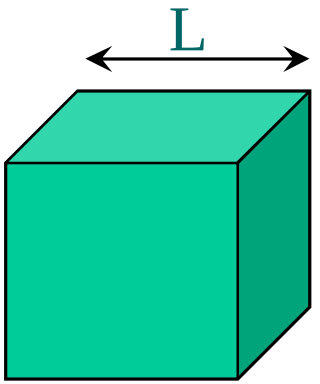


Calculate $a(t+\delta t)=F/m$

The cost of running atomistic simulations



But MD is very costly …



fcc lattice, L~30 monolayers $\Rightarrow 10^5$ atoms

Time step~ 10^{-15} s $\Rightarrow 10^{-11}$ s= 10^4 steps

Speed of typical MD code (short range force field) is ~5 10^{-6} s/(atom*time step)

1 iteration:

50 10^{-6} *10^5*10^4 = 5 10^4 s ~ 14 hours

20 iterations:

Need statistics ….

Total time ~ **12 days (in single core)**

Many MD codes can now use GPU acceleration

LAMMPS (*Large-scale Atomic/Molecular Massively Parallel Simulator*):
http://lammps.sandia.gov/ . MPI ofr several GPUs/cores (LJ: 1.2 ~10^7 atoms max Tesla C2070)
GPULAMMPS: http://code.google.com/p/gpulammps/ CUDA + OpenCL

HOOMD-Blue (*Highly Optimized Object-oriented Many-particle Dynamics***)**:
http://codeblue.umich.edu/hoomd-blue/index.html OMP for several GPUs in single board.

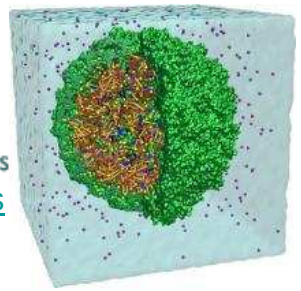
DL_POLY:
http://www.cse.scitech.ac.uk/ccg/software/DL_POLY/ F90+MPI, CUDA+OpenMP port.

GROMACS : http://www.gromacs.org/Downloads/Installation_Instructions/Gromacs_on_GPUs
Uses OpenMM libs (https://simtk.org/home/openmm). No paralelization. ~10^6 atoms max.

AMBER (*Assisted Model Building with Energy Refinement*): http://ambermd.org/gpus/
Ross Walker (keynote). MPI for several GPUs/cores. TIP3P, PME, ~10^6 atoms max Tesla C2070)

NAMD (*"Not another" MD*): http://www.ks.uiuc.edu/Research/namd/
GPU/CPU clusters.
VMD (Visual MD): http://www.ks.uiuc.edu/Research/vmd/

Many more!!!!



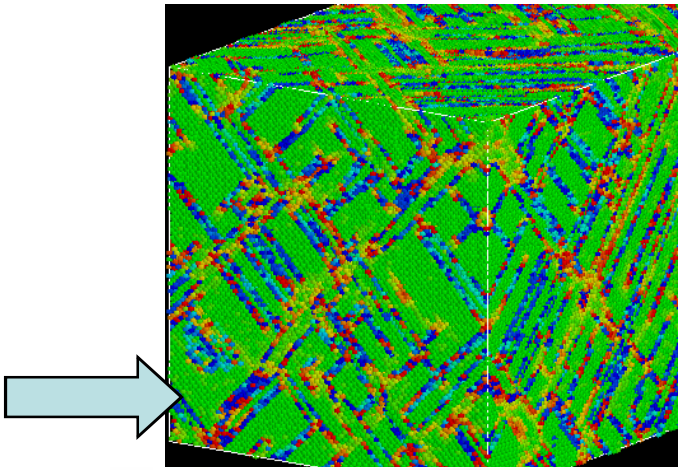
NAMD Scalable Molecular Dynamics
1,000,000+ atom Satellite Tobacco Mosaic Virus
Freddolino *et al.*, *Structure*, 14:437-449, 2006.

GTC 2010 Archive: videos and pdf's: http://www.nvidia.com/object/gtc2010-presentation-archive.html#md

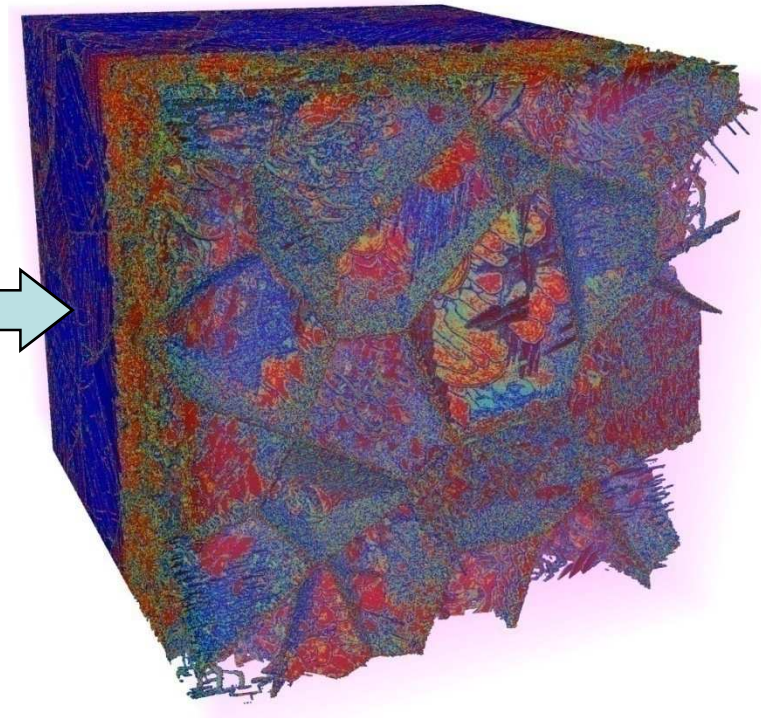
Visualization tools (que uso yo)

• PovRay (http://www.povray.com):
up to few million atoms, very fancy,
not interactive

• Rasmol
http://www.umass.edu/microbio/rasmol
up to few tens of millions of atoms,
very fast, not fancy but interactive



• LibGen, by M. Duchaineau (LLNL),
http://www.cognigraph.com/LibGen
viz + analysis tools, including parallel execution, interactive tools, etc.



Library for compact snapshots (TBs size).

• VMD,VISIT,TecPlot, GnuPlot, etc.

ParaView

- Open-source, multi-platform data analysis and visualization application.
- Developed to analyze extremely large datasets using *distributed memory computing* resources. It can be run on *supercomputers to analyze datasets of terascale* as well as on laptops for smaller data.

- **2010 Editors' Choice Award for Best HPC Visualization Product or Technology**
- **2010 Readers' Choice Award for Best HPC Visualization Product of Technology**

HPC wire

Distribution of workload for parallel computing in MD

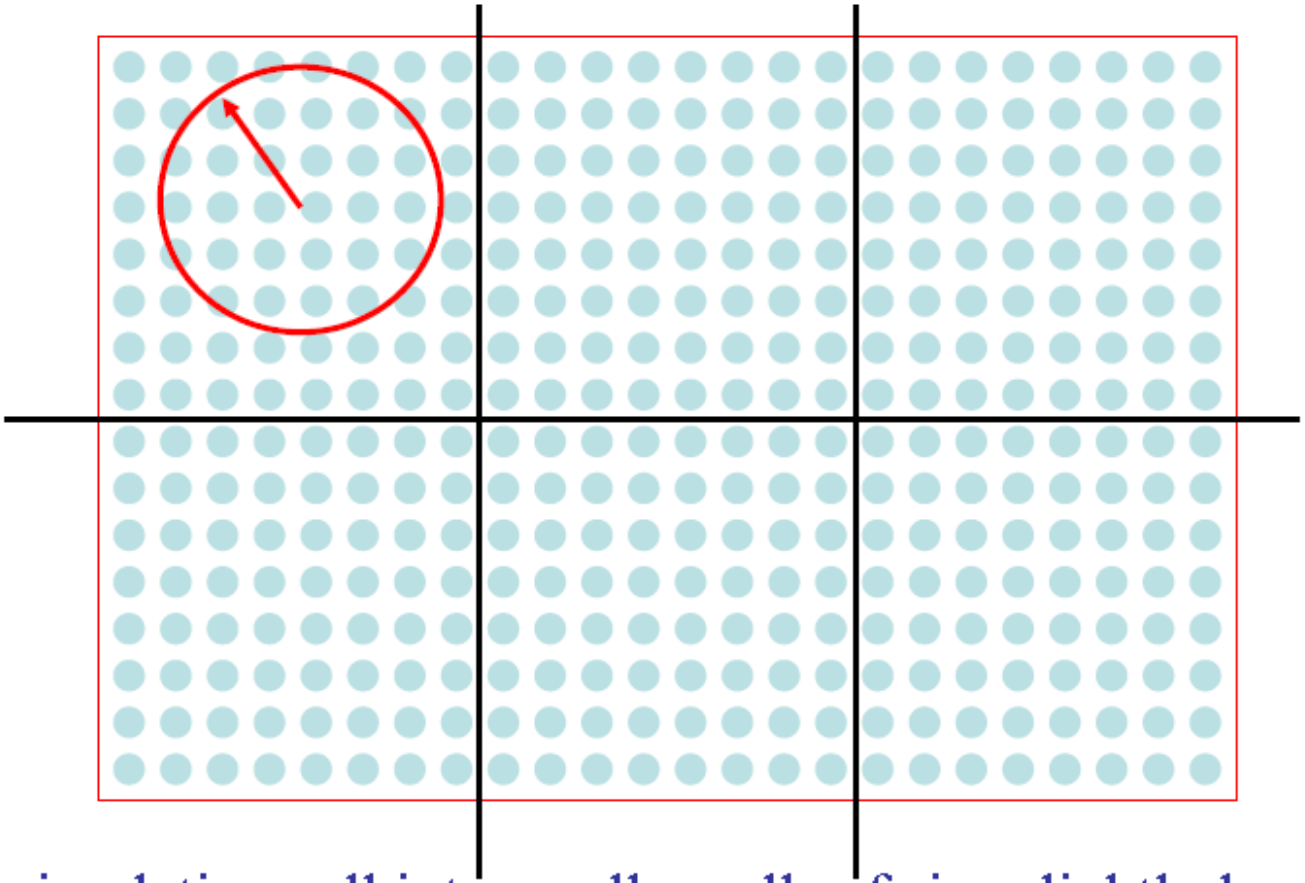
- The replicated data (RD) algorithm is the easiest to implement. Each node stores a copy of all atomic data in the whole system, but performs only its own part of calculational workload. The pay-off for the simplicity of this algorithm is poor scaling, regarding both memory and communication.

