Computation of positron lifetime with ABINIT :

Two-Component Density-Functional Theory

M. Torrent, G. Jomard

CEA, DAM, DIF, F-91287 Arpajon cedex, France

Outline

Introduction,

Positron Annihilation Spectroscopy,

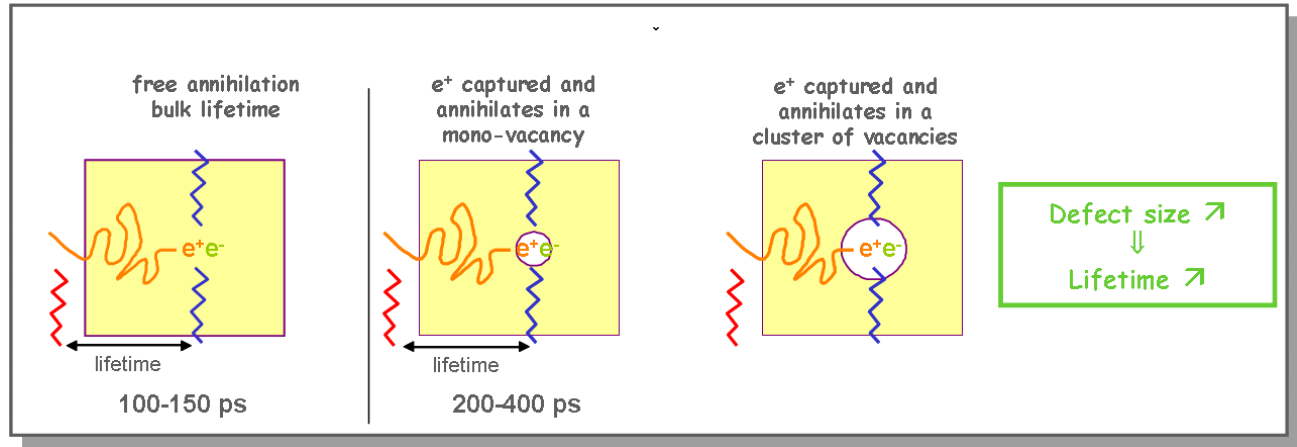*Two-component Density-Functional Theory*

Implementation in ABINIT code

Validation on bulk systems

Results on UO_2 point defects

How to use

Conclusions

Context

Positron Annihilation Spectroscopy (PAS) for defects detection

| free annihilation bulk lifetime | e⁺ captured and annihilates in a mono-vacancy | e⁺ captured and annihilates in a cluster of vacancies |

Defect size ↗
⇓
Lifetime ↗

lifetime
100-150 ps

lifetime
200-400 ps

$$\tau_{measured} = \sum_i I_i \tau_i = \sum_j f_j \tau_j$$

- Vacancy-type defects act as efficient positron traps : positron lifetime increases

- The correct identification of defects with PAS requires the knowledge of accurate positron lifetimes for the various kinds of defects. That can be provided by numerical calculations.
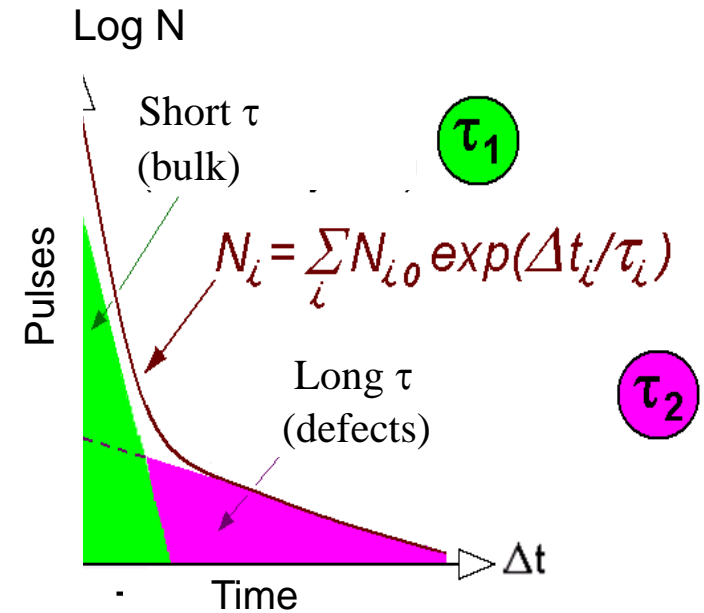
PAS : principes

