

Loriano Storchi

**HPC in Chemistry
and in Physics**

Contents

- Grid Computing
- Green's function and the ADC
- Relativistic DFT
- GRID force-field pKa prediction
- GPGPU and DSP
- Sphere packing for Dye-sensitized solar cells

Contents

- Grid Computing
- Green's function and the ADC
- Relativistic DFT
- GRID force-field pKa prediction
- GPGPU and DSP
- Sphere packing for Dye-sensitized solar cells

Grid Computing - Introduction

Single Machine

APPLICATIONS

OPERATING SYSTEM

HARDWARE

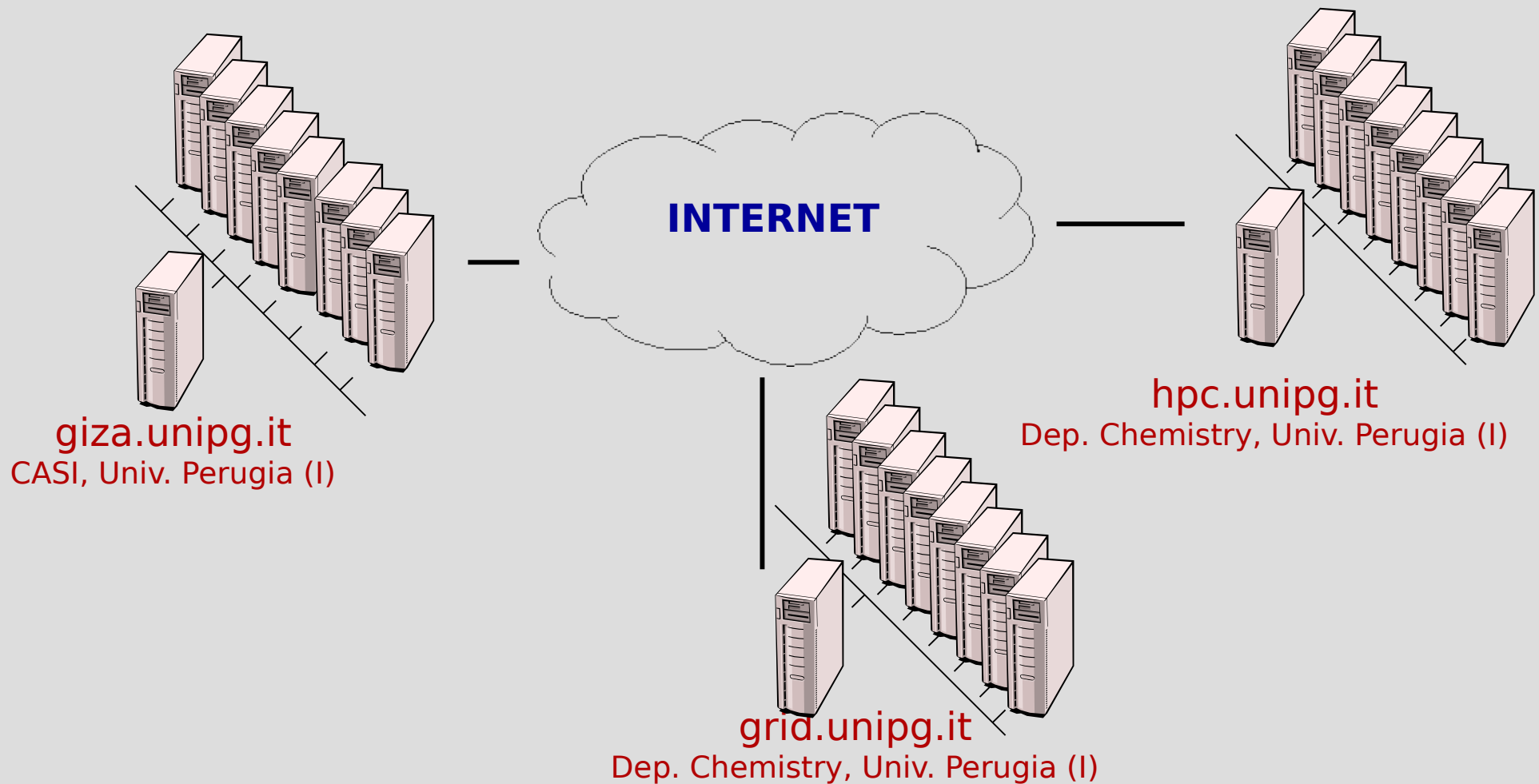
Grid

APPLICATIONS

MIDDLEWARE

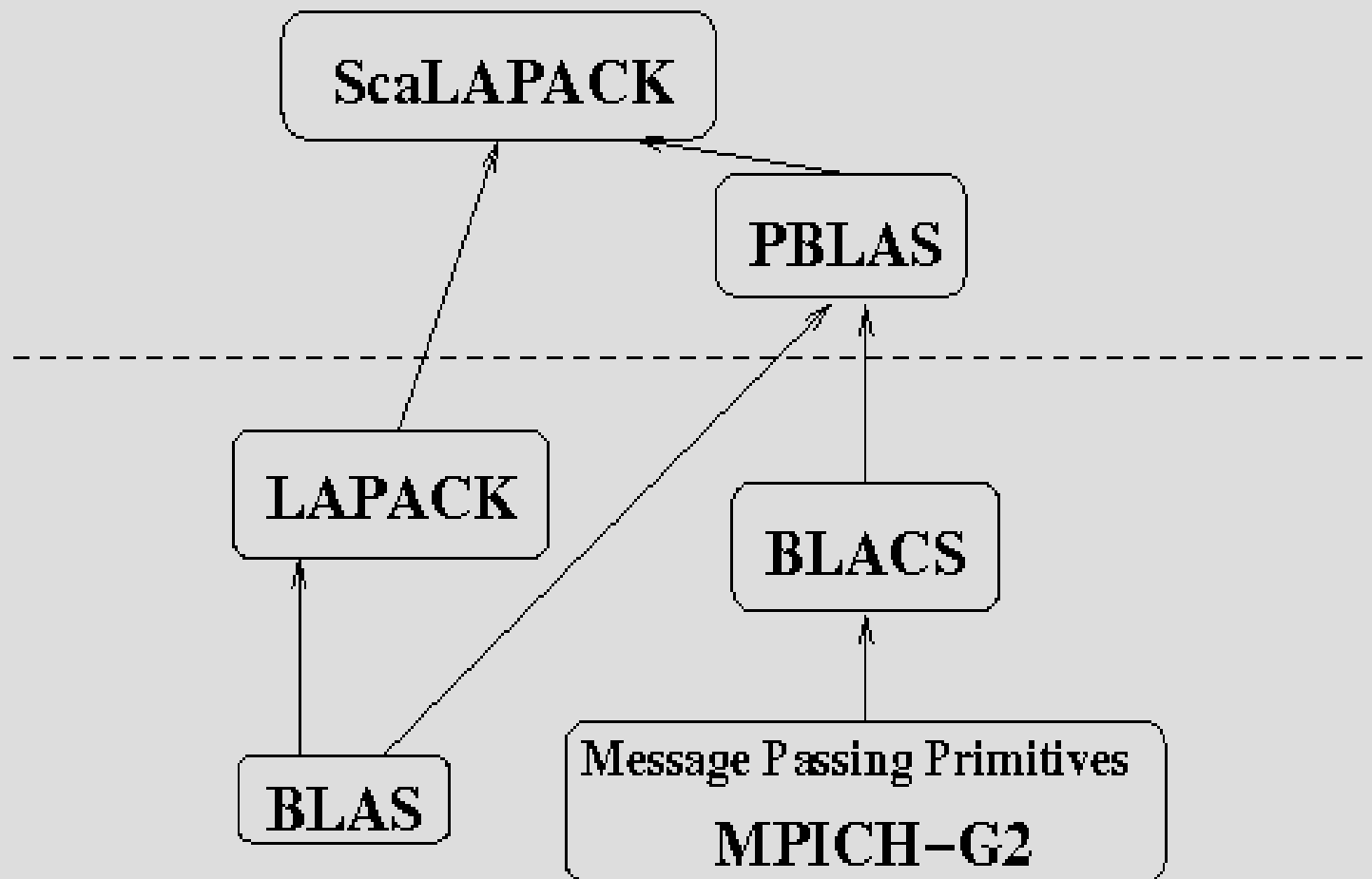
HARDWARE (NETWORK +
MACHINES)

Grid Computing - Model Grid platforms and Applications



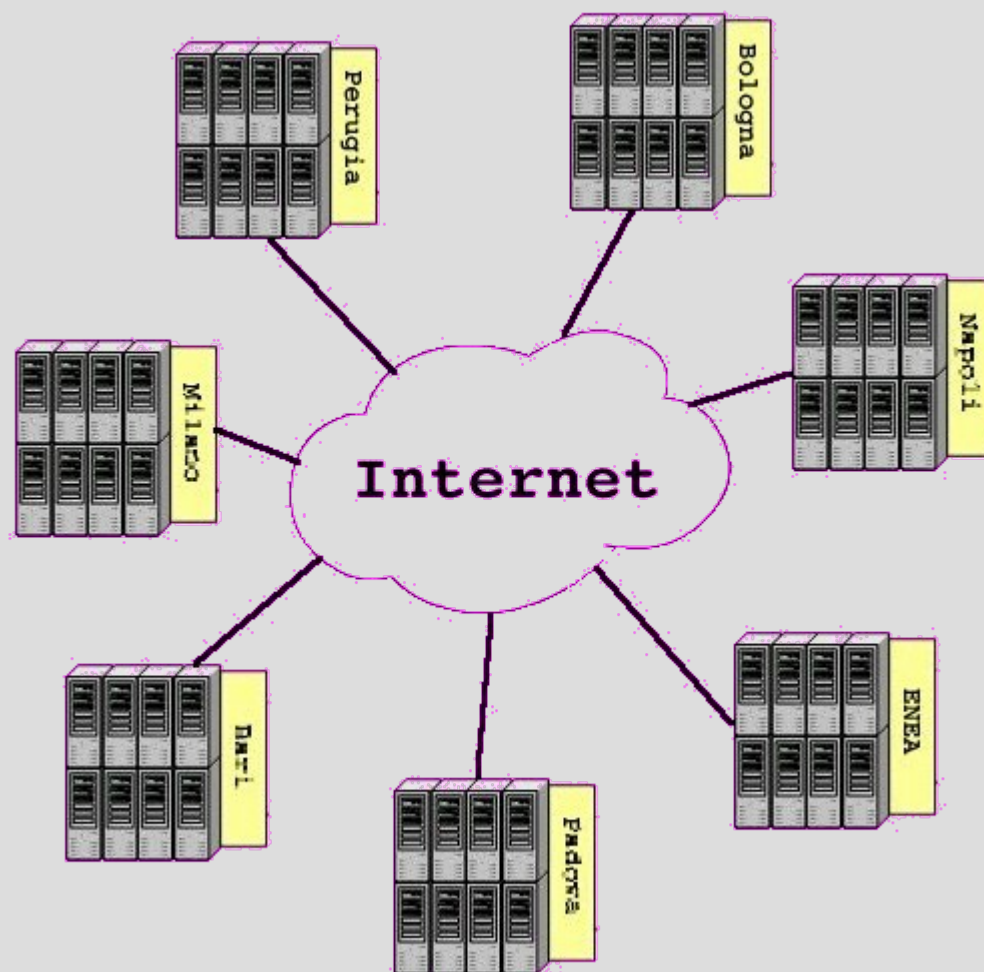
Grid Computing - Model Grid platforms and Applications

Software Hierarchy

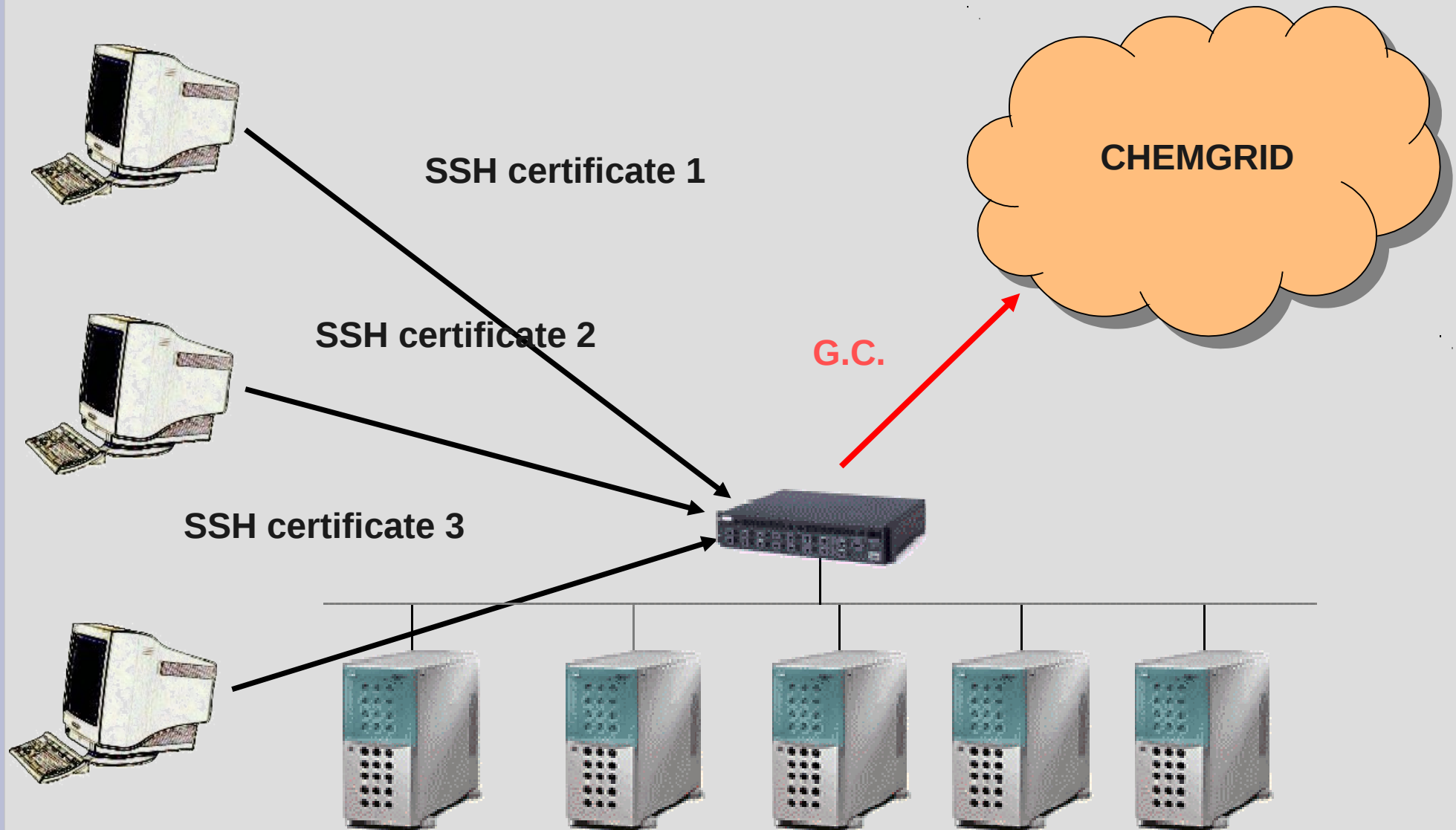


Using a big enough matrix we obtained an effective speed of 2.6 Gflops

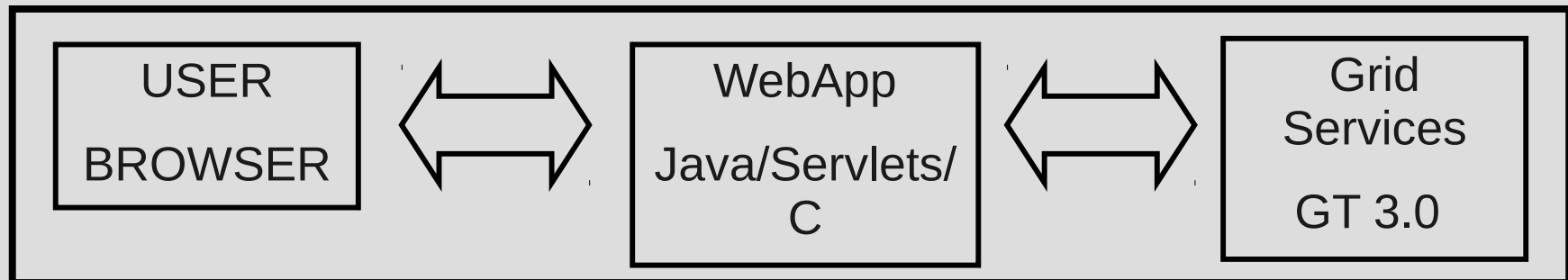
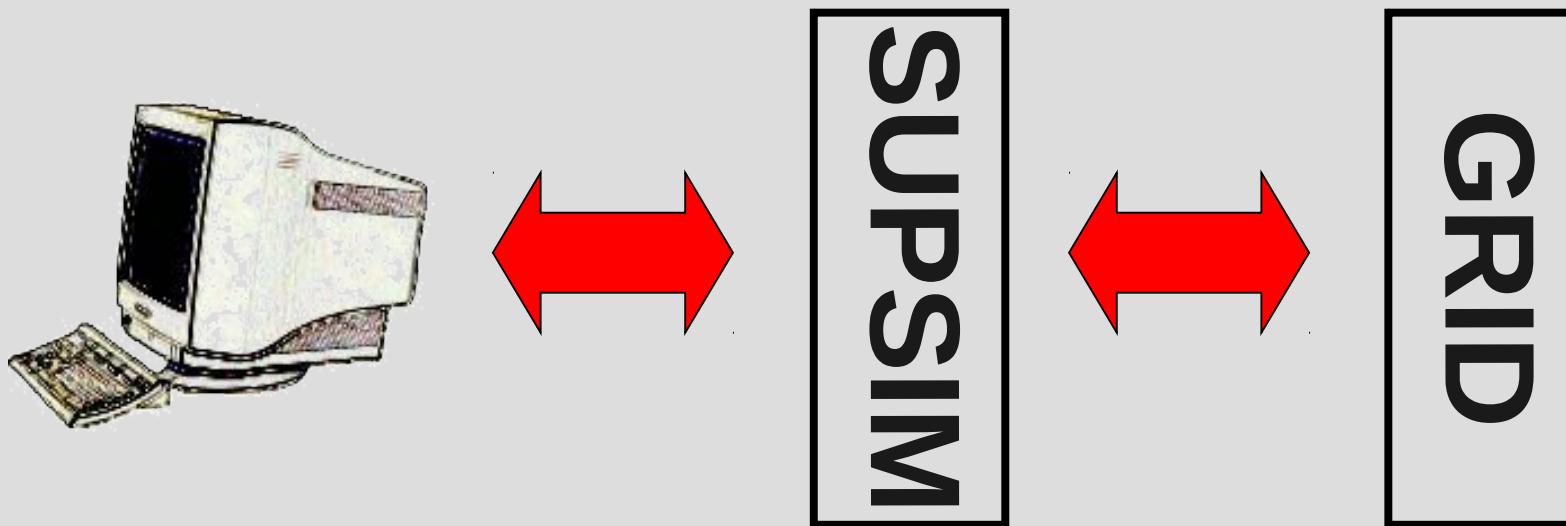
Grid Computing- Model Grid platforms and Applications



Grid Computing- Model Grid platforms and Applications



Grid Computing- Model Grid platforms and Applications




Grid Computing- Model Grid platforms and Applications

File Edit View Go Bookmarks Tools Window Help


http://banquo:8080/supsim/servlet/post/calc/SubCalc Search

Mail AIM Home Radio My Netscape Search Bookmarks



Università degli Studi di Perugia
Chemistry Department
Mathematics and Informatics Department
Simulation of Molecular Beam Experiments

Main Menu
If new, please register

Partners


Set grid and spin multiplicity

Grid:

	min.	max.	points
R ₁ :	<input type="text" value="0.900000"/>	<input type="text" value="2.000000"/>	<input type="text" value="5"/>
R ₂ :	<input type="text" value="0.900000"/>	<input type="text" value="2.000000"/>	<input type="text" value="5"/>
Alpha:	<input type="text" value="10.0"/>	<input type="text" value="180.0"/>	<input type="text" value="2"/>

Spin Multiplicity:

Atoms

H C O

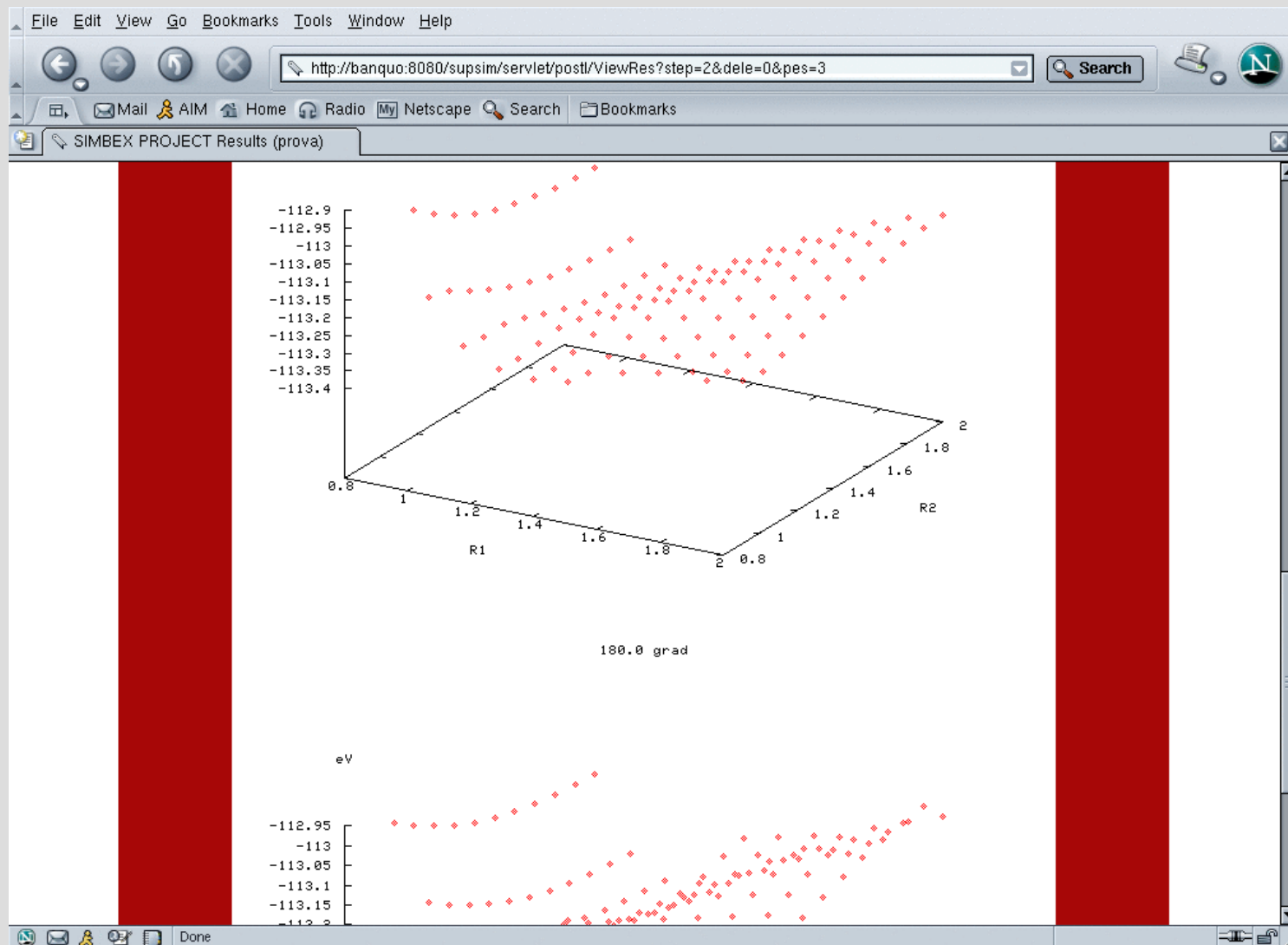
Molecules

HCO

Reset Submit

Done

Grid Computing- Model Grid platforms and Applications



Contents

- Grid Computing
- Green's function and the ADC
- Relativistic DFT
- GRID force-field pKa prediction
- GPGPU and DSP
- Sphere packing for Dye-sensitized solar cells

ADC(3), Non-Dyson scheme

$$G_{pq}(\omega) = \sum_{n \in N+1} \frac{\langle \Psi_0^N | \hat{a}_p | \Psi_n^{N+1} \rangle \cdot \langle \Psi_n^{N+1} | \hat{a}_q^\dagger | \Psi_0^N \rangle}{\omega + E_0^N - E_n^{N+1}} + \sum_{n \in N-1} \frac{\langle \Psi_0^N | \hat{a}_q^\dagger | \Psi_n^{N-1} \rangle \cdot \langle \Psi_n^{N-1} | \hat{a}_p | \Psi_0^N \rangle}{\omega + E_n^{N-1} - E_0}$$

$$\mathbf{G}(\omega) = \mathbf{G}^-(\omega) + \mathbf{G}^+(\omega) = \mathbf{x}^+ (\omega \mathbf{1} - \Omega)^{-1} \mathbf{x}$$

ADC(3), Non-Dyson scheme

The Dyson equation

$$\mathbf{G}(\omega) = \mathbf{G}^0(\omega) + \mathbf{G}^0(\omega) \Sigma(\omega) \mathbf{G}(\omega)$$

$$\Sigma(\omega) = \Sigma(\infty) + \mathbf{M}(\omega)$$

The ADC

$$M_{pq}(\omega) = U_p^+ (\omega \mathbf{1} - \mathbf{K} - \mathbf{C})^{-1} U_q$$

$$U_q = U_q^{(1)} + U_q^{(2)} + \dots$$

$$\mathbf{C} = \mathbf{C}^{(1)} + \mathbf{C}^{(2)} + \dots$$

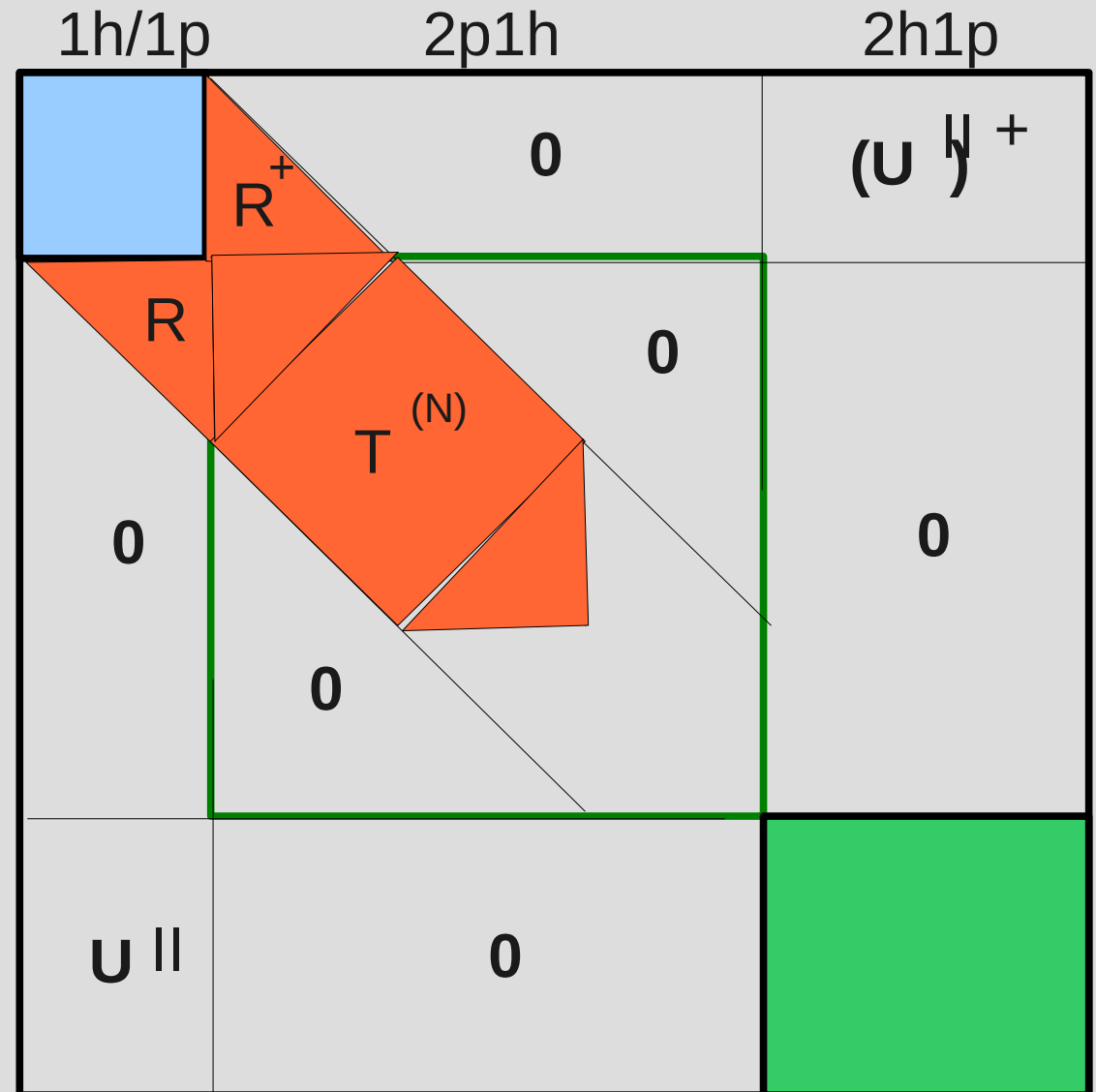
ADC(3), Non-Dyson scheme

The ADC(3) Matrix

	1h/1p	2p1h	2h1p
		(U^+)	(U^+)
U^I			0
U^{II}		0	

ADC(3), Non-Dyson scheme

The ADC(3) Matrix
(with affinity reduction)



ADC(3), Non-Dyson scheme

$$\mathbf{G}(\omega) = \mathbf{G}^{-}(\omega) + \mathbf{G}^{+}(\omega) = \mathbf{x}^{+}(\omega \mathbf{1} - \Omega)^{-1} \mathbf{x}$$

We adopt the ADC approach directly for the (N-1)-particle part of the one-particle

GF

$$\mathbf{G}^{-}(\omega) = \left(\mathbf{x}^I \right)^{+} (\omega \mathbf{1} - \Omega^I)^{-1} \mathbf{x}^I$$

$$\mathbf{G}^{-}(\omega) = \mathbf{f}^{+}(\omega \mathbf{1} - \mathbf{K} - \mathbf{C})^{-1} \mathbf{f}$$

$$\mathbf{f} = \mathbf{f}^{(0)} + \mathbf{f}^{(1)} + \dots$$

$$\mathbf{C} = \mathbf{C}^{(1)} + \mathbf{C}^{(2)} + \dots$$

ADC(3), Non-Dyson scheme

The nD-ADC(3) Matrix

1h	2h1p
$\mathbf{K}_1 + \mathbf{C}_{11}^{(2-3)}$	$\mathbf{C}_{12}^{(1-2)}$
$\mathbf{C}_{21}^{(1-2)}$	$\mathbf{K}_2 + \mathbf{C}_{22}^{(2-3)}$

