A DFT nanostructure calculation case study

For BES/NERSC large scale simulation workshop

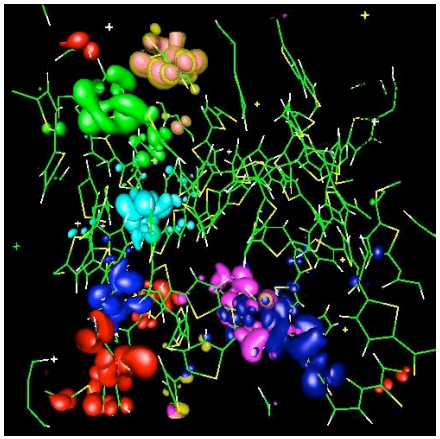
Lin-Wang Wang

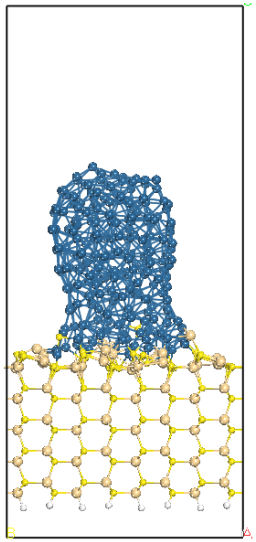
Lawrence Berkeley National Lab

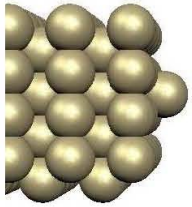
A summary of mp304 NERSC account

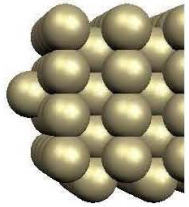
- Allocated and used computer time in 2009: ~ 1 M hours
- Total number of users: ~ 7 active users
- Main codes used: VASP, LAMMP, Petot, Escan, LS3DF
- Number of topics (number of published papers): ~ 15
- Number of processors for typical jobs: 16 to 1000, sometimes 10,000
- Duration of the jobs: 20 minutes, to several hours, to a few days
- The main considerations which determine the jobs we run:
  the physics problem, queue time and computer time.
- Other facilities: no group cluster, INCITE project at NCCS and
  ALCF (but not discussed here).

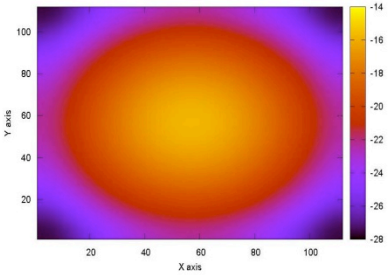
A few examples of topics studied in 2009 under mp304