- The force decomposition (FD) algorithm is similar in design to the RD algorithm, but aims at reducing the amount of data flowing between nodes. This is done by using a permutation matrix for pair forces which allows the storage of less atomic data per processor. It is especially effective for simulation of moderately sized systems.

- The domain decomposition (DD) algorithm divides the entire simulation box geometrically and assigns each subdomain to one node. Individual processors must exchange information with their neighbors for the boundary particles, but afterwards each node is able to compute the forces and potentials in parallel. This algorithm is especially effective for short-ranged interactions.

Reference:
A domain decomposition molecular dynamics program for the simulation of flexible molecules with an arbitrary topology of Lennard–Jones and/or Gay–Berne sites. Jaroslav Ilnytskyi, Mark R. Wilson. Computer Physics Communications 134 (2001) 23–32

Spatial decomposition & cell lists



- Divide simulation cell into smaller cells of size slightly larger than r_c
- Atoms only interact with other atoms in the same of nearest neighbor cell

- Allocating atoms to cells can be done in order N
- Calculating interactions is also order N

Alejandro Strachan, http://nanohub.org/resources/5838#series

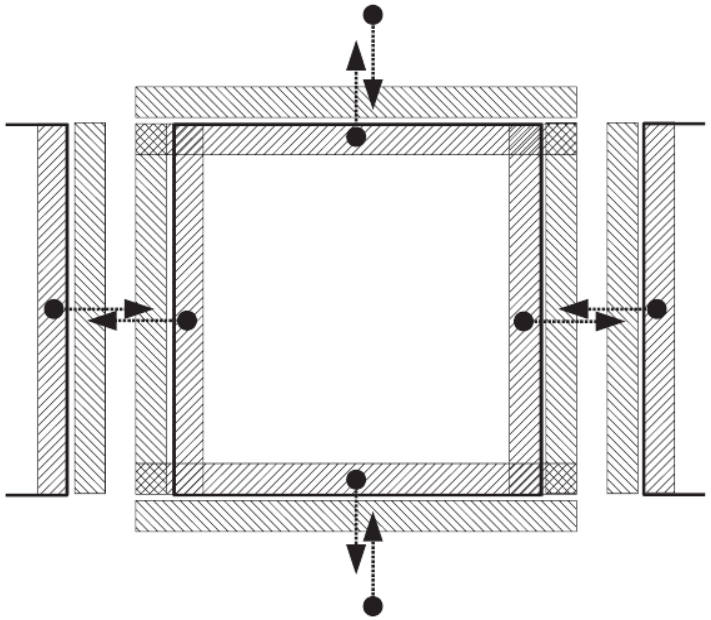
Domain decomposition (DD)

- To apply the DD algorithm the simulational box is divided spatially into equal-sized cuboidal subdomains (regions), one for each computing node. The subdomain dimension in any direction must be no less than the maximal cutoff for the non-bonded interactions. At the start of a simulation each node reads the molecular topology file and a copy of the topology for each different molecule type is stored on each node.
- All the atoms that are in a given subregion at some moment in time reside in the processor responsible, and when an atom moves between subregions all the associated variables are explicitly transferred from one processor to another. Thus there is economy insofar as memory is concerned, and also in the communication required to allow atoms to transfer between processors, since comparatively few atoms make such a move during a single timestep.

References:

- *A domain decomposition molecular dynamics program for the simulation of flexible molecules with an arbitrary topology of Lennard–Jones and/or Gay–Berne sites. Jaroslav Ilnytskyi, Mark R. Wilson. Computer Physics Communications 134 (2001) 23–32*
- *THE ART OF MOLECULAR DYNAMICS SIMULATION. D. C. Rapaport*

Domain decomposition (DD)



- The portion of the simulation region (for a two-dimensional subdivision) represented by the square outline is handled by a single processor; it contains shaded areas denoting subregions whose atoms interact with atoms in adjacent processors, and is surrounded by shaded areas denoting subregions from adjacent processors whose interactions must be taken into account; arrows indicate the flow of data between processors.

Reference:
THE ART OF MOLECULAR DYNAMICS SIMULATION. D. C. Rapaport

Neighbor lists

- For short-range force calculations in MD, the force summations are restricted to atoms within some small region surrounding each particle. This is typically implemented using a cutoff distance r_c , outside of which particles are not used for force calculation. The work to compute forces now scales linearly with the number of particles. This approach requires knowing which particles are within the cutoff distance r_c at every timestep. The key is to minimize the number of neighboring atoms that must be checked for possible interactions.
- Two basics techniques are typically used to accomplish this: The Verlet neighbor list and the link-cell method.
- Verlet [1967] suggested a technique that maintains a list of neighbors of a particular atom/molecule, which is updated at intervals. Between updates of the list the program only checks the j atoms/molecules in the list.

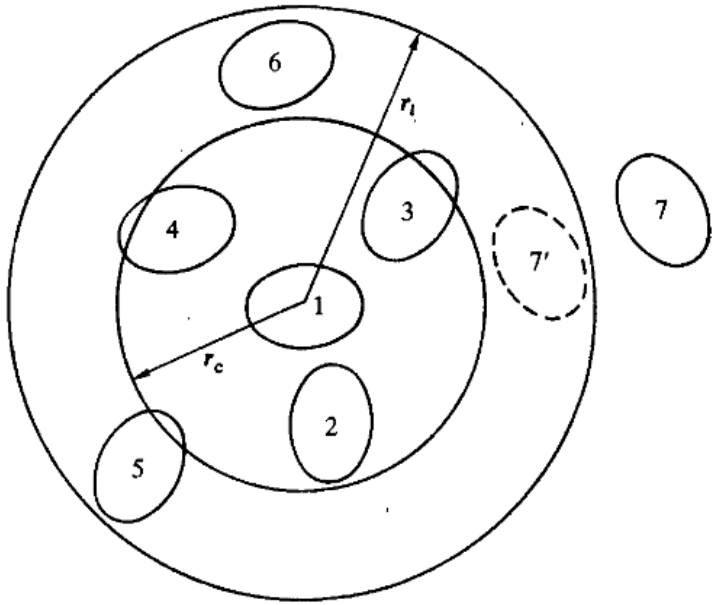
References:

- *Implementing molecular dynamics on hybrid high performance computers – short range forces. W. Michael Brown, Peng Wang, Steven J. Plimpton, Arnold N. Tharrington. Computer Physics Communications 182 (2011) 898–911.*
- *Computer Simulation of Liquids. M.P. Allen and D.J. Tildesley. 1989*

Verlet neighbor list

- For each atom, a list of nearby atoms is kept. Typically, when the list is formed, all neighboring atoms within an extended cutoff distance $r_i = r_c + \gamma$ are stored. The list can be used for multiple timesteps until an atom has moved from a distance $r > r_i$ to $r < r_c$. The optimal value for γ will depend on simulation parameters, but is typically small relative to r_c.

- The cutoff sphere and its skin, around molecule 1. Molecules 2,3,4,5 and 6 are on the list of molecule 1; molecule 7 is not. Only molecules 2,3 and 4 are within the range of the potential at the time the list is constructed.