ADC(3), Non-Dyson scheme

nD-ADC(3) and ADC(3) direct comparison

Dyson ADC(3)

$$\mathbf{K}_1 + \Sigma(\infty)$$

Non-Dyson ADC(3)

$$\mathbf{K}_1 + \mathbf{C}_{11}^{(2)} + \mathbf{C}_{11}^{(3)}$$

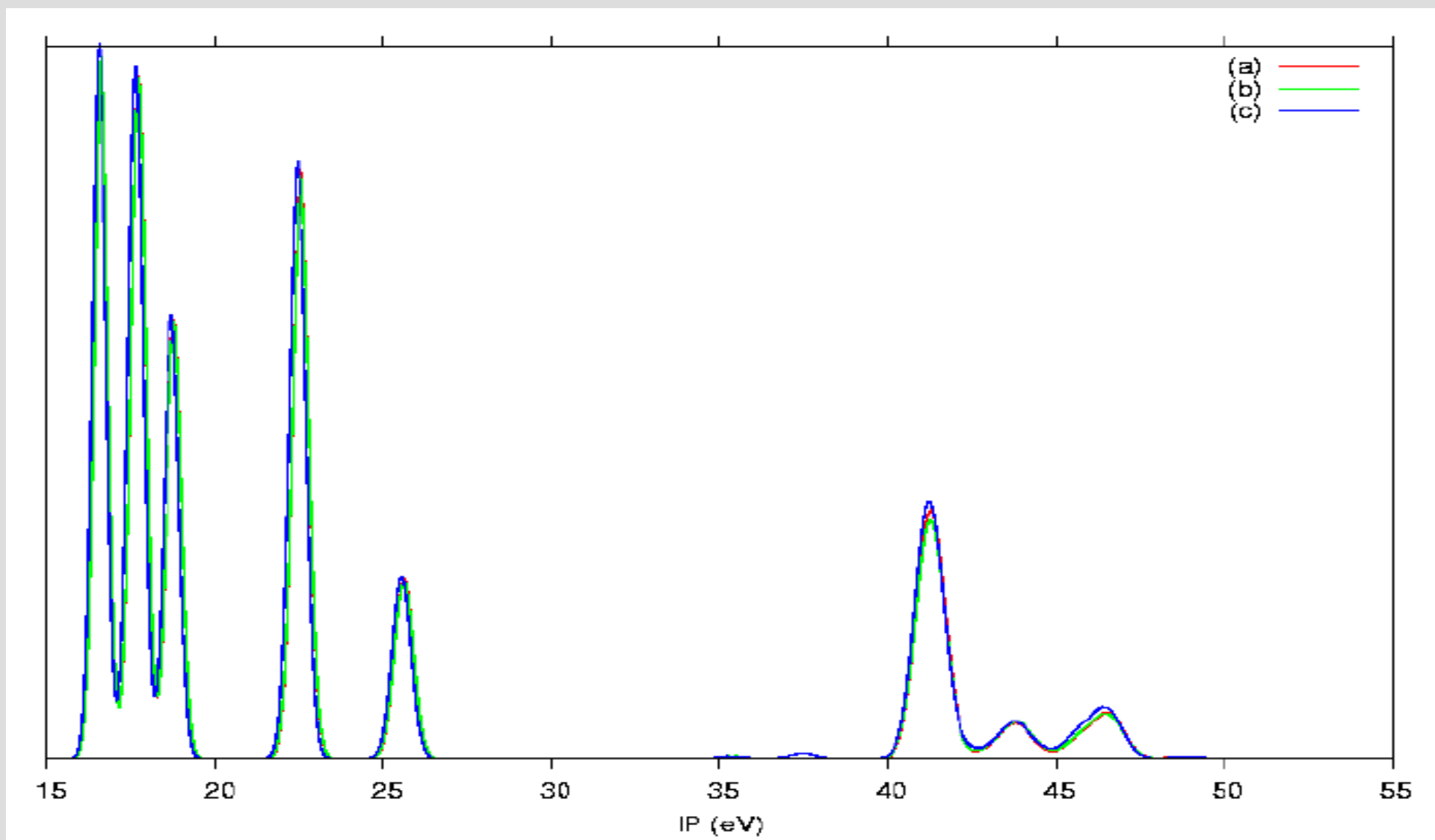
$$\mathbf{C}_{11}^{(3)} = \mathbf{C}_{11}^{(A)} + \mathbf{C}_{11}^{(B)} + \mathbf{C}_{11}^{(C)} + \mathbf{C}_{11}^{(D)} + \Sigma(\infty)$$

ADC(3), Non-Dyson scheme

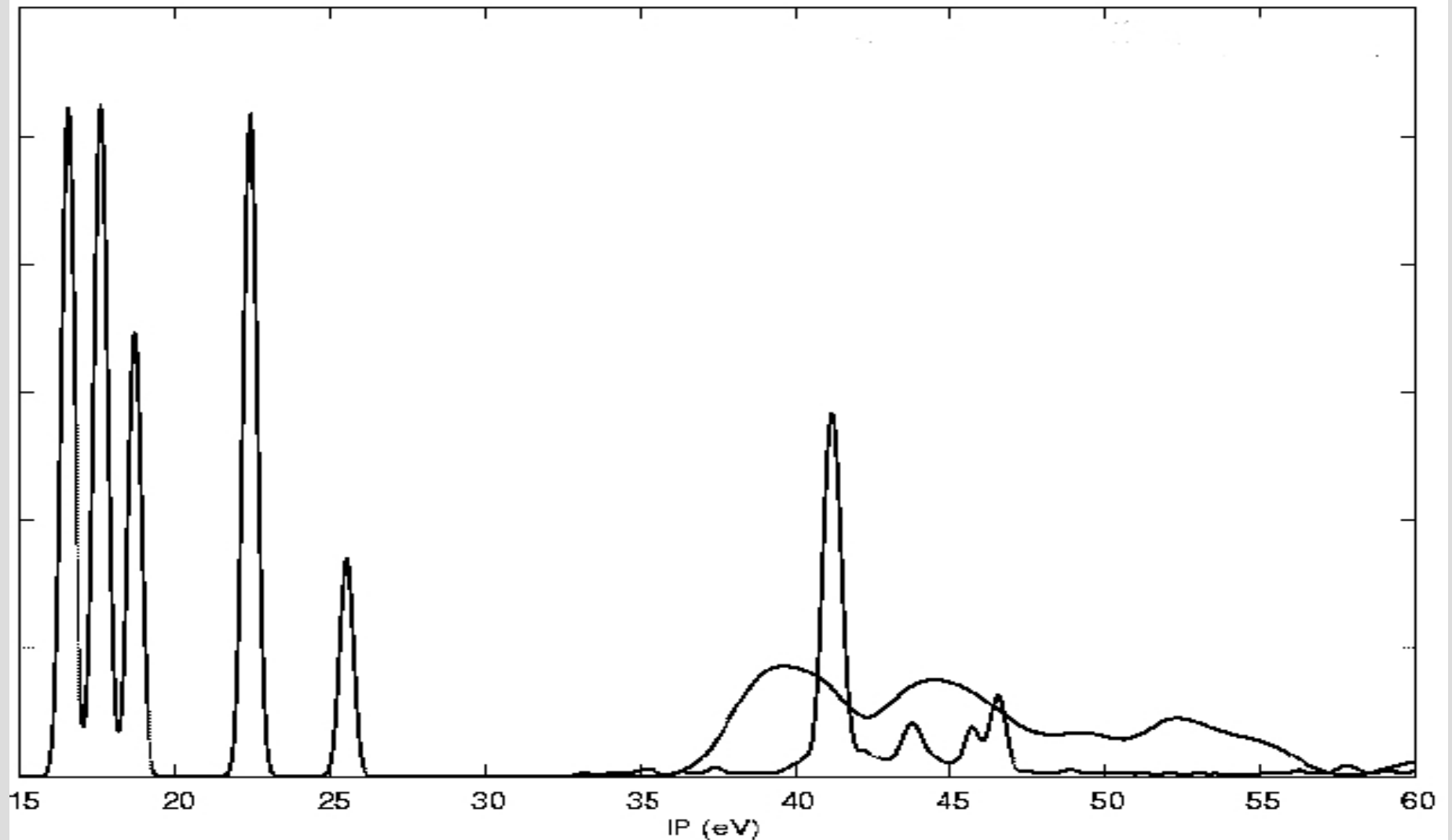
$C_{ij}^{(A)}$ is a function of the 04 integrals

- To make the code simpler and more readable, the core of the integral-driven code is constituted by one routine which implements the equation (except of the loop over virtual indices) and is called as many times as needed for each integral, with the appropriate set and permutation of argument indices.
- We actually use the $\Sigma(\text{DEM})$ approximation for the static self-energy, we will use the $\Sigma(4+)$ to avoid completely the use of the $(N+1)$ -particle space.

ADC(3), Non-Dyson scheme



ADC(3), Non-Dyson scheme



ADC(3), Non-Dyson scheme

num. Lanczos Iterations: 200

num. Lanczos iterations for affinity reduction 50

num. 1h confs. 16

num. 1p confs. 89

num. 2h1p confs. 5696

num. 2p1h confs. 31684

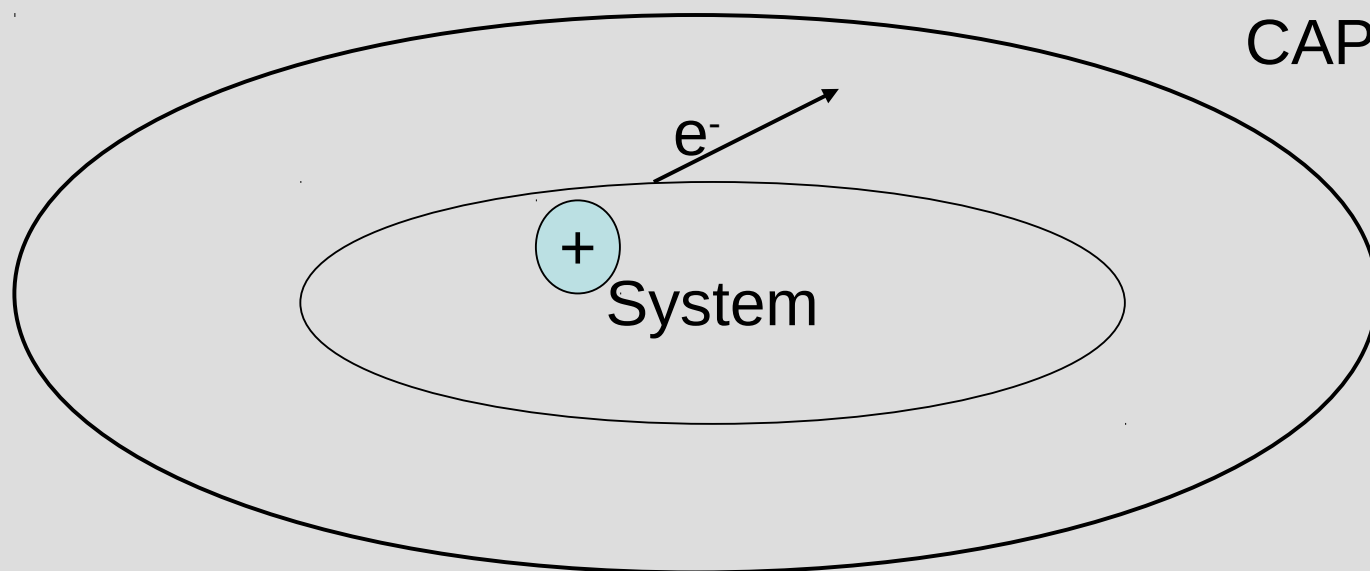
	Step 1 s.	Step 2 s.	Step 3 s.	Tot s.
ADC(3)	719.4	8168.5	371.8	9259.7
ADC(3) redu.	719.8	4499.5	352.4	5204.3
ND-ADC(3)	2882.3	305.5	0.4	3188.2

J. Comp. Chem., 30 (5), 818 (2009)

J. Comp. Chem (2011) in preparation

Green's function and the ADC – CAP/ADC

A CAP is introduced to absorb the electron that is emitted in the decay process. In this way the wave function of the scattered electron becomes square-integrable. The scattering problem may be described within the framework of standard quantum chemistry methods. The CAP method can be combined with virtually any electronic correlation method.



Green's function and the ADC – CAP/ADC

In the CAP method a an absorbing potential is added to the physical hamiltonian H , and one works with a parametrized operator:

$$\hat{H}(\eta) = \hat{H} - i \eta \hat{W}$$

W is typically a real “soft” boxlike potential in the dissociation coordinate, and η is a strength parameter.

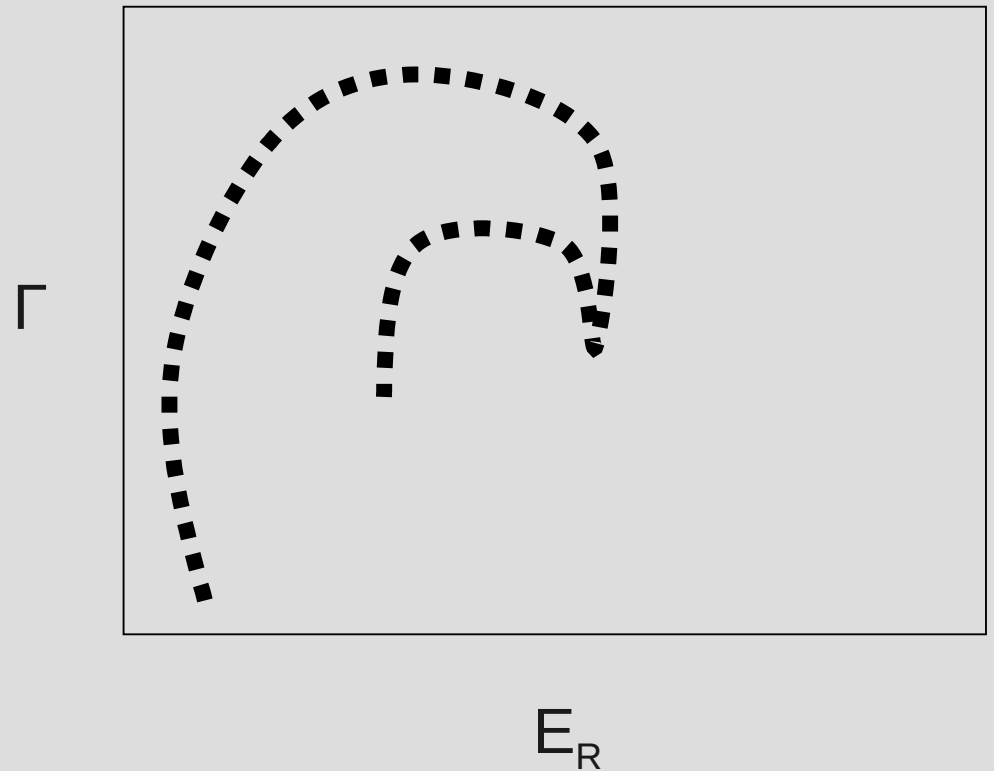
Green's function and the ADC – CAP/ADC

In the framework of a finite basis set E_{res} will depend on W . One typically studies the trajectories of the complex eigenvalues $E_i(\eta)$ of $H(\eta)$ and identifies a resonance by a minimum of the “velocity”:

$$v_i(\eta) = \left| \eta \frac{\partial E_i}{\partial \eta} \right|$$

Green's function and the ADC – CAP/ADC

In other words for each resonance there is an optimal CAP strength (η_{OPT}) that yields the best approximation to the resonance parameters for the chosen basis set and W .



Green's function and the ADC – CAP/ADC

- The constants c_i define the size L of the box.
- The box is defined in order to spatially confine the basis set.

$$W(\mathbf{r} ; \mathbf{c} ; n) = \sum_{i=1}^{i=3} W_i(x_i ; c_i ; n)$$

$$W_i(x_i ; c_i ; n) = \begin{cases} 0, & |r_i| \leq c_i \\ (|r_i| - c_i)^n & |r_i| > c_i \end{cases}$$

$$i = x, y, z$$

Green's function and the ADC – CAP/ADC

CAP/ADC(3) $A' + i\eta$

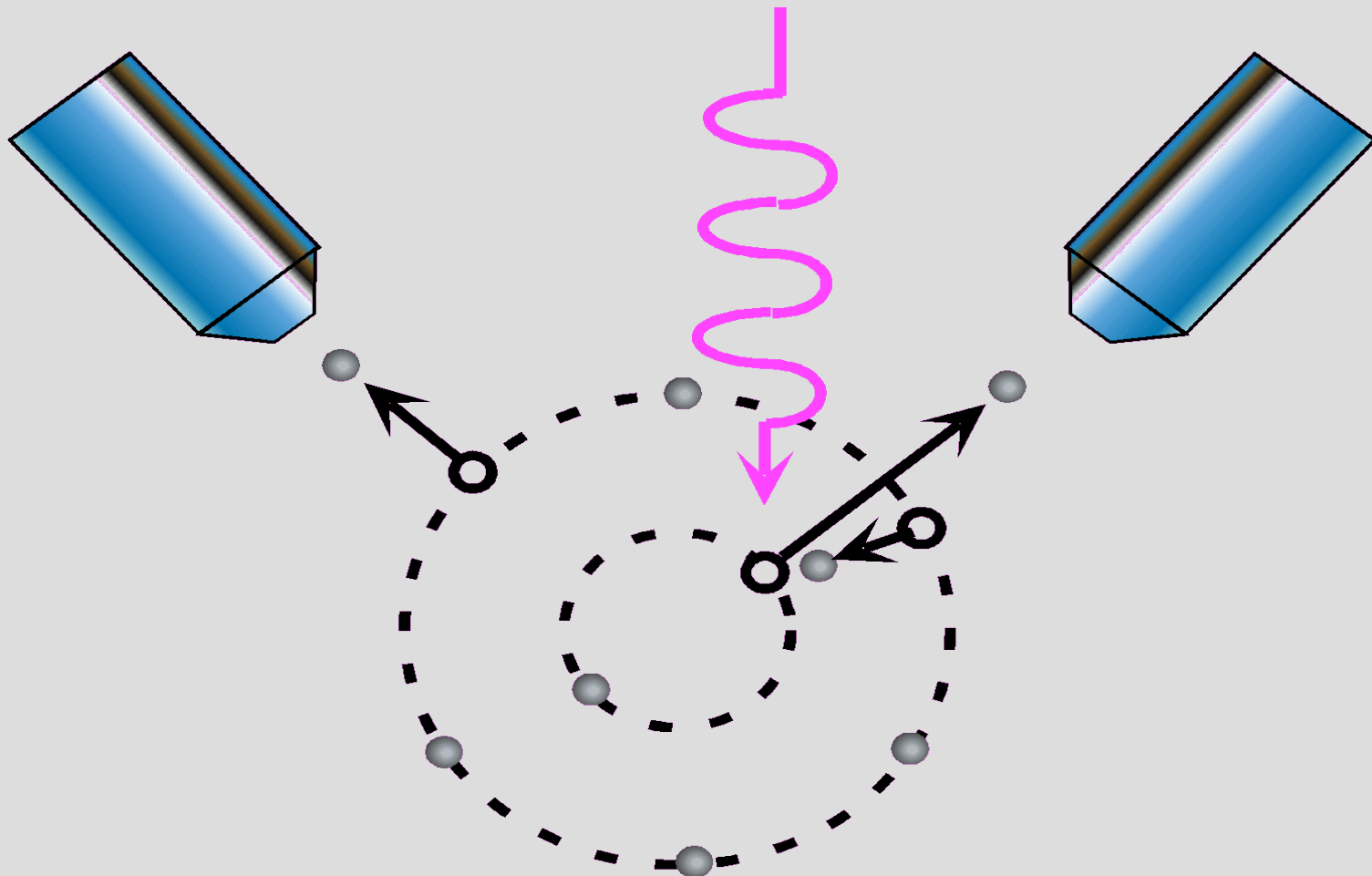
	1h/1p	2p1h	2h1p
1h/1p	$W_{qq'}$	0	0
2p1h	0	$\begin{bmatrix} W_{aa'} \delta_{bb'} - W_{ab'} \delta_{ba'} \\ - W_{ba'} \delta_{ab'} + W_{bb'} \delta_{aa'} \end{bmatrix} \delta_{ii'}$	0
2h1p	0	0	$W_{aa'} \delta_{ii'} \delta_{jj'}$

Green's function and the ADC – CAP/ADC

- It is computationally hard to obtain the resonance parameters. One needs to diagonalize the “big” complex symmetric CAP/ADC(3) matrix many times to search for the minimum of the velocity.
- Subspace projection technique. The result of the subspace projection is a smaller complex symmetric matrix. It is easier to diagonalize for many values of the η parameter
- This approach is simple to use with the filter diagonalization method.

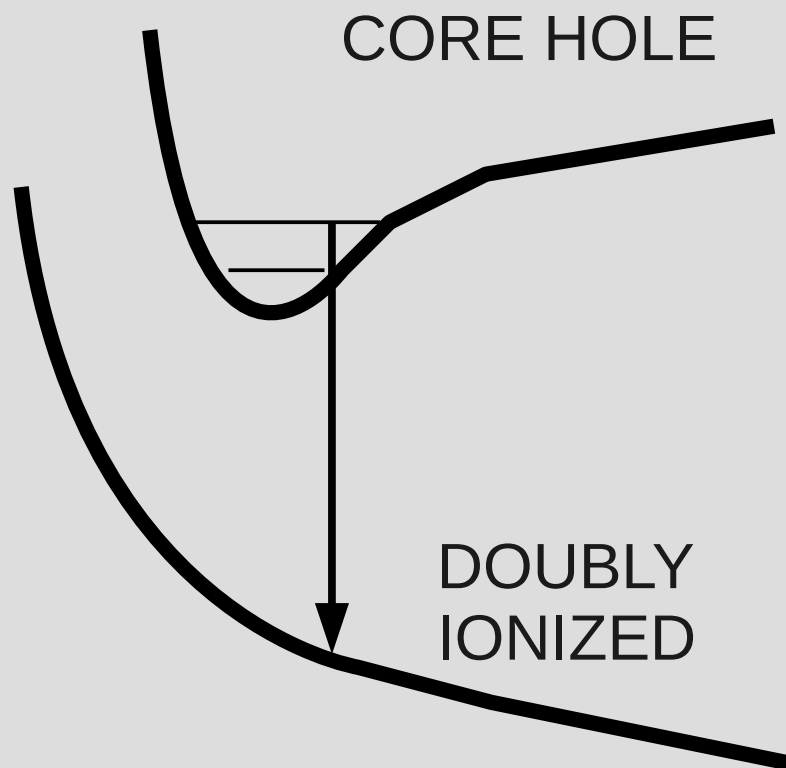
ADC(2) and applications - Auger

AUGER SPECTROSCOPY



ADC(2) and applications - Auger

The population analysis provides a qualitative as well as a quantitative description of the charge distribution for each dicationic state and estimates of spectral intensities

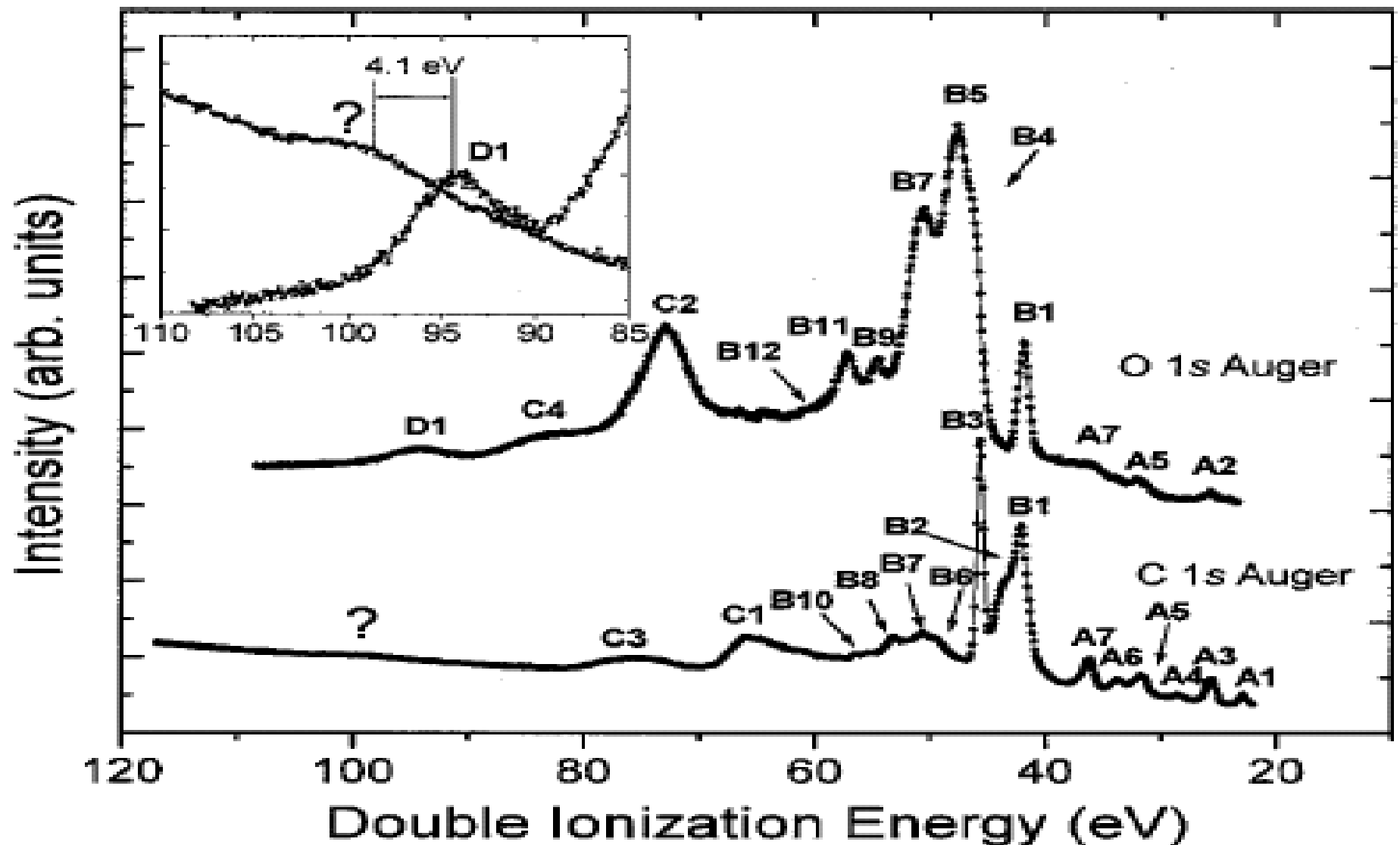


The nuclear dynamics analysis provides the correct energy centroids and broadenings of the bands. These can be very different in different spectra

ADC(2) and applications - Auger

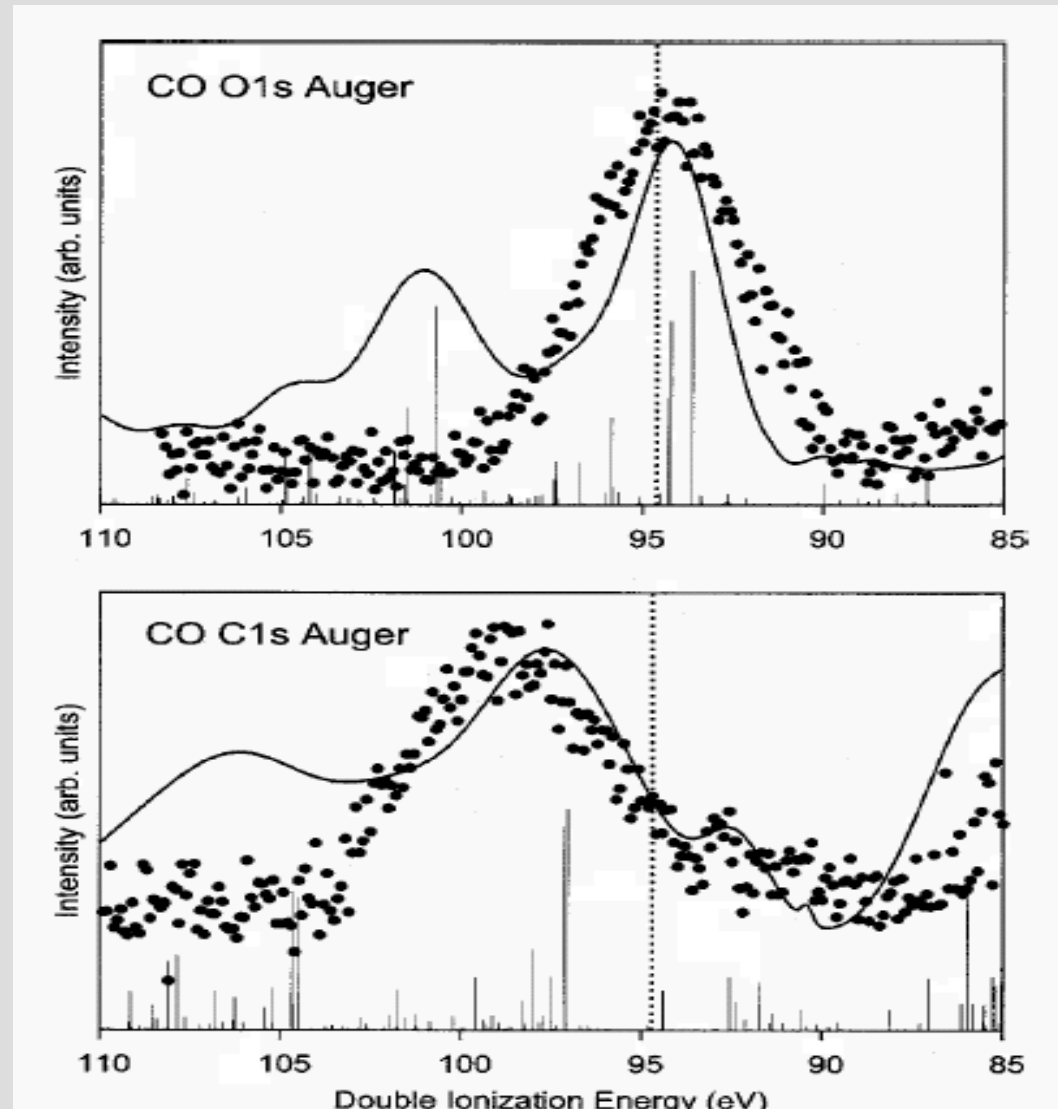
CO and CO₂ Auger spectra

ADC(2) and applications - Auger



ADC(2) and applications - Auger

The new observed feature is not an interatomic effect, but an effect of the nuclear dynamics. The shift between the bands in the two spectra was theoretically predicted.