PAS is sensistive to electronique density

- ✓ Presence of vacancy-type defects
- ✓ Size of defects
- ✓ Defect concentration

An average lifetime is measured

$$\tau_{moyen} = \sum_i I_i \tau_i$$

Individual contributions have to be separated

Log N

Short τ (bulk)

$$N_i = \sum_i N_{i0} \exp(\Delta t_i / \tau_i)$$

τ_1

Long τ (defects)

τ_2

Δt

Time

Theory/experiment coupling necessary to analyse measurments

Two-component Density-Functional Theory

Goal: determine properties of an electrons/positron system

Method: solve the many-body problem : e⁻, e⁺ and nuclei

$$E[n_+, n_-] = F[n_+] + F[n_-] - \int d\vec{r} \int d\vec{r}\,' \frac{n_-(\vec{r})n_+(\vec{r})}{|\vec{r}-\vec{r}\,'|} + E_c^{e-p}[n_+, n_-]$$

One-component DFT functionals
(electrons or positron)

Electrons/positron
Coulomb interaction

Electrons/positron
correlation energy

$$\lambda = \frac{1}{\tau} = \pi r_0^2 c \int d\vec{r}\, n_+(\vec{r}) n_-(\vec{r}) g\big(n_+(\vec{r}), n_-(\vec{r})\big)$$

TC-DFT : generalisation of the DFT (80's)
Chakraborty & Siegel 1983, Lundqvist & March 1983, Boroński & Nieminen 1986

TC-DFT in ABINIT

2005-2006 :   TC-DFT limited to NC- pseudopotentials
              No self-consistency, no forces, …
              Specific pseudopotentials required

2010 :   TC-DFT implementation within PAW
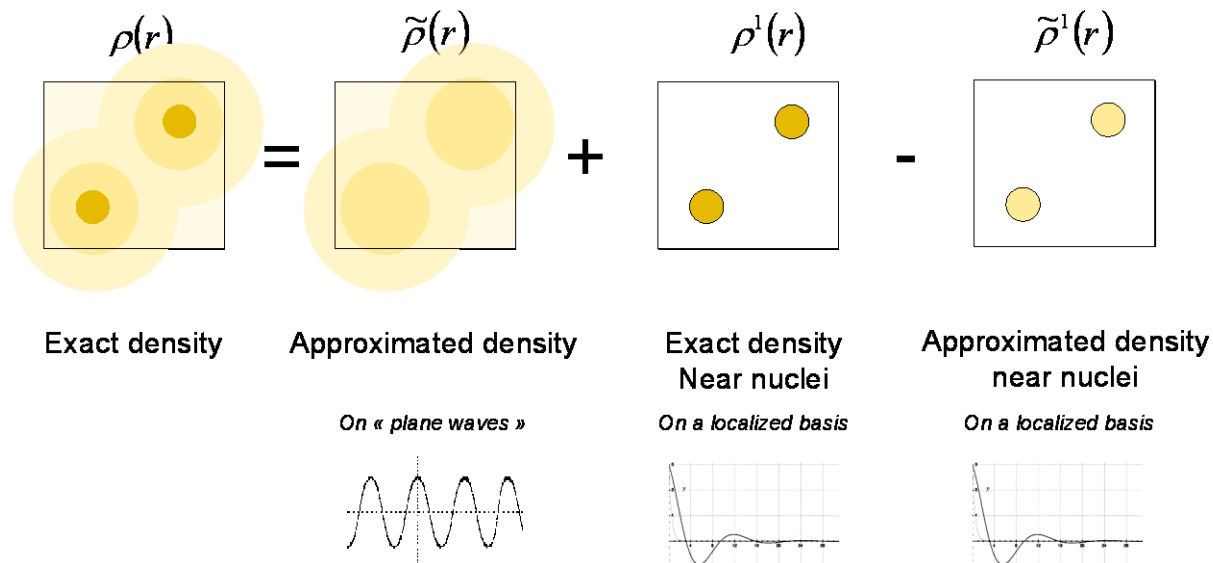   Fully self-consistent, forces, …