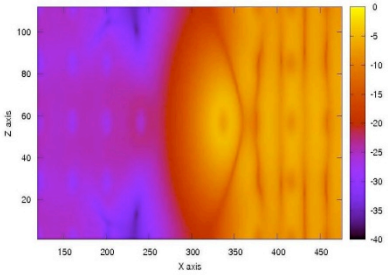


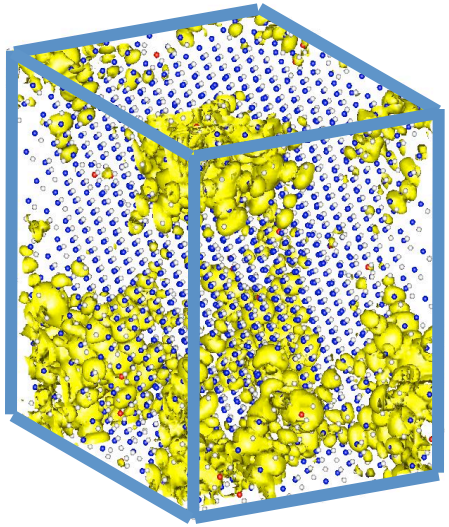


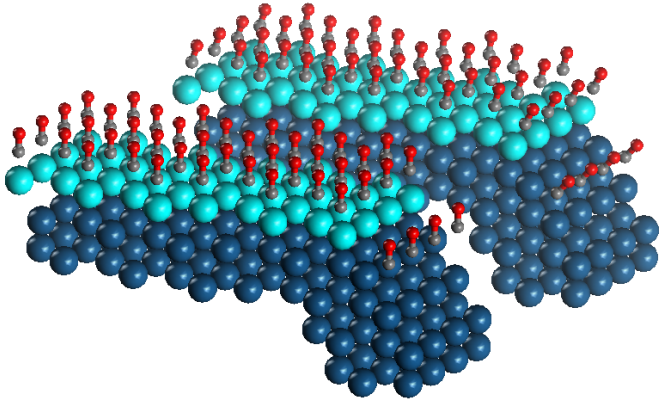


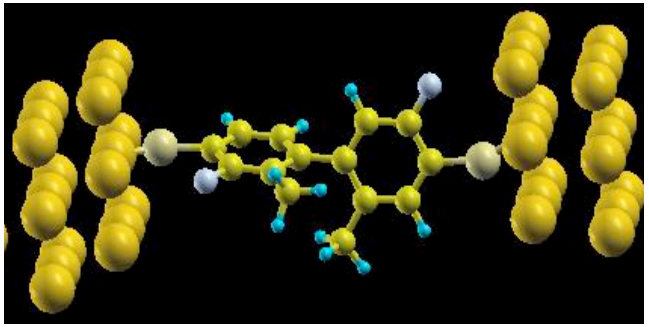




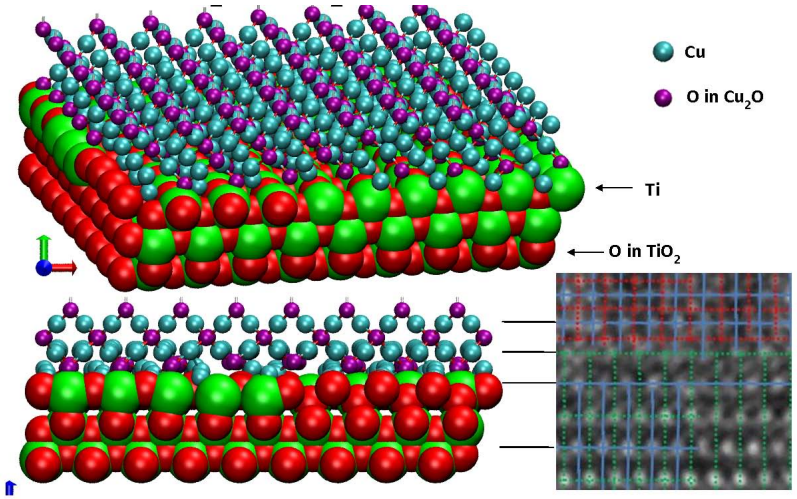


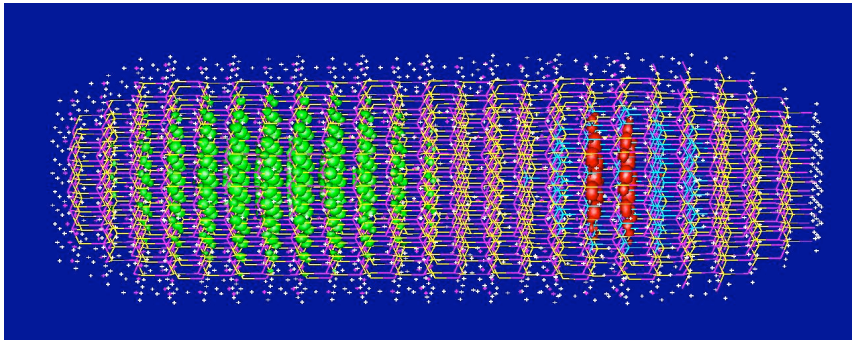


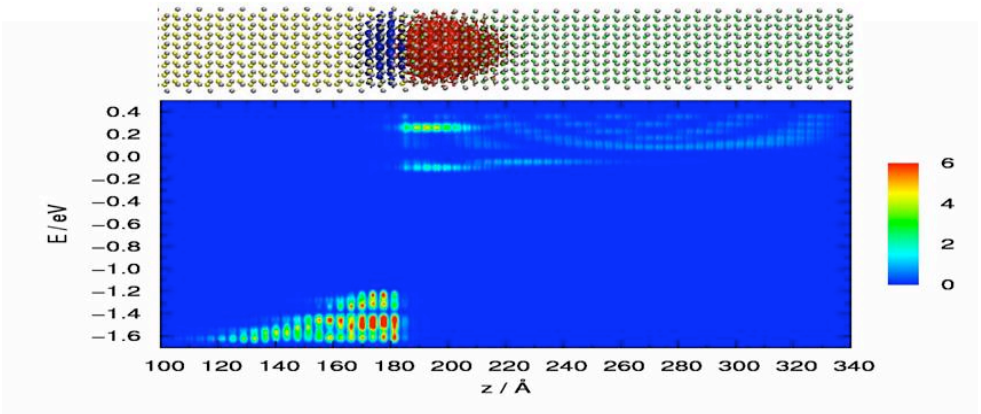


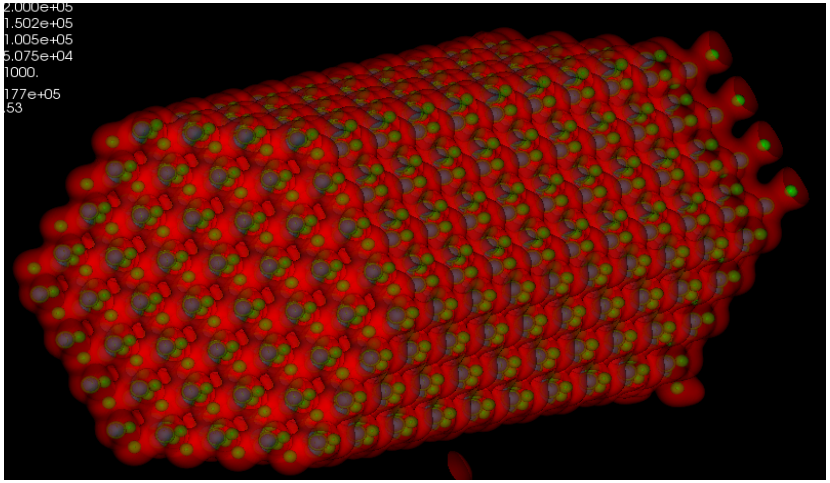


A few examples of topics studied in 2009 under mp304









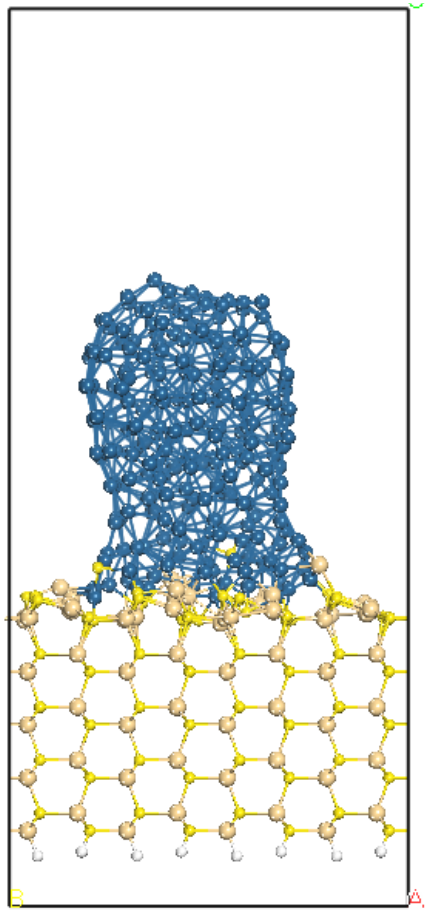
A more detail example: Pt nanocluster on CdS (101-0) surface

- Purpose: to understand the amorphous interface and the resulting Schottky barrier

- Motivation: there are experimental work in LBNL

- Question to address:
  what determines the Schottky barrier? MIGS or interface state?
  How sensitive does the barrier depend on the interface atomic positions?
  Does the nanosize (nanocontact) make any difference compared to bulk Schottky barrier?