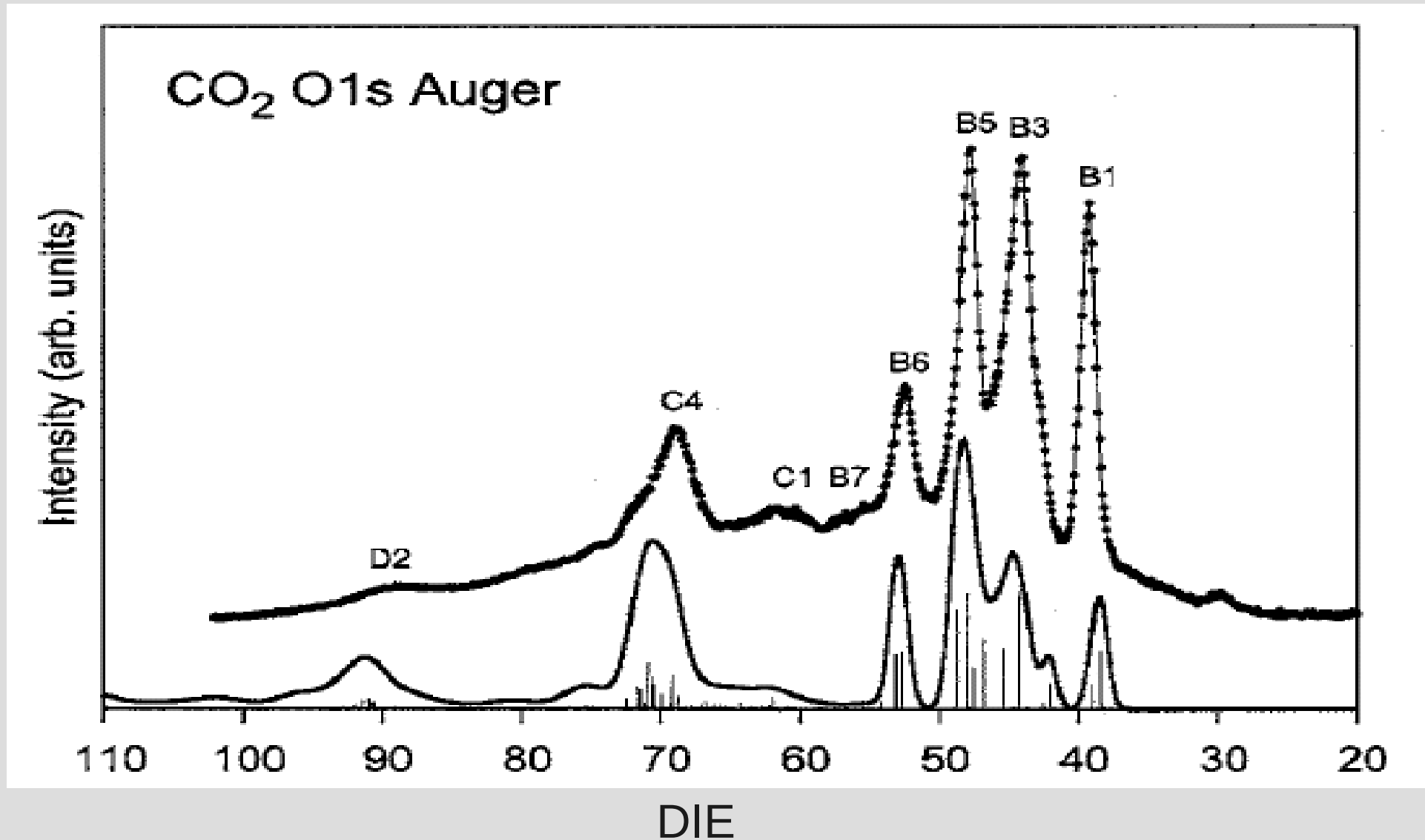


ADC(2) and applications - Auger

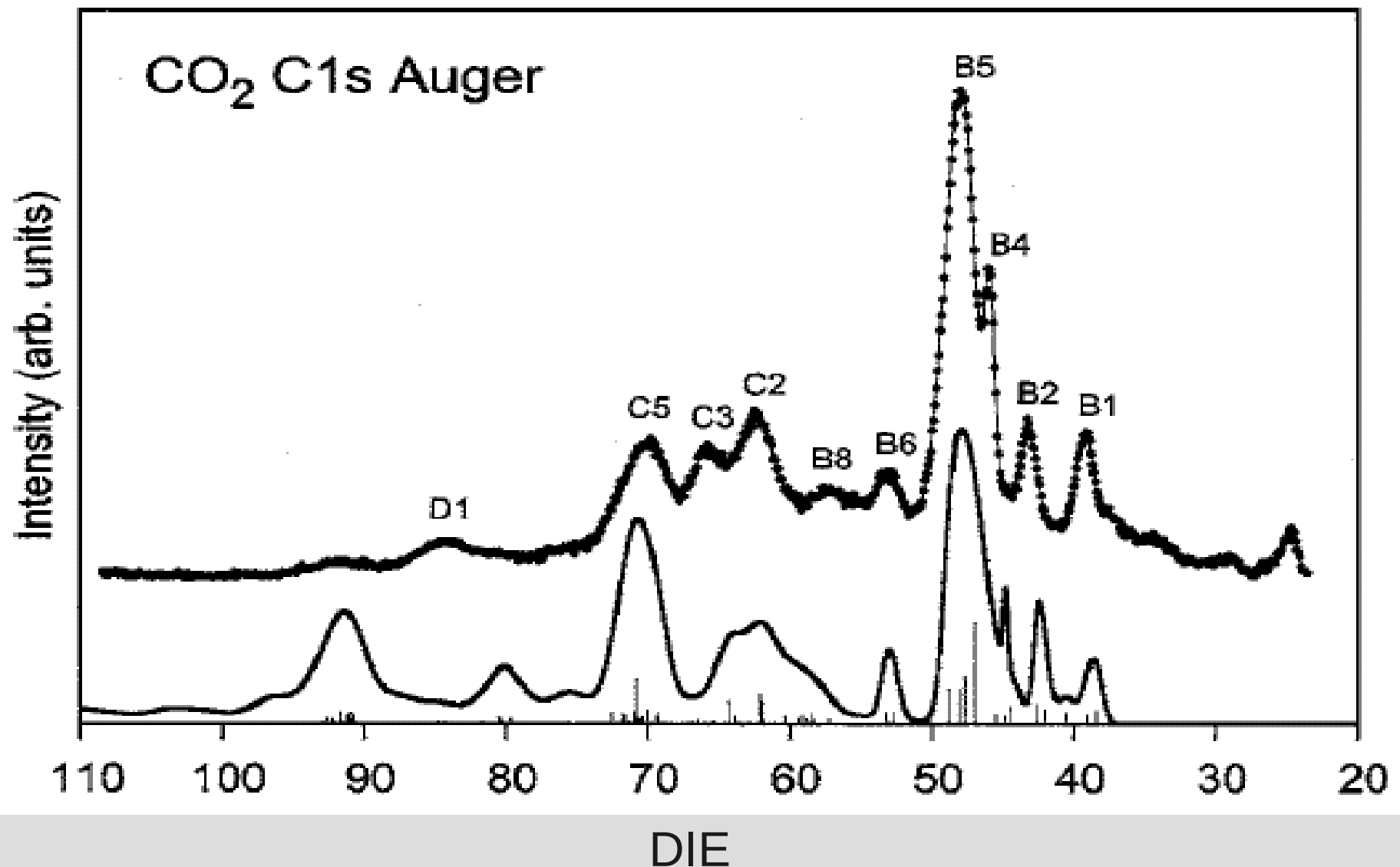


- A single total symmetric normal mode (symmetric stretching)
- We used the aug-cc-pVTZ basis set
- We performed a full diagonalization of the ADC(2) matrix
- The number of dicationic states found to be relevant for reproducing the Auger spectra is about 2000 up to a DIE (Double Ionization Energy) of 110 eV

ADC(2) and applications - Auger



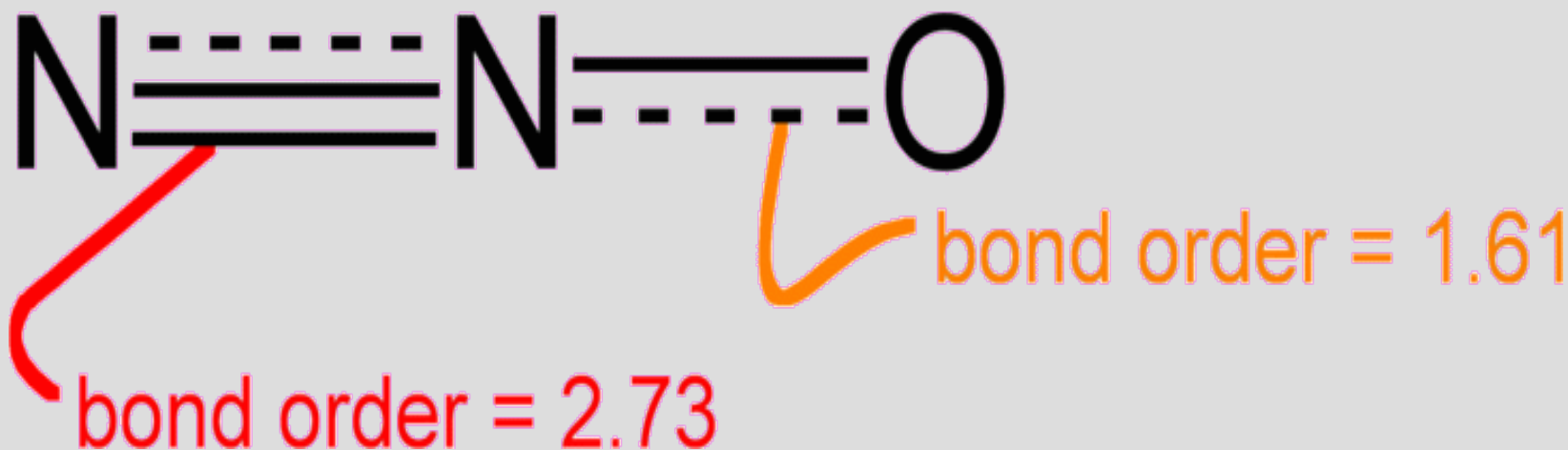
ADC(2) and applications - Auger



ADC(2) and applications - Auger

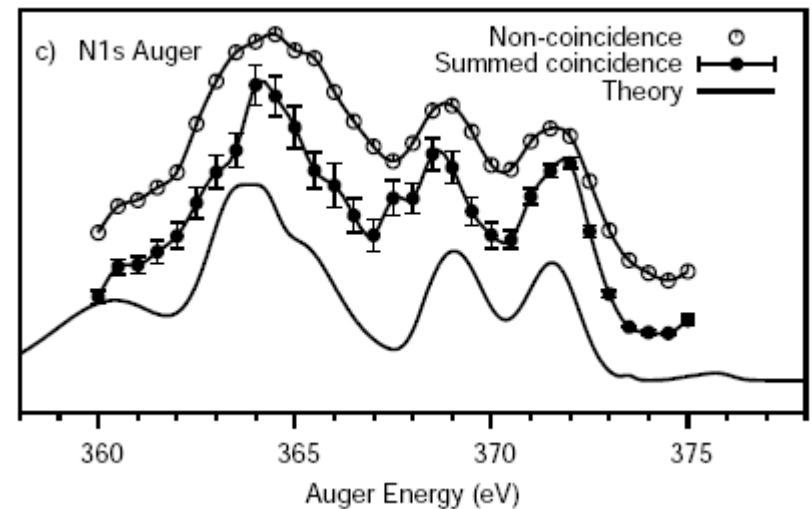
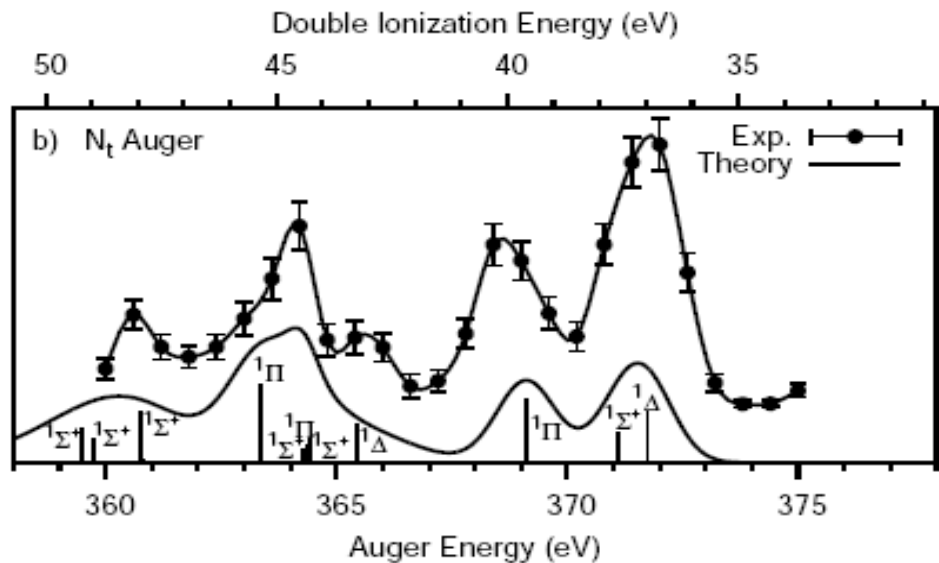
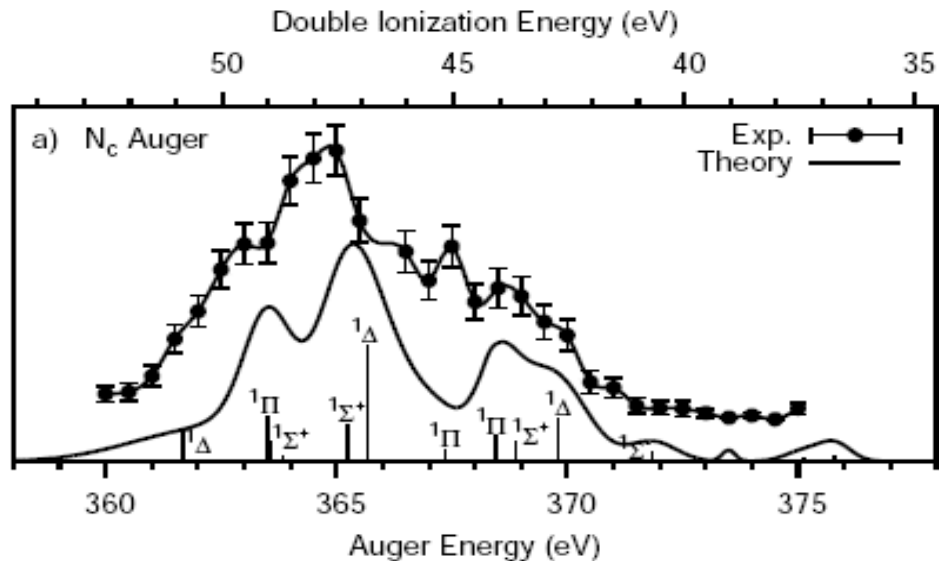
N_2O Auger spectra

ADC(2) and applications - Auger



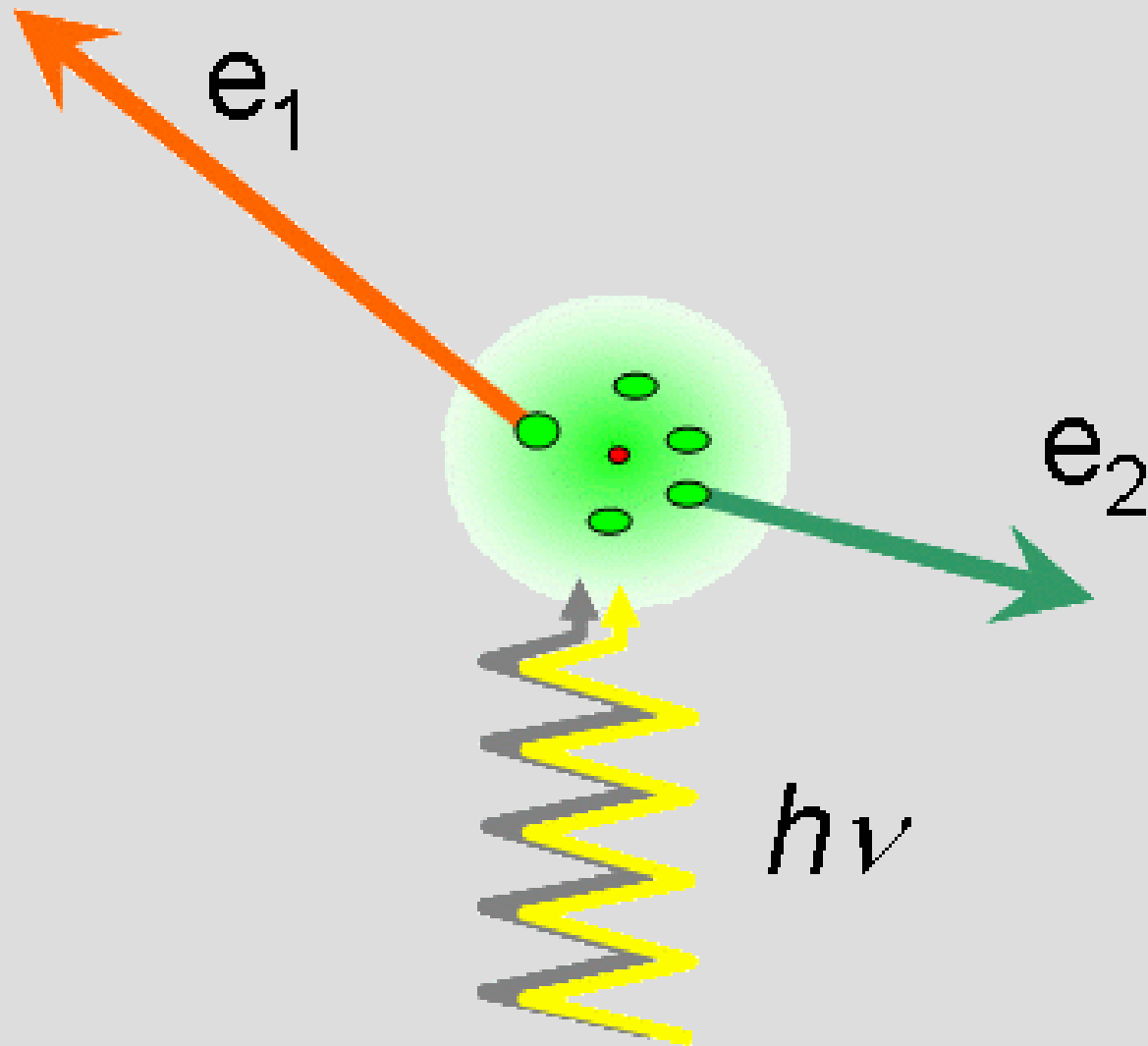
- We used the aug-cc-pVTZ basis set.
- We performed a full diagonalization of the ADC(2) matrix.
- The N atoms 1s ionization energies differ by 3.9 eV.

ADC(2) and applications - Auger



A more precise assignment than previous attempts as well as a better understanding of the relative intensities of the different features in the unselected spectrum have been achieved.

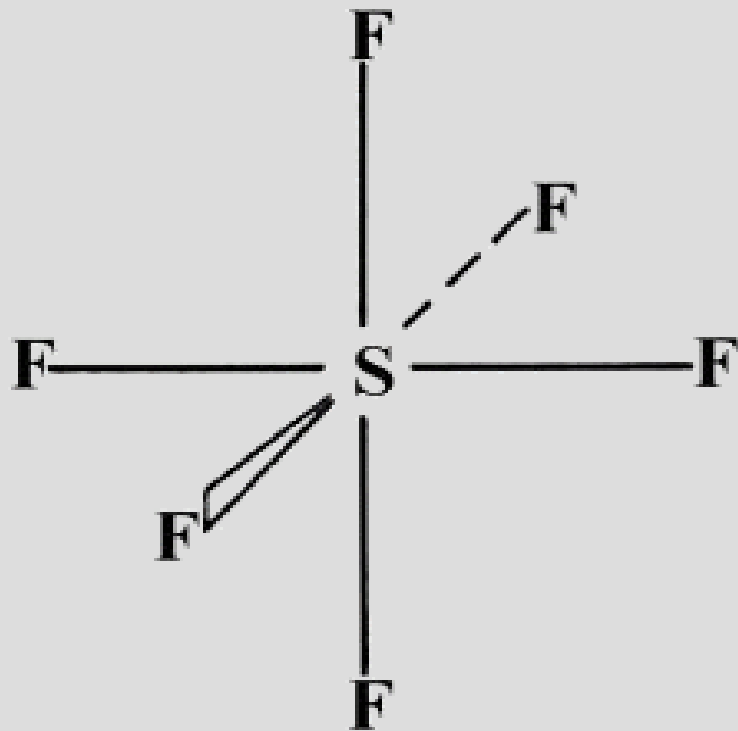
ADC(2) and applications - Direct double photoionization



ADC(2) and applications - Direct double photoionization

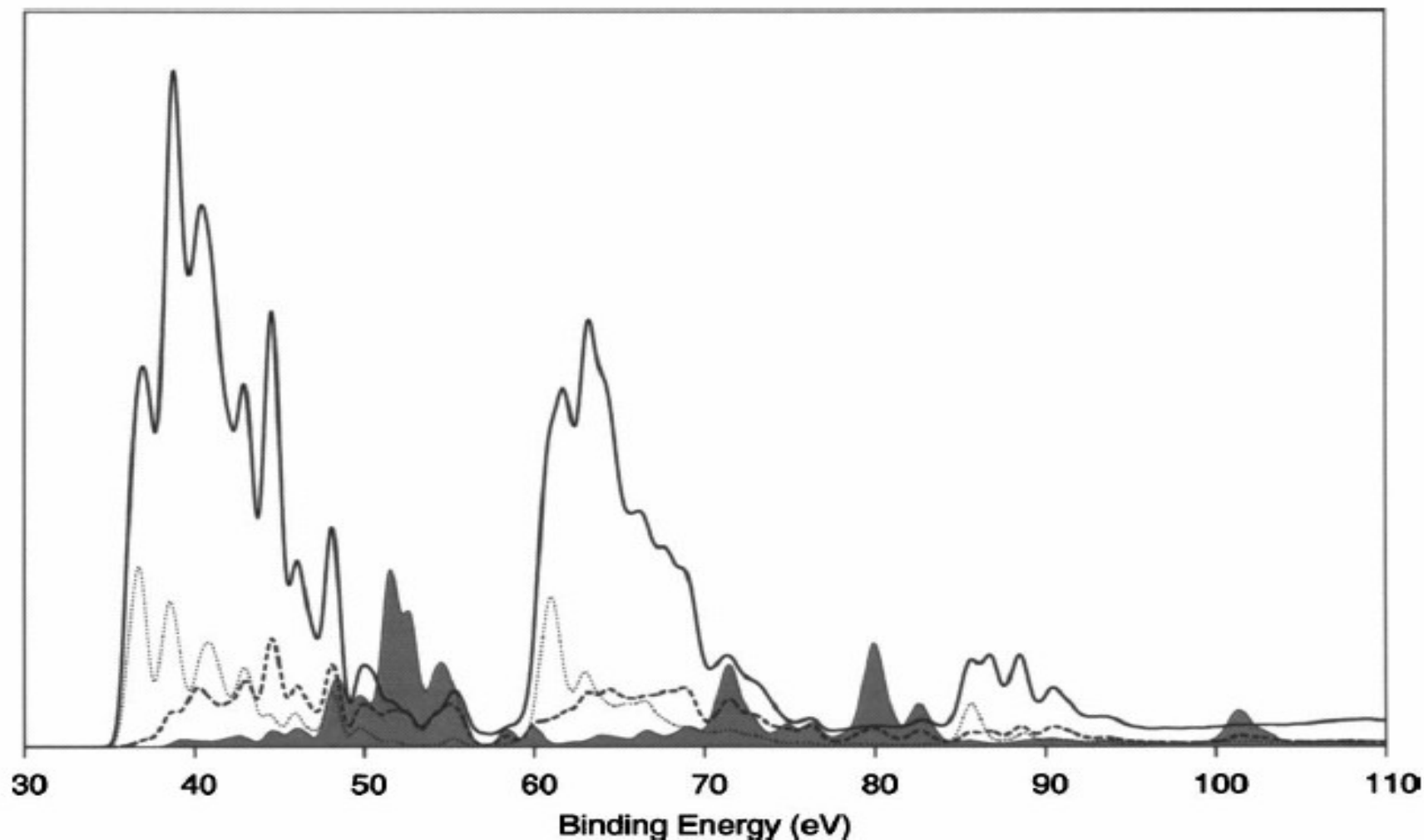
SF₆ direct double photoionization spectra

ADC(2) and applications - Direct double photoionization

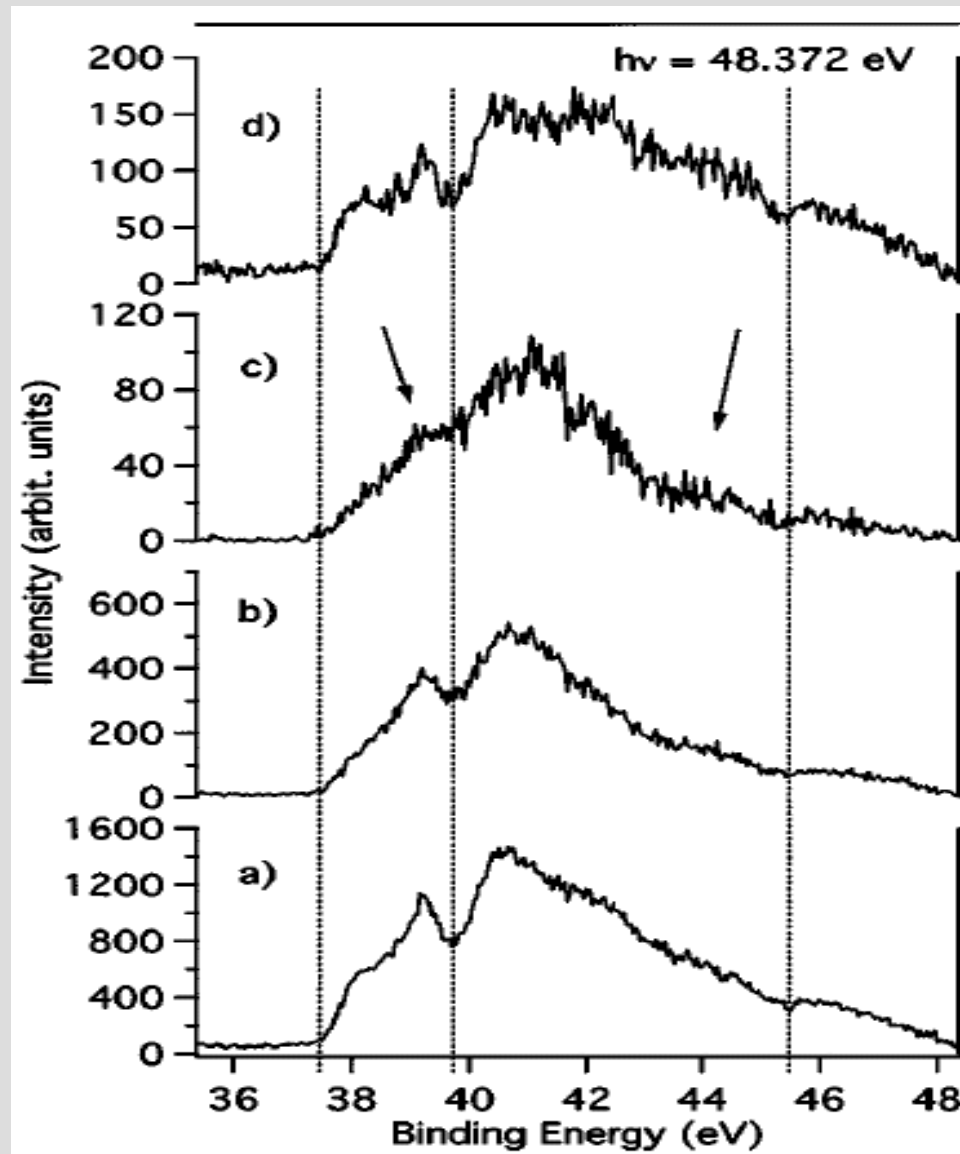


- We used a dzp basis set.
- We used the block-Lanczos iterative diagonalizer.
- The computed spectrum comprises nearly 29000 electronic states up to a binding energy of 110 eV.

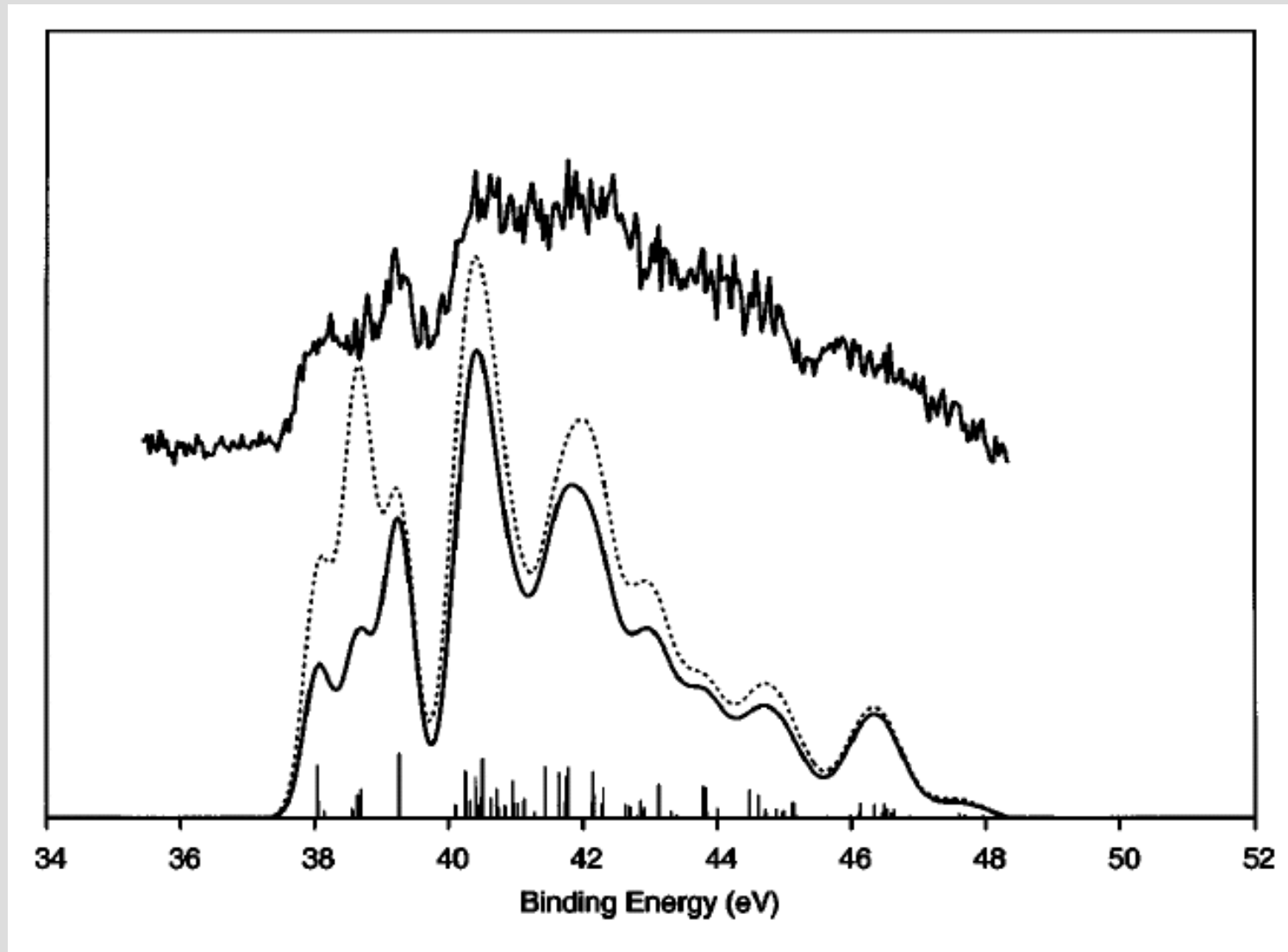
ADC(2) and applications - Direct double photoionization



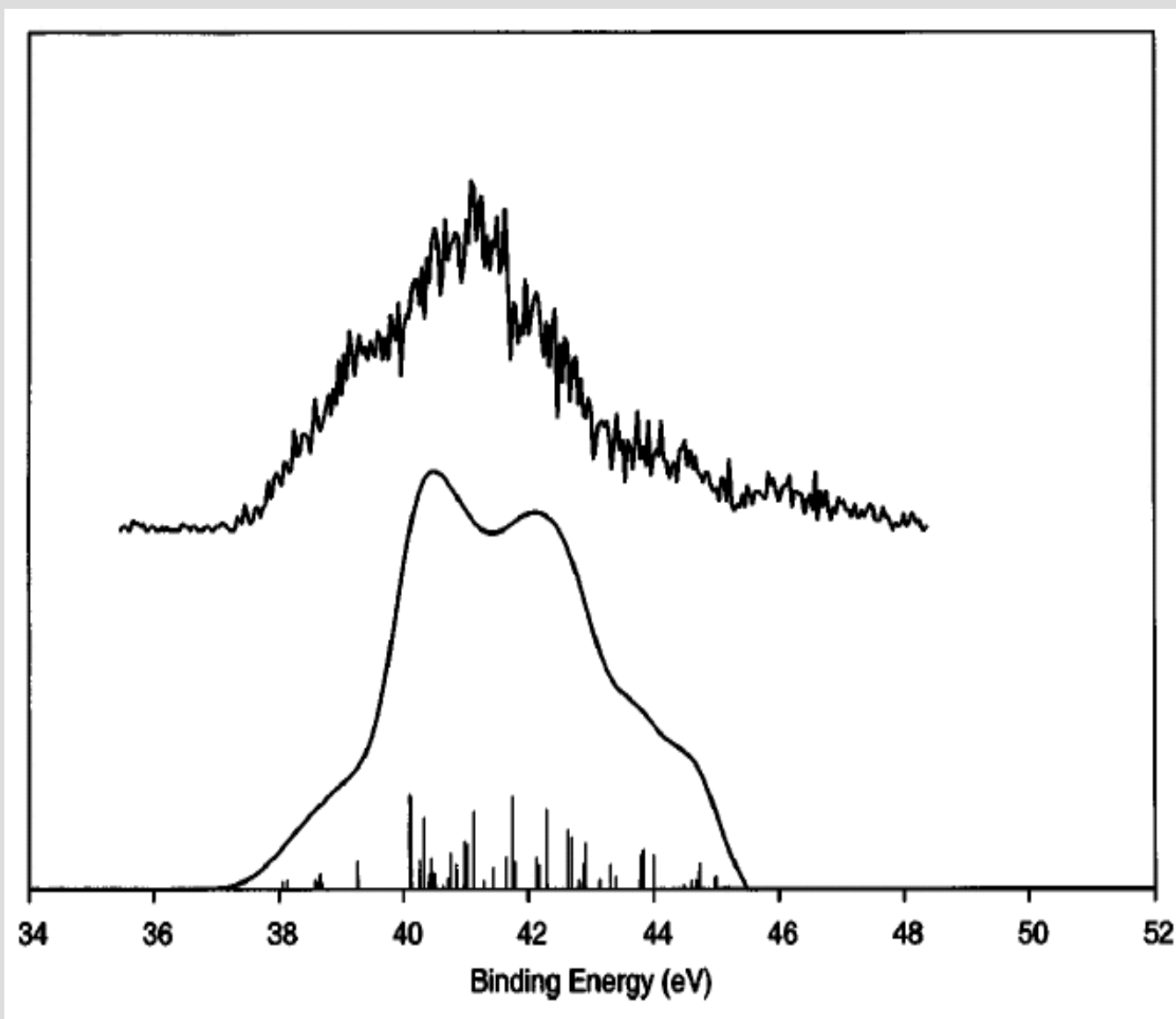
ADC(2) and applications - Direct double photoionization



ADC(2) and applications - Direct double photoionization



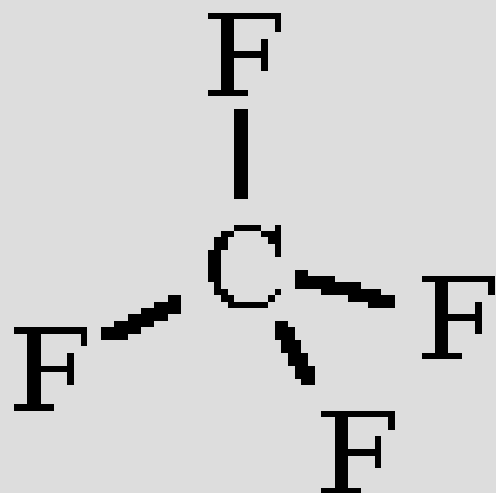
ADC(2) and applications - Direct double photoionization