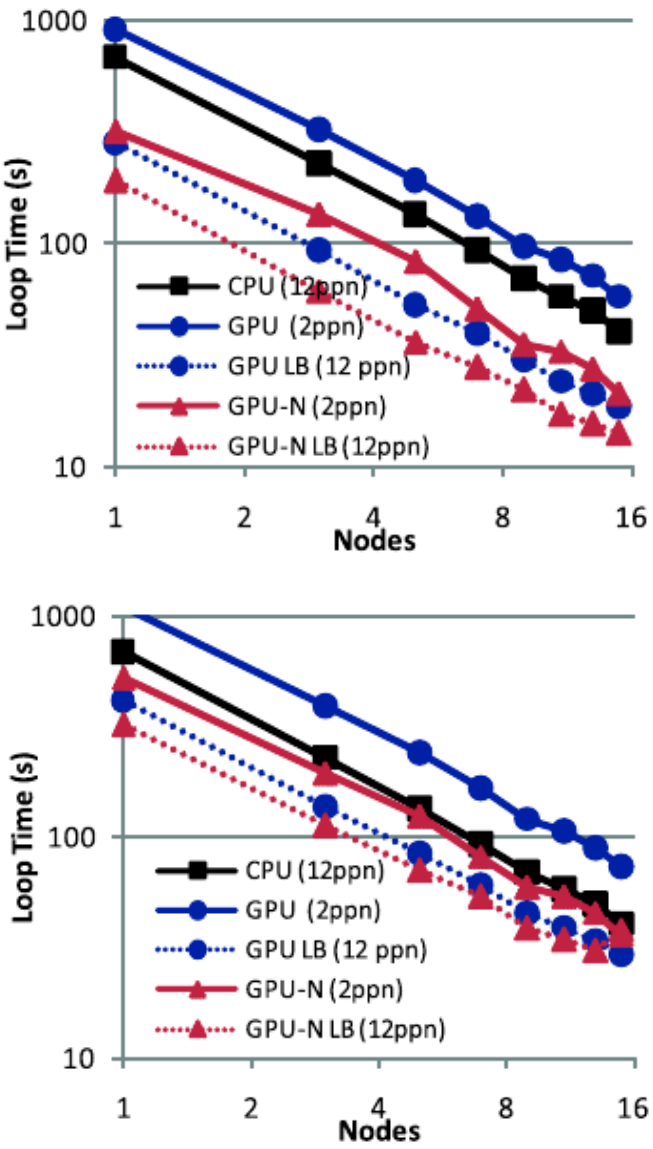


References:

- *Implementing molecular dynamics on hybrid high performance computers – short range forces. W. Michael Brown, Peng Wang, Steven J. Plimpton, Arnold N. Tharrington. Computer Physics Communications 182 (2011) 898–911.*
- *Computer Simulation of Liquids. M.P. Allen and D.J. Tildesley. 1989*

GPU vs CPU performance

- Strong-scaling for the Lennard-Jones test case with and without acceleration. Top: Comparison of loop time without acceleration (CPU), acceleration with 1 process per GPU (2 ppn), and load balancing (LB) with 6 processes per GPU (12 ppn) for single precision. Neighbor calculations are performed on the GPU for the GPU-N cases.
- Bottom: double precision.
- Loop times are the wall time required to complete the entire simulation loop.



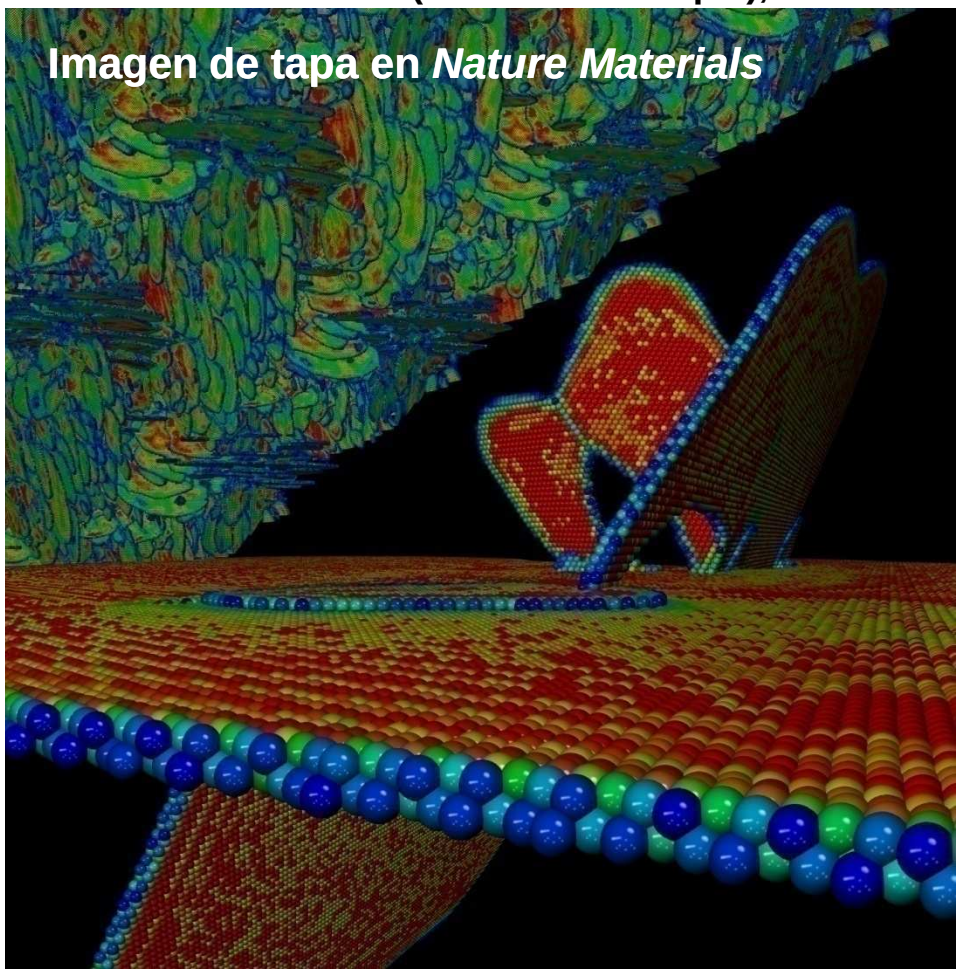
Reference:

•*Implementing molecular dynamics on hybrid high performance computers – short range forces. W. Michael Brown, Peng Wang, Steven J. Plimpton, Arnold N. Tharrington. Computer Physics Communications 182 (2011) 898–911.*

Ejemplo: simulaciones a la misma escala que el experimento descubren cómo relajan los defectos bajo alta presión y altas velocidades de deformación

1980 CPU's en MCR, ~15 días, 30 TB of datos

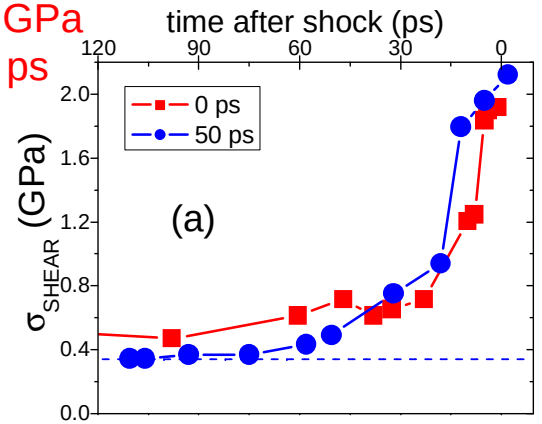
256-352 million atoms (~0.07×0.07×0.8 μm), Cu



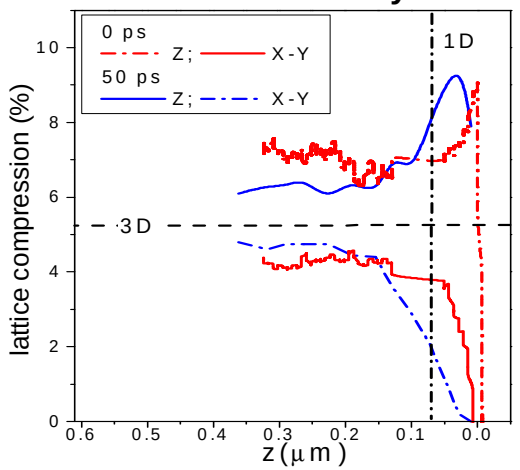
Bringa et al., *Nature Materials* 5, 805 (2006)
Shehadeh, Bringa, et al., *APL* 89, 171918 (2006)

movimiento de dislocaciones cambia el estrés de 1D a ~3D

P~35 GPa
t_R=50 ps

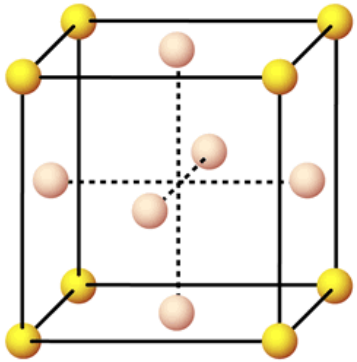


Difraccion simulada de rayos X



Centro-Symmetry Parameter (CSP)*

- **Centro-symmetry parameter** *(CSP):* a parameter to measure the local disorder, particularly useful to study cubic structures.

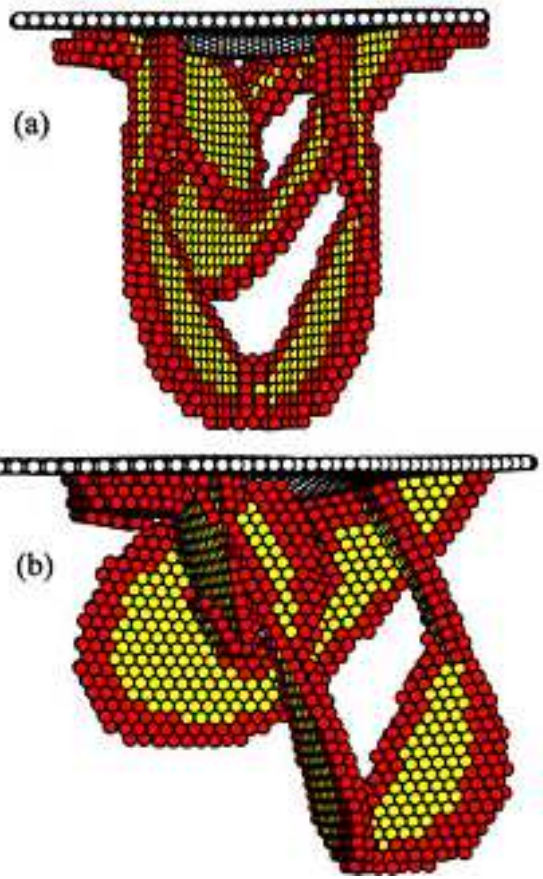


f.c.c structure

$$C = \sum_{i=1}^{6} \left| \vec{R}_i + \vec{R}_{i+6} \right|^2$$

CSP expression for a f.c.c. unit cell

This is done for every atom in the sample ➔ high computational cost



Kelchner et al, FIG. 2, partial view. Defect structure at the first plastic yield point during indentation on Au (111), (a) view along [112], (b) rotated 45° about [111]. The colors indicate defect types as determined by the centrosymmetry parameter: partial dislocation (red), stacking fault (yellow), and surface atoms (white). Only atoms with P>0.5 are shown.