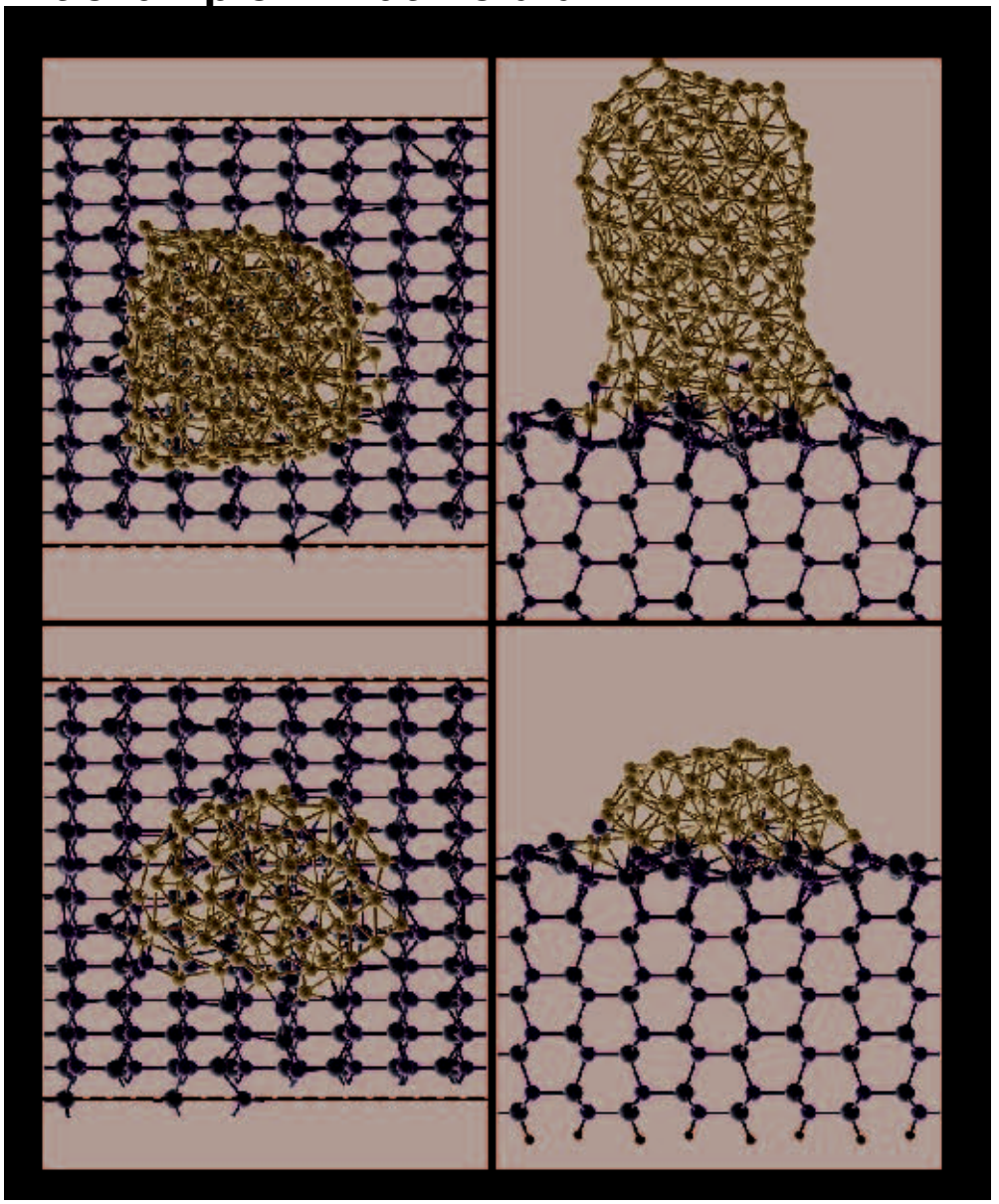
- Approach: DFT molecular dynamics (using VASP), then calculate the local density of states.



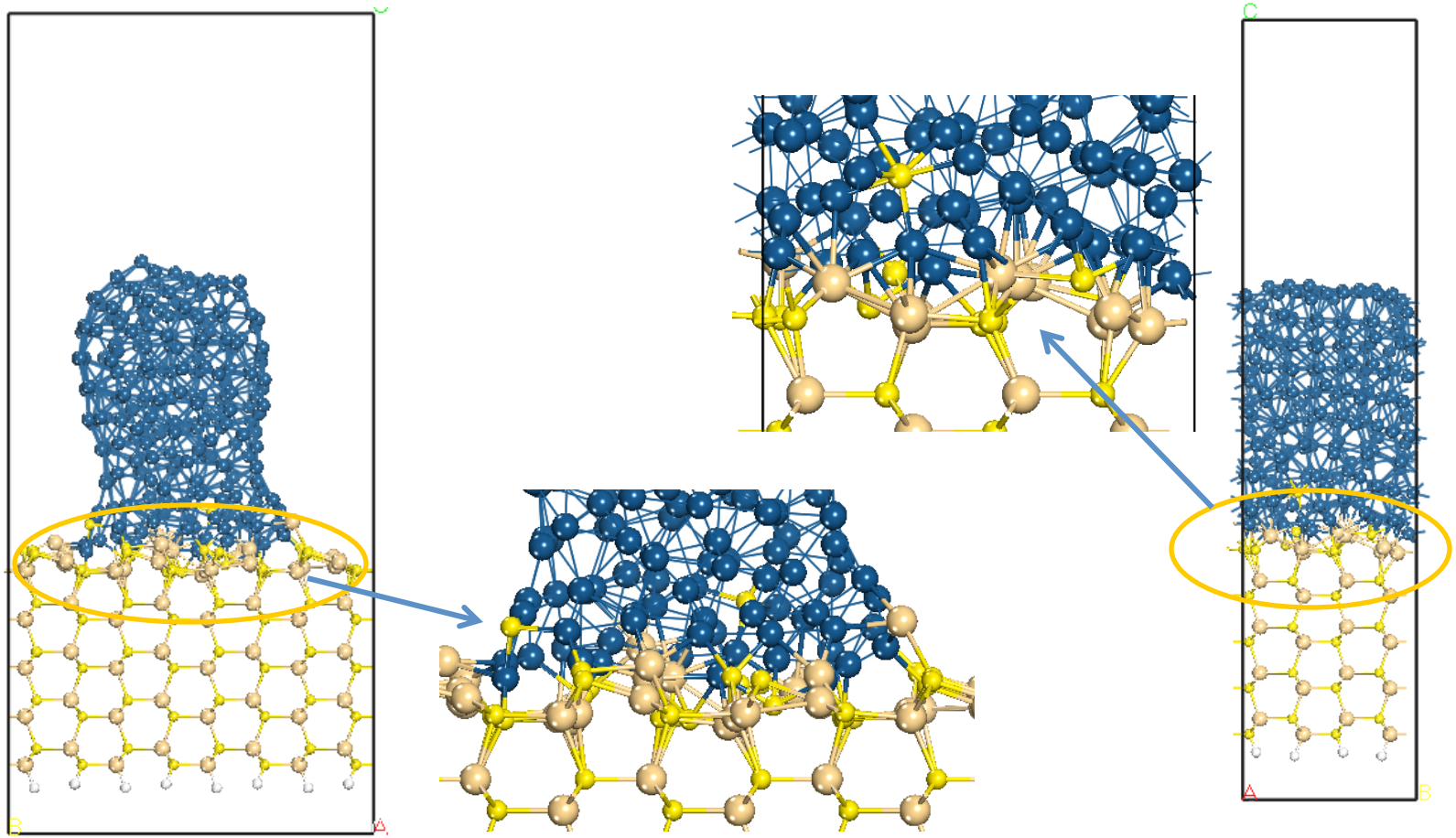
- Challenges: large system size (close to a thousand atoms) and long simulation time.

