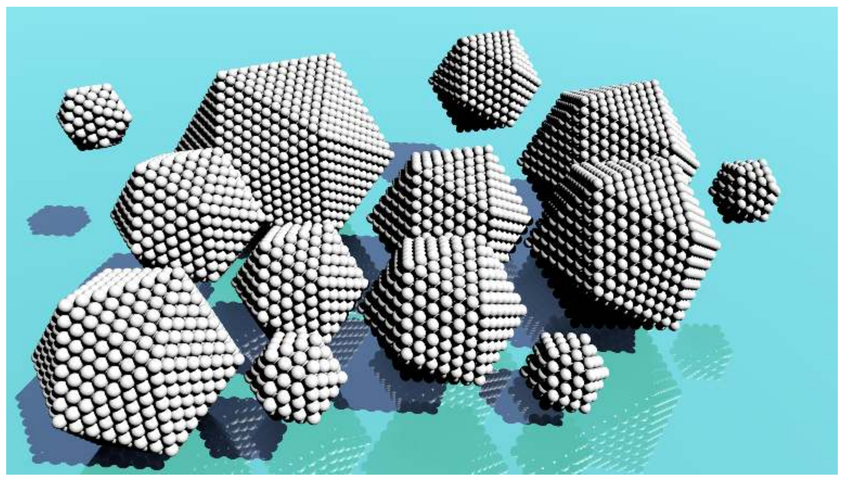
Improving performance of an ab-initio TDDFT program using GPUs: application to large silver clusters

M. Barbry[1], P. Koval[2], D. Sánchez-Portal[1,2]

[1]Centro de Fisíca de Materiales (CFM), CSIC-UPV/EHU, San Sebastian, Spain
[2]Donostia International Physics Center (DIPC), San Sebastian, Spain

Introduction



Electronic excitation, e.g., either caused by irradiation with electrons, conventional or modern light sources (synchrotron, ultrafast lasers), are instrumental for the study of materials, ranging from solids to atoms, from surfaces to nanoscale systems. An improved understanding and prediction of the interaction of radiation with matter is instrumental for the development of new technologies. It is in this framework that we are currently developing an efficient ab-initio program called “mbpt-lcao” (many body perturbation theory with localized basis) that implements TDDFT, GW and BSE calculations for finite systems and is now extend to periodic systems [1]. The implementation of TDDFT in the linear response regime uses the locality of atomic orbitals to expand the wave functions and employs the results of our DFT code Siesta [2], allowing the study of the optical response of systems up to thousands of atoms [3, 4].
The iterative algorithm implemented into this method has an $O(N^3)$ computational complexity, where N is the number of electrons in the system, and requires an $O(N^2)$ memory usage, enabling a relatively fast calculation of the interacting polarizability in plasmonic systems. Although our algorithm possesses a relatively high asymptotic computational complexity, the method is relatively inexpensive in terms of computational resources. We successfully managed to parallelized our code with CUDA and to run some tests with silver clusters of increasing sizes (from 147 to 2057 atoms) that we present here using only few processors. Furthermore, with the same systems, using GPU, we succeed to get a speedup up to 4.

A bit of theory

The main goal of the code is to calculate the density change δn induced by an external perturbation $V_{\text{ext}}(r)$:

$$\delta n(r,\omega) = -\int \chi(r,r',\omega) V_{\text{ext}}(r') dr', \quad (1)$$

where χ is the interacting response function calculated from the non-interacting response function by,

$$\chi = \chi_0 + \chi_0 K \chi \quad (2)$$

The most expensive part of our program is to calculate χ_0, this part scales as $O(N^3)$ with N the number of orbitals. It is these expensive calculations that we parallelized with GPUs.

Previous limitations

In previous work, we worked with Na dimers containing up to 760 atoms and 561 atoms for Ag clusters.