* Kelchner, Plimpton, Hamilton, Phys Rev B, 58, 11085 (1998)

Common Neighbor Analysis

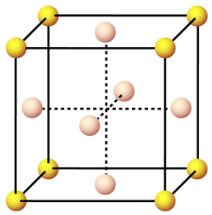
- **CNA**: a parameter to measure the local disorder
- Sensitive to cutoff radius
- 12 nearest neighbor for perfect FCC and HCP crystals, 14 nearest neighbors for perfect BCC crystals

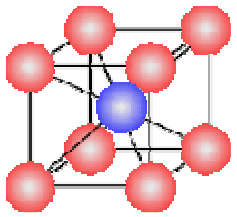
This is done for every atom in the sample → high computational cost

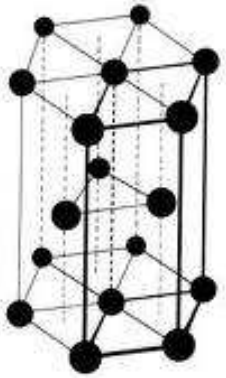
$$r_c^{fcc} = \frac{1}{2}\left(\frac{\sqrt{2}}{2}+1\right)\mathrm{a} \simeq 0.8536\,\mathrm{a}$$

$$r_c^{bcc} = \frac{1}{2}(\sqrt{2}+1)\mathrm{a} \simeq 1.207\,\mathrm{a}$$

$$r_c^{hcp} = \frac{1}{2}\left(1+\sqrt{\frac{4+2x^2}{3}}\right)\mathrm{a}$$







• Faken, Jonsson, Comput Mater Sci, 2, 279 (1994).
• Tsuzuki, Branicio, Rino, Comput Phys Comm, 177, 518 (2007).

DXA (Dislocation eXtraction Algorithm)

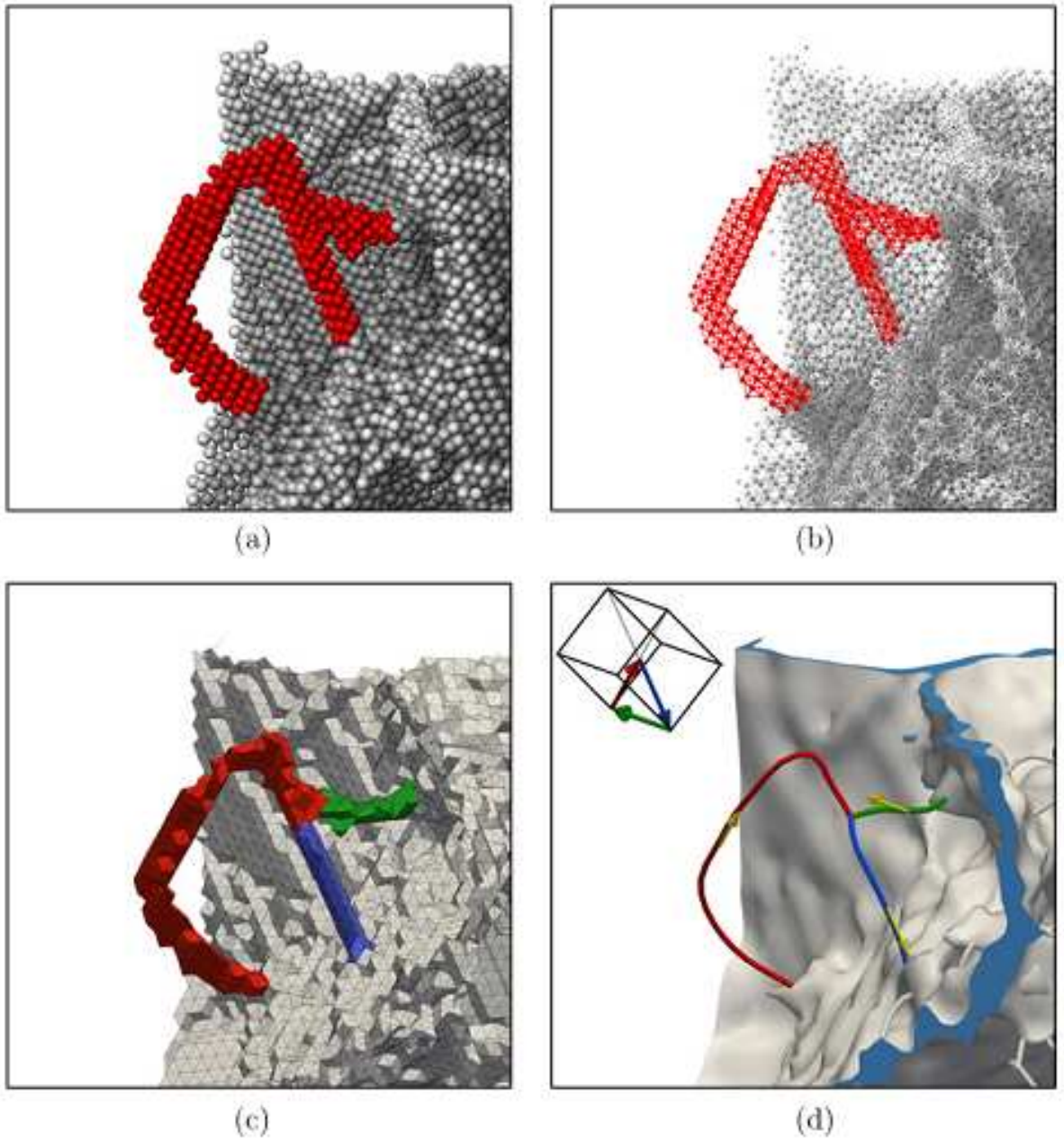
Stepwise conversion of three atomistic dislocation cores into a geometric line representation.

(a) Atomistic input data.

(b) Bonds between disordered atoms.

(c) Interface mesh.

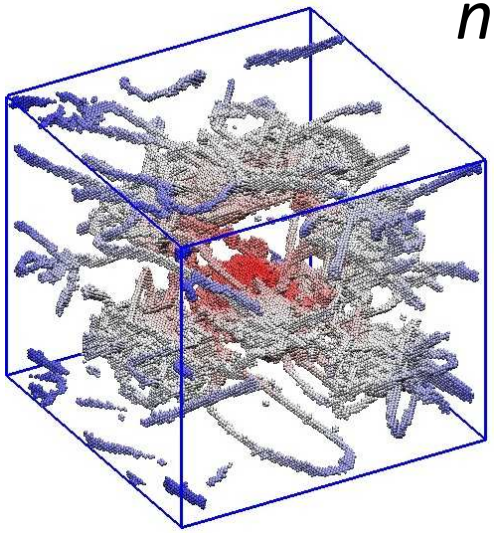
(d) Smoothed output.