CdS/Pt example: What we did?

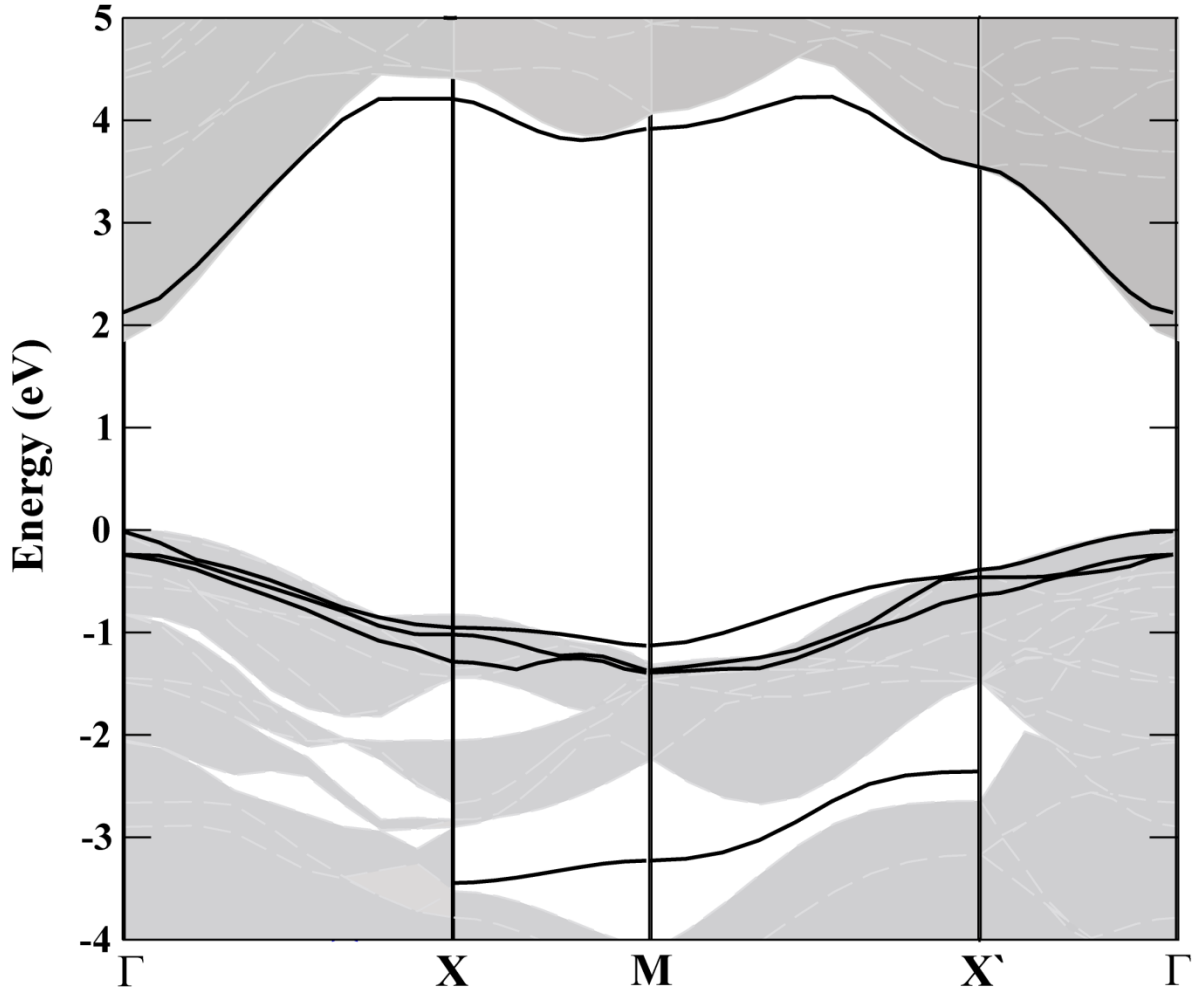
- 528 CdS substrate atoms
- small system: 74 Pt atom
- large system: 246 Pt atom
- 5ps MD simulation for the small system, fix the bottom layers of CdS atoms
- Add more Pt atoms on top, do MD for a separated Pt cluster with bottom Pt atoms fixed.
- Using VASP, MD simulations using ~1000 processors, 100-200 hours.
- The most challenging part: waiting in the queue
- Computer time used: ~ 100,000 hours