TC-DFT: implementation

- The TC-DFT has been implemented within the PAW (Projector Augmented Wave) method in the ABINIT code.
- Electrons and positron densities are optimized self-consistently and positron-induced forces are accurately calculated.
- Relaxed geometries of defects that trapped positron are correctly determined.

Electronic density

$$\rho(r) \qquad \tilde{\rho}(r) \qquad \rho^1(r) \qquad \tilde{\rho}^1(r)$$

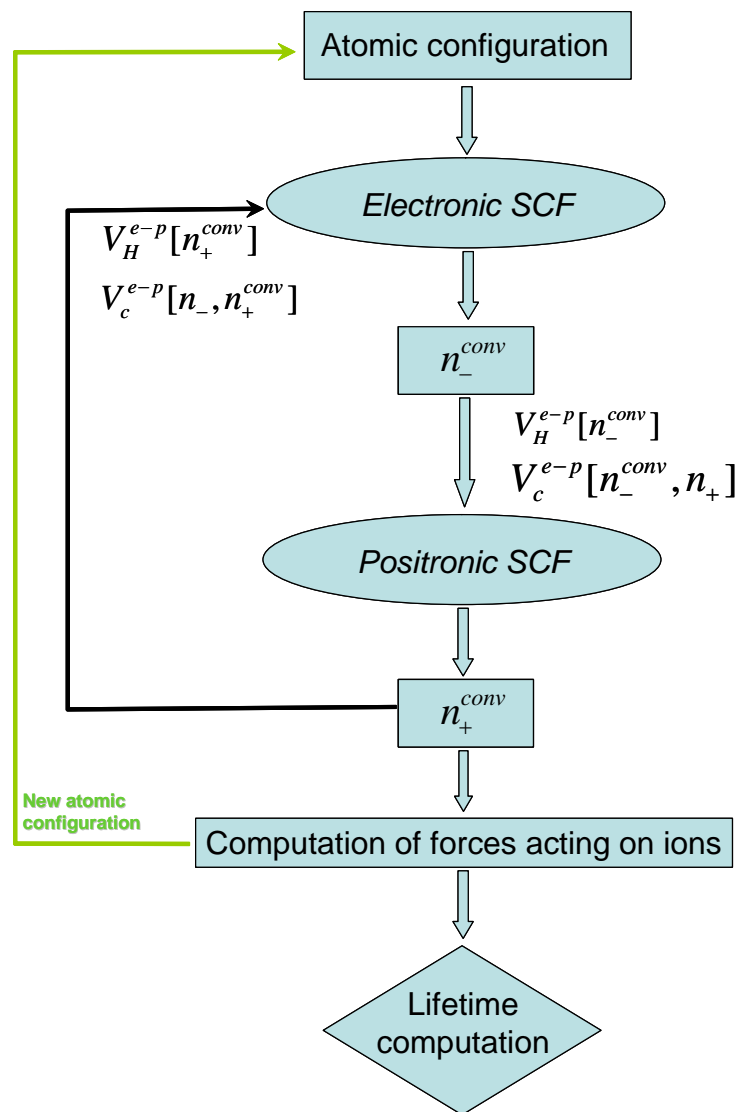| Exact density	Approximated density	Exact density Near nuclei	Approximated density near nuclei
On « plane waves »		On a localized basis	On a localized basis