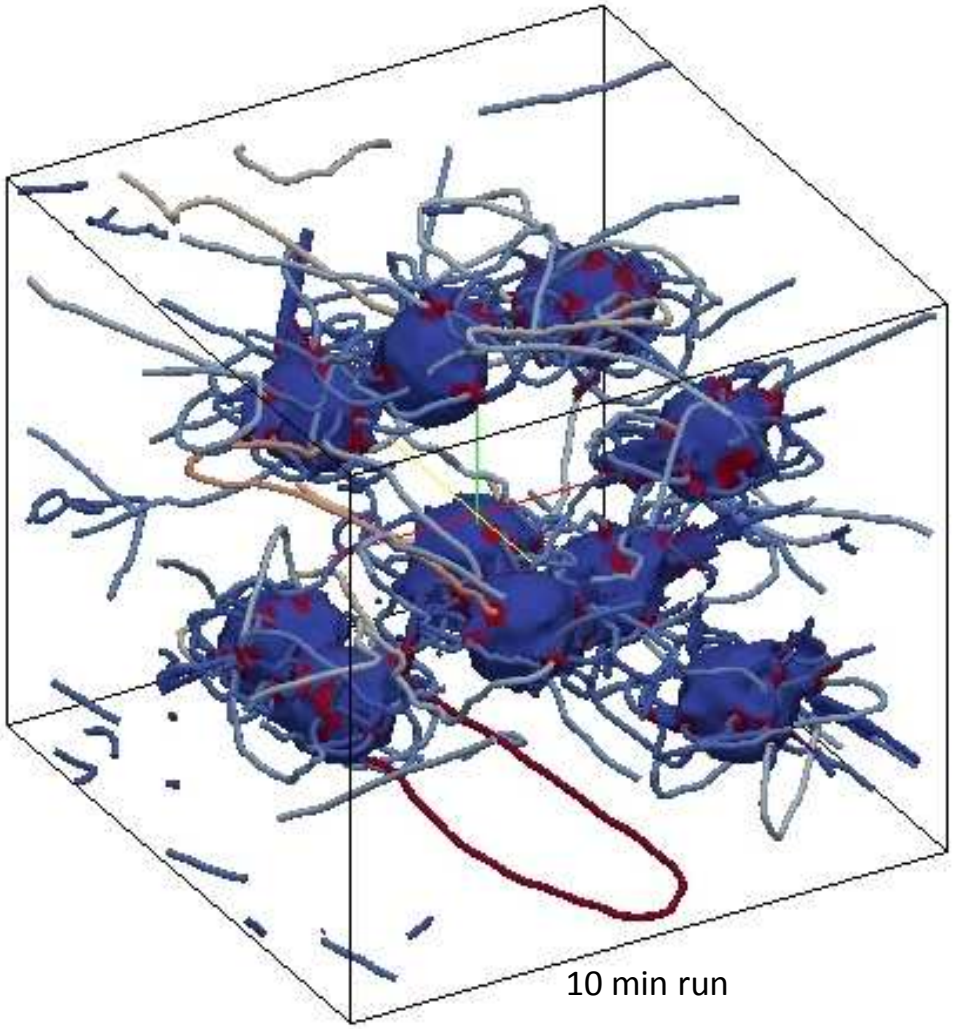
A. Stukowski and K. Albe *Extracting dislocations and non-dislocation crystal defects from atomistic simulation data* [Modelling Simul. Mater. Sci. Eng. 18 (2010) 085001].

Modified DXA + ParaView

*Atomistic simulation of the mechanical properties of a nanoporous b.c.c. metal **



Typical view using well known VMD



10 min run

ParaView visualization of the results provided by DXA for a nanoporous Ta sample subjected to a 10^9/s uniaxial compressive strain rate at an 8% strain. Preprocessed sample has 1.9 million atoms.
Run: 3 days in 32 cores
Analysis of each snapshot: 10 min run on AMD M520 + 4Gb RAM (dual core)

* Ruestes et al. To be submitted to Acta Materialia (2011)

Modified DXA + ParaView

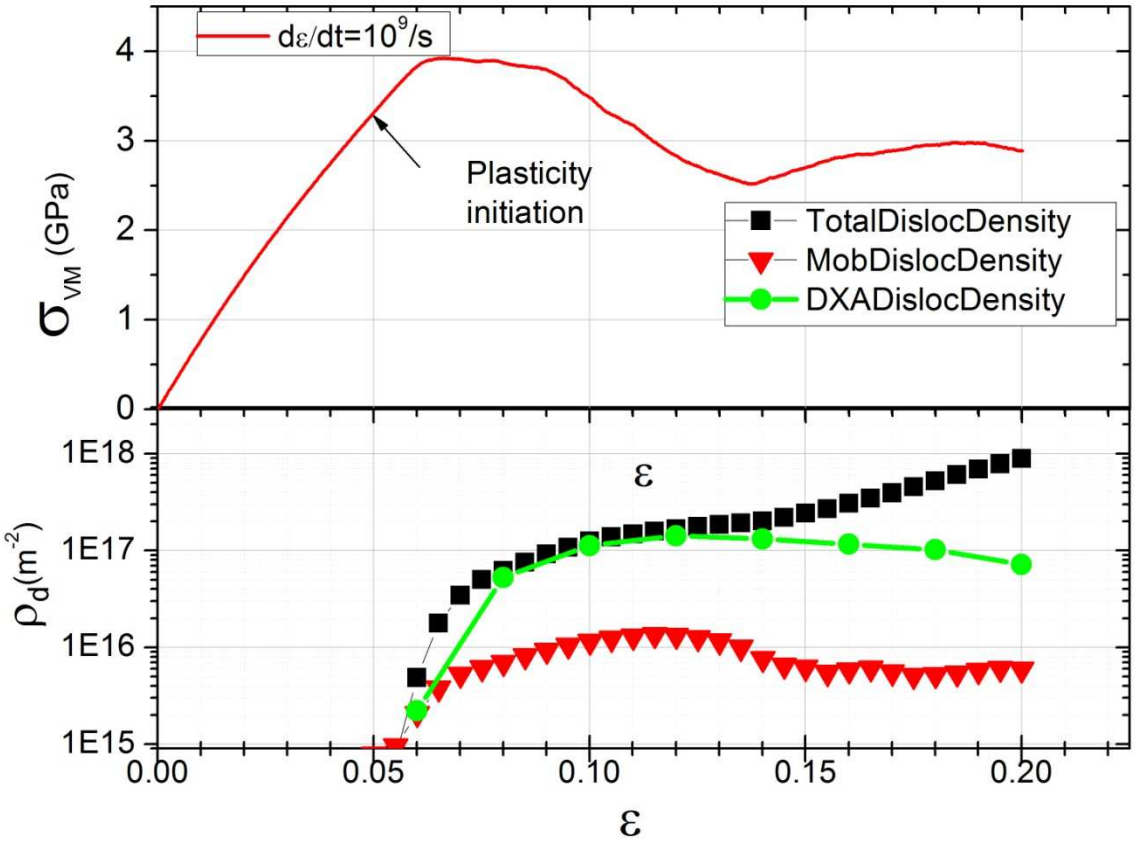
*Atomistic simulation of the mechanical properties of a nanoporous b.c.c. metal **

Every black dot involves filtering non bcc atoms from a 1.9 million atoms sample by means of a CNA analysis run in parallel using LAMMPS
Every green dot involves a DXA run on the sample.

Run: 3 days in 32 cores

Analysis of each snapshot (off-line): 10 min run on AMD M520 + 4Gb RAM (dual core)

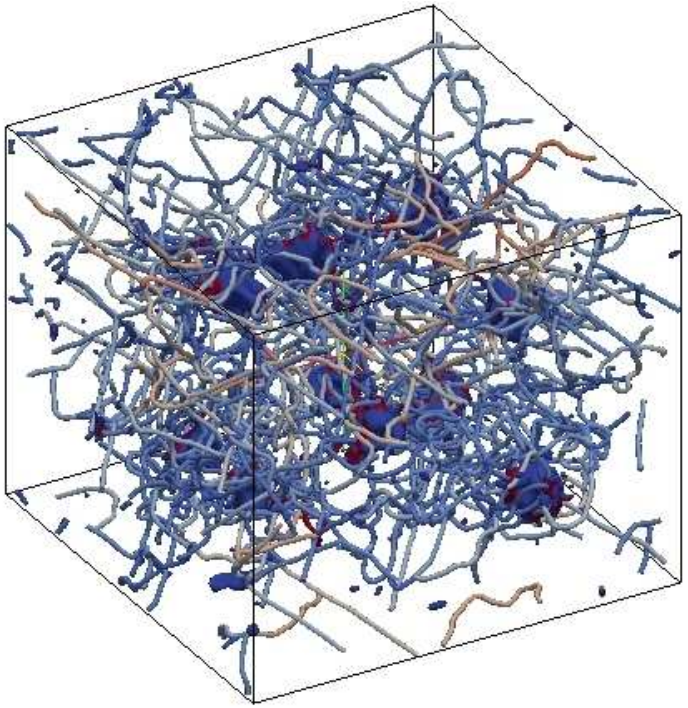
Analysis during simulation would push run time beyond maximum allowed time.



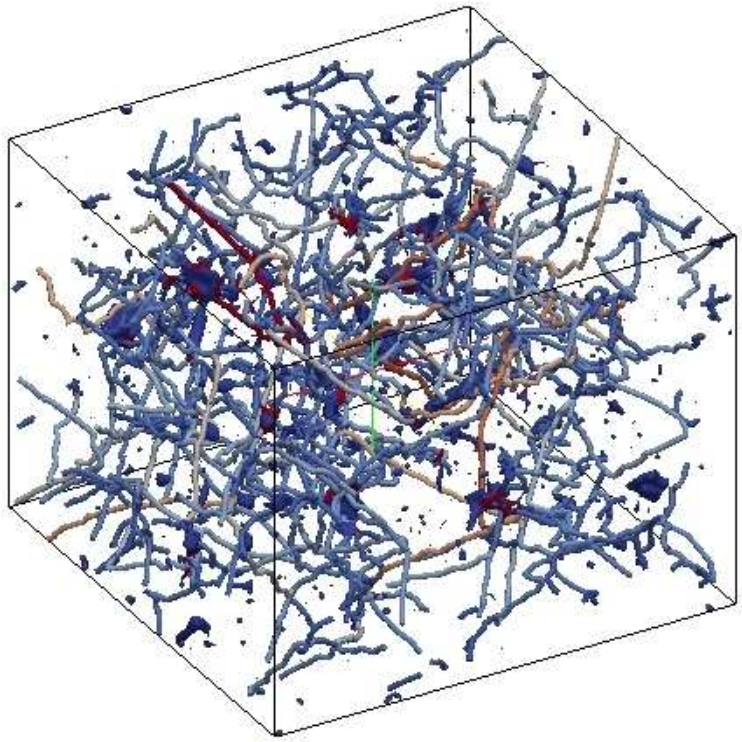
* Ruestes et al. To be submitted to Acta Materialia (2011)