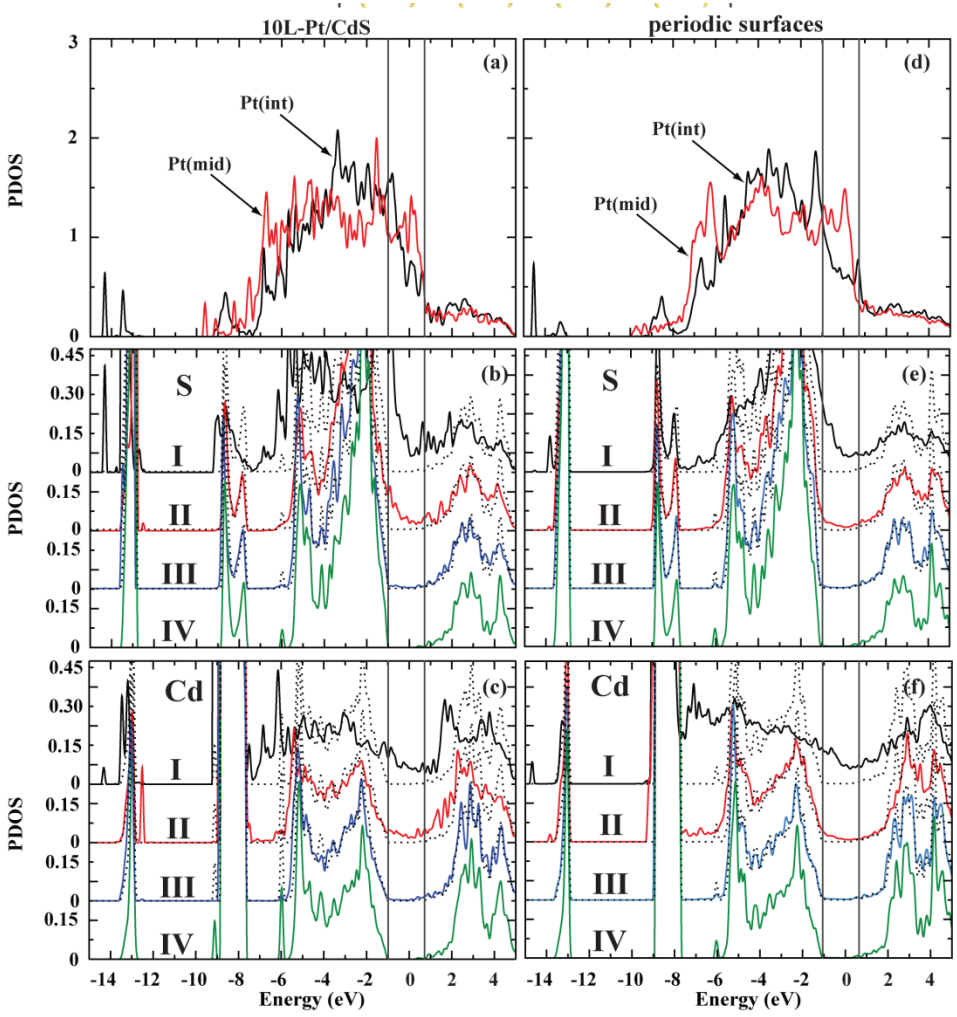
The CdS/Pt example: Cluster versus bulk

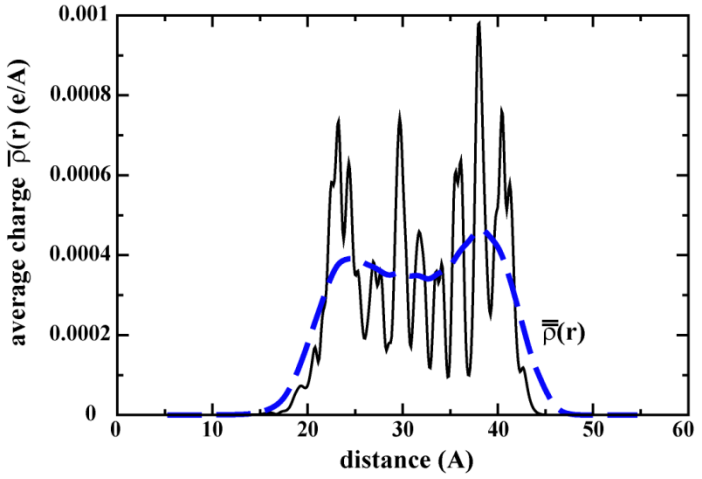


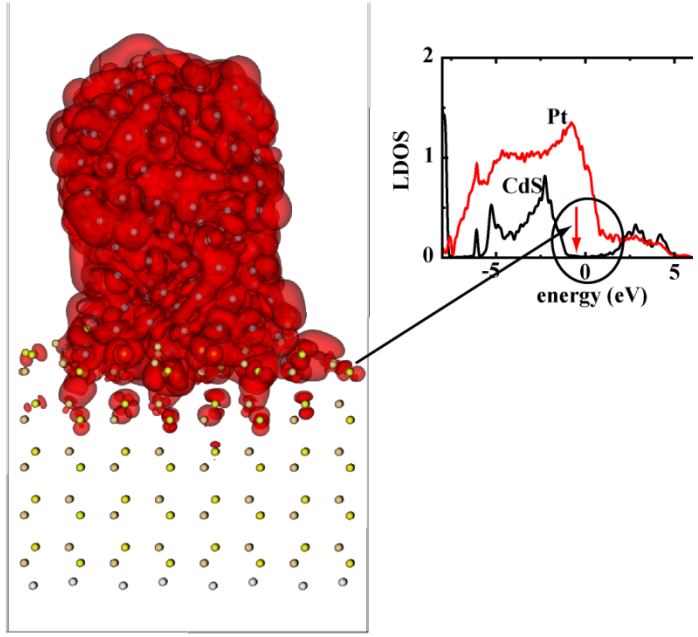