ADC(2) and applications - Direct double photoionization

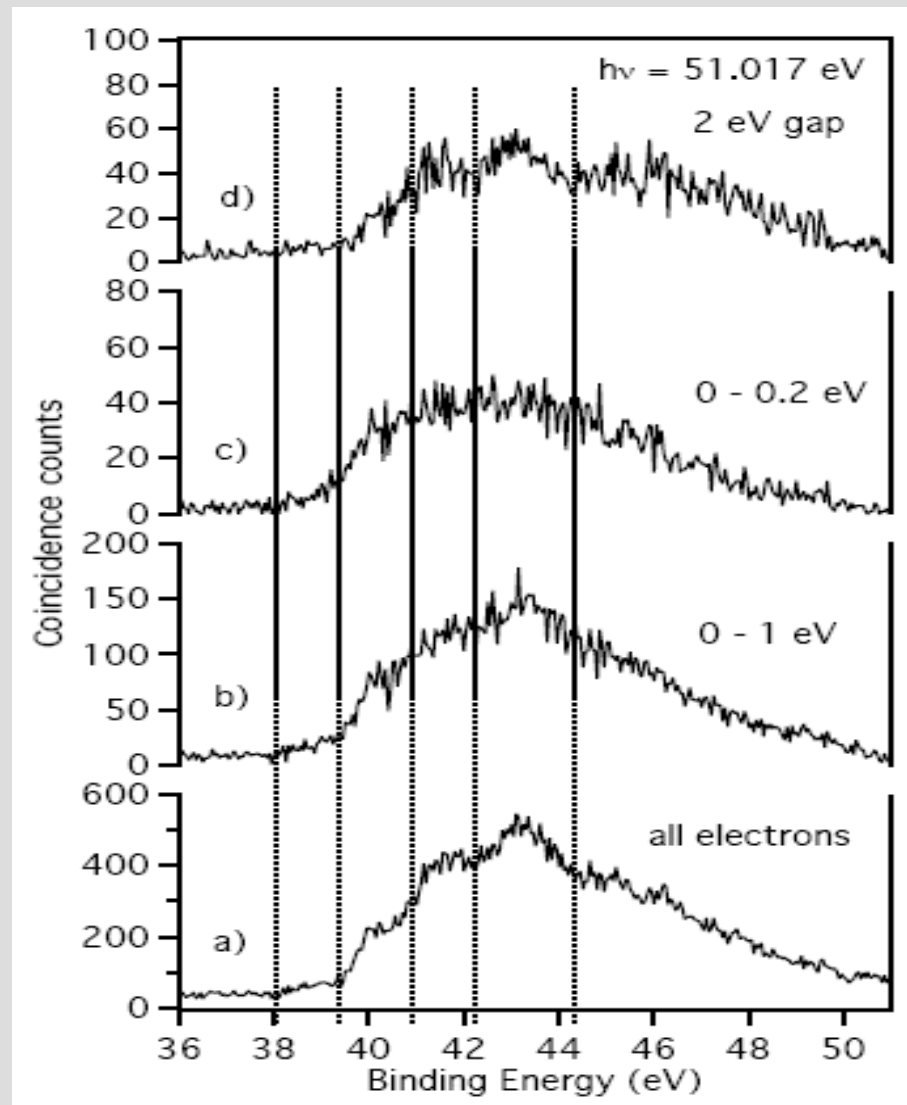
CF₄ direct double photoionization spectra

ADC(2) and applications - Direct double photoionization

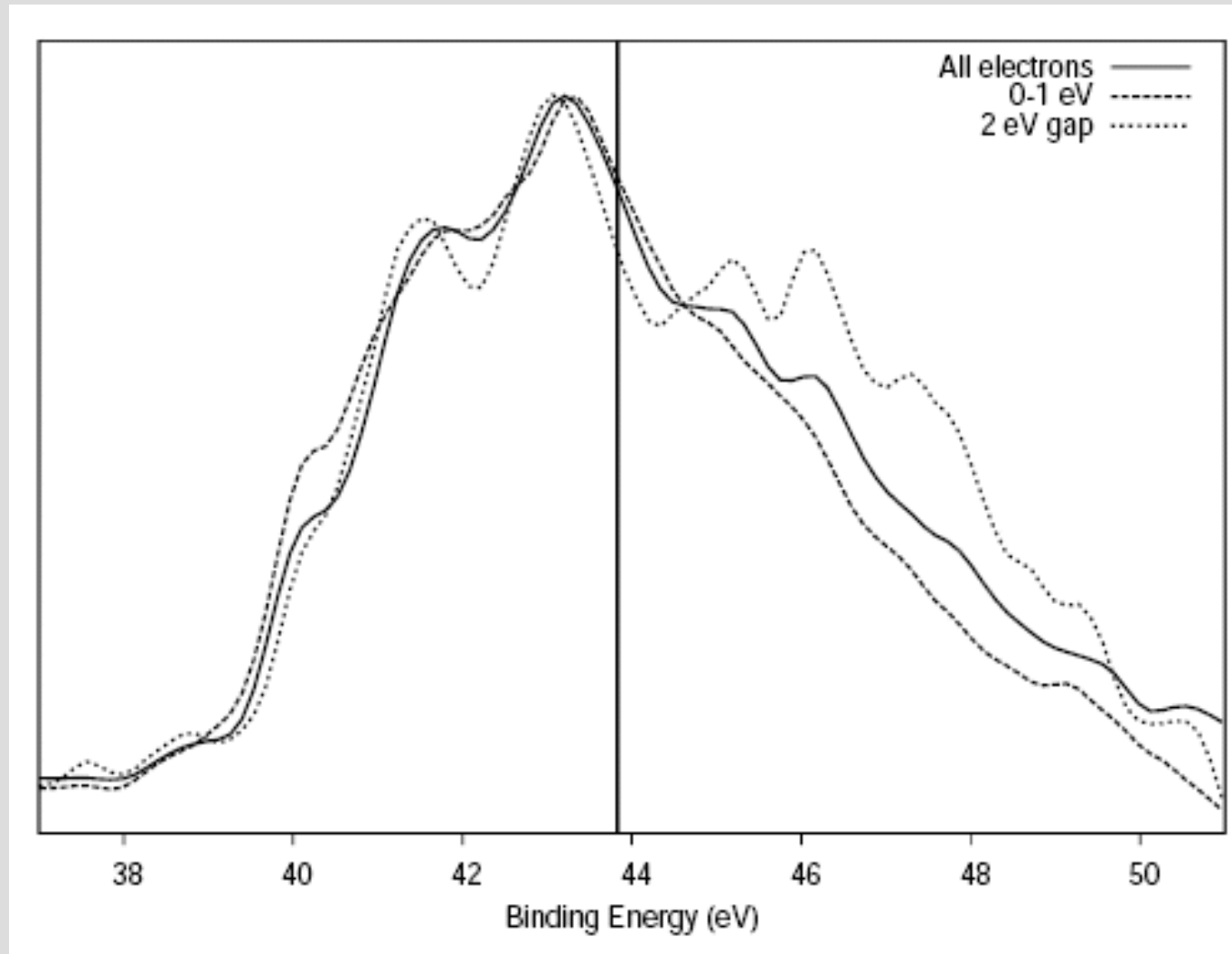


- We used aug-cc-pVDZ basis set.
- We used the block-Lanczos iterative diagonalizer.

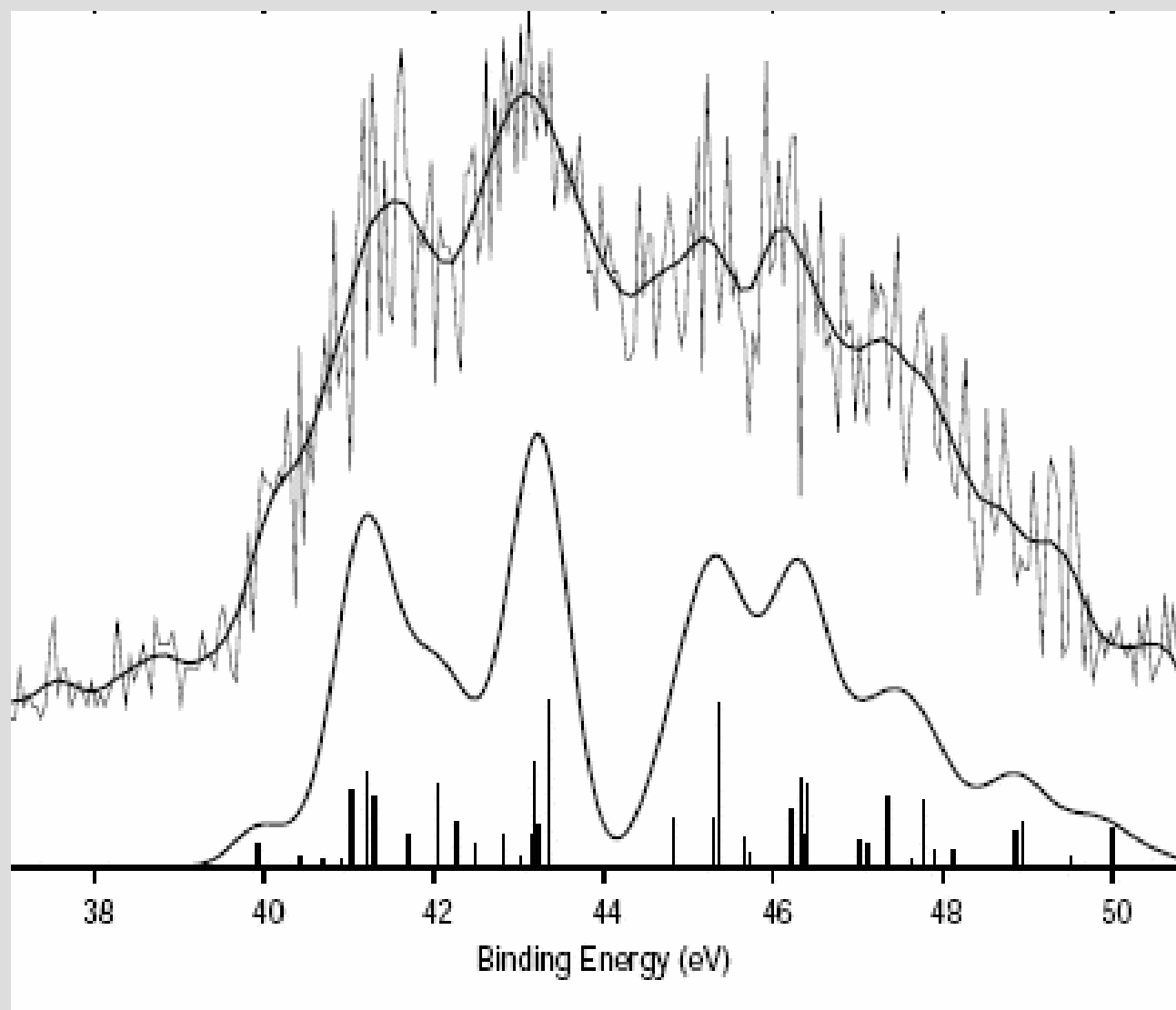
ADC(2) and applications - Direct double photoionization



ADC(2) and applications - Direct double photoionization



ADC(2) and applications - Direct double photoionization



Contents

- Grid Computing
- Green's function and the ADC
- **Relativistic DFT**
- GRID force-field pKa prediction
- GPGPU and DSP
- Sphere packing for Dye-sensitized solar cells

Parallelization of BERTHA

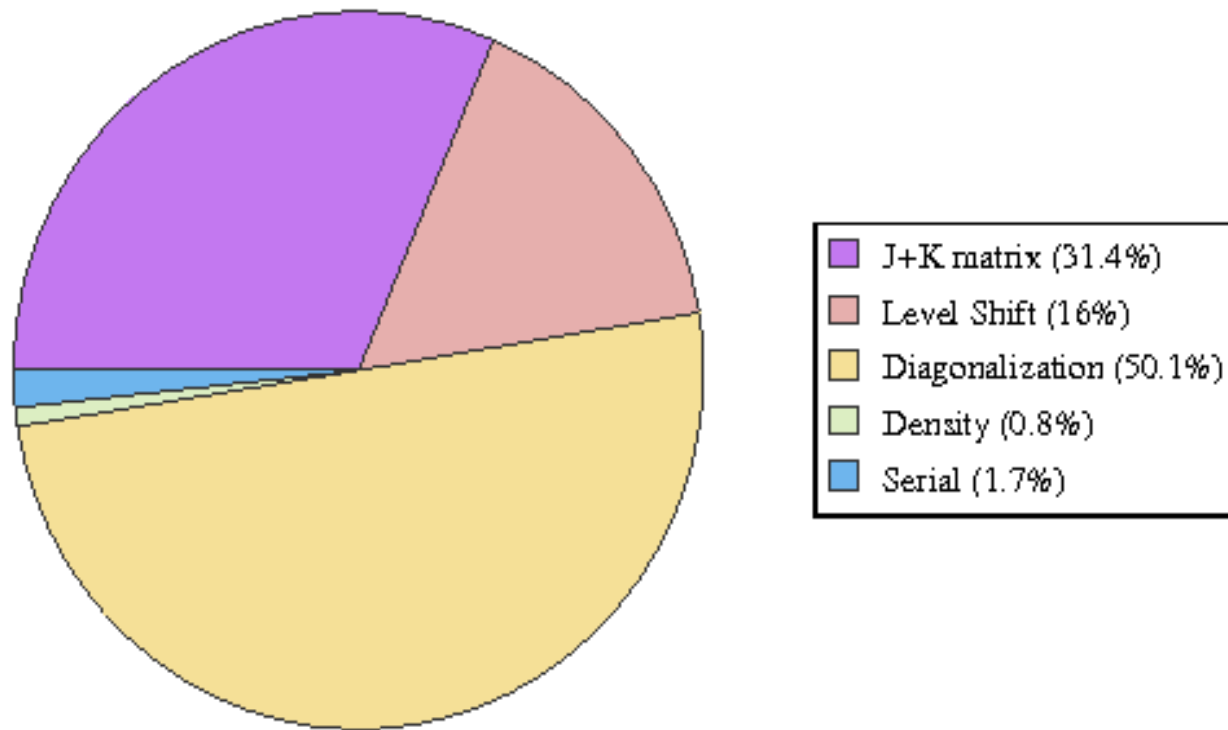
Each iteration can be subdivided in several steps:

1. **Serial Part**
2. **J+K matrix**: the DKS matrix is computed.
3. **Level Shift**: The level shift is applied to the DKS matrix by adding a weighted virtual orbital projection operator. This procedure involves two matrix-matrix multiplications.
4. **Diagonalization**: The hermitian DKS matrix diagonalization is carried out.
5. **Density**: The density matrix is computed as a matrix-matrix multiplication of positive occupied eigenvectors.

JCTC Journal of Chemical Theory and Computation, 6, 384, (2010)

•Parallel Computing: From Multicores and GPU's to Petascale, ed.: B. Chapman et al. p. 501-512 (IOS Press, Amsterdam, 2010)

Parallelization of BERTHA



First step: use a profile and optimize the serial code.

Second step: parallelization of the code

Parallelization of BERTHA

Step 2:

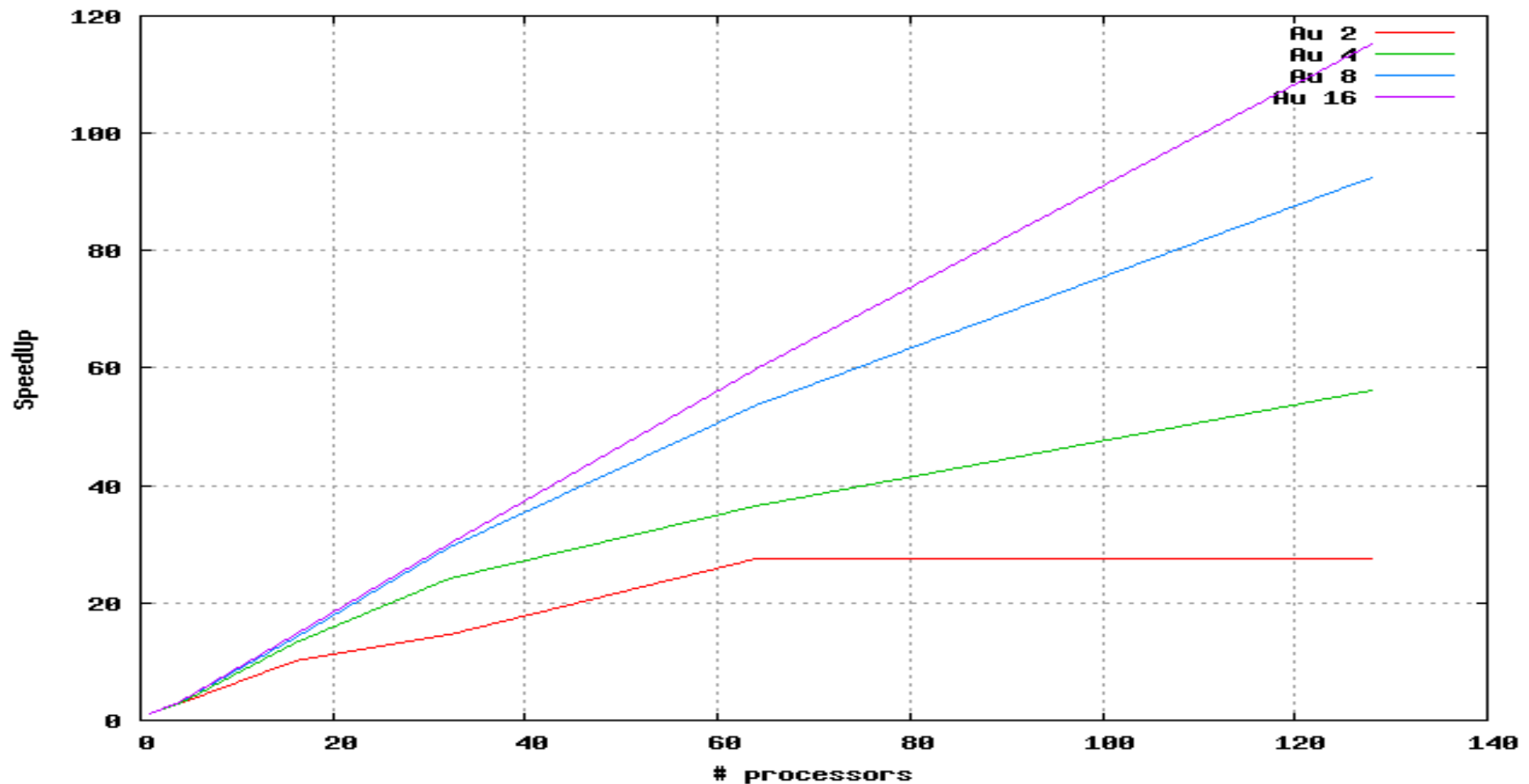
```
IF ( my_rank .eq. master ) THEN
  RECV FROM ANY_SLAVES (blockdim, block_position)
  IF (blockdim < 0) then
    EXIT
  ELSE
    RECV FROM slave (block)
    copy block into J+K matrix
  ENDIF
ELSE
  DO block_num = 1, max_nom_of_block
    IF (func(my_rank, block_num))
      ALLOCATE block
      COMPUTE block
      SEND (blockdim, block_position)
      SEND (block)
    END IF
  ENDDO
  SEND (blockdim = -1)
  EXIT
END IF
```

- Communication time is more or less independent from the number of processors involved.

- The small dimension of each block, guarantee by itself a good load balancing and good overlapping of communication and computation time.

- Last but not least, the small dimension of each block guarantee a better cache reuse, respect to the serial algorithm

Parallelization of BERTHA



Next step try to use AMPI (Charm++)

Parallelization of BERTHA

Step 3, 4, 5 (using ScaLAPACK) :

call DISTRIBUTE(DKSmat, master)

call LEVELSHIFT(DKSmat)

call DIAGONALIZE(DKSmat)

call COLLECT(EIGENVEC, master)

call DENSITY(EIGENVEC)

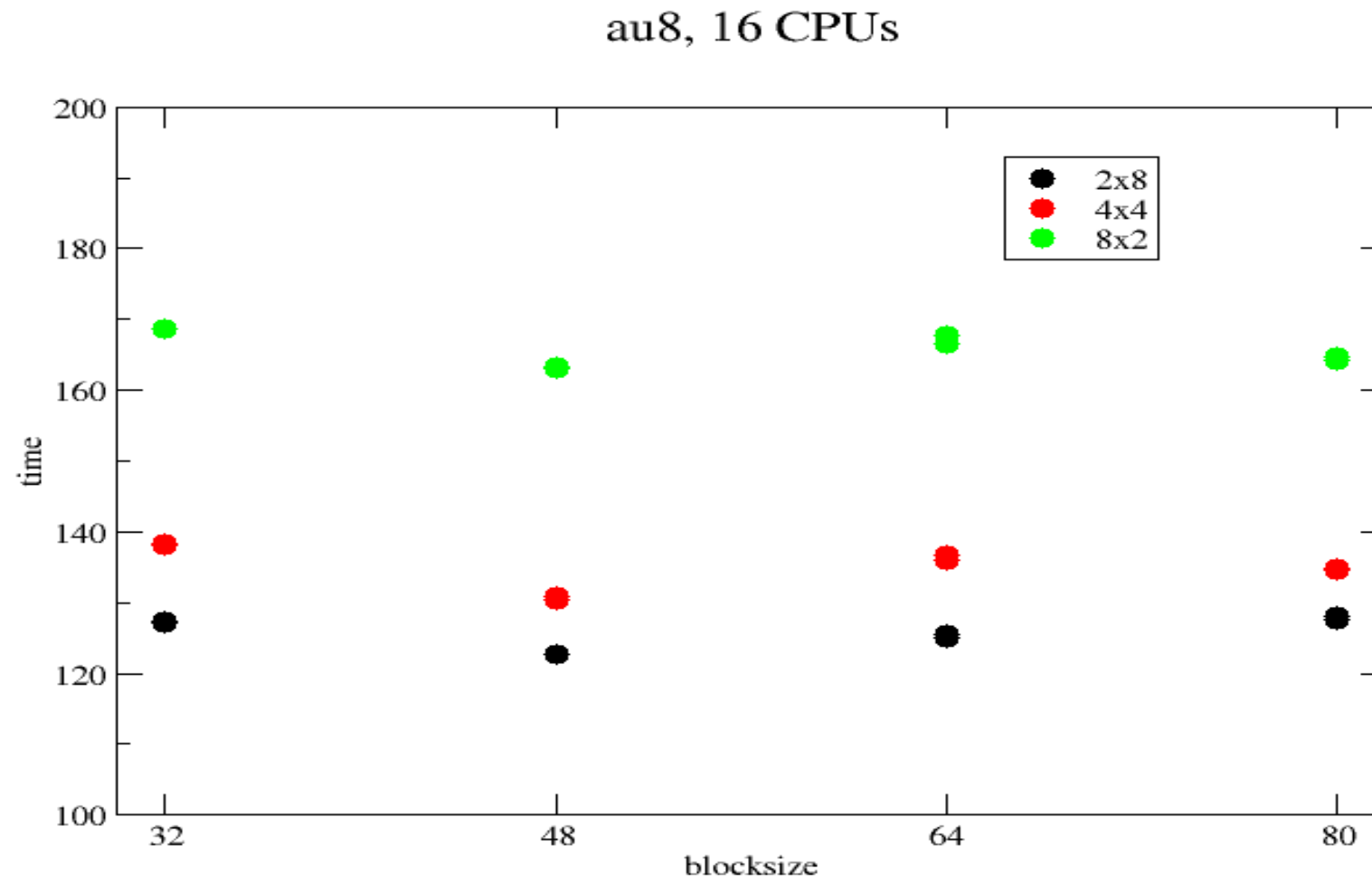
call COLLECT(DENSmat, master)

Parallelization of BERTHA

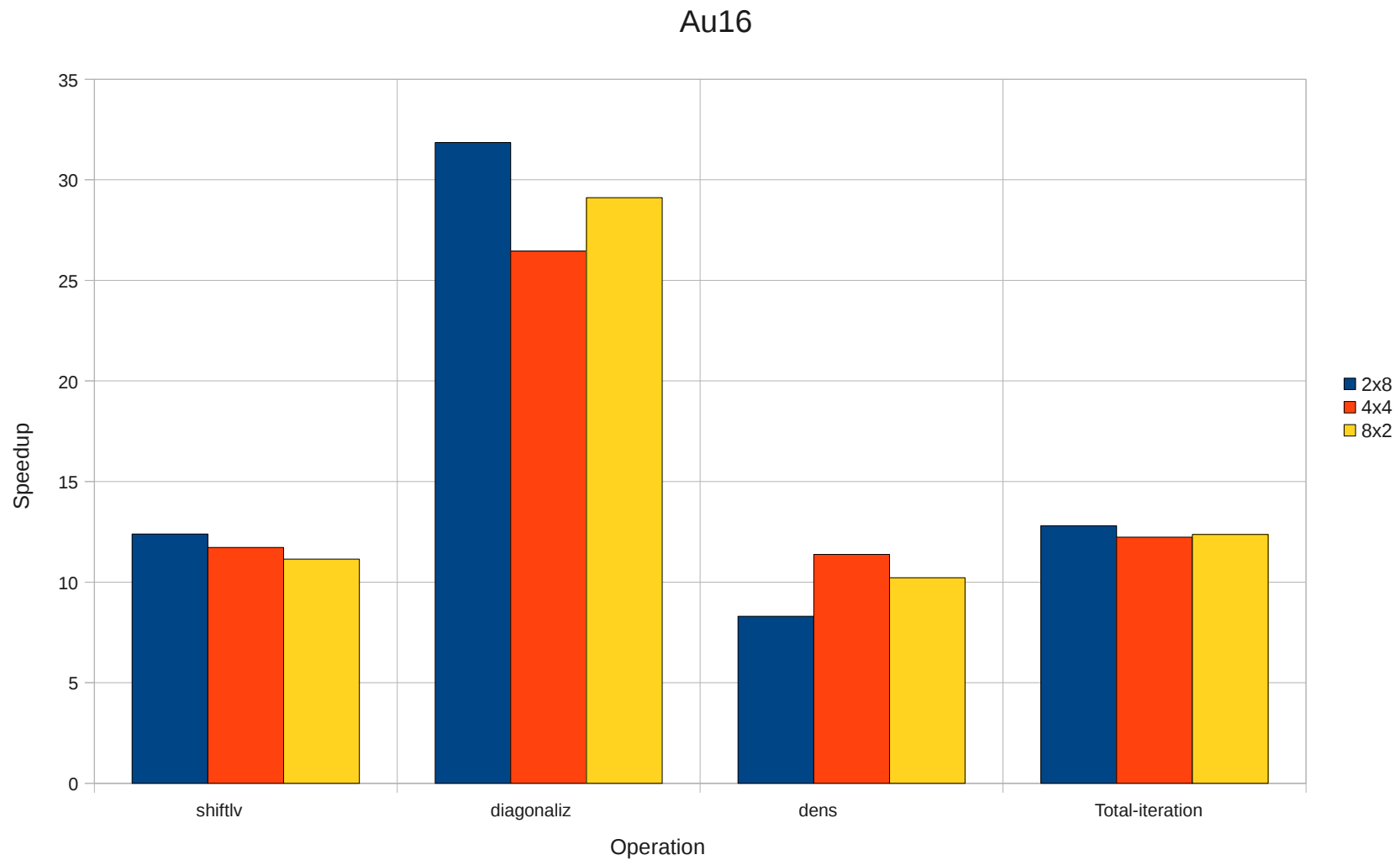
p,q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3
1	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3
2	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3
3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3
4	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3
5	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3
6	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3
7	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3
8	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3
9	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3
10	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3
11	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3

(a) Assignment of global block indices, (B, D) , to processes, (p, q) .

Parallelization of BERTHA



Parallelization of BERTHA

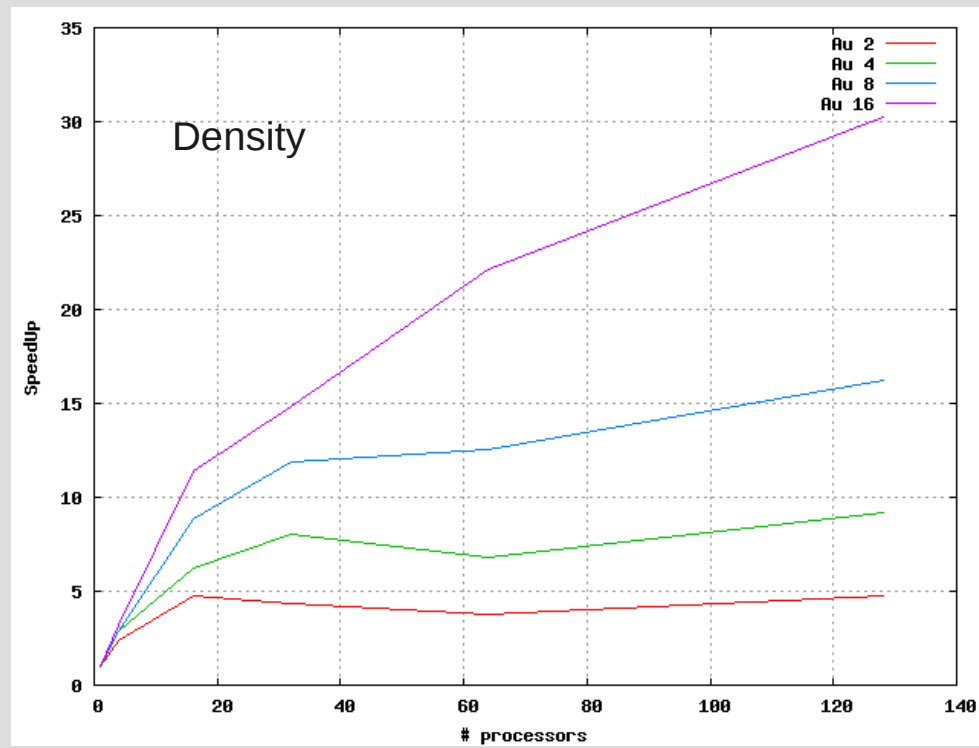
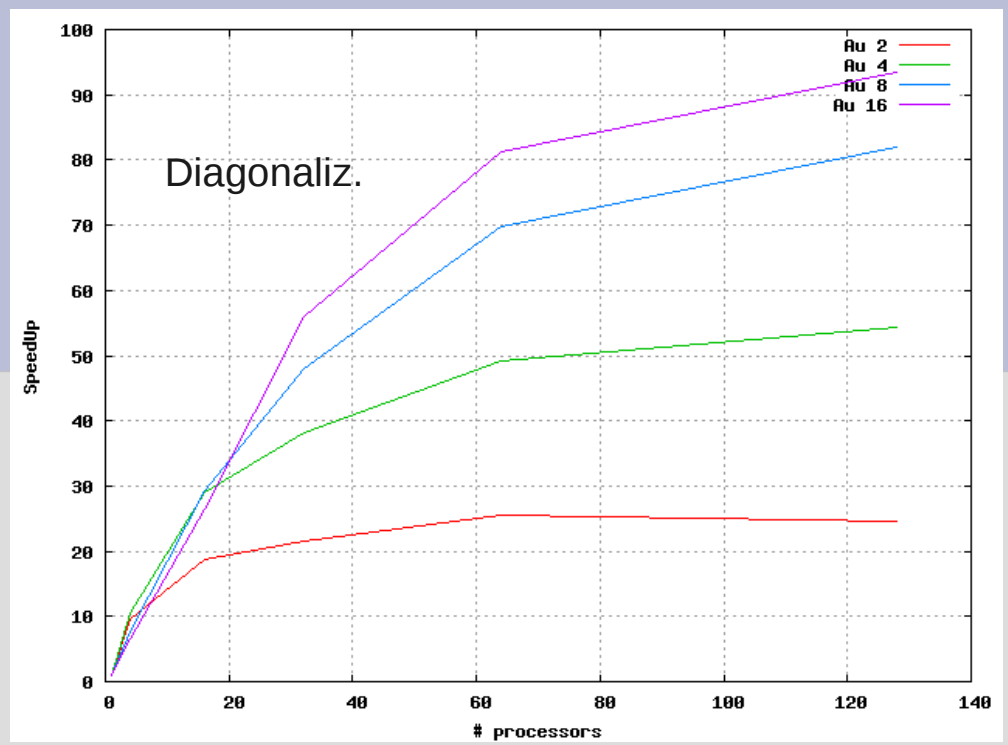
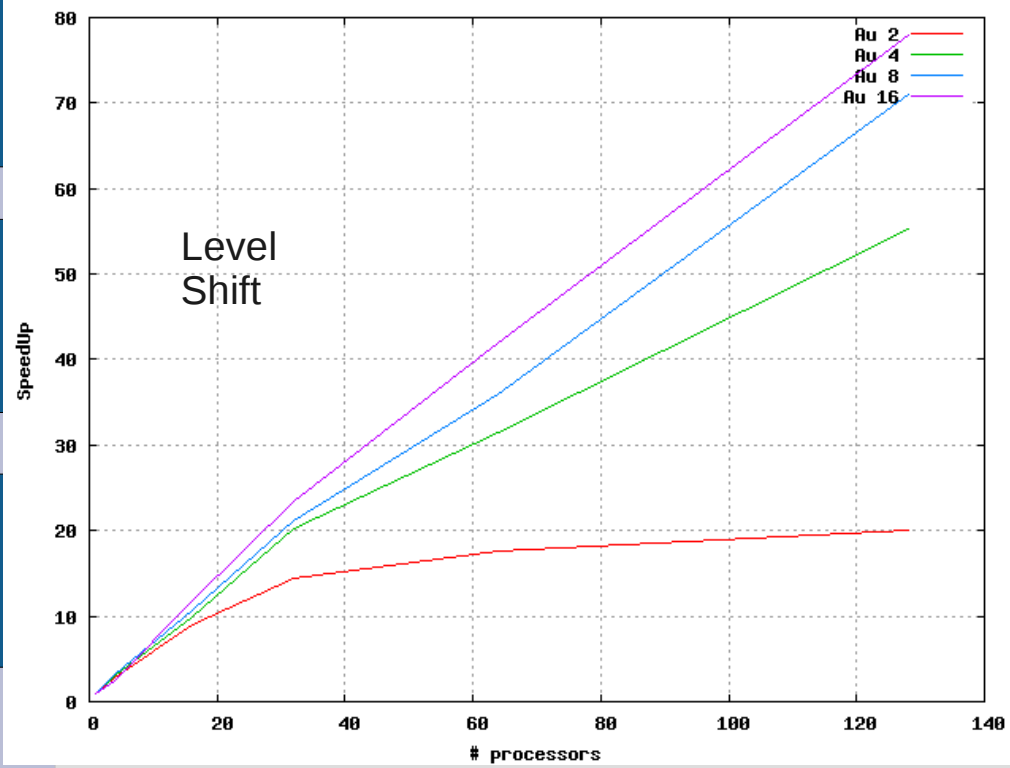