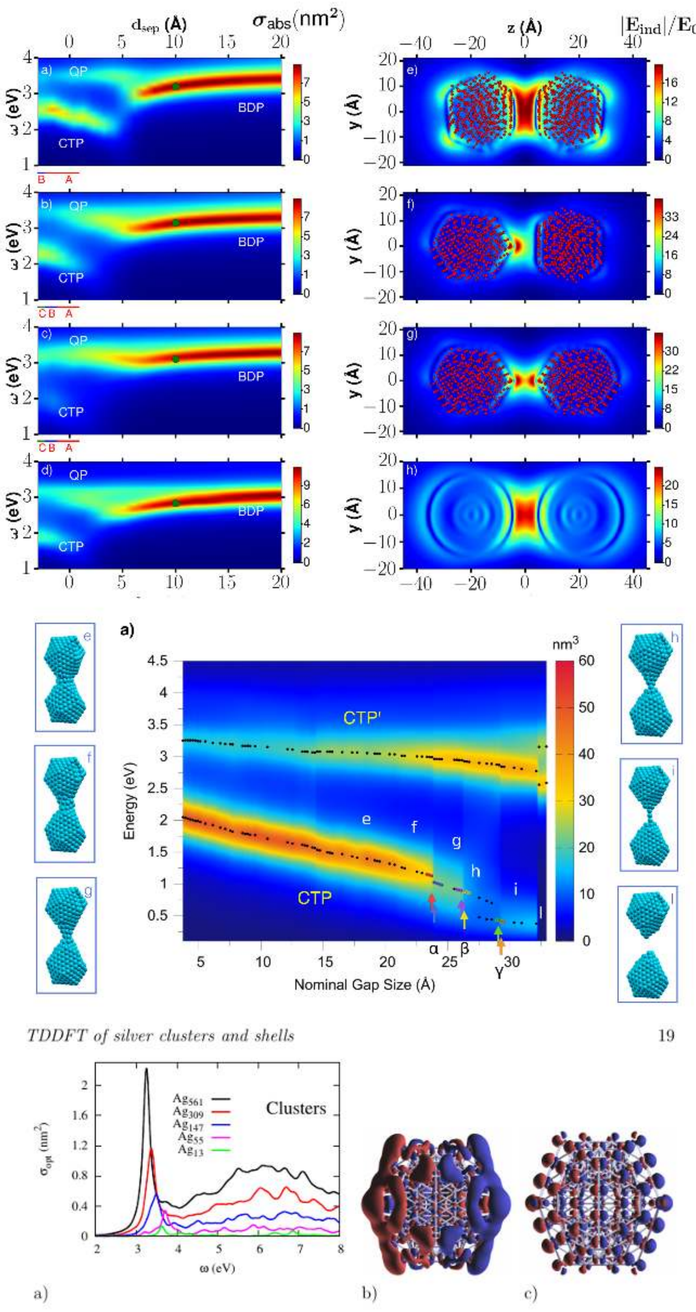


Figure 1 : Top plot represent the polarizability of 380 Na dimers with different cavity sizes (left column) and the electric field distribution along the dimer axis at resonances for a distance of 10 Å(right column) from [3]. The middle plot represent the polarizability of a Na_{380} dimer but with relaxed geometry while the two clusters that have entered into contact are retracted. The pelicular behavior of the charge transfer plasmon mode (CTP) is due to the formation of a connecting metal neck of atomic dimensions [4]. The last row represents the polarizability of Ag clusters (left column) and the density change distributions (right column) from [1].

References

[1] P. koval et al., J. Phys.: Condens. Matter 28, 2016
[2] J. M. Soler et al., J. Phys.: Condens. Matter 14, 2002
[3] M. Barbry et al., Nanolett. 15, 2015
[4] F. Marchesin et al., ACS Photonics 3, 2016

Latest improvements: Silver cluster up to 2057 atoms

We have been working hard on the improvement of the code, mainly concerning the memory consumption. And we managed to run calculations for icosahedral Ag clusters up to 2057 atoms with one node and using less than 120GB of memory (with the improved memory requirements new memory and the parallel method using MPI, we hope to be soon able to run up to 5000 atoms).