CdS/Pt example: Dimmerized CdS surface band structure



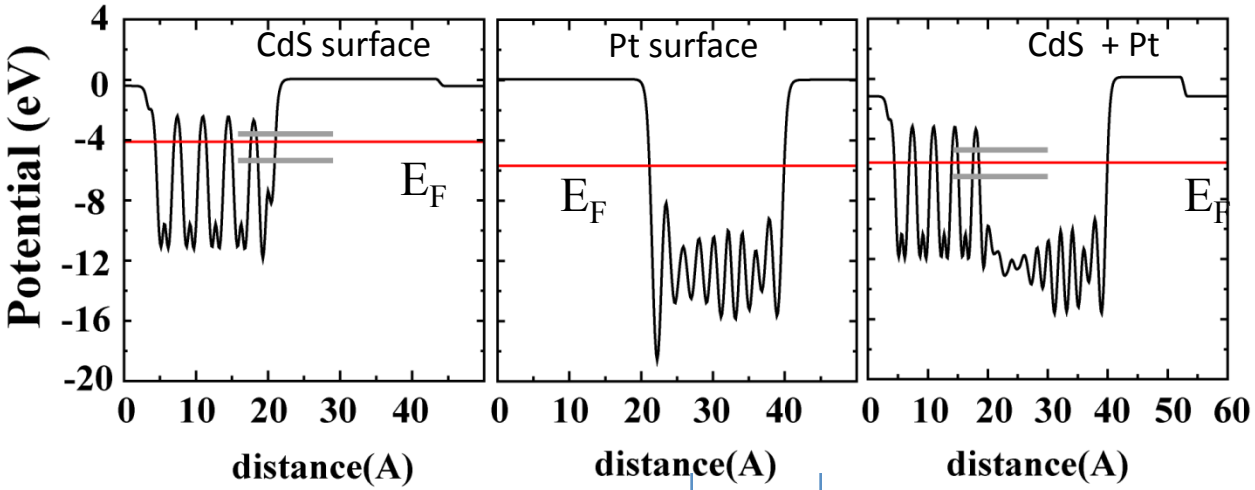
Determine the Schottky barrier by looking at local density of state, and check possible interfacial states.