Modified DXA + ParaView

*Atomistic simulation of the mechanical properties of a nanoporous b.c.c. metal **



Useful for analysing highly distorted structures, full of dislocations, junctions and surface defects. (a) 12 % strain and (b) 20 % strain

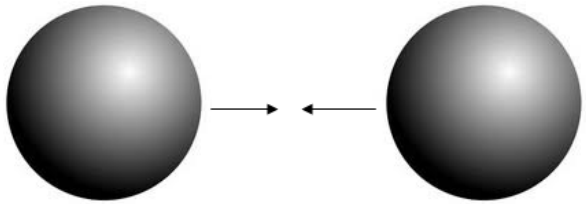


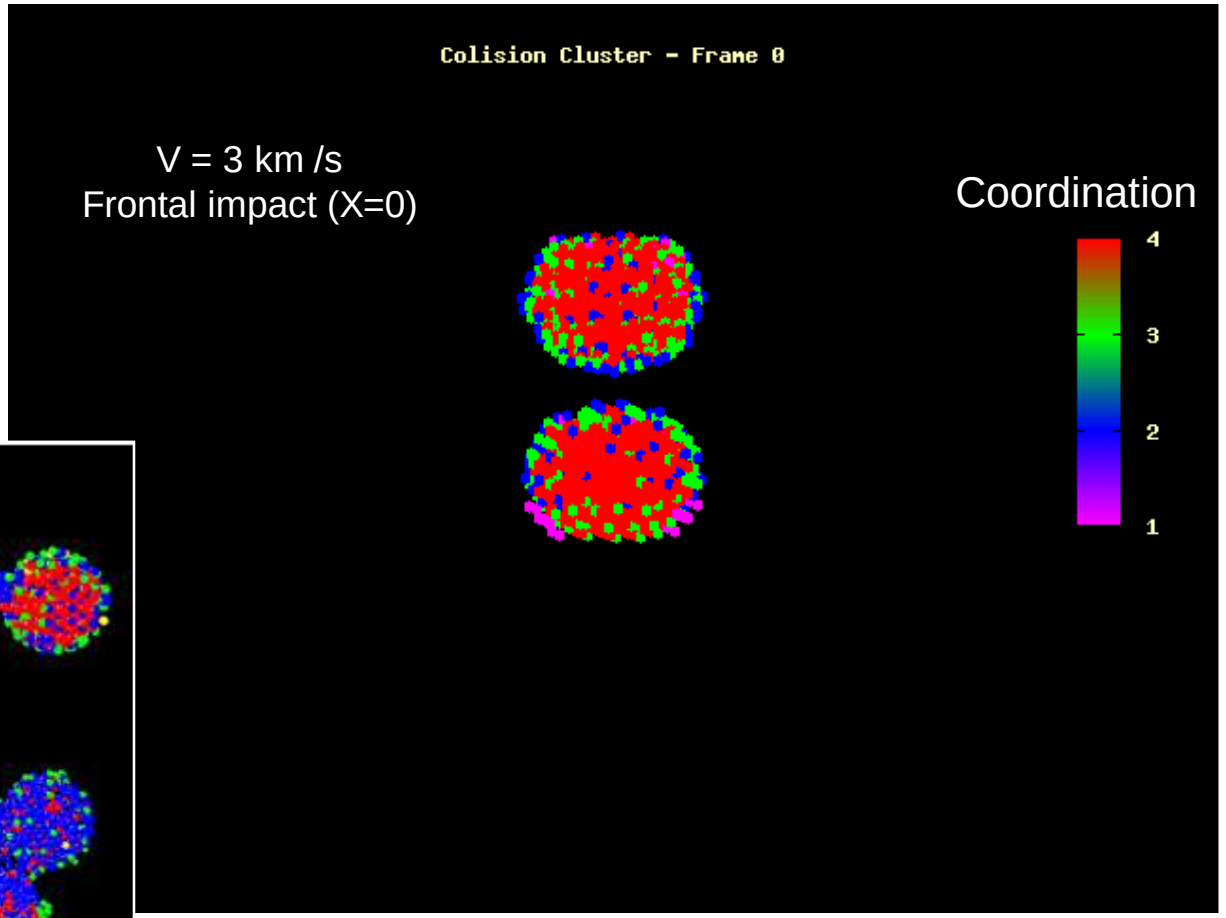
CNA analysis takes about 1/3 of the total analysis time

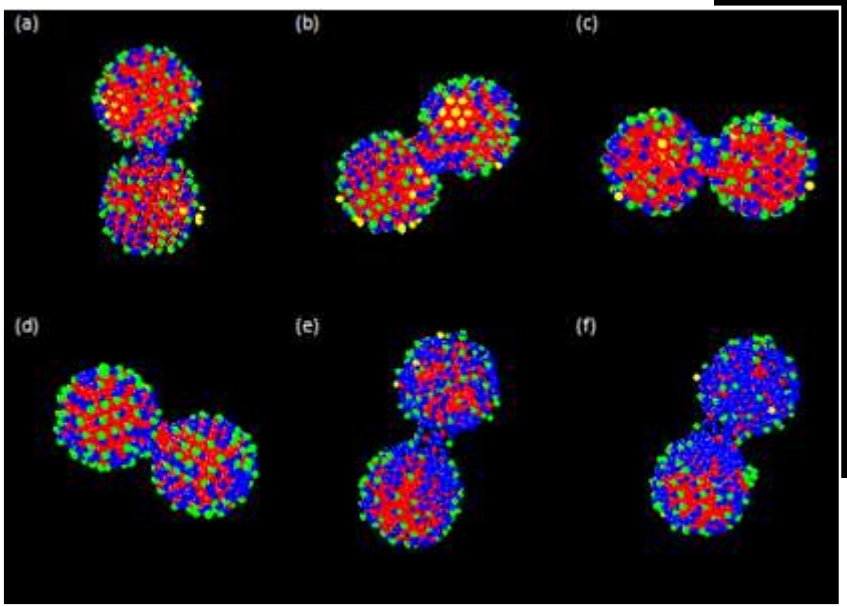
* Ruestes et al. To be submitted to Acta Materialia in 2011

Another example: ATOMIC CLUSTER COLLISION

Diamond clusters





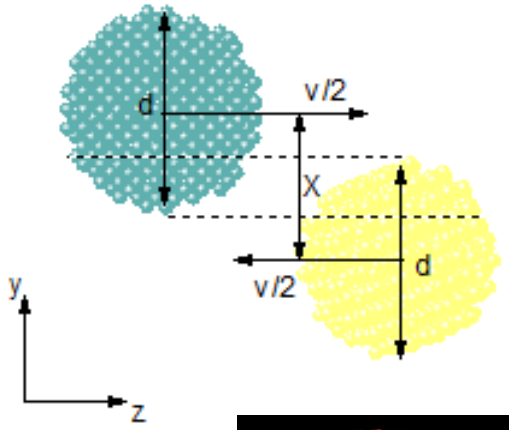


V=2, 2.5, 3, 3.5, 4, 4.5 km/s
X=1,8 nm

Change of coordination:
Diamond → graphite

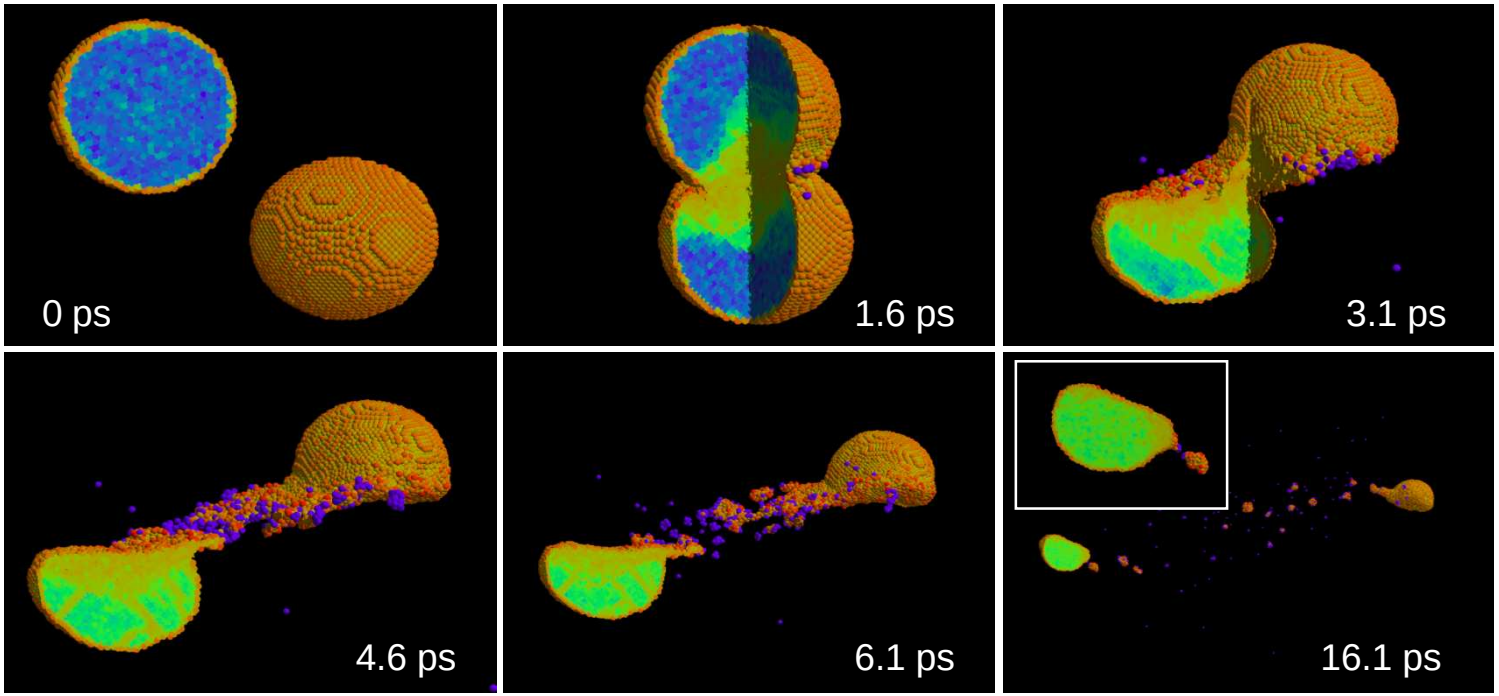
D. Bertoldi and E. Bringa (2010). Run in 20 cores (ITU)