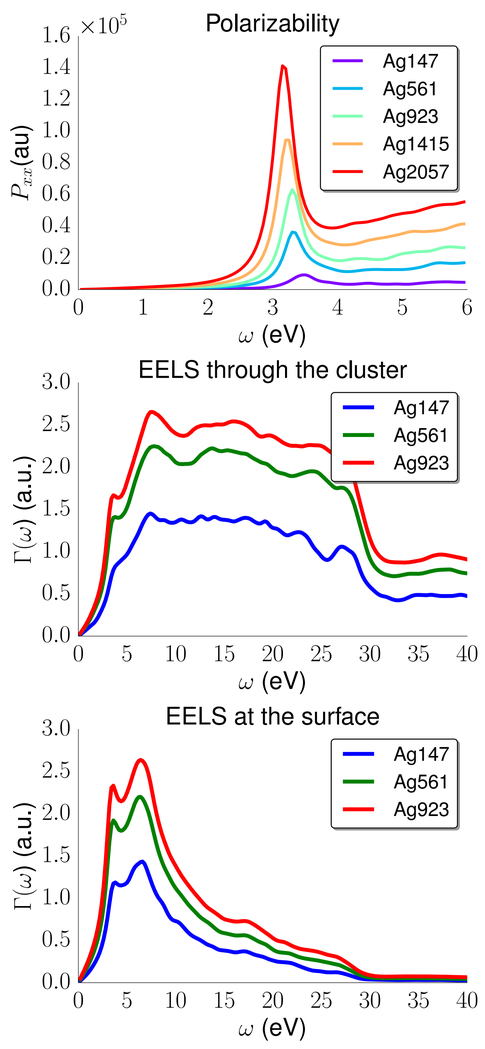


Figure 2 : The top row represents the polarizabilty of icosahedral Ag clusters from 147 to 2057 atoms. The second and third row represent the Electron Energy Loss Spectroscopy (EELS) signal for Ag cluster from 147 to 923 atoms for an electron velocity of 5.0 a.u. The middle row is the loss probability for an electron passing through the center of the clusters, while the last row is for an electron passing at 3.0 Bohr from the surface.

CUDA implementation

The iterative procedure of our implementation is done for each frequency.

```
do i=f1, f2
  call Chi0_mv(args)
  call fini_Chi_Solver(args)
enddo
```

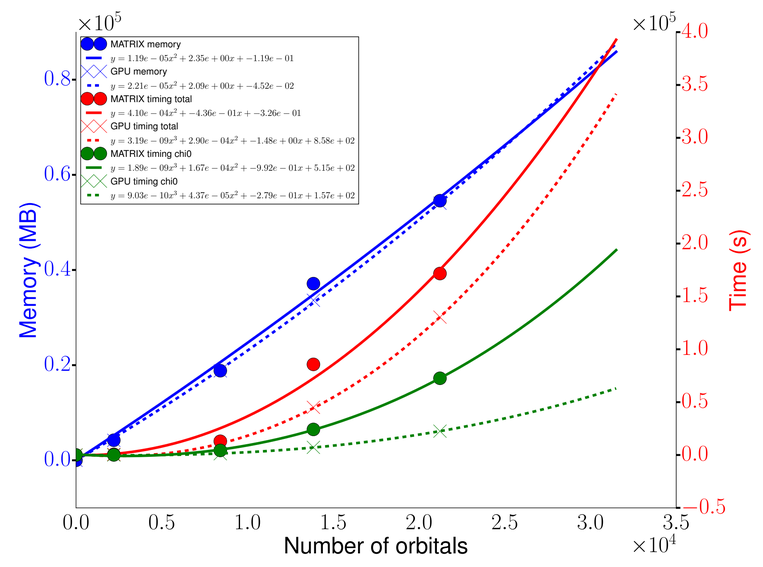
The Chi0_mv subroutine is the one calling CUDA for solving heavy matrix operations (GEMM). The subroutine is the following

```
subroutine Chi0_mv(args)
  call calculate_VV_matrix(args)
  call GEMM(VV, XVV)
  call GEMM(XVV, XXVV)
  call temp_effect(XXVV)
  call GEMM(XXVV, XVV)
  call GEMM(XVV, VV)
  call calculate_Chi0(VV, Chi0)
end subroutine
```