Parallelization of BERTHA

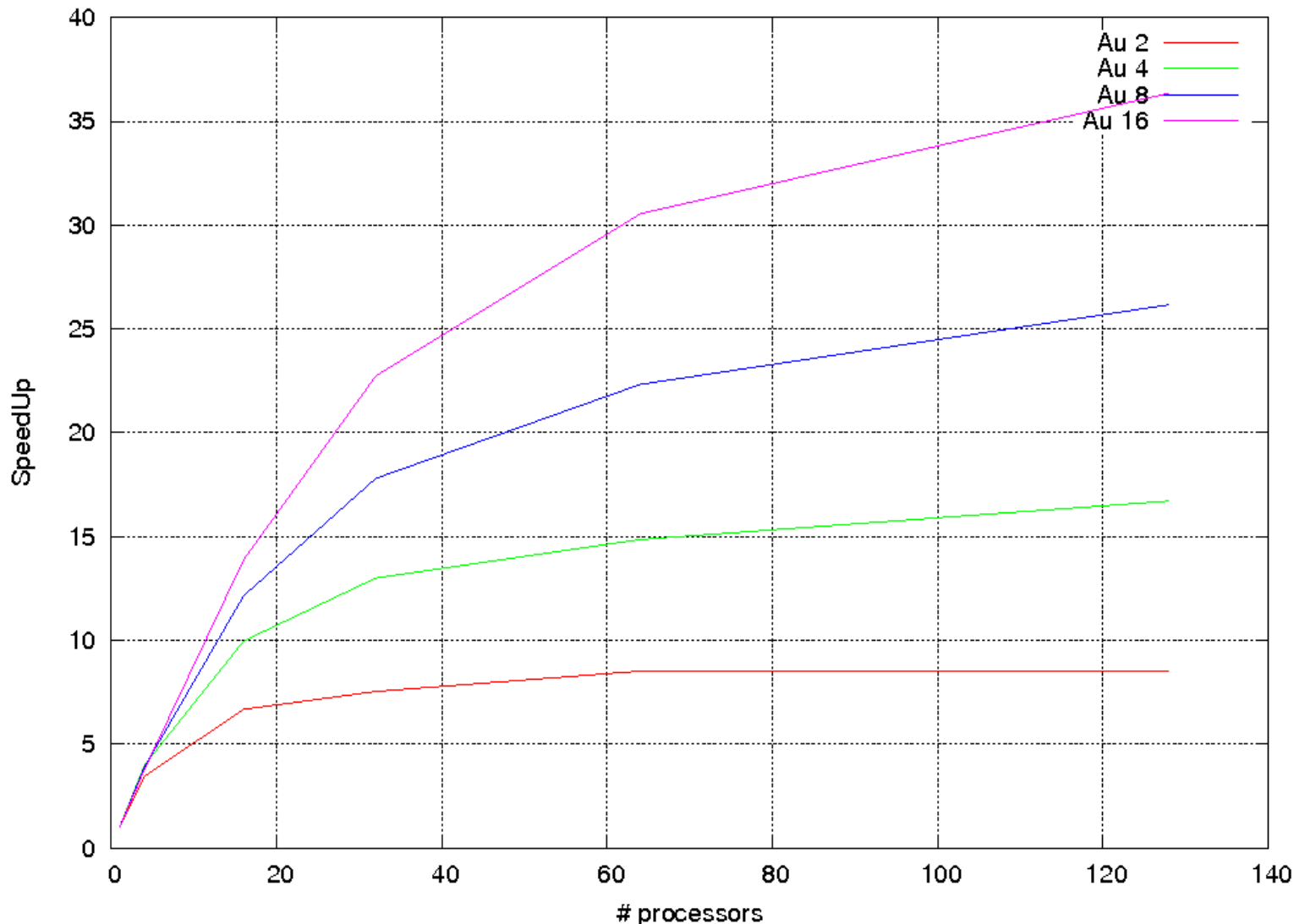
# of processors	<i>Time(s) COLLECT</i>			
	<i>dim. 1560</i>	<i>dim. 3120</i>	<i>dim. 6240</i>	<i>dim. 12480</i>
4	0.12	0.36	1.41	5.90
16	0.12	0.44	1.60	6.26
32	0.10	0.39	1.56	6.32
64	0.13	0.51	2.12	8.23
128	0.14	0.52	2.13	8.62

# of processors	<i>Time(s) DISTRIBUTE</i>			
	<i>dim. 1560</i>	<i>dim. 3120</i>	<i>dim. 6240</i>	<i>dim. 12480</i>
4	0.12	0.33	1.29	5.07
16	0.22	0.76	1.29	5.04
32	0.27	0.98	1.38	5.40
64	0.27	1.07	3.78	6.03
128	0.28	1.13	4.47	5.97

Communication time is “independent” from the number of processors



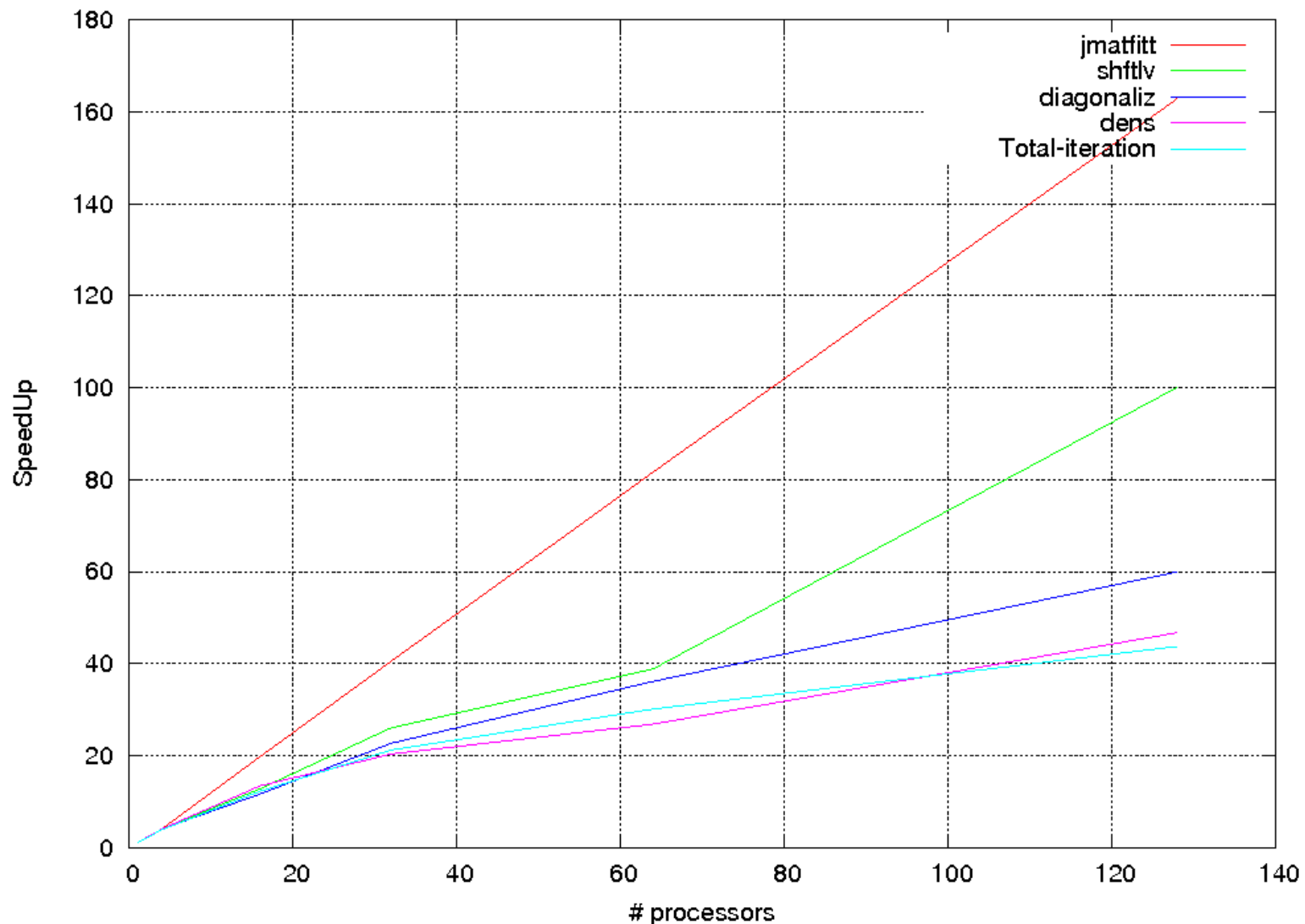
Parallelization of BERTHA



If we consider the Amdahl's law using 128 processors we reached roughly the 60% of the maximum theoretical speedup. (AMPI and use more processors)

Amdahl's law
 $\text{Speedup} = 1 / F$

Parallelization of BERTHA



Relative
speedup Au_{32}

With 128
processors
140 GB

Parallelization of BERTHA

```
DO block_num = 1, max_nom_of_block
  IF ((my_rank+((num_of_processors)*my_block_num)) == block_num)
    my_block_num = my_block_num + 1
    ALLOCATE block
    COMPUTE block
  END IF
ENDDO
```

CALL DISTRIBUTESPARSE

Each process compute a subset of all blocks constituting the entire J+K matrix, and store the result in a local array. At the end of the procedure the matrix is naturally distributed among all the processes,

DISTRIBUTESPARSE re-distributes the matrix obeying the two-dimensional block cyclic distribution, so that it can be used by the ScaLAPACK routines.

Parallelization of BERTHA

```
do l = 1, NOB
  2D_CBD_BLOCK(l) = transform (NDBLOCK(l))
  collect process rank in processor_ranks vector
enddo
```

```
call barrier
```

```
do i=0,numprocs-1
  i-th process broadcast processor_ranks vector
```

```
  if (myrank.eq.i) then
    for each j in processor_ranks
      if (j.ne.myrank)
        send new_block to j
      endif
    endfor
```

```
  else
    if (myrank is in processor_ranks)
      receive new_block
    endif
```

```
  call barrier
enddo
```

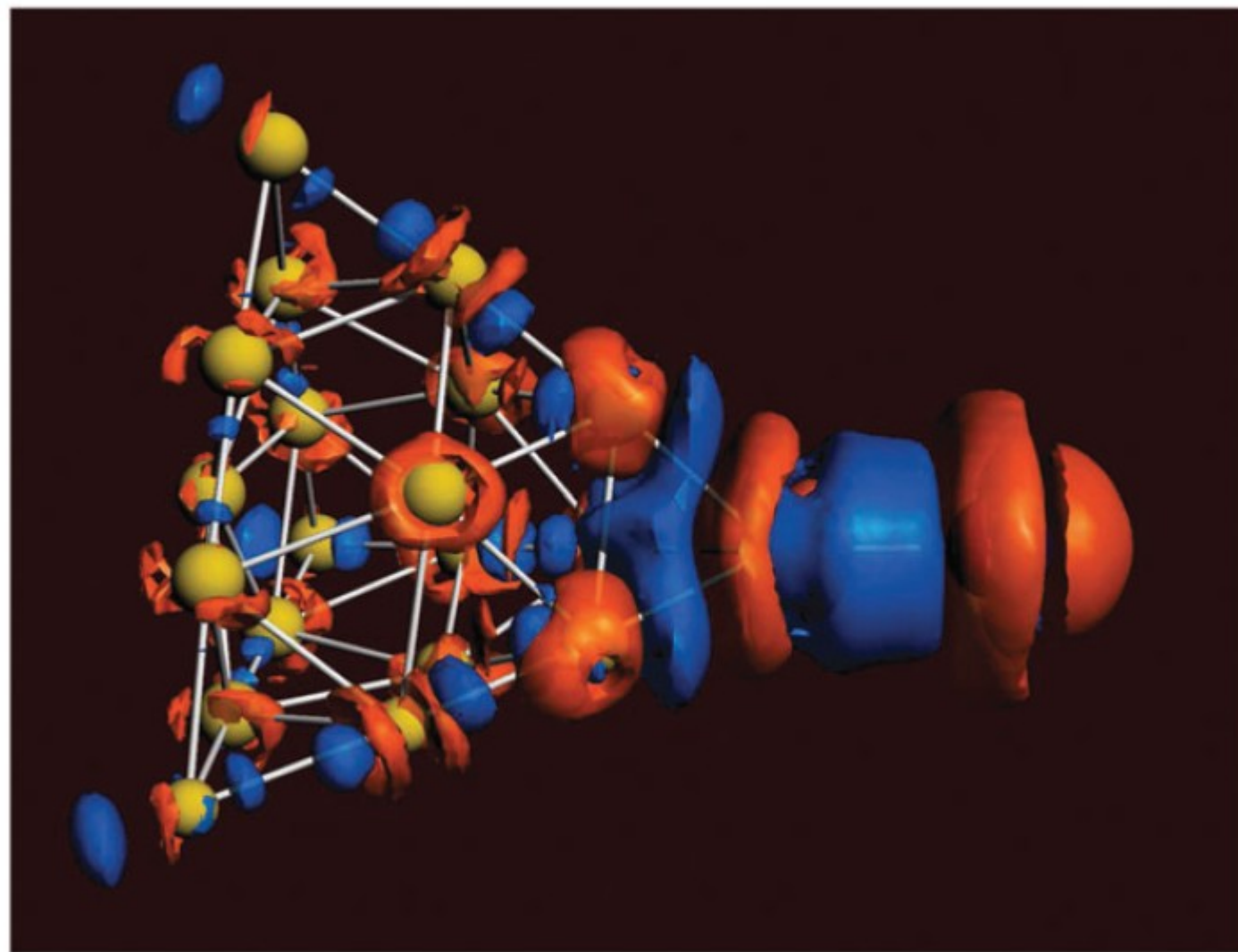
Parallelization of Betha

Cluster	Parallel scheme	Processor grid				
		2×2	4×4	4×8	8×8	8×16
Au ₂	MS	2.9	10.2	14.6	27.5	27.5
	DP	3.8	9.7	14.5	23.3	21.1
Au ₄	MS	3.0	13.3	24.3	36.2	55.8
	DP	3.9	14.4	23.7	33.1	46.3
Au ₈	MS	3.0	14.8	30.8	63.8	117.7
	DP	3.9	15.6	31.9	63.7	103.9
Au ₁₆	MS	3.0	15.0	31.5	62.8	124.2
	DP	4.0	16.1	32.4	63.8	123.8

Parallelization of Bertha

- The DKS arrays are always explicitly distributed among all processes and there is no need for a single global allocation. This also mean that, in the $J + K$ matrix construction, no process is lost to the task of coordinating the workload, which is a non-negligible gain in case only few processes are available.
- This explicit matrix distribution and the effective computation parallelism make the applicability range of the code essentially “open-ended” and, most of all, essentially portable on any parallel architecture including low-cost clusters

Parallelization of Bertha



DKS/BLYP
contour plot of the
electron density
difference upon
bond formation
between
Copernicium and
an Au₂₀ cluster.
Red
isodensity
surfaces identify
zones of density
decrease, blue
ones of
density increase.

Contents

- Grid Computing
- Green's function and the ADC
- Relativistic DFT
- GRID force-field pKa prediction
- GPGPU and DSP
- Sphere packing for Dye-sensitized solar cells

GRID: Applications and Parallelization

GRID programme:

a computational procedure for determining energetically favourable binding sites on molecules for functional groups of known structure through the use of **PROBES**.

The PROBE is moved through a grid of points superimposed on the target molecule. Its interaction energy with the target molecule is computed by an empirical energy function.

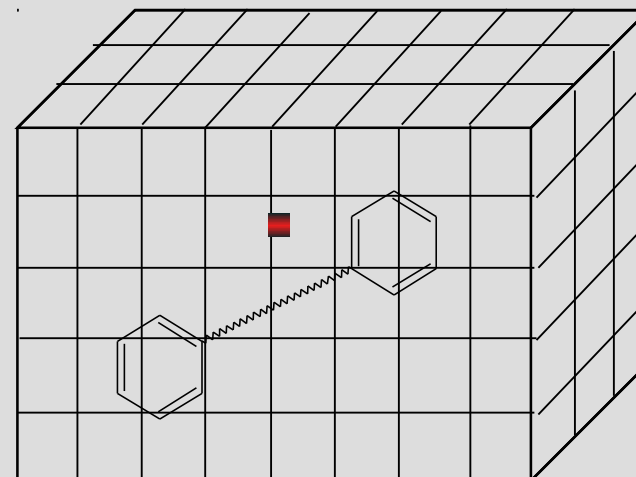
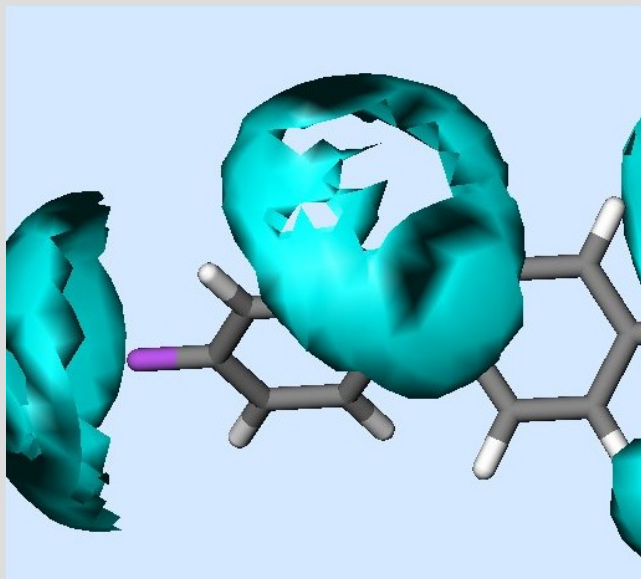
$$E_{XYZ} = \sum[E_{LJ}] + \sum[E_{HB}] + \sum[E_Q] + [S]$$

E_{LJ} = Lennard-Jones potential

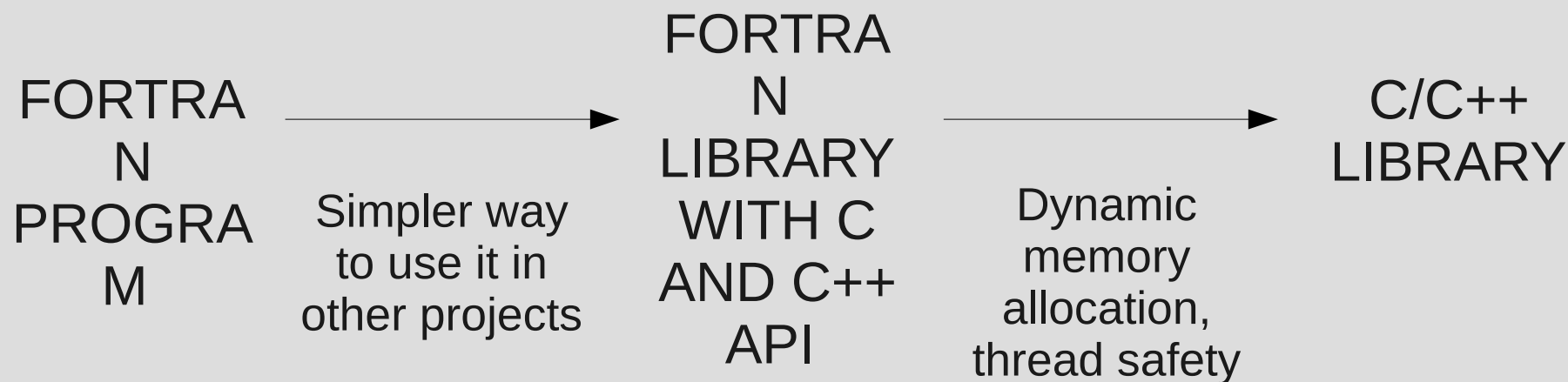
E_{HB} = hydrogen bonding interaction energy

E_Q = electrostatic function

S = entropic term



GRID: Applications and Parallelization



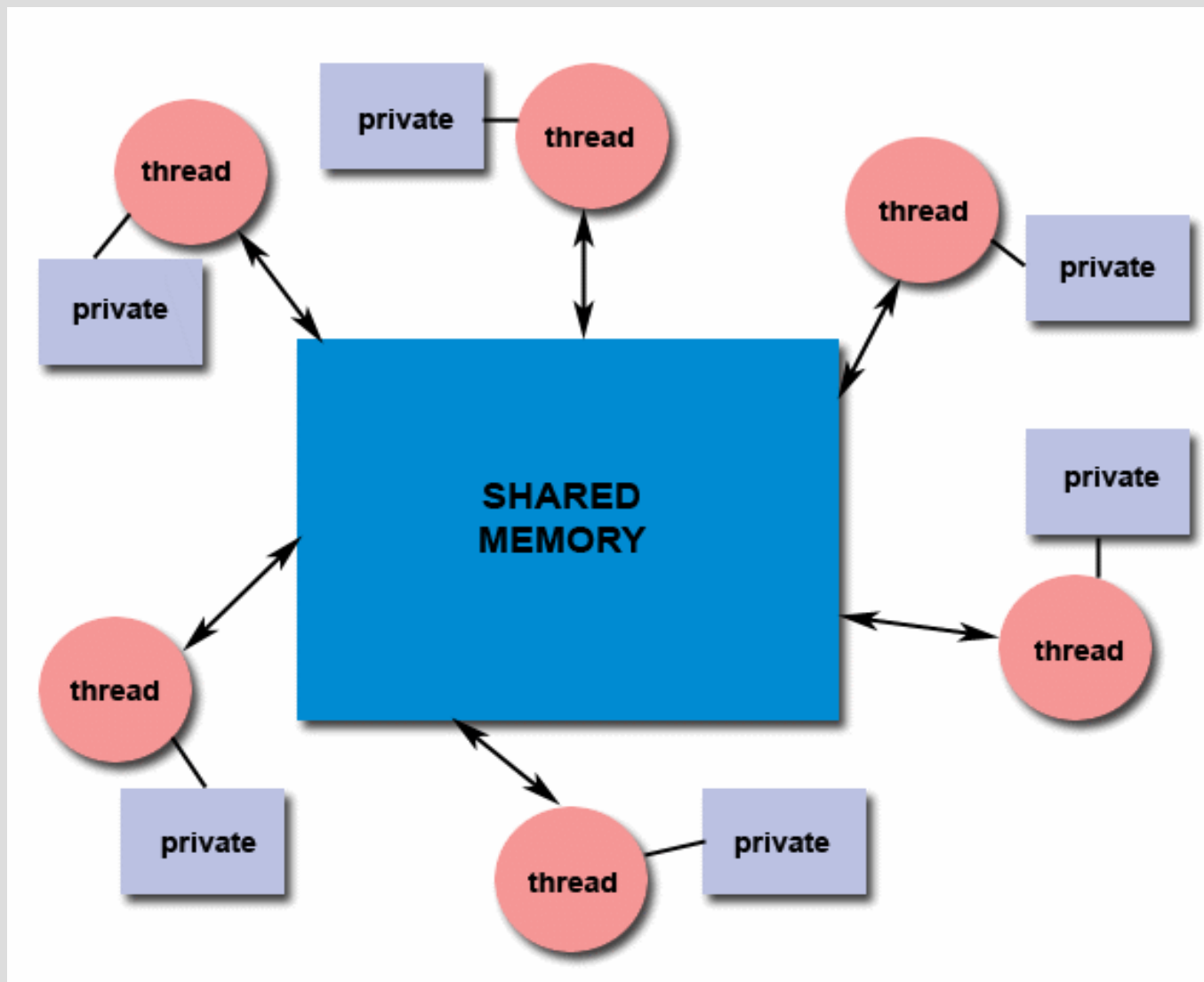
1. pKa prediction for small molecules (and proteins)
2. VolSurf, Almond and SHOP use molecular descriptors from 3D Molecular Interaction Fields (MIFs) produced by GRID
3. MetaSite is a computational procedure that predicts metabolic transformations related to cytochrome-mediated reactions in phase I metabolism

J. Chem. Inf. Model., 47 (6), 2172 (2007)

J. Chem. Inf. Model., 49 (1), 68 (2009)

PROTEINS: Structure, Function, and Bioinformatics, (2009)

GRID: Applications and Parallelization



GRID: Applications and Parallelization

- Easiest way is one thread for each probes (excellent speedup)
- One thread for each XY plane of the grid

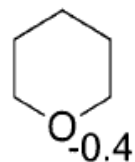
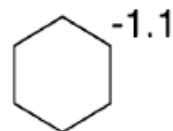
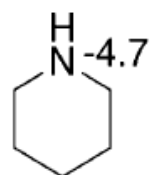
N threads	CPU Time	Wall Time	Speedup
Serial	234.40	234.40	1
2	235.05	119.67	1.96
3	235.81	83.15	2.82
4	237.24	62.89	3.35

Intel Core2 Quad CPU Q6600
2.40GHz

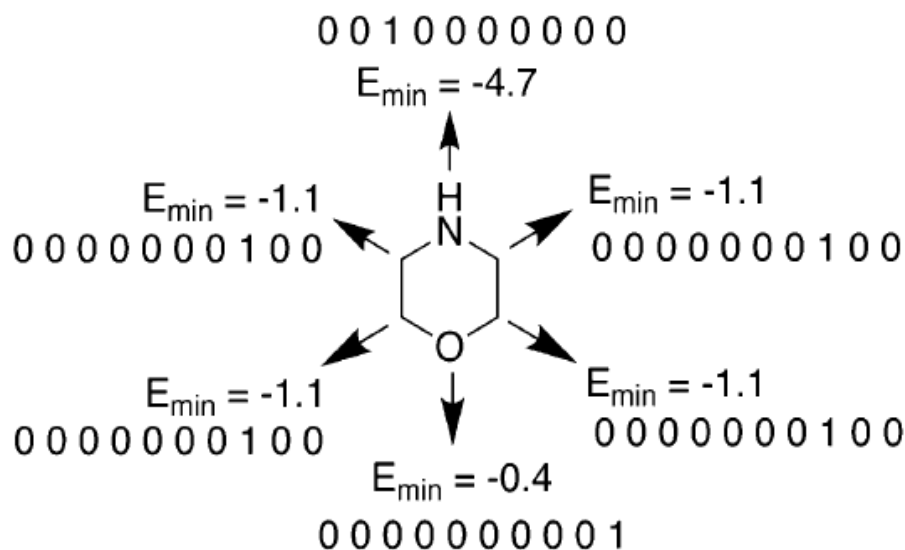
N threads	CPU Time	Wall Time	Speedup
Serial	181.54	181.57	1
2	185.28	95.12	1.91
3	184.20	64.72	2.81
4	182.68	48.50	3.74
5	183.58	39.72	4.57
6	184.78	33.46	5.43
7	184.51	28.45	6.38
8	185.46	24.76	7.33

Intel Xeon E5420 2.50GHz

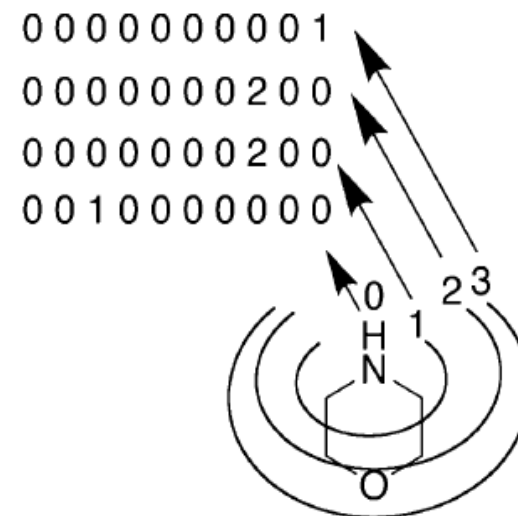
Pka prediction



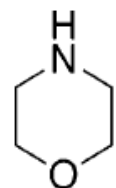
a)



b)



c)



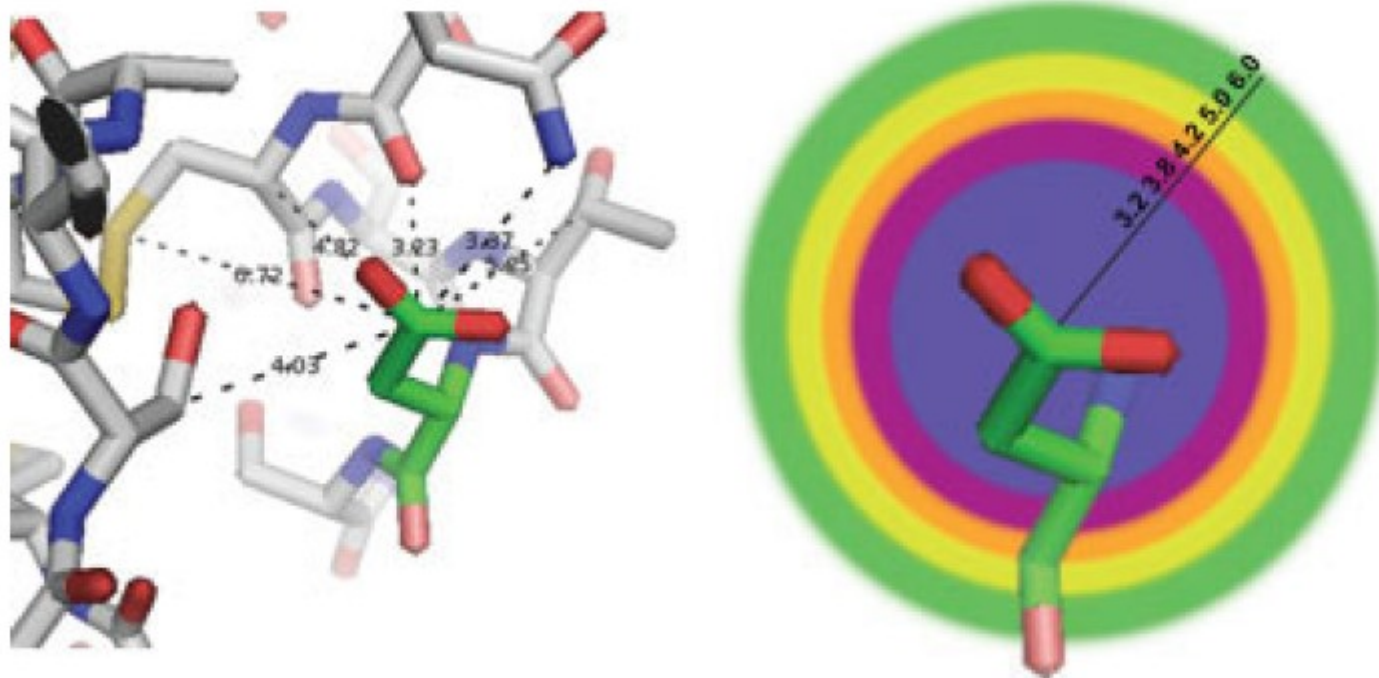
Binned E_{\min} at each topological distance using Probe "O"