COMPLEXITY in cluster collisions



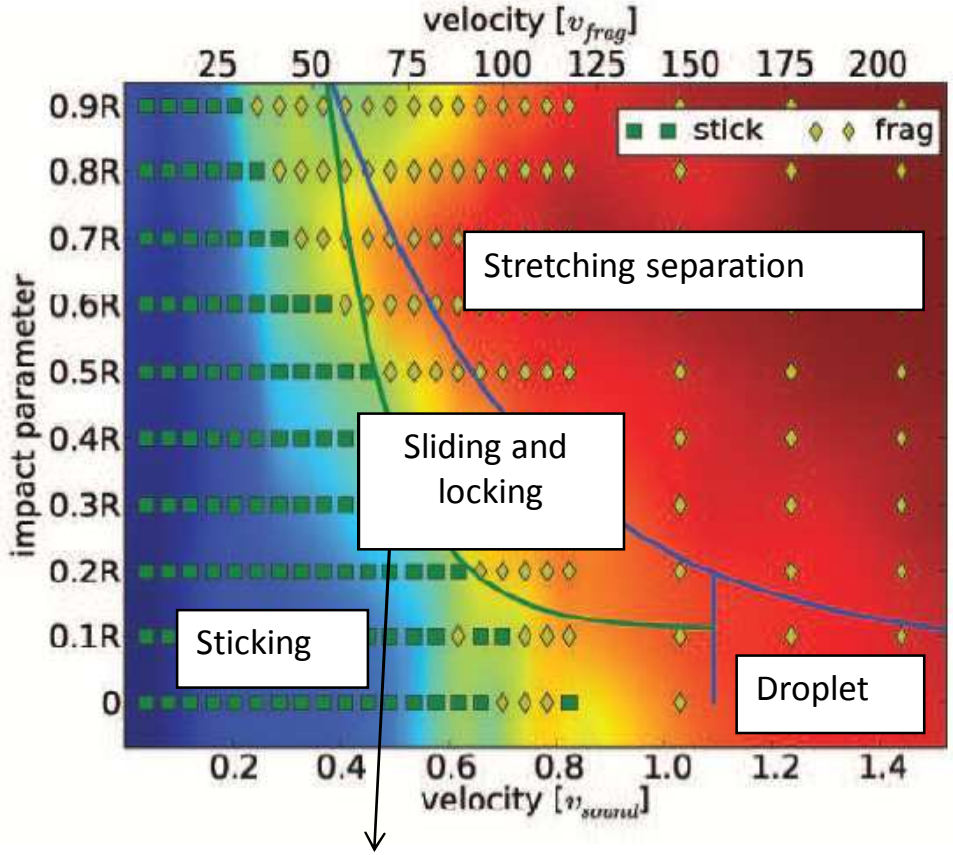
Parameters:

- ✓ **Velocity (v)**
- ✓ **Impact parameter (x)**
- ✓ **Radius (d/2)**
- ✓ **Structure**
- ✓ **Orientation of the lattice**



N. Ohnishi, et al. “Numerical analysis of nanograin collision by classical molecular dynamics,” J. Phys. Conf. Series **112** (2008) 042017. Run in 256 cores (LLNL)

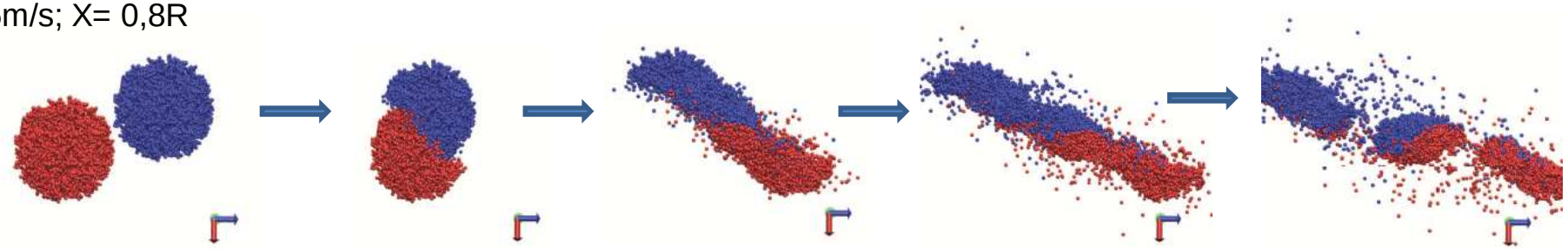
COMPLEX CLUSTER STRUCTURE



Additional parameters:

- ✓ Shape
- ✓ Porosity
- ✓ Speed of sound in the material
- ✓ Fragmentation velocity

V=5m/s; X= 0,8R

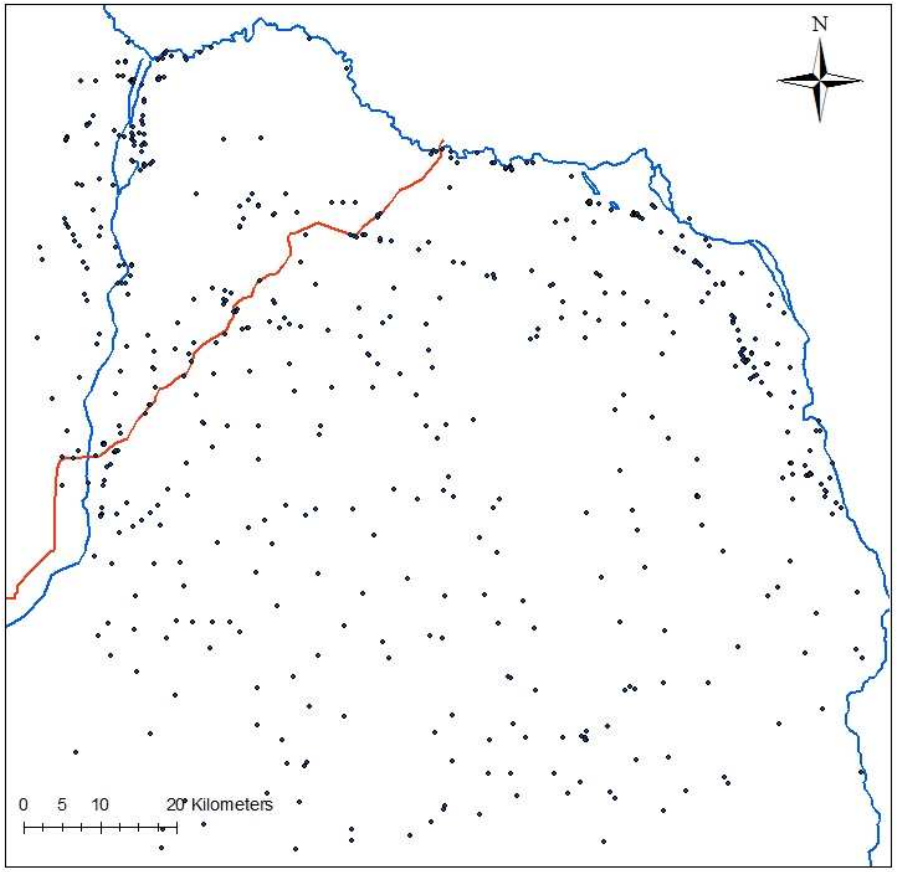


Ringl, Bringa, Bertoldi, and Urbaseek Submitted to Astrophysics Journal (2011). **Run in 48 cores, 128 MB RAM (Germany).**
Kalweit and Drikakis PHYSICAL REVIEW B **74,** 235415 (2006)

Another example: Dynamics of settlement in arid environments

Problem: Which factors influence the livestock post spatial distribution in the NE of Mendoza?

E. Millán, S. Goirán, J. Aranibar



Because livestock spatial distribution changes may negatively affect vegetation and ecosystem sustainability, It is important to understand and predict the consequences of the changes of land use due to spatial pattern of settlements and the environment.