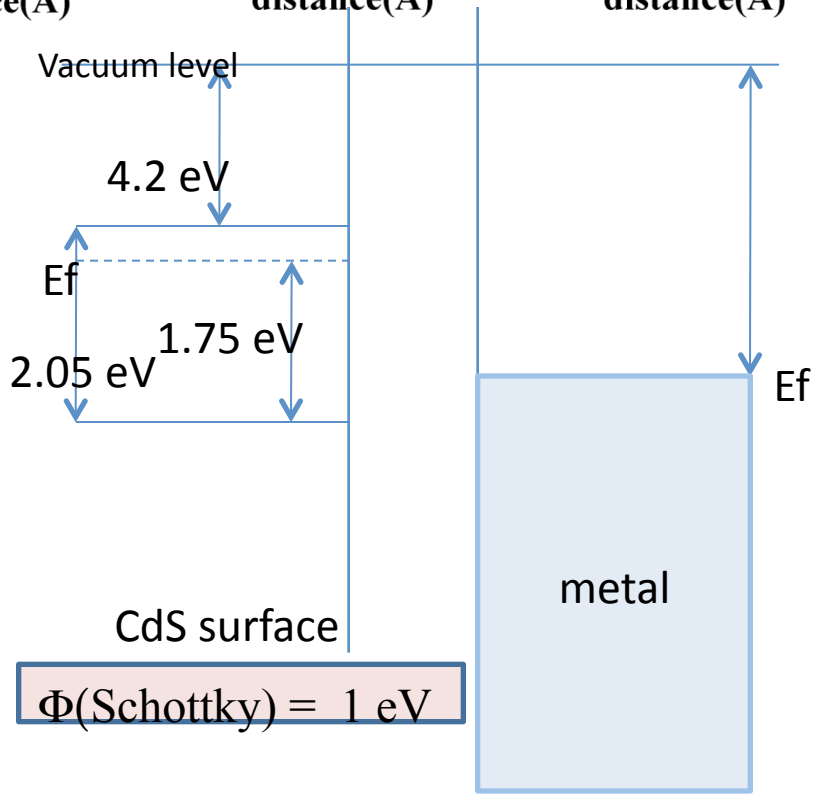






CdSe/Pt example: analysis and conclusions





Future requirement (for next five years)

- What we will do depends on how much computer time and what kind of machine we will have (so the dependence is mutual).

- For mp304, we will see more projects (topics) in the follow areas:

  (1) Colloidal quantum dot surface atomic structure and ligand passivation
  (2) Electron-phonon interaction and carrier dynamics in nanostructures
  (3) Polymer blend atomic and electronic structures
  (4) Alloy and impurity electronic structure calculations
  (5) Coherent+incoherent carrier dynamics

- The methods/algorithms we will use:

  (1) Limited ab initio MD simulations
  (2) Ab initio atomic relaxations for many different configurations, for ~1000 atom systems.
  (3) Long (~10ns) classical MD simulations for 10,000 atom systems
  (4) Electronic structure calculations for large systems (~10,000 atoms) using LS3DF or charge patching methods
  (5) Some approximated time-domain simulations

Future requirement (for next 5 years)

- The code we will use, and computer processors/times (for one typical project):

  (1) VASP: 2000 processors, 100 hours for ab initio MD
  (2) Petot: 2000 processors, 10 hours for electronic structures
  (3) Escan: 5000 processors, 5 hours for electronic structures
  (4) LS3DF: 50,000 processors, 10 hours for electronic structures and atomic relaxations
  (5) LAMMP: 500 processors, 100 hours for classical MD
  (6) new codes for time domain simulation (time, scale unknown!)

- Total annual computer time requirement for mp304: ~5 to 10 M hours

Some thoughts of new algorithms in the future

The communication reliability and power consumption dictate that the future architecture (beyond petascale) is likely consisted with multiprocessor nodes (while the total number of nodes might be the same as the current petascale computer).

Divide and conquer scheme (e.g., LS3DF) might be a good strategy in general, with each local pieces to be solved within each nodes. Maps physical locality into computer architecture locality.

Data-base based algorithm. For example, pre-calculate (or generate on the flight) a large number of “local bonding situations”, which provide the atomic forces, then the MD engine will derive the atomic forces from an interpolation scheme from data sets in the data base.

(in both cases, take the advantage of the physical locality in the system)

More complicated (e.g., explicit double integral based) algorithms for beyond DFT methodology developments?