TC-DFT: implementation

energie atomique · energies alternatives

- ✓ Fully self-consistent Two-Component DFT

- ✓ PAW applied for both electrons and positron

- ✓ Various electron-positron correlation functionals available

- ✓ Positron induced forces accurately computed

- ✓ Optimized density prediction and SCF mixing schemes

Atomic configuration

Electronic SCF

$V_H^{e-p}[n_+^{conv}]$

$V_c^{e-p}[n_-, n_+^{conv}]$

n_-^{conv}

$V_H^{e-p}[n_-^{conv}]$

$V_c^{e-p}[n_-^{conv}, n_+]$

Positronic SCF

n_+^{conv}

New atomic configuration

Computation of forces acting on ions

Lifetime computation

$$E[n_T^+, n_T^-] = F[n_T^-] + \sum_{n,k} f_{n,k}^+ \left\langle \Psi_{n,k} \left| T \right| \Psi_{n,k} \right\rangle - \int d\vec{r} \int d\vec{r}' \frac{n_T^+(\vec{r}) n_T^-(\vec{r})}{\left| \vec{r} - \vec{r}' \right|} + E_c^{e-p} \left[n_T^+, n_T^- \right]$$

$$E_{PAW} = \left(\tilde{E}^- + E^{1-} - \tilde{E}^{1-} \right) + \left(\tilde{E}^+ + E^{1+} - \tilde{E}^{1+} \right)$$

$$
\begin{cases}
\tilde{E}^+ = \left\langle \tilde{\Psi}^+ \left| -\frac{\Delta}{2} \right| \tilde{\Psi}^+ \right\rangle \quad - \quad \int V_H \left(\tilde{n}^- + \hat{n}^- + \tilde{n}_{Zc} \right) \left(\tilde{n}^+ + \hat{n}^+ \right) d\vec{r} \quad + \quad E_c^{e-p} \left(\tilde{n}^- + \tilde{n}_c, \tilde{n}^+ \right) \\[2ex]
E^{1+} = \sum_{ij} \rho_{ij}^+ \left\langle \Phi_i \left| -\frac{\Delta}{2} \right| \Phi_i \right\rangle \quad - \quad \int V_H \left(n^{1-} + n_{Zc} \right) \left(n^{1+} \right) d\vec{r} \quad + \quad E_c^{e-p} \left(n^{1-} + n_c, n^{1+} \right) \\[2ex]
\tilde{E}^{1+} = \sum_{ij} \rho_{ij}^+ \left\langle \tilde{\Phi}_i \left| -\frac{\Delta}{2} \right| \tilde{\Phi}_i \right\rangle \quad - \quad \int V_H \left(\tilde{n}^{1-} + \hat{n}^{1-} + n_{Zc} \right) \left(\tilde{n}^{1+} + \hat{n}^{1+} \right) d\vec{r} \quad + \quad E_c^{e-p} \left(\tilde{n}^{1-} + \hat{n}^{1-} + \tilde{n}_c, n^{1+} + \hat{n}^{1+} \right)
\end{cases}
$$