Complexity: large parameter space + neighbor search

Variables:

- Distance to road
- Distance to river
- Settlements distance
- Water table depth
- Vegetation degradation

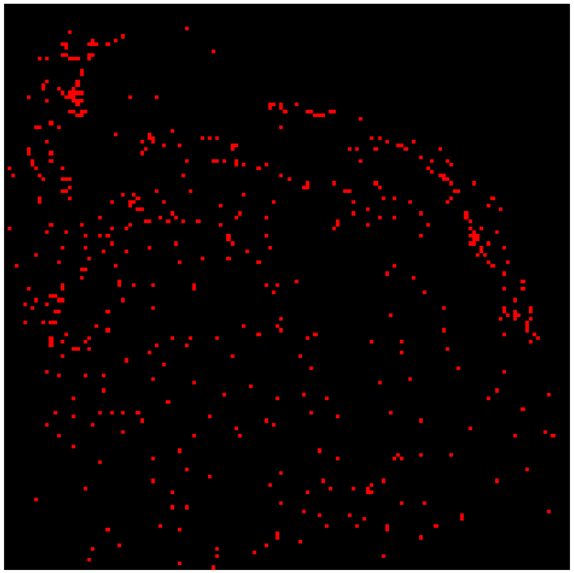
(need neighbors!!)

- **Objetive: Find optimal solution, minimizing error.**

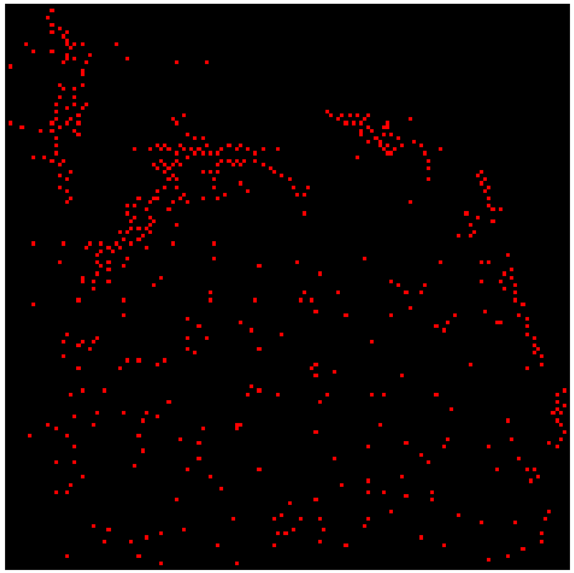
E. Millán, S. Goirán, J. Aranibar

Results: spatial distributions (simulations averaged over 20+ cases)

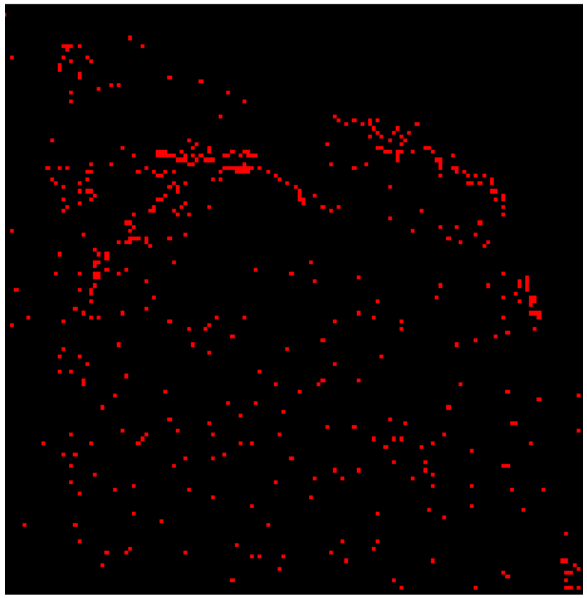
E. Millán, S. Goirán, J. Aranibar



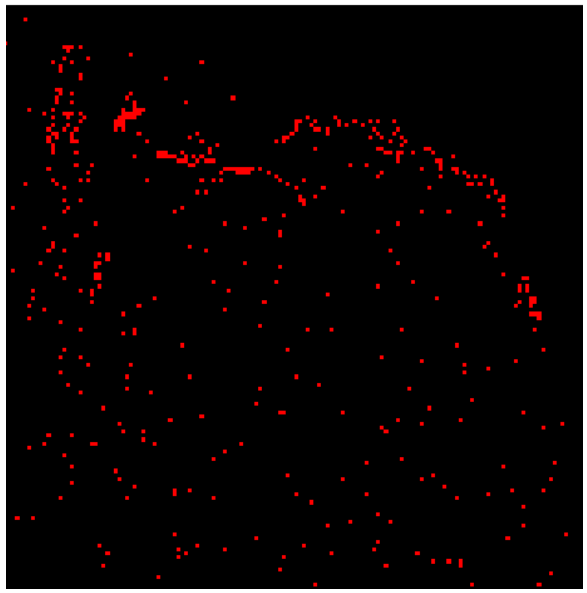
real settlements



output 30

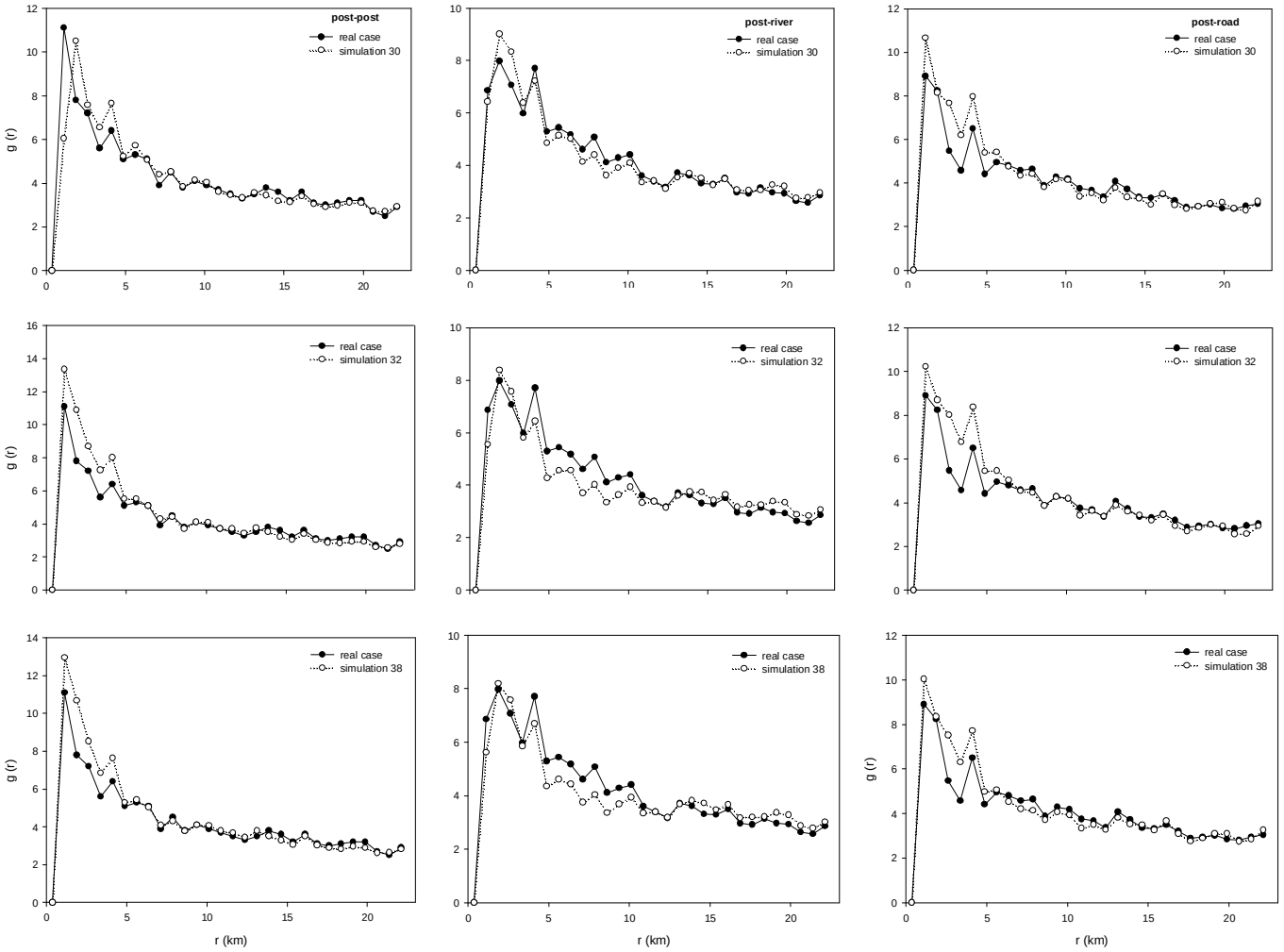


output 32



output 38

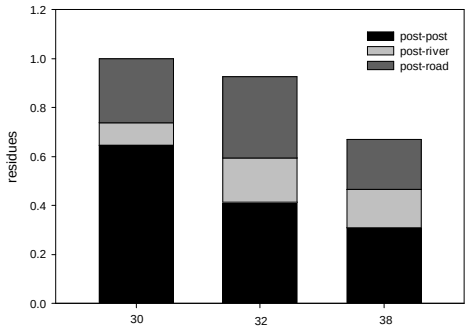
Results: how do I pick the best parameter set?



Settlement exclusion area (1 pixel)

No settlement exclusion area

No settlement exclusion, less probability and greater area of road influence



residues

E. Millán, S. Goirán, J. Aranibar

Summary and future perspectives

- MD simulations often require more CPU time for analysis than for time evolution (months versus days) ➔ processing has to be done in parallel.
- Need smart algorithms which scale well in parallel archs taking advantage of link-cells.
- Need to use parallel viz tools for samples with +few million atoms (generally not the case in chemistry/biology).
- New computers and novel algorithms are allowing comparison of simulation and experimental data.
- GPU processing has bright future!