To improve performance, we allocate on the device all the matrices once before the calculations and transfer the constant matrices as well only once. Obviously, this method requires more memory on the device, but it avoids the constant data transfer between host and device memory.

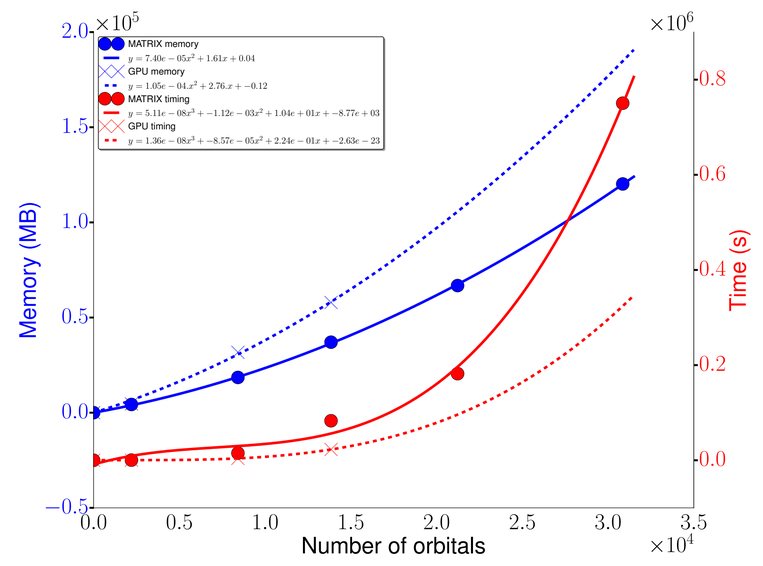
GPU Benchmark with silver clusters: 1 Tesla K40

The computer used for these tests has 12 CPUs Intel(R) Xeon(R) CPU E5-2630 v2 at 2.60GHz with a cache of 15.360 MB, 275 GB of Memory and 1 Tesla K40 GPU (The Tesla K40 used for these calculations was donated by the NVIDIA Corporation.). We performed tests for silver clusters going from a size of 147 to 1415 atoms. With this computer, we tested the MATRIX parallel method. In this case Openmp is used inside of the frequency loop, in particular for Blas routines. In this case only one GPU can be used, but this method uses much less memory.



GPU Benchmark with silver clusters: 2 Tesla K80

The computer used for this tests has 24 CPUs Intel(R) Xeon(R) CPU E5-2680 v3 at 2.50GHz with a cache of 30.0 MB, 120 GB of Memory and 2 Tesla K80 GPUs (The computer use for this calculations are from the CESGA center). We performed tests for silver clusters going from a size of 147 to 923 atoms. With this Computer we tested the FREQ parallel method of the code. In this case the frequency loop is paralleled with Openmp. Therefore, multiple GPU can be used (in this case 4) giving maximal performance but limitation in memory.



Conclusions

We presented in this poster

- The recent progress that we performed in the code that we are implementing. Recent progress that allows the study of very large silver clusters that could not be addressed in the past with such small computational resources. We believe that we will be soon able to perform calculations with systems up to 5000 atoms.
- We also showed how we could increase the performances of our calculation by using GPU instead of CPU at the bottleneck of the program. By doing so, we could increase the performance by a factor 4.