0010000000 00000000200 00000000200 00000000001

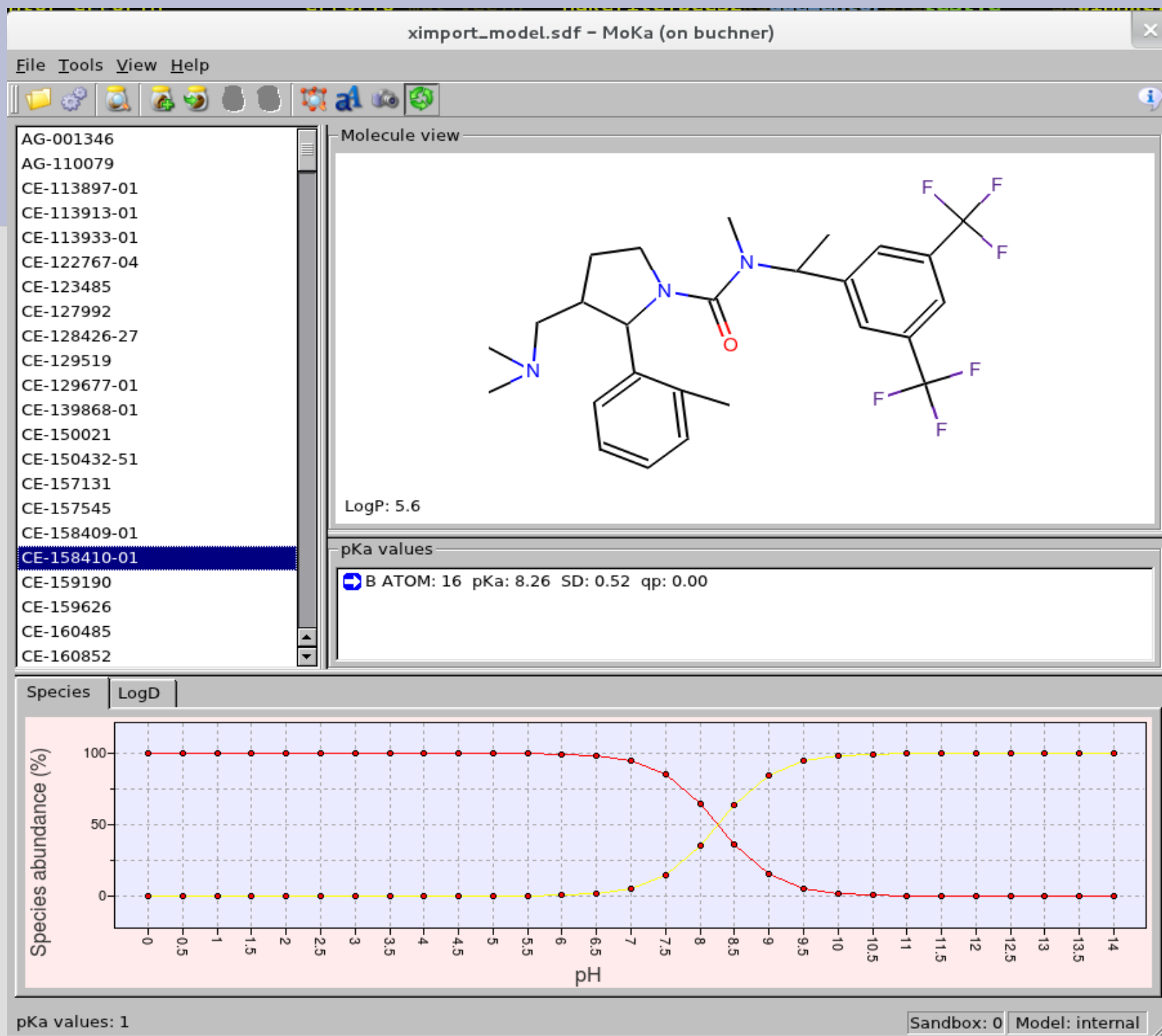
0 1 2 3

J. Chem. Inf. Model., 47 (6), 2172 (2007)
Chemistry and Biodiversity, 6 (11), 1812, (2009).

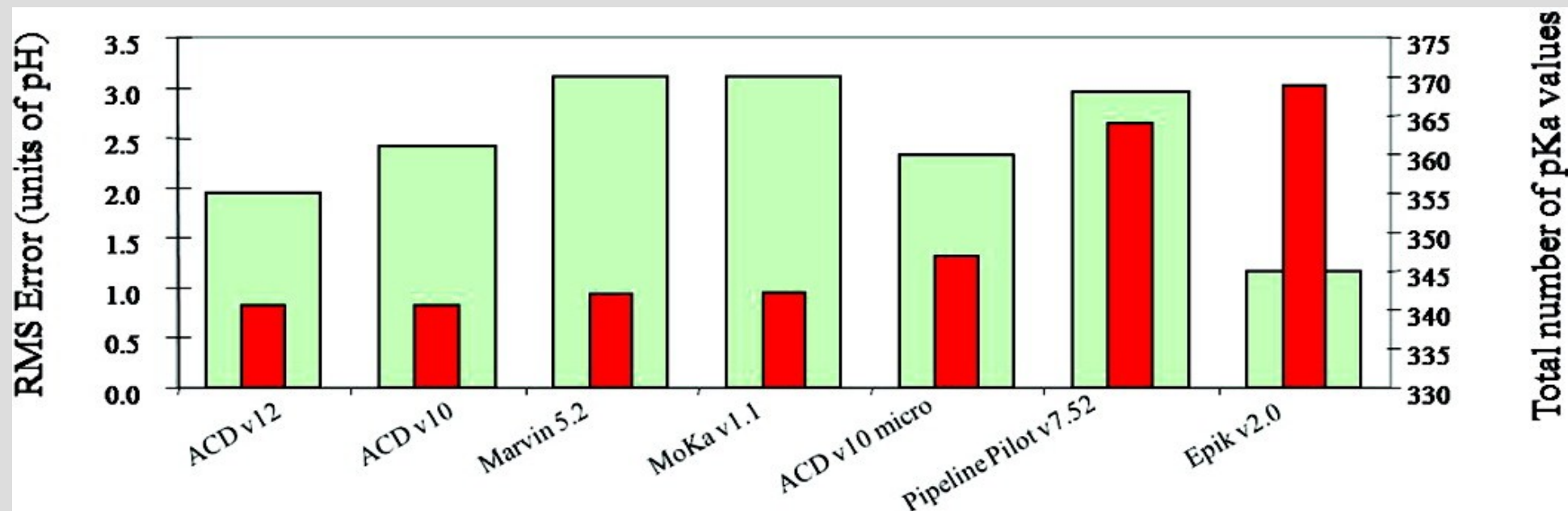
Pka prediction



PROTEINS: Structure, Function, and Bioinformatics, (2009)



Pka prediction



Even though the topological method MoKa was noticeably faster than ACD, the accuracy of those two methods and Marvin was statistically indistinguishable, with a root-mean-squared error of about 1 pKa unit compared to experiment

ximport_model.sdf - MoKa Kibitzer (on buchner)

File Edit Tools View Help

Objects

Name	Accuracy
<input checked="" type="checkbox"/> AG-001346	2
<input checked="" type="checkbox"/> AG-110079	1
<input checked="" type="checkbox"/> CE-113897-01	2
<input checked="" type="checkbox"/> CE-113913-01	1
<input checked="" type="checkbox"/> CE-113933-01	1
<input checked="" type="checkbox"/> CE-122767-04	1
<input checked="" type="checkbox"/> CE-123485	2
<input checked="" type="checkbox"/> CE-127992	1
<input checked="" type="checkbox"/> CE-128426-27	1
<input checked="" type="checkbox"/> CE-129519	1
<input checked="" type="checkbox"/> CE-129677-01	1
<input checked="" type="checkbox"/> CE-139868-01	1
<input checked="" type="checkbox"/> CE-150021	2
<input checked="" type="checkbox"/> CE-150432-51	1
<input checked="" type="checkbox"/> CE-157131	1
<input checked="" type="checkbox"/> CE-157545	2
<input checked="" type="checkbox"/> CE-158409-01	1
<input checked="" type="checkbox"/> CE-158410-01	1
<input checked="" type="checkbox"/> CE-159190	1
<input checked="" type="checkbox"/> CE-159626	1
<input checked="" type="checkbox"/> CE-160485	1
<input checked="" type="checkbox"/> CE-160852	3
<input checked="" type="checkbox"/> CE-186912-01	2
<input checked="" type="checkbox"/> CE-210666	1

Molecule view

pKa values

	Type	Atom	Exp pKa	MoKa pKa	QP	Info
1	<input checked="" type="checkbox"/> B	2	6.32	5.87	0.00	
2	<input checked="" type="checkbox"/> A	39	8.09	9.03	0.00	
3	A	22	N/A	11.40	-0.23	
4	A	8	N/A	11.73	-0.07	
5	A	16	N/A	12.39	1.44	

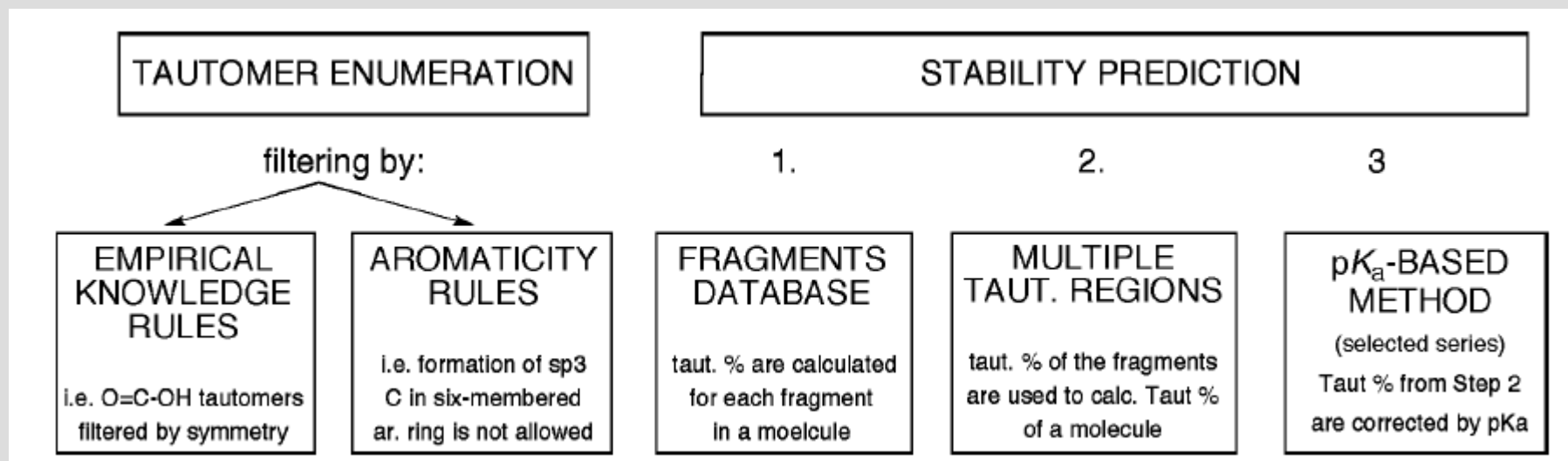
File loaded. 250 structure(s) available

Model: internal

1000000



Tautomer enumeration and stability

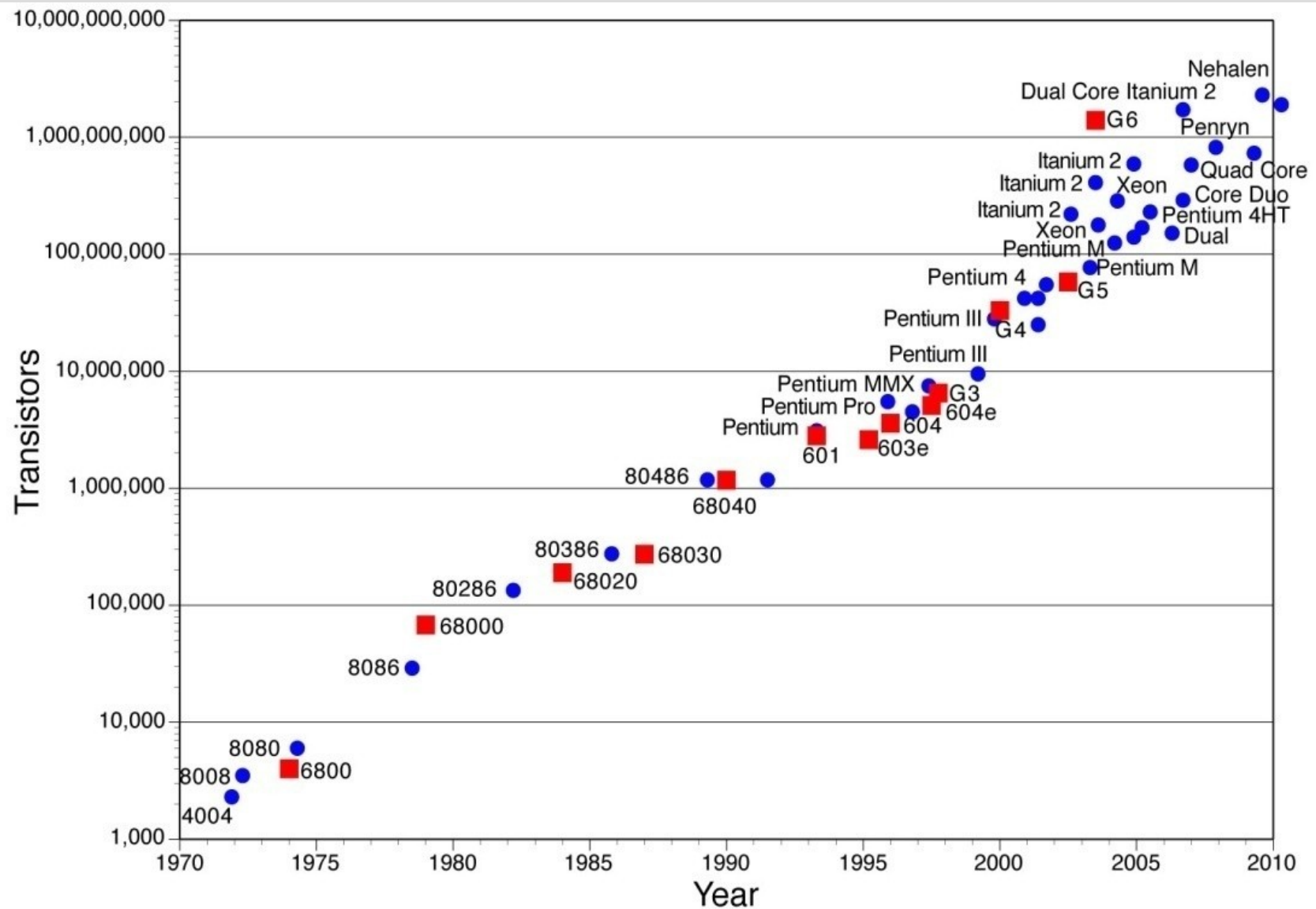


We introduced also an initial “normalization” procedure, to be sure to produce always the same series of tautomers.

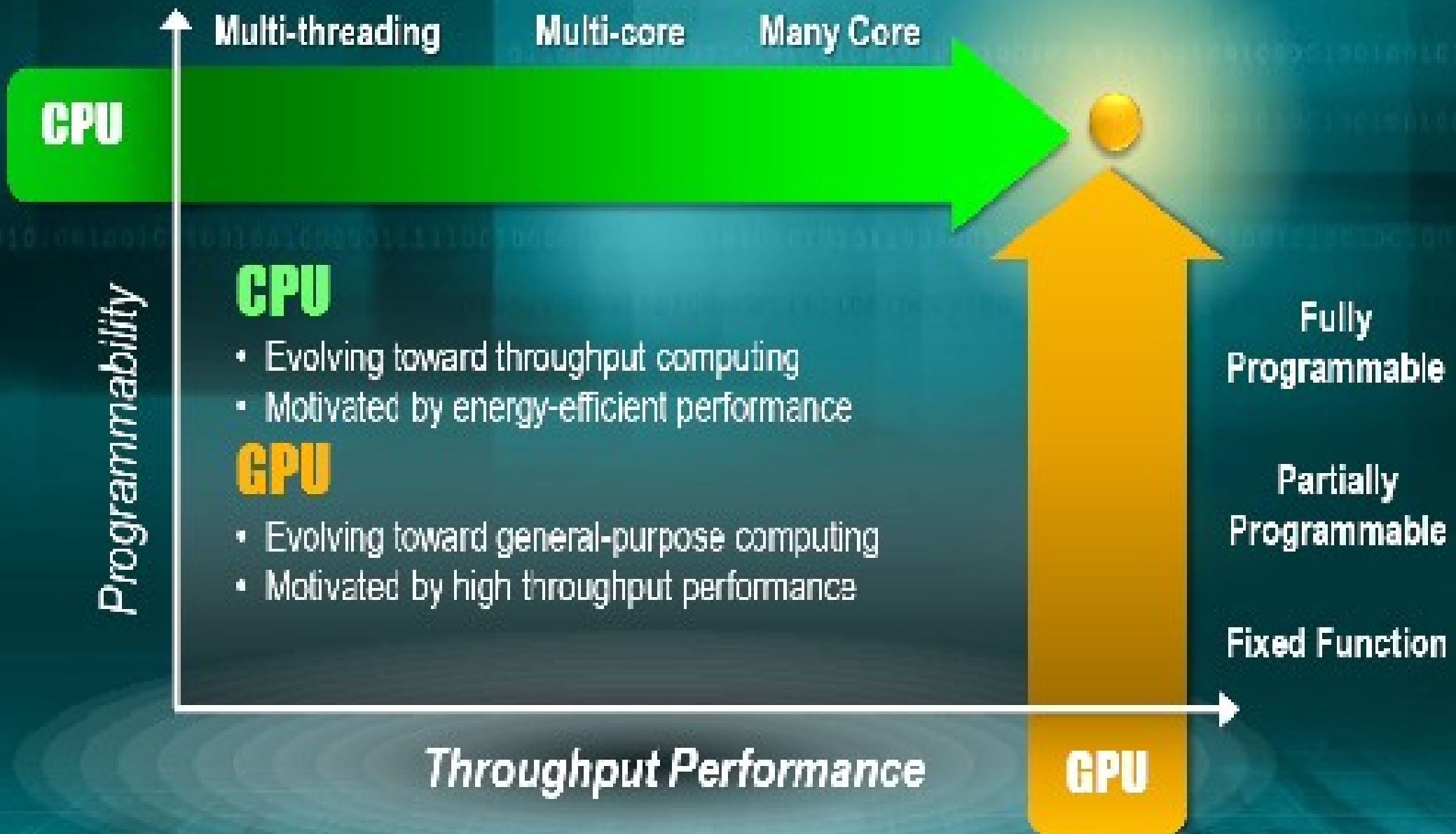
Contents

- Grid Computing
- Green's function and the ADC
- Relativistic DFT
- GRID force-field pKa prediction
- **GPGPU and DSP**
- Sphere packing for Dye-sensitized solar cells

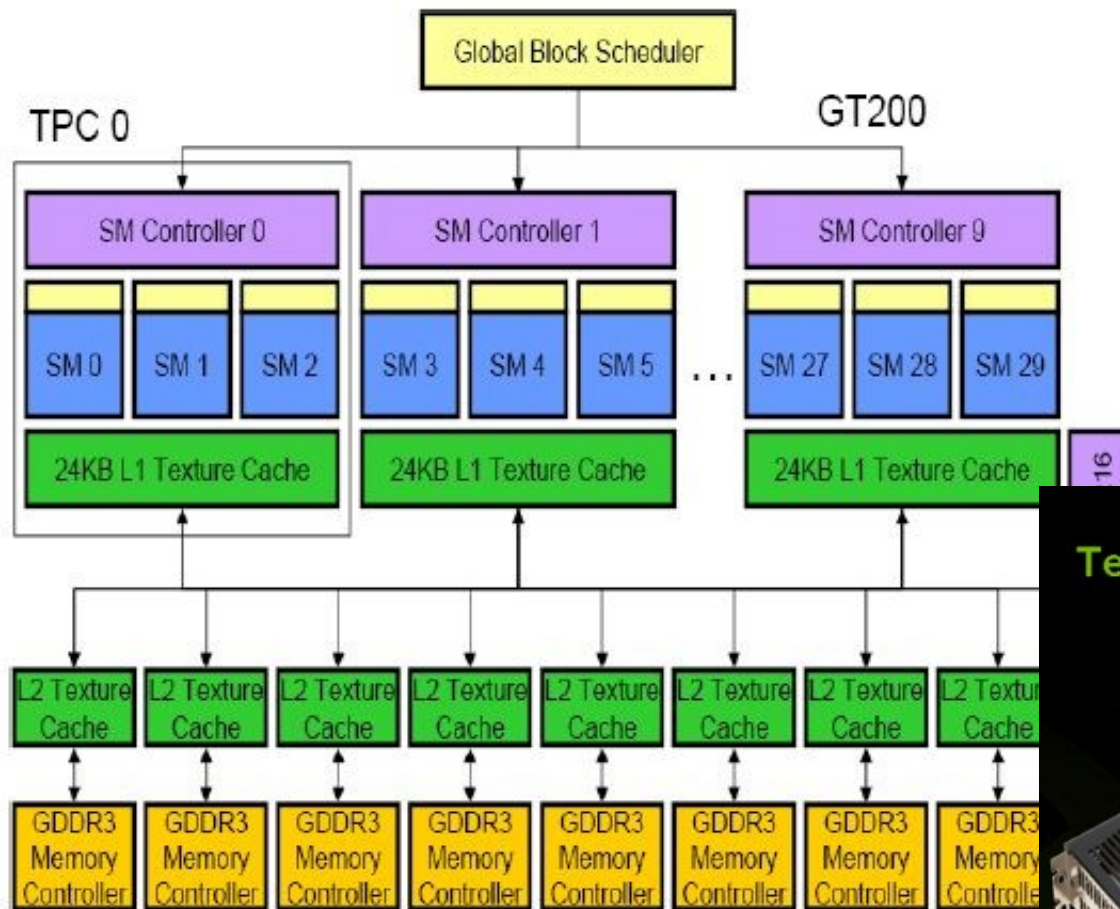
Moore's law



Architecture Evolution: A Collision Course



GPU

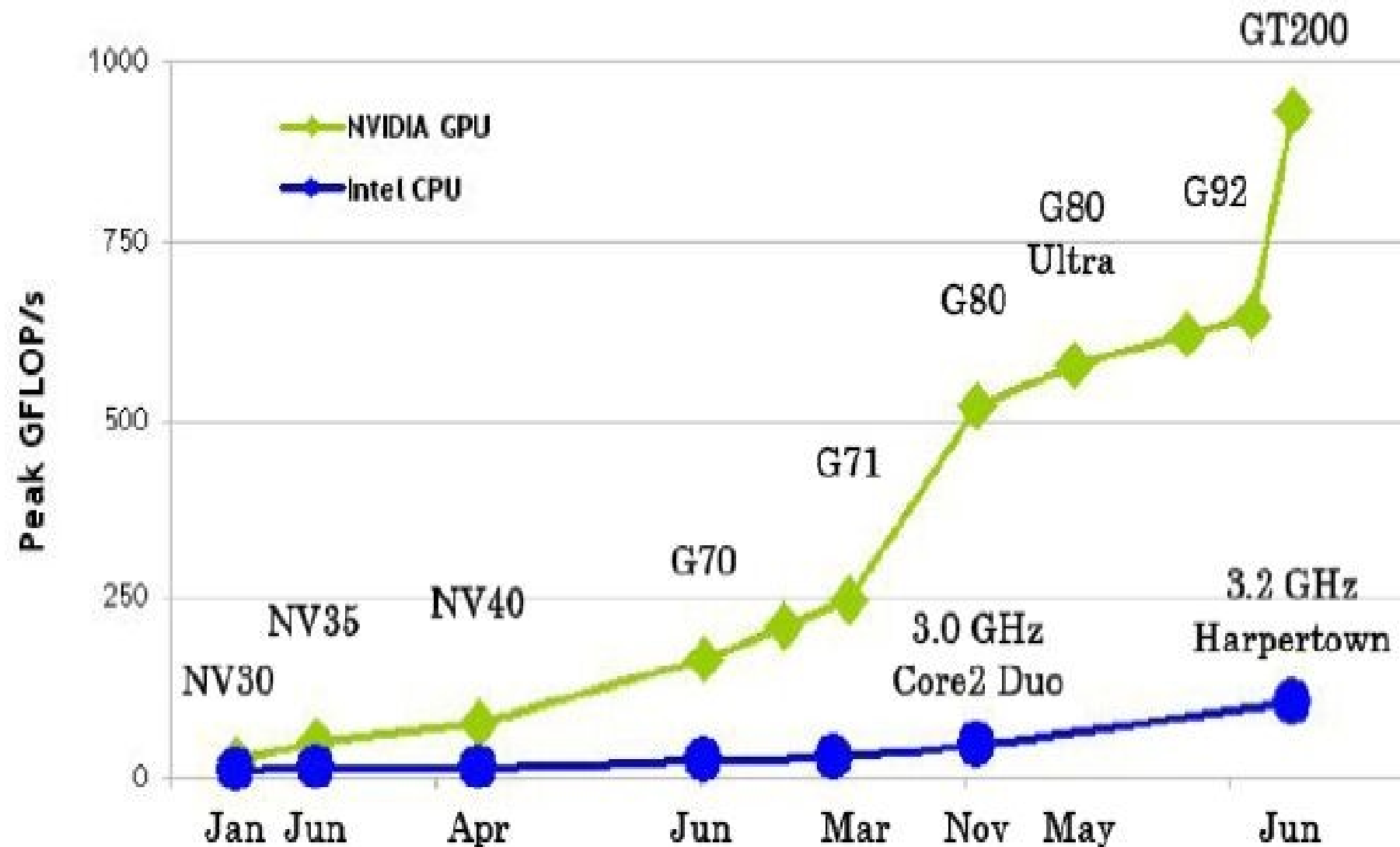


Tesla C1060 Computing Processor

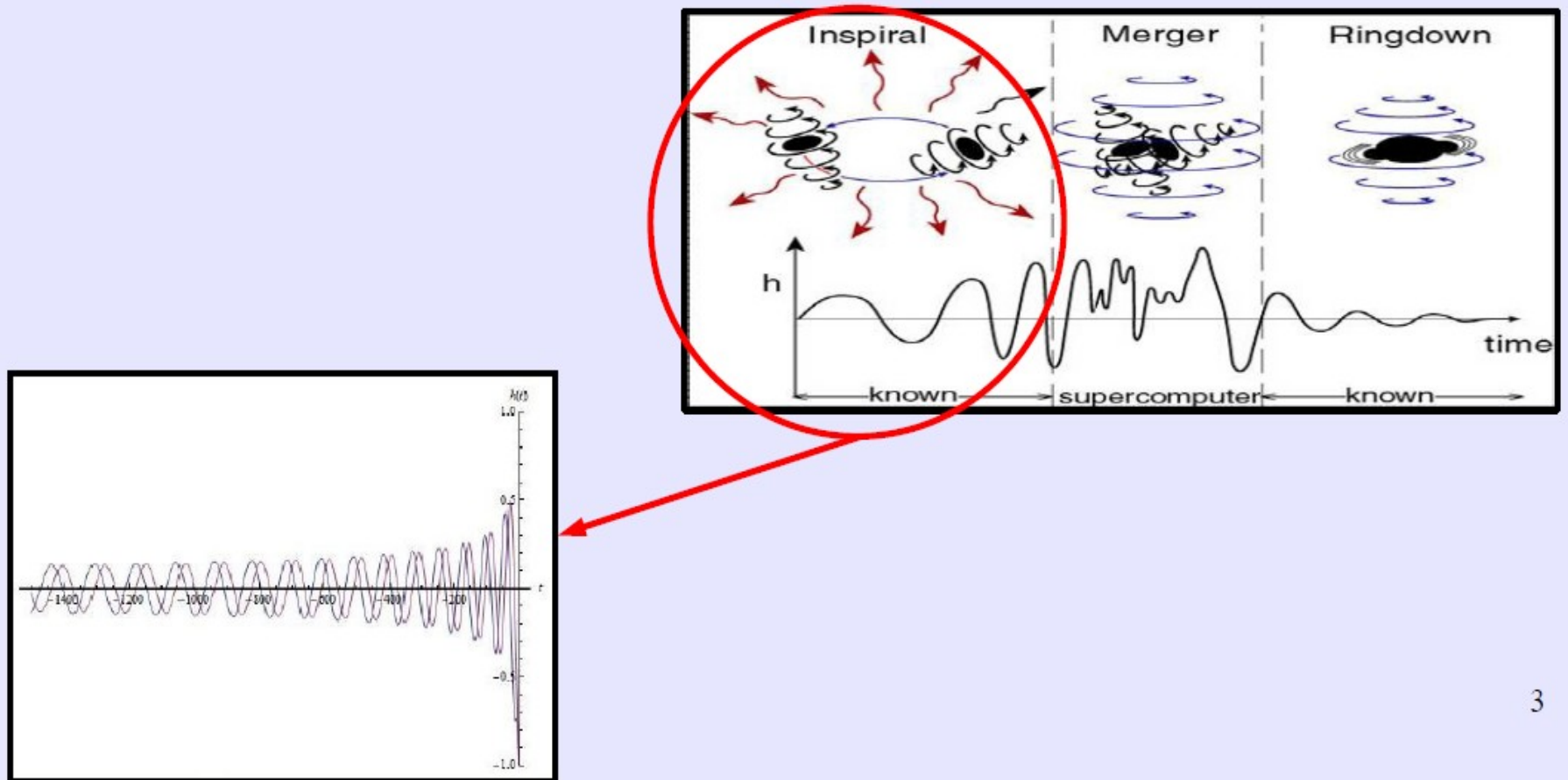


Processor	1 x Tesla T10
Number of cores	240
Core Clock	1.33 GHz
On-board memory	4.0 GB
Memory bandwidth	102 GB/sec peak
Memory I/O	512-bit, 800MHz GDDR3
Form factor	Full ATX: 4.736" (H) x 10.5" (L) Dual slot wide
System I/O	PCIe x16 Gen2
Typical power	160 W

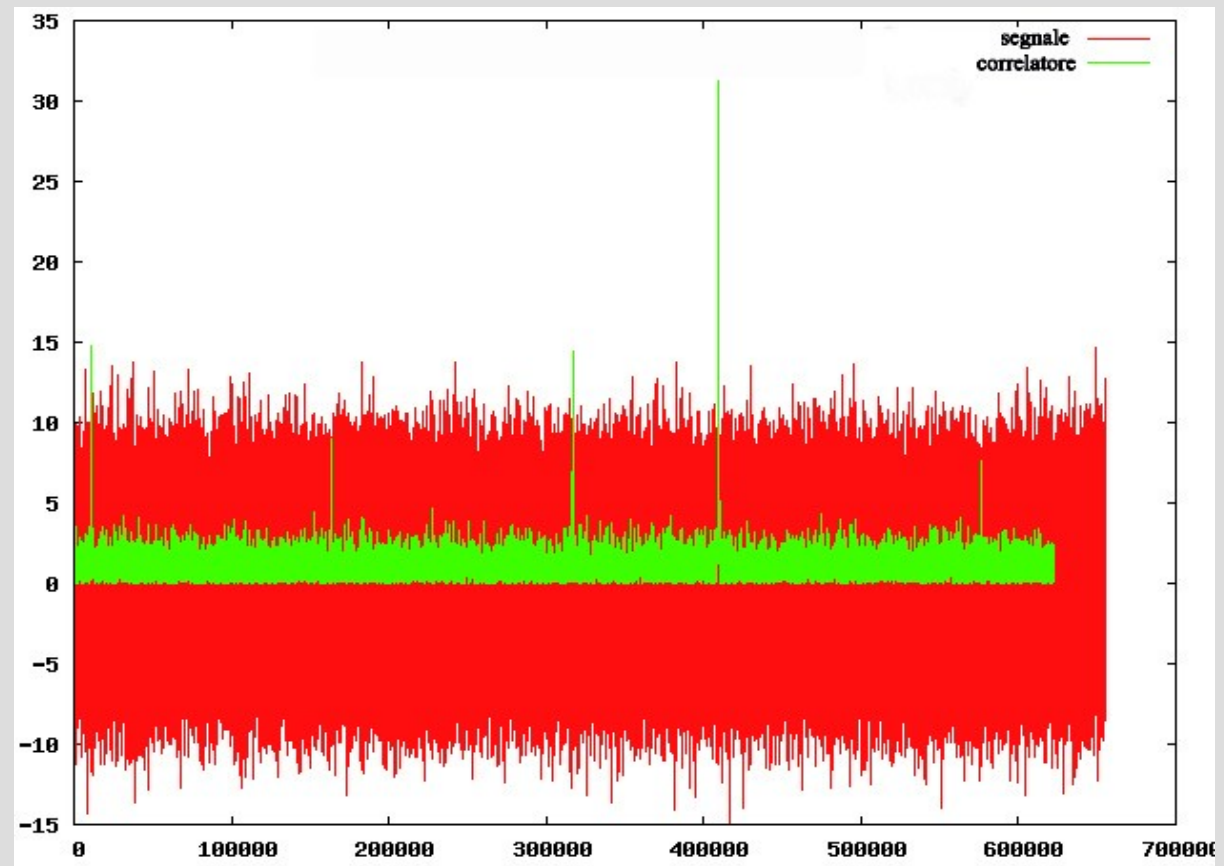
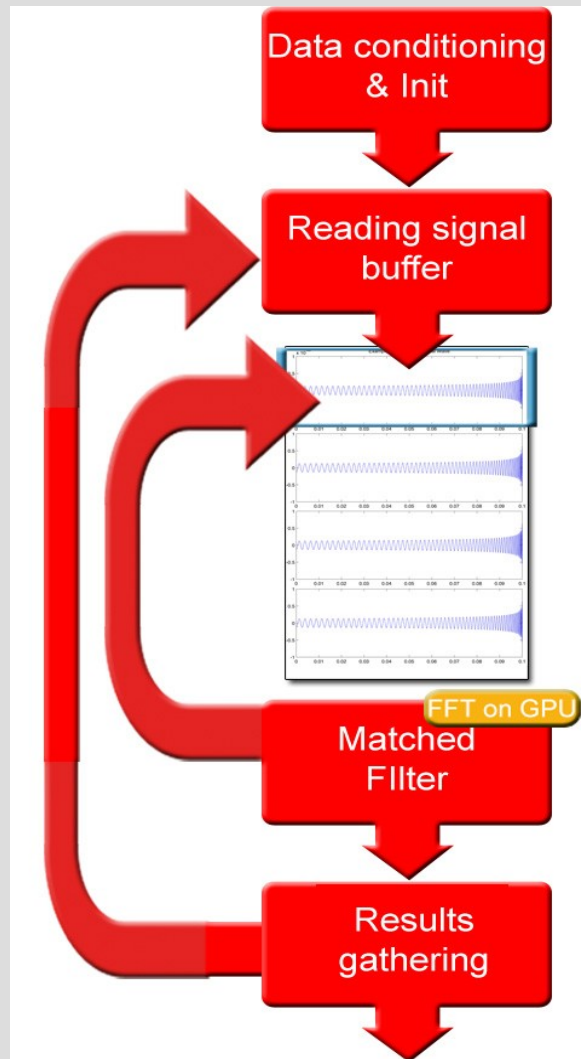
GPU



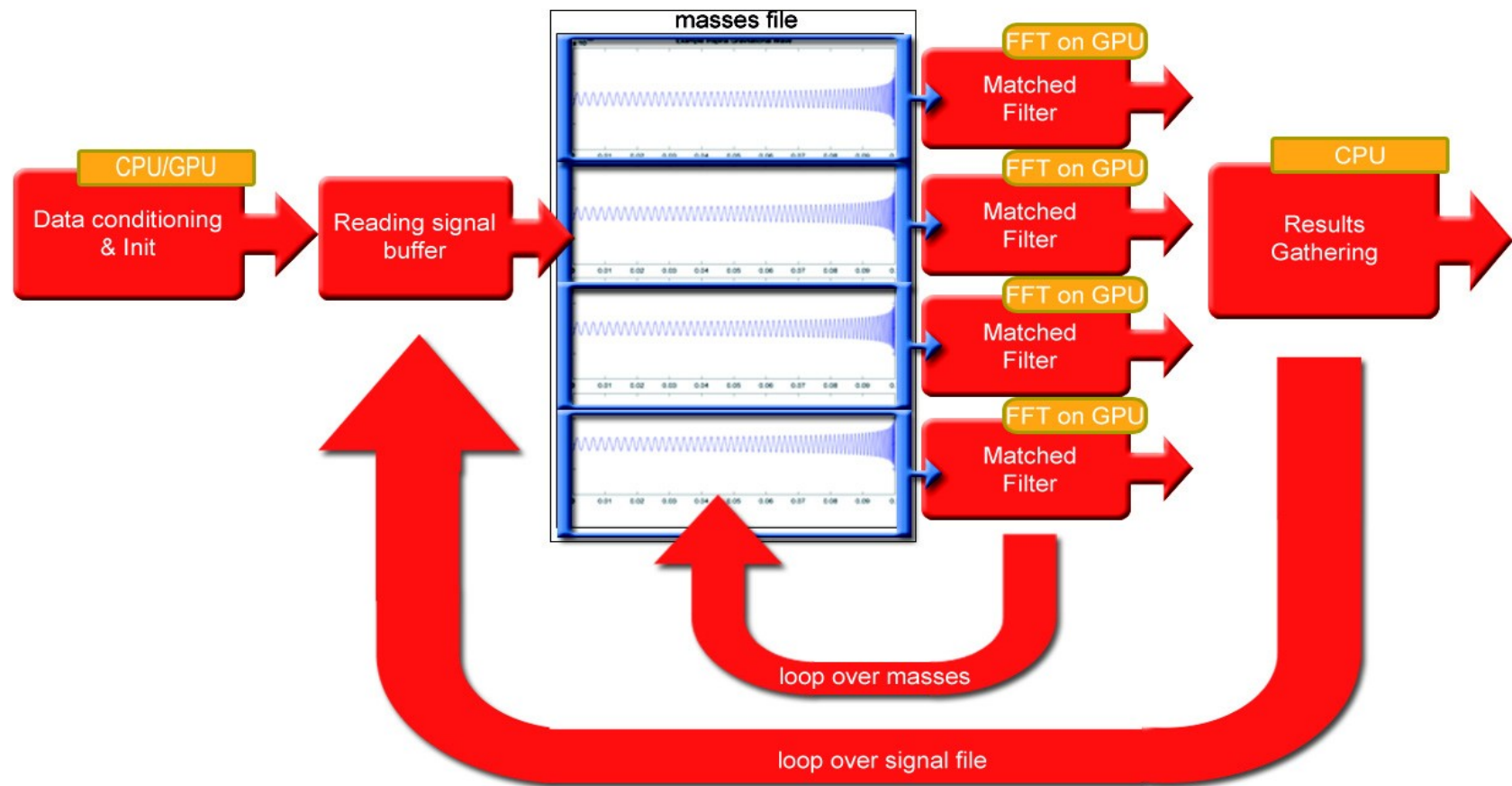
Coalescing binaries



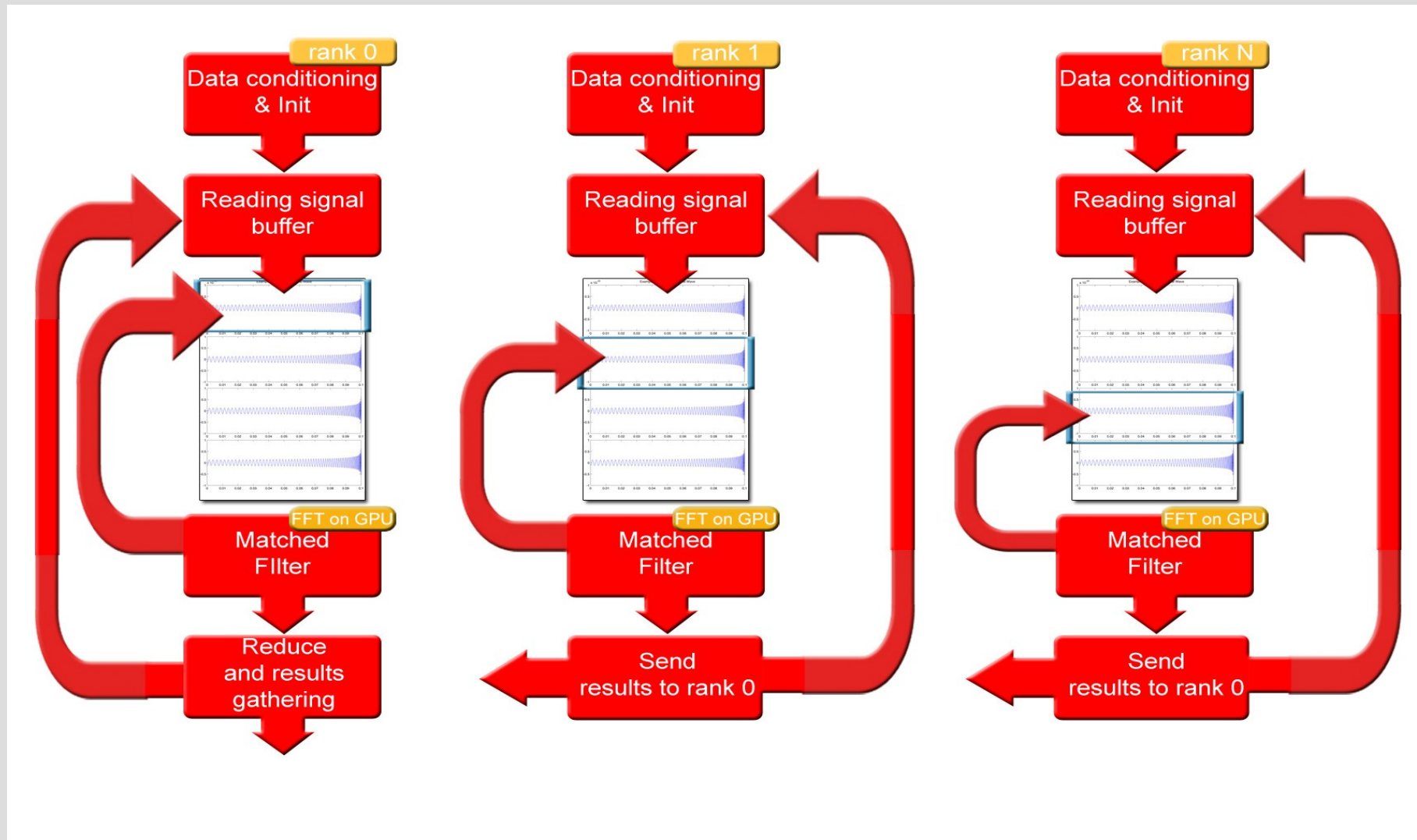
Signal detection - Single GPU



MultiGPU - Multithreading



MultiGPU - MPI



Results

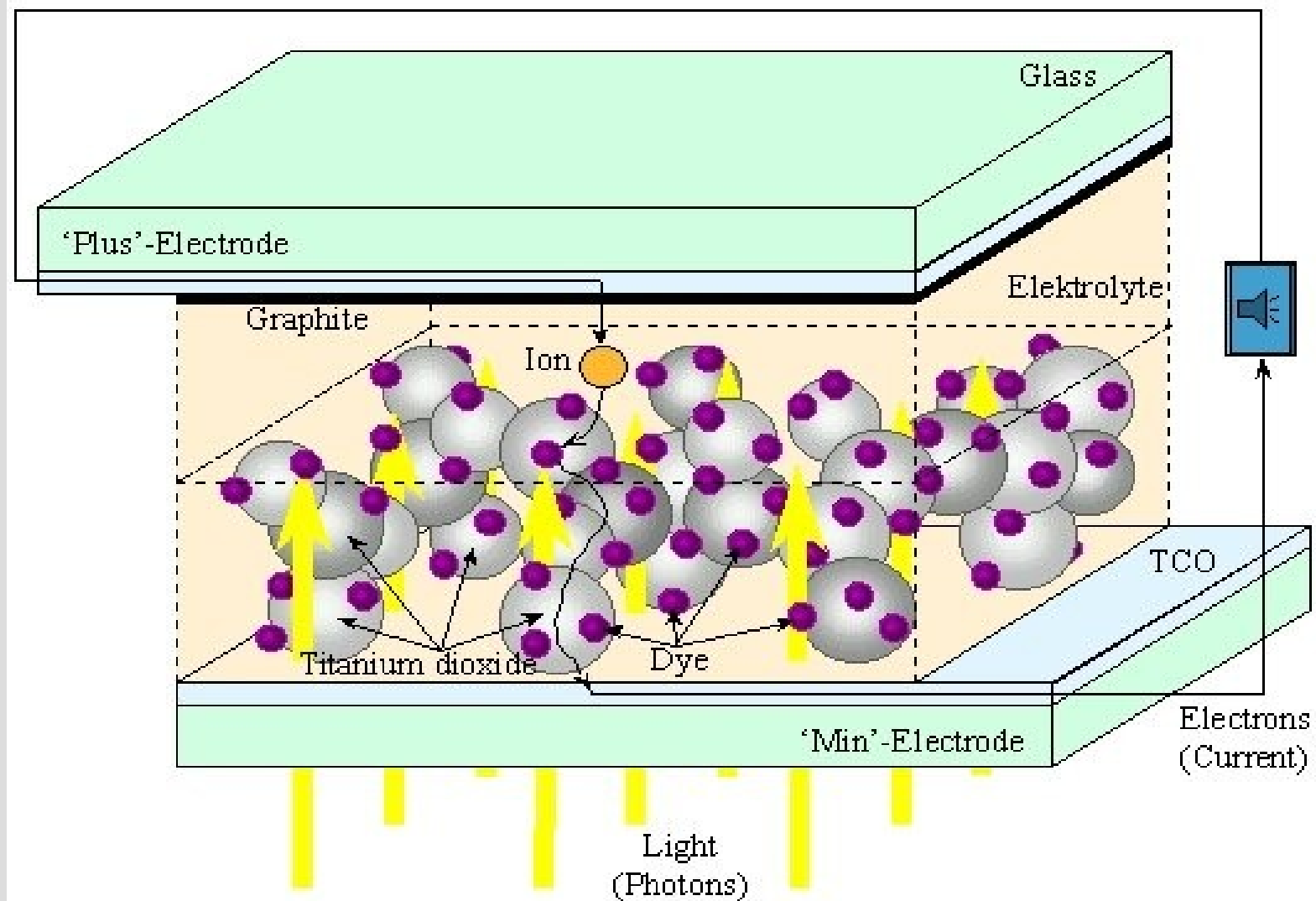
- The CPU implementation of the algorithm was able to compute 3 templates per seconds, the GPU implmentation, taking advantage of up to 4 GPUs, we achieved an improvement by 2 orders of magnitude being able to process about **400** templates per second.

Einstein gravitational wave Telescope conceptual design study", ET-0106A-10

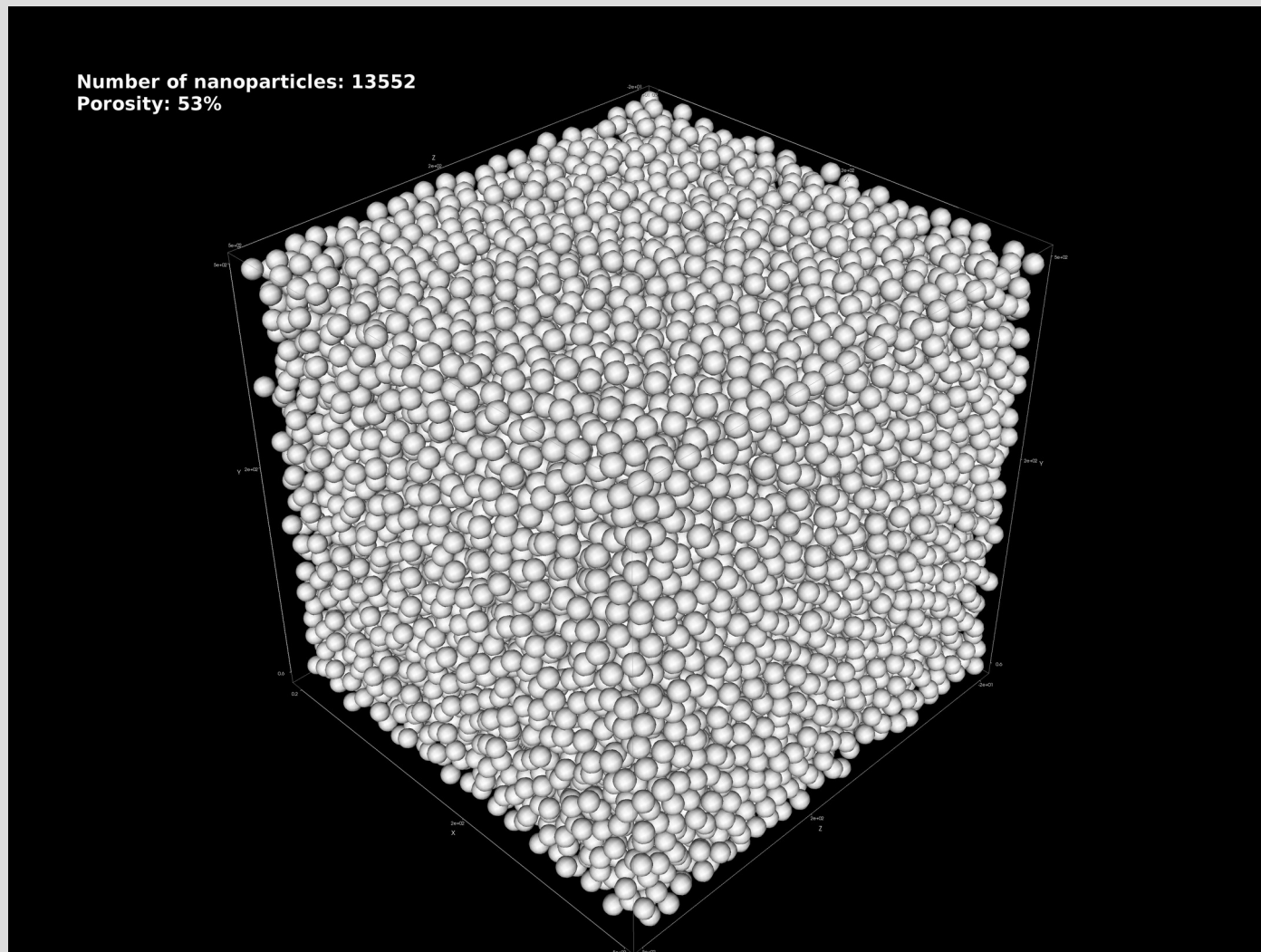
Contents

- Grid Computing
- Green's function and the ADC
- Relativistic DFT
- GRID force-field pKa prediction
- GPGPU and DSP
- Sphere packing for Dye-sensitized solar cells

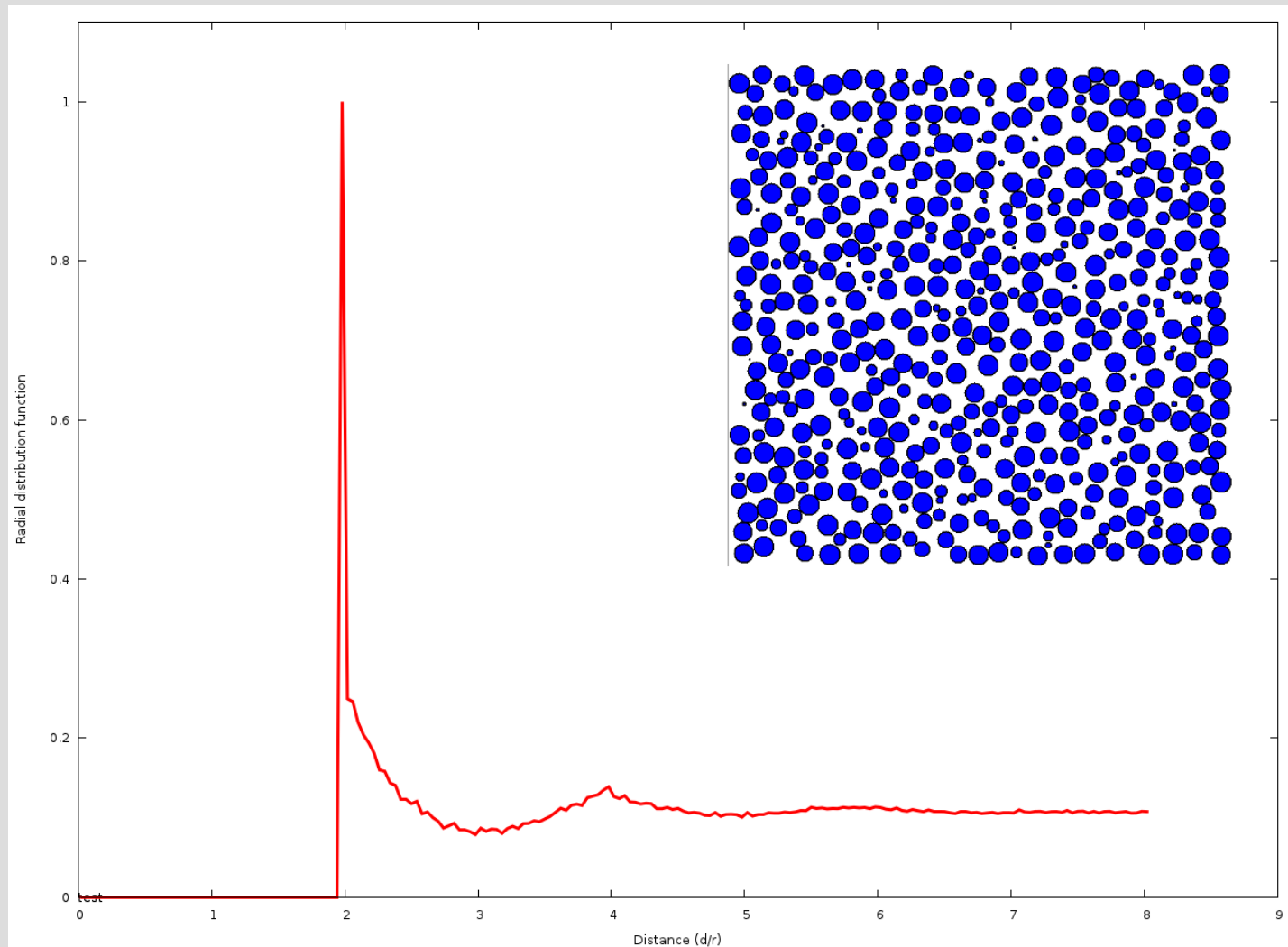
Dye-sensitized solar cells



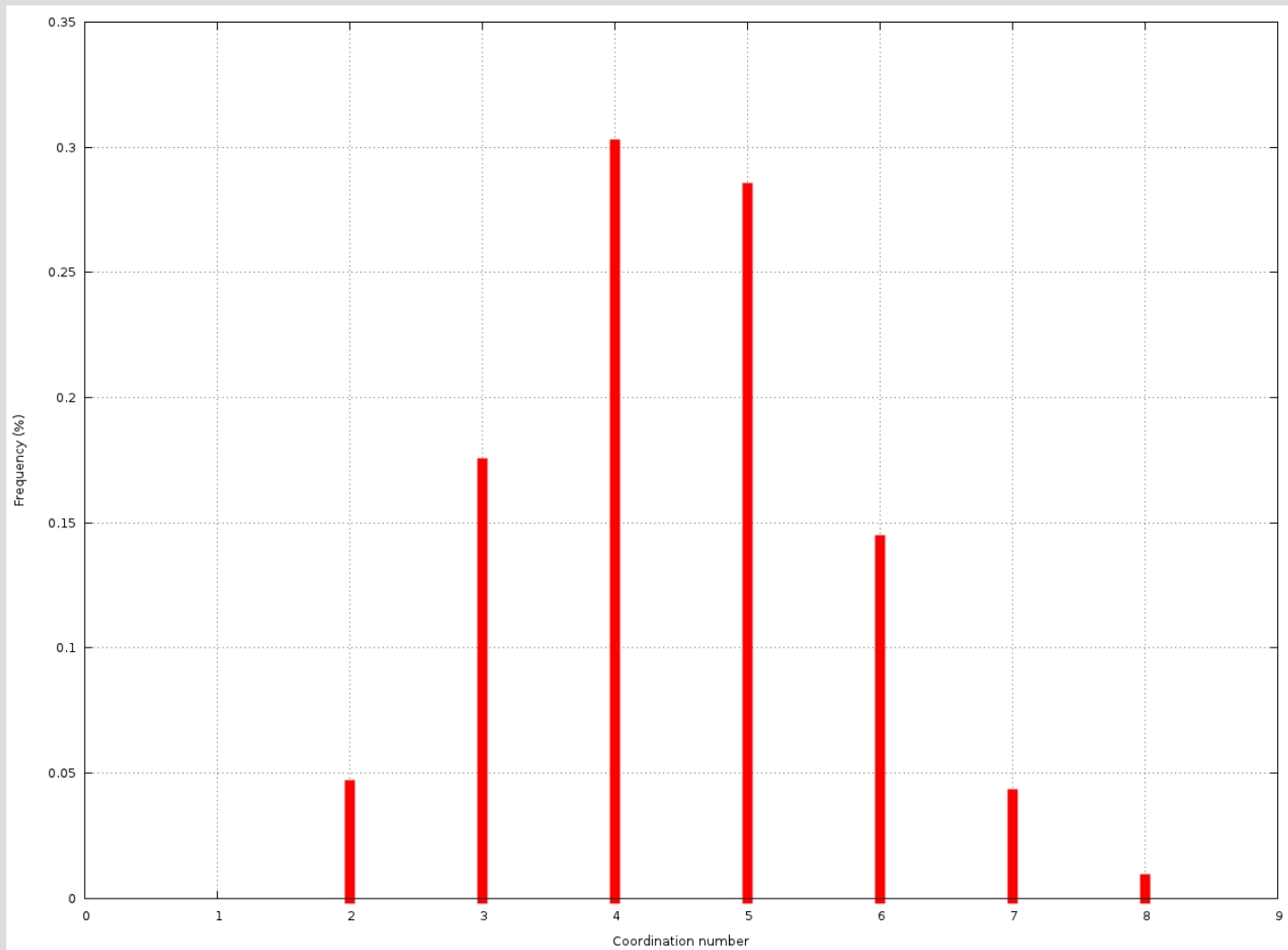
Random Sphere Packing



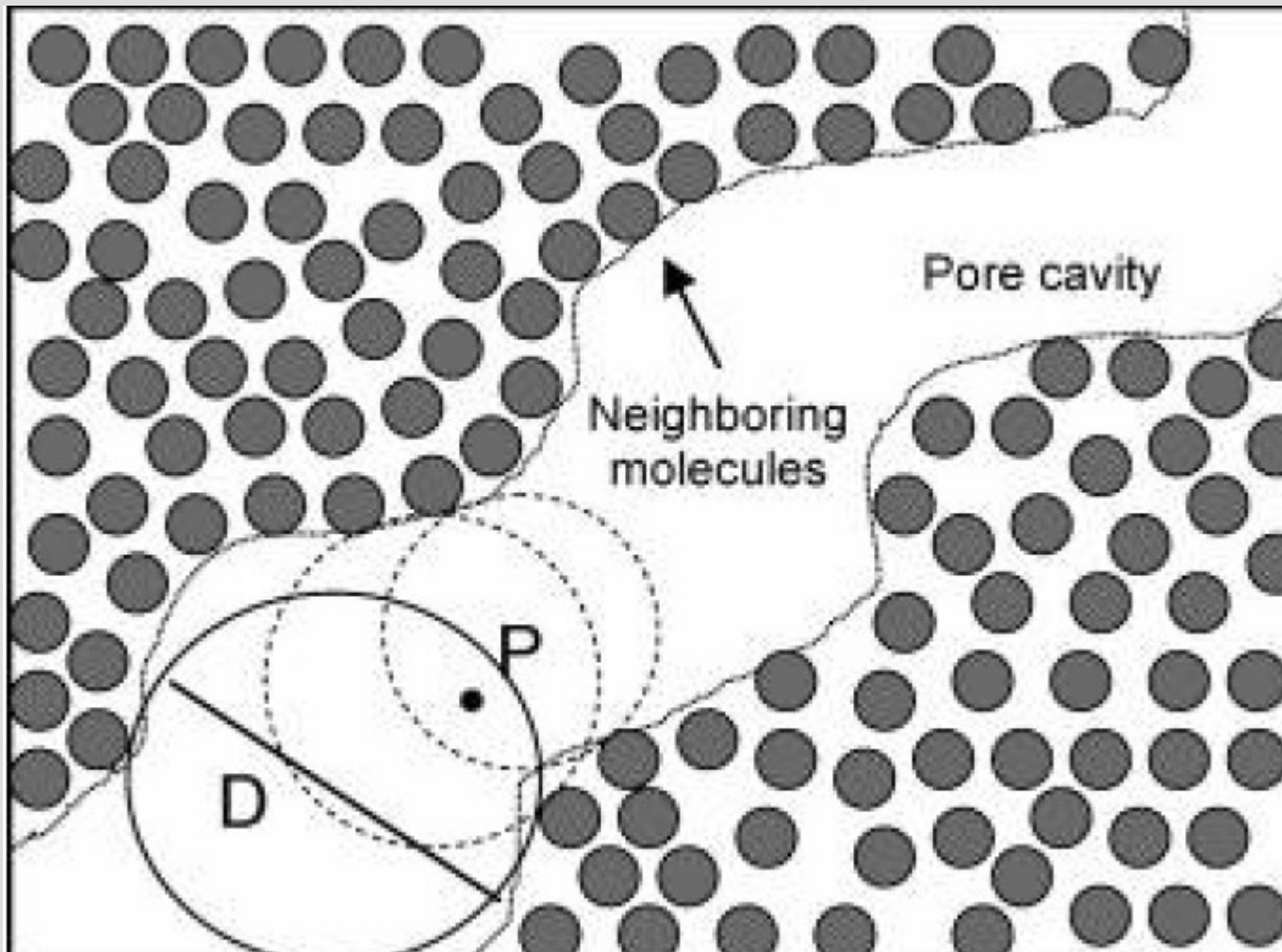
Random Sphere Packing



Coordination Number

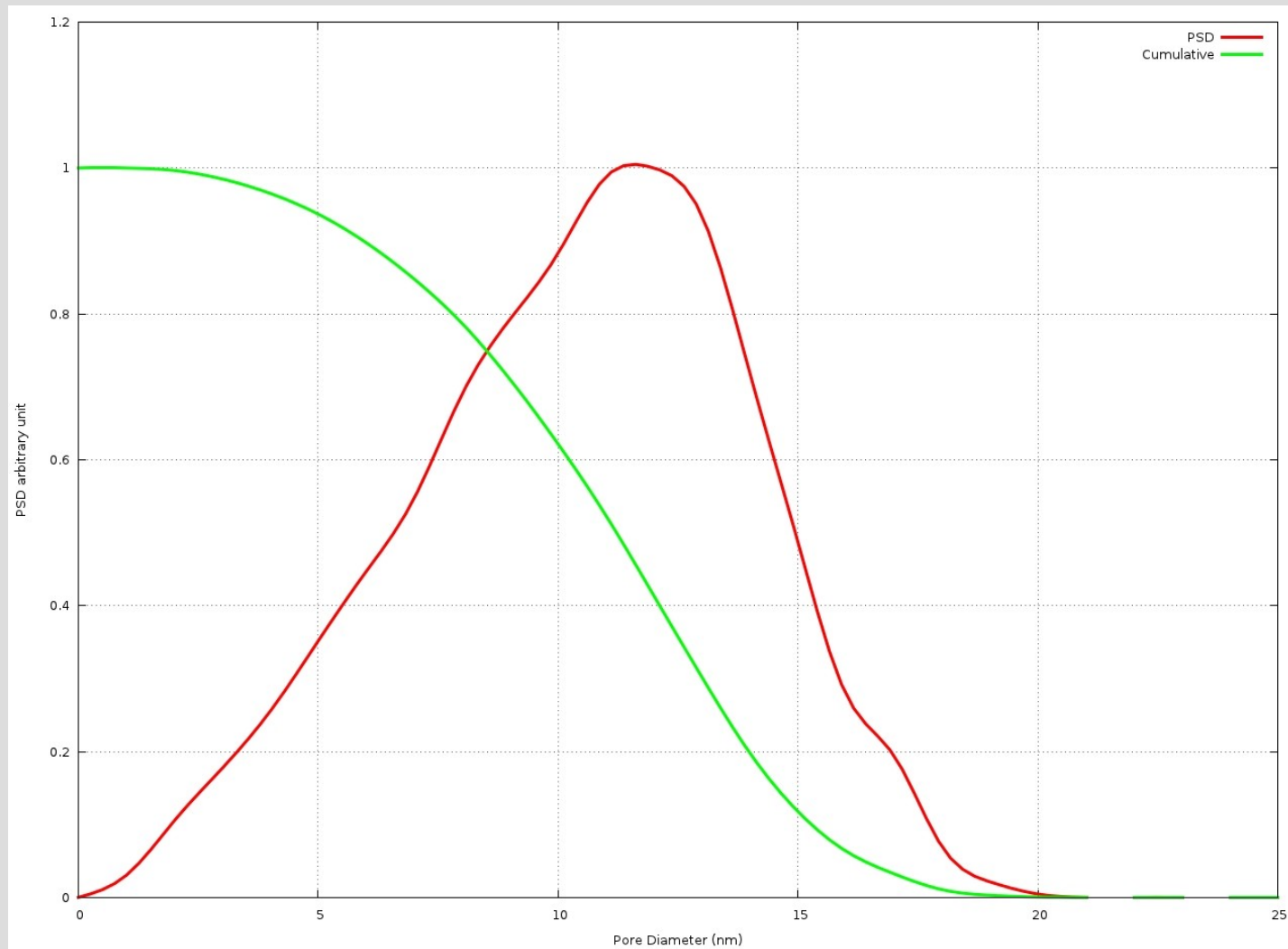


PSD calculation

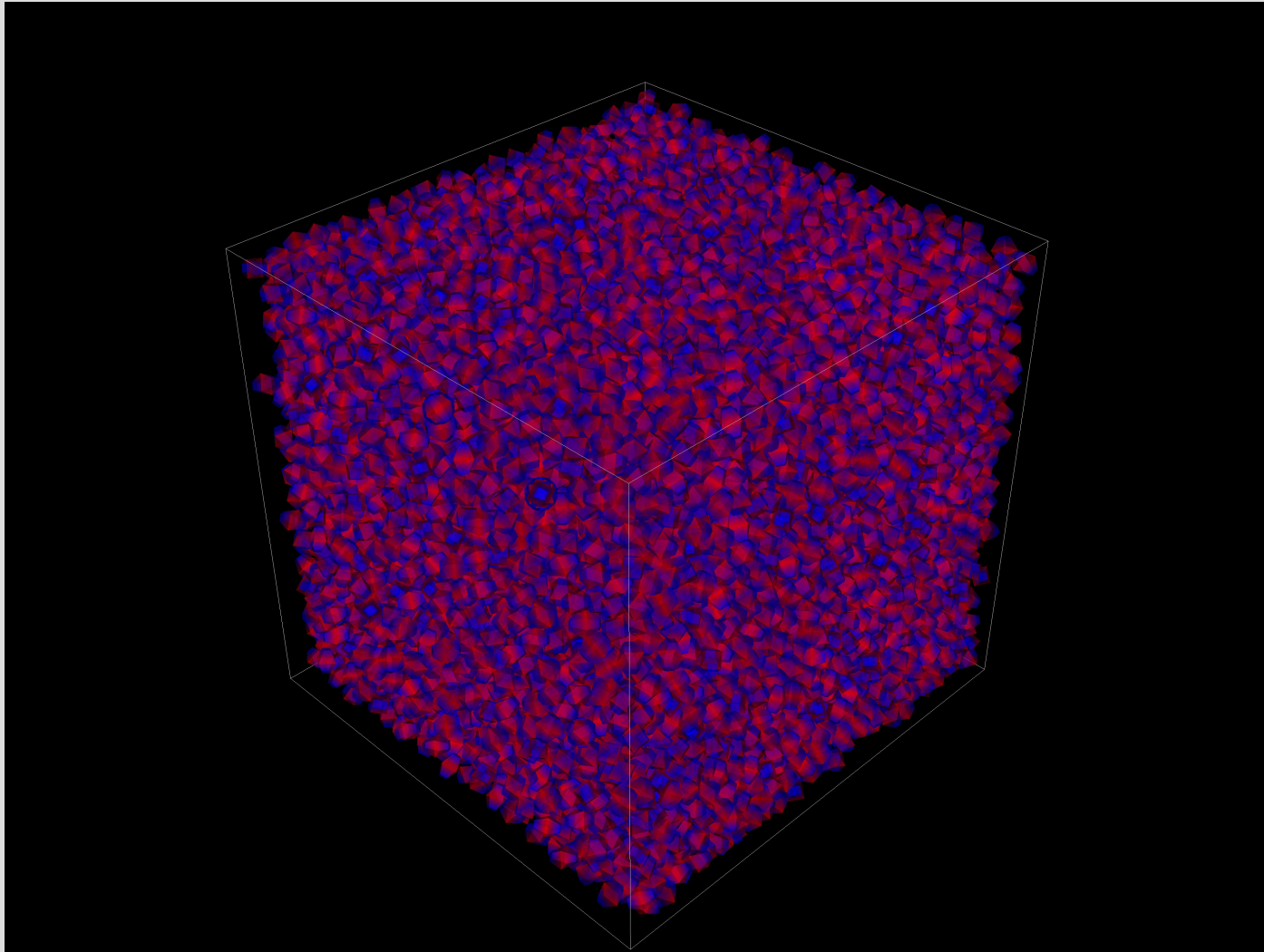


To solve the problem I used pyOpt is a Python-based package for formulating and solving nonlinear constrained optimization problems

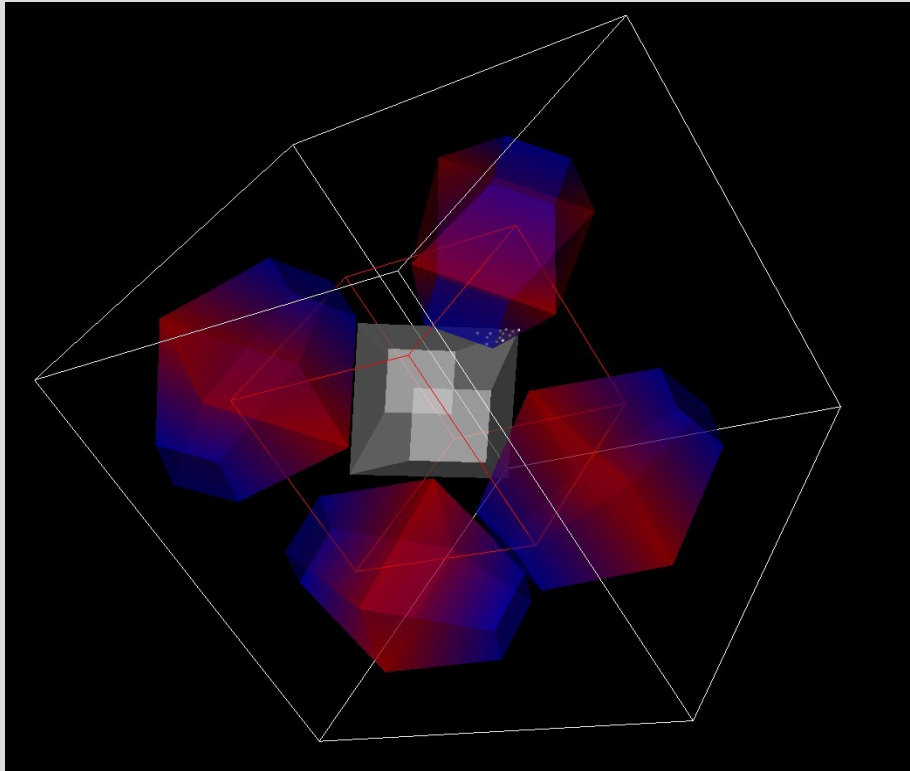
PSD



Random Nanoparticle packing

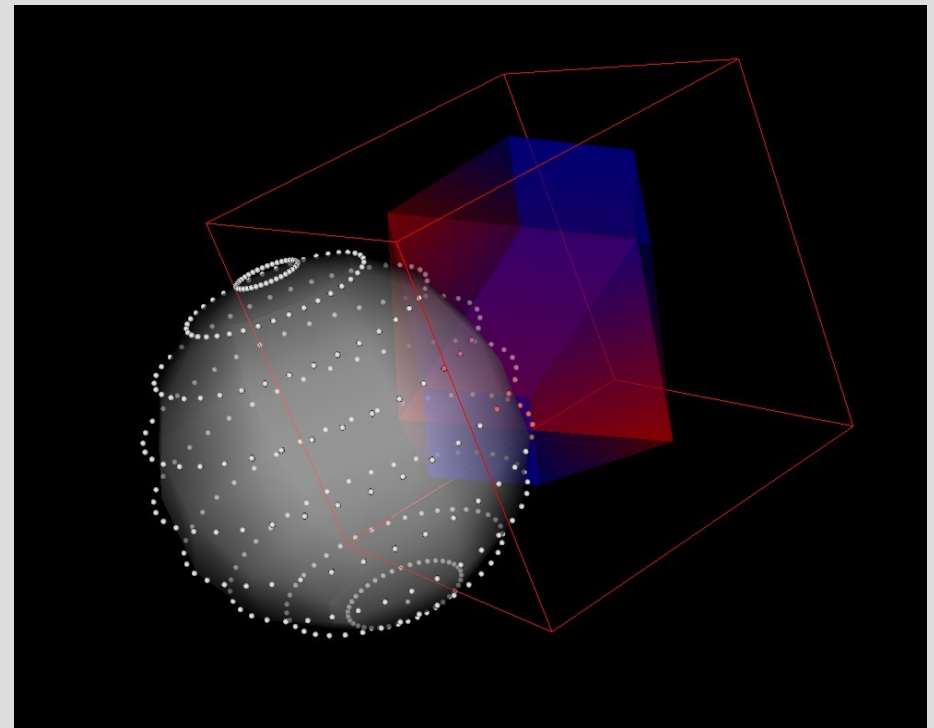


Nanoparticle packing properties

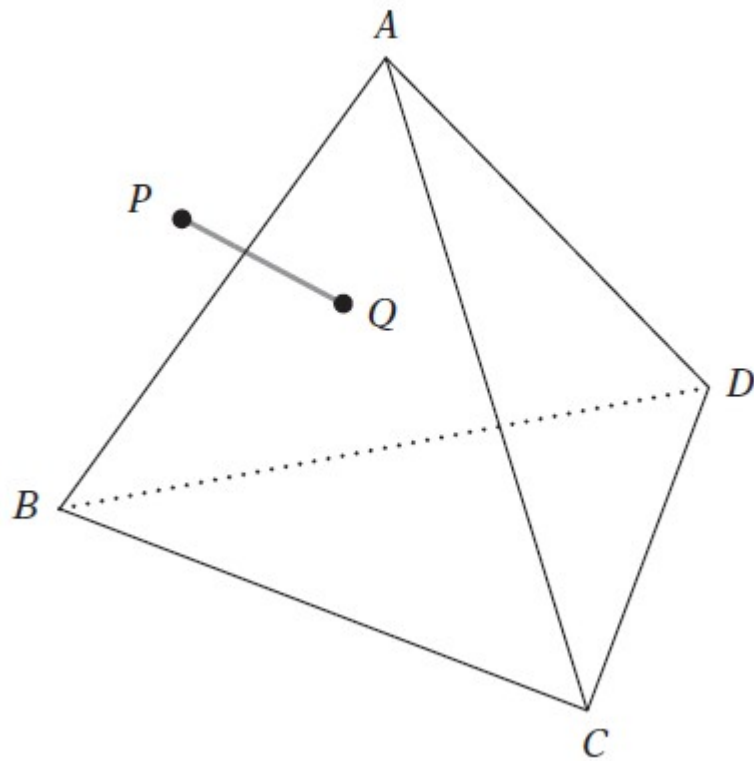


Coordination number
intersection of nanoparticles

Pore Size Distribution

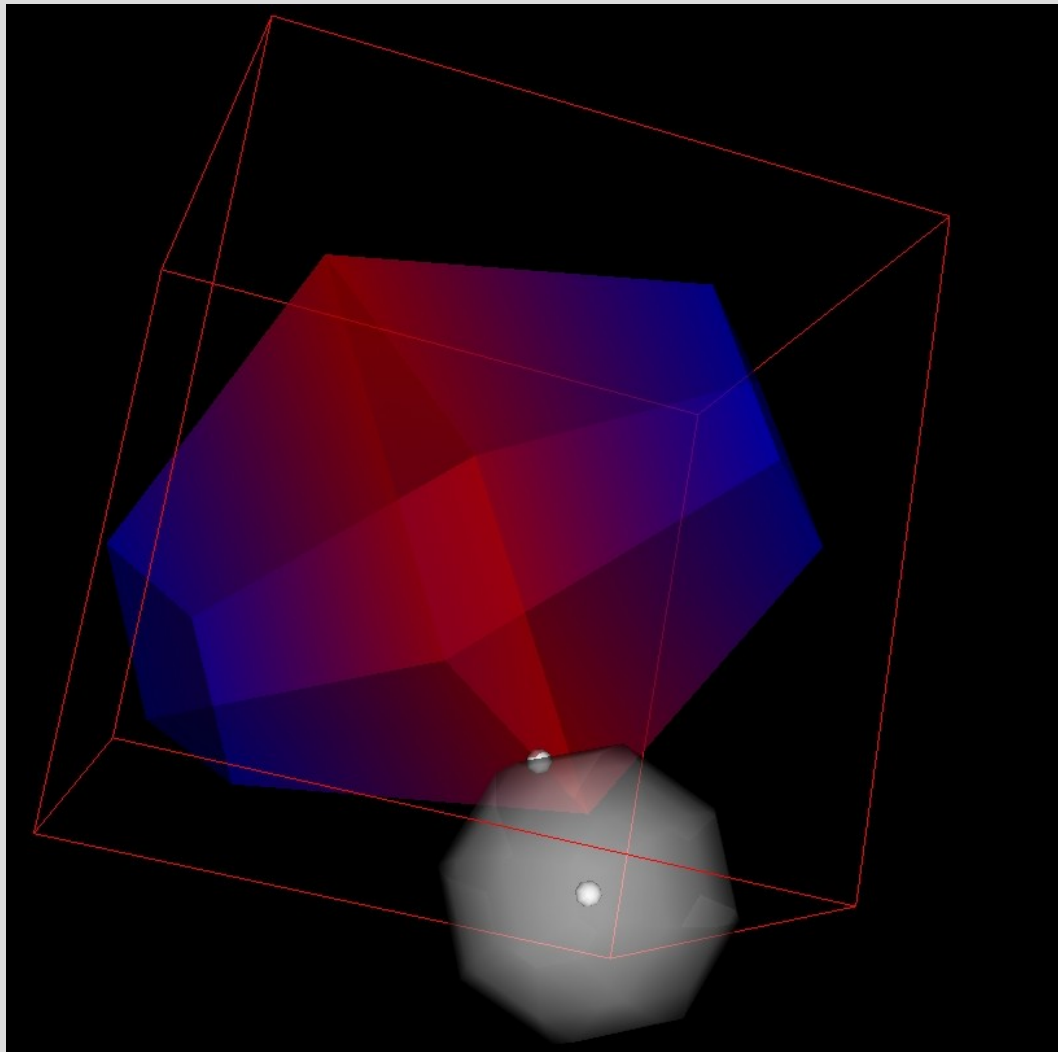


Real Time Collision Detection



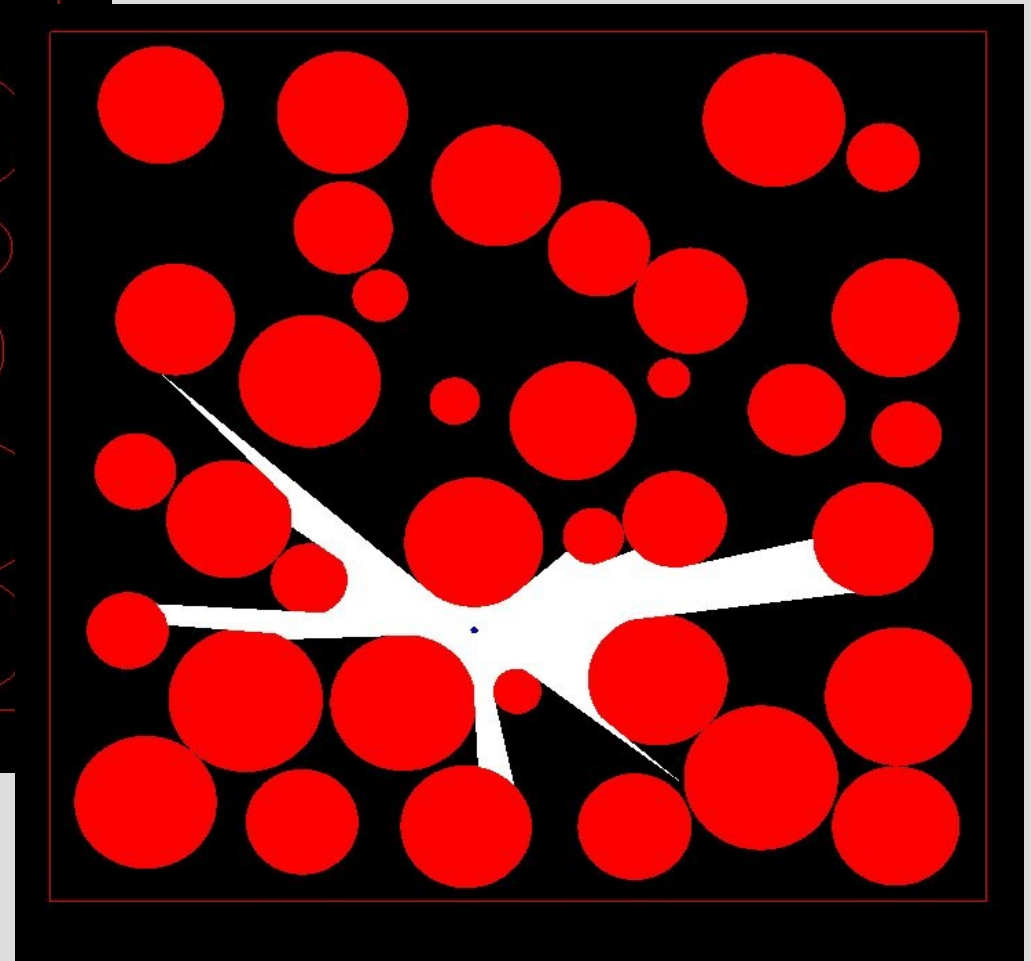
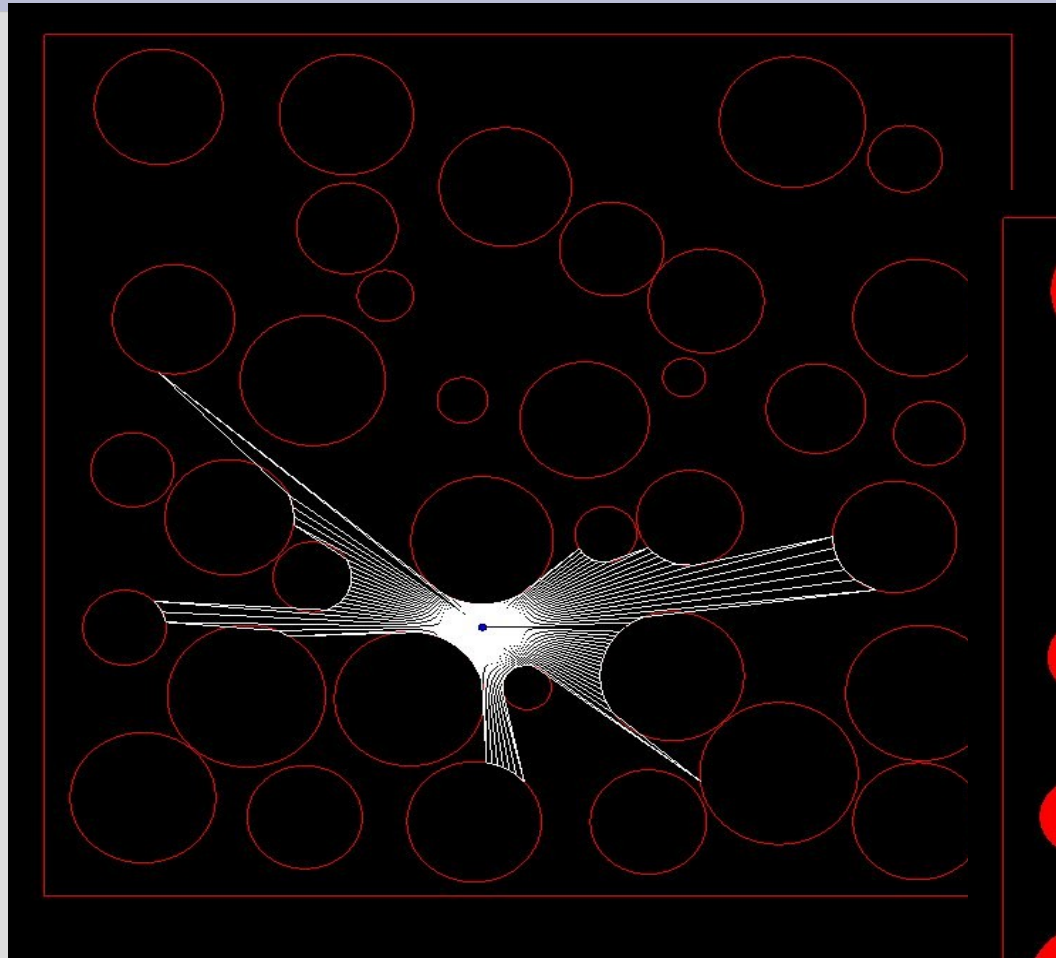
The point Q on the tetrahedron $ABCD$ closest to P .

Real Time Collision Detection

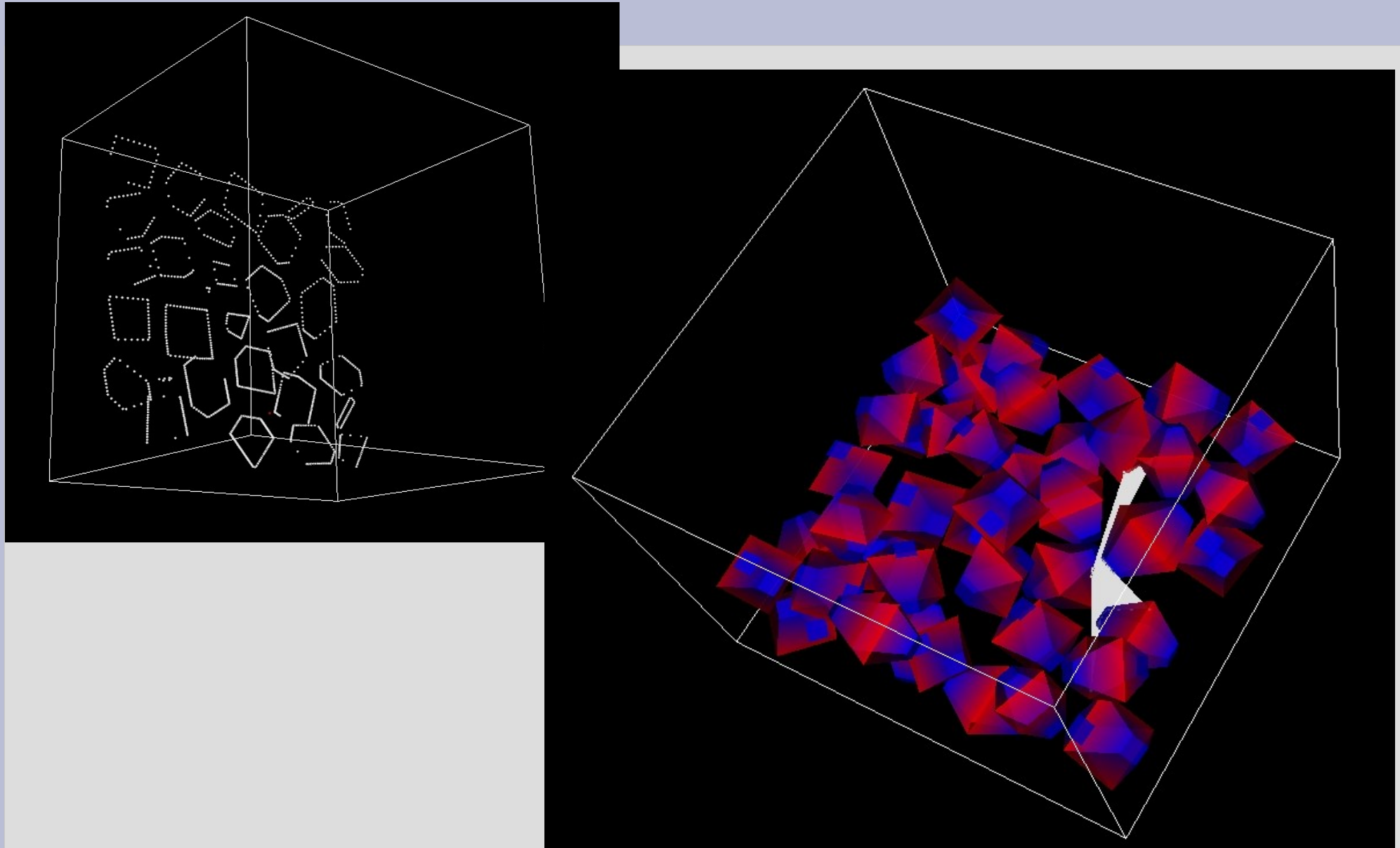


Computation time
reduced by more
than two orders of
magnitude

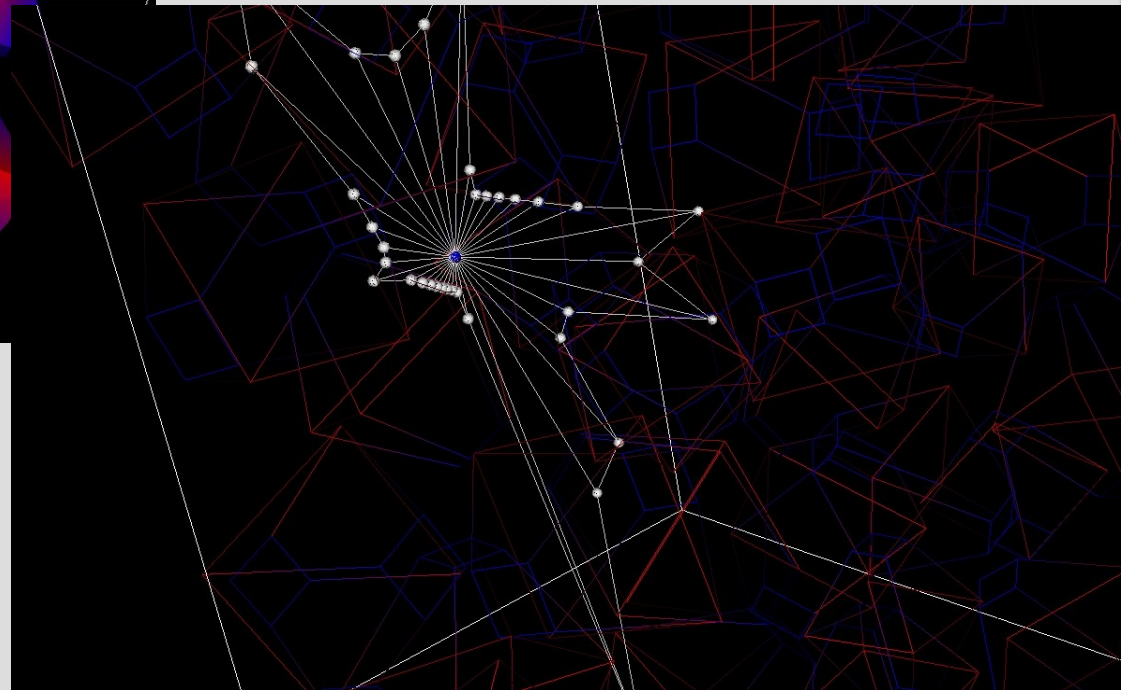
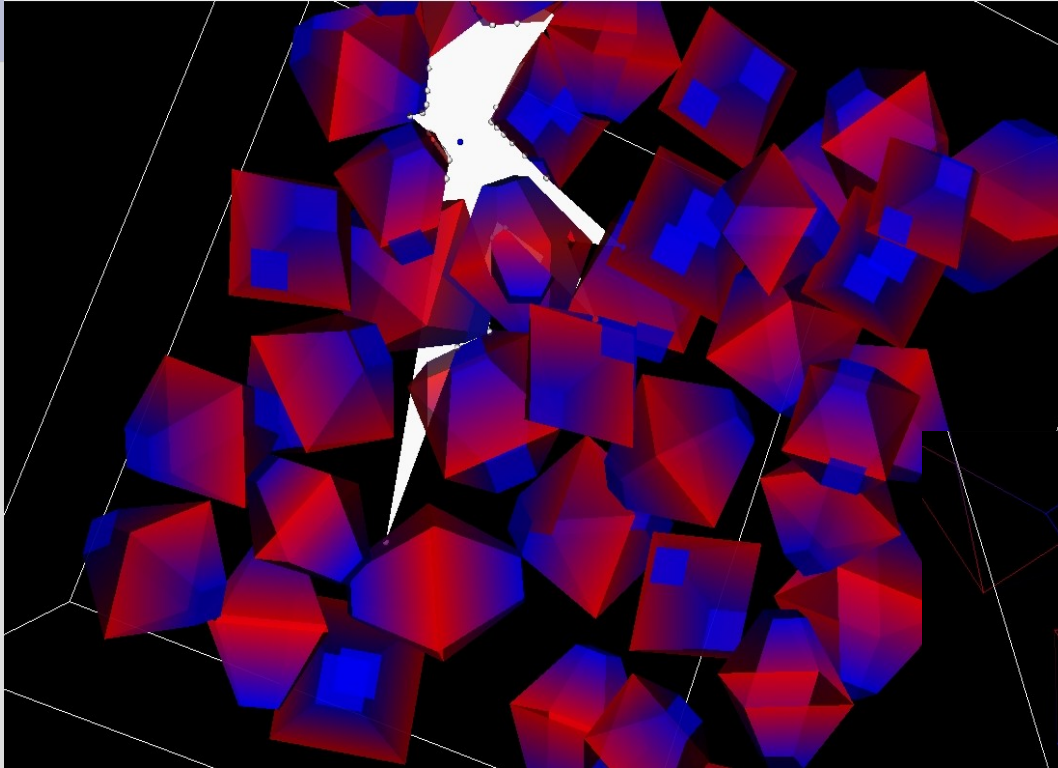
PSD using TR



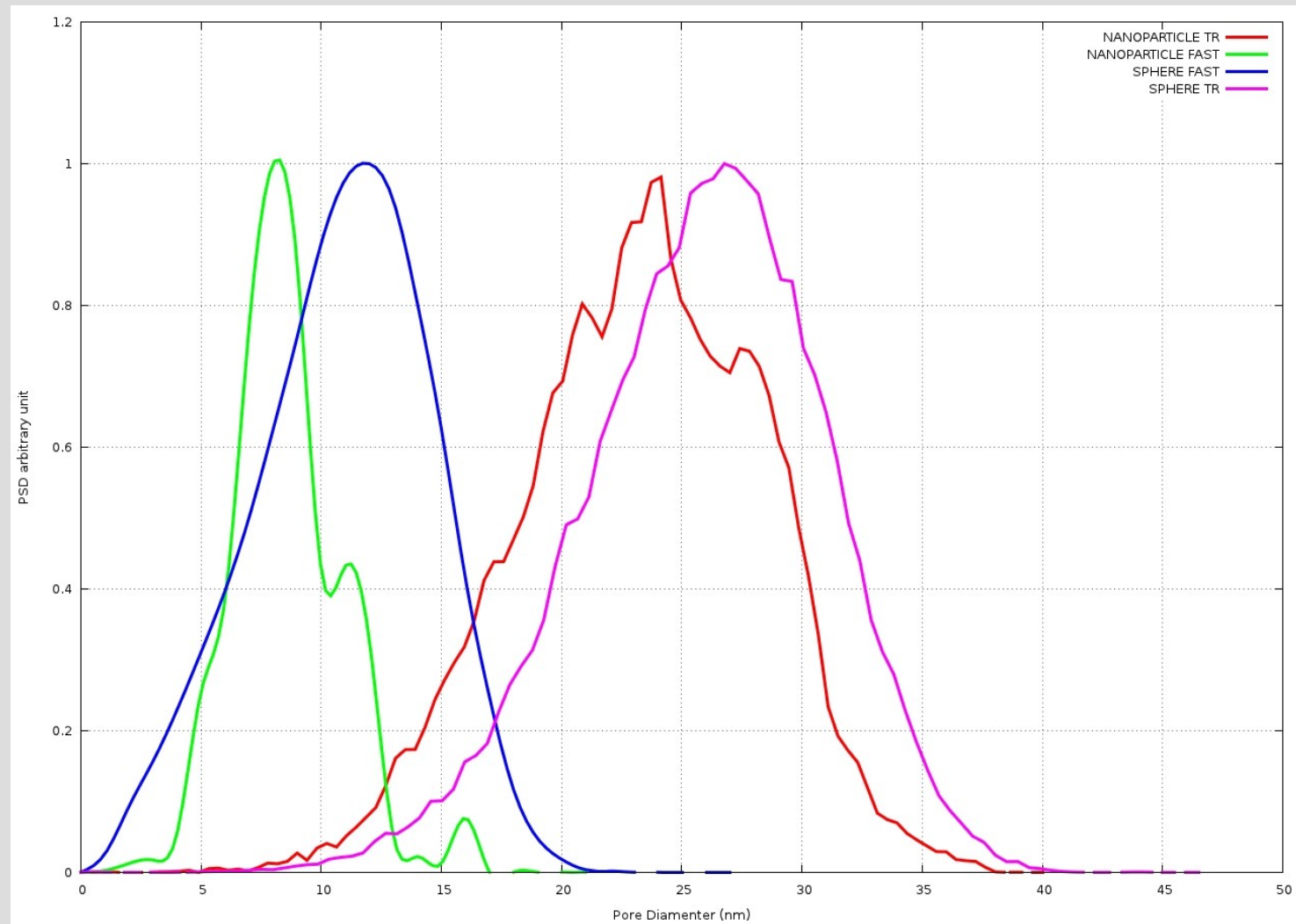
PSD using TR for nanoparticles



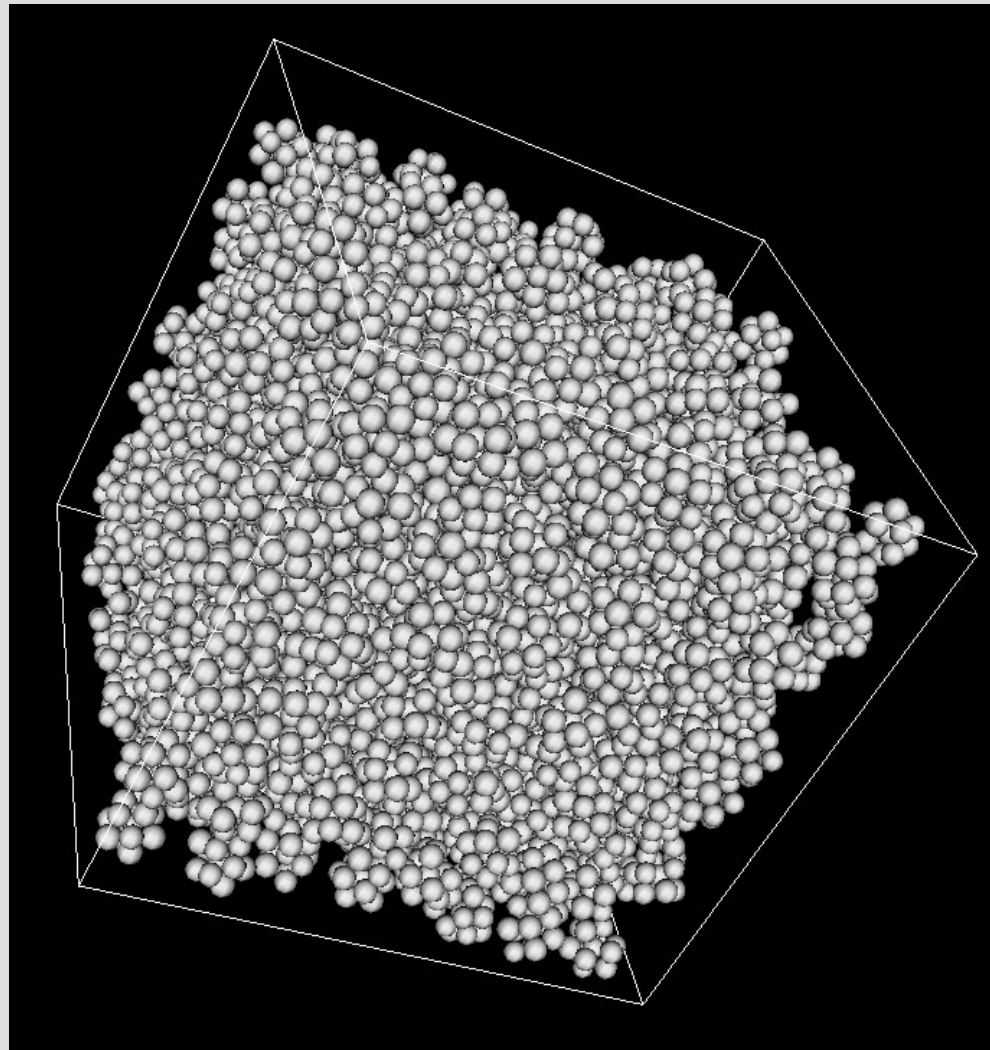
PSD using TR for nanoparticles



PSD



Different Packing Procedure



**Thanks for your
attention**