TC-DFT : focus on implementation

```
!{\src2tex{textfont=tt}}
!!****f* ABINIT/setup_positron
!! NAME
!! setup_positron
!!
!! FUNCTION
!! Do various initializations for the p
!!
!! NOTE
!!
!! COPYRIGHT
!! Copyright (C) 1998-2011 ABINIT group
!! This file is distributed under the t
!! GNU General Public License, see ~abi
!! or http://www.gnu.org/copyleft/gpl.t
!! For the initials of contributors, se
!!
!! INPUTS
!!  atindx(natom)=index table for atoms
!!  atindx1(natom)=index table for atoms, inverse of atindx
!!  dtfil <type(datafiles_type)>=variables related to files
!!  dtset <type(dataset_type)>=all input variables for this dataset
!!  ecore=core psp energy (part of total energy) (hartree)
!!  etotal=current value of total energy
!!  forces_needed=if >0 forces are needed
!!  fred(3,natom)=forces in reduced coordina
!!  gprimd(3,3)=dimensional primitive transl
!!  gmet(3,3)=reciprocal space metric
!!  grewtn(3,natom)=d(Ewald)/d(xred) (hartree
!!  gsqcut=cutoff value on G**2 for sphere i
!!  hdr <type(hdr_type)>=the header of wf, de
!!  ifirst_gs= 0 if we are in a single ground
!!     or in the first ground-state calculati
!!  indsym(4,nsym,natom)=index showing trans
!!                     under symmetry opera
!!  istep=index of the number of steps in the
!!  istep_mix=index of the number of steps f
!!  kg(3,mpw*mkmem)=reduced (integer) coordin
!!  kxc(nfft,nkxc)=exchange-correlation kerne
!!  maxfor=maximum absolute value of fcart (
!!  mgfft=maximum size of 1D FFTs
!!  mpi_enreg=informations about MPI paralle
!!  n3xccc=dimension of the xccc3d array (0
!!  nattyp(ntypat)= # atoms of each type.
!!  nfft=(effective) number of FFT grid poin
!!  ngfft(18)=contain all needed information
!!  nhat(nfftf,nspden*usepaw)= -PAW only- co
!!  nkxc=second dimension of the array kxc, s
!!  npwarr(nkpt)=number of planewaves in bas
!!  nvresid(nfftf,nspden)=array for the residual of the density/potential
!!  optres=0 if the potential residual has to be used for forces corrections
```

```
!{\src2tex{textfont=tt}}
!!****m* ABINIT/m_electronpositron
!! NAME
!!  m_electronpositron
!!
!! FUNCTION
!!  This module provides the definition of the electronpositron_type used
!!  used to store data for the electron-positron two-component DFT
!!  as methods to operate on it.
!!
!! COPYRIGHT
!! Copyright (C) 2008-2011 ABINIT group (MT)
!! This file is distributed under the terms of the
!! GNU General Public License, see ~abinit/COPYING
!! or http://www.gnu.org/copyleft/gpl.txt .
```

This routine manages TC-DFT

g

g

quantities are stored in
electronpositron datastructure

```
type, public :: electronpositron_type

! Integer scalars
  integer :: calctype        ! type of electron-positron calculation:
                             !   0: no calculation
                             !   1: positron in the electrons potential
                             !   2: electrons in the positron potential
  integer :: dimcg           ! Dimension of cg array dimcg=dtset%mpw*dtset%nspinor*dtset%mband*dtset%mkmem*
  integer :: dimcprj         ! Dimension of cprj array dimcprj=dtset%nspinor*dtset%mband*dtset%mkmem*dtset%
  integer :: dimeigen        ! Dimension of eigen array dimeigen=dtset%mband*dtset%nkpt*dtset%nsppol
  integer :: dimocc          ! Dimension of occ array dimocc=dtset%mband*dtset%nkpt*dtset%nsppol
  integer :: has_pawrhoij_ep ! flag for pawrhoij_ep (0: not allocated, 1: allocated, 2: computed)
  integer :: has_pos_ham     ! flag: 1 if current Hamiltonian is the positronic hamiltonian, 0 is it is the
  integer :: ixcpositron     ! XC type for electron-positron correlation
  integer :: istep           ! Current index of TC-DFT SCF step
  integer :: istep_scf       ! Current index of DFT SCF step  in current electron/positron minimization
  integer :: lmmax           ! Max. number of (l,m) moments over all types of atom
  integer :: natom           ! Number of atoms
  integer :: nfft            ! Number of points in FFT grid
  integer :: nspden          ! Number of spin density components
  integer :: nstep           ! Max. number of steps for the TC-DFT SCF cycle
```

TC-DFT+PAW : difficulties
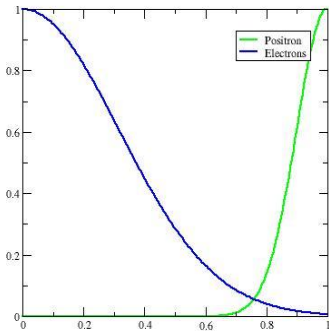
Several numerical issues (completeness of the basis):

✓ Is the local basis (atomic orbitals) able to reproduce a positronic wave-function ?

$$\widetilde{\Psi}^+(\vec{r}) - \sum_i \left\langle \widetilde{\Psi}^+ \middle| \widetilde{p}_i \right\rangle \cdot \Phi_i(\vec{r}) \stackrel{?}{=} 0$$

✓ Are the different basis able to reproduce a zero-density ?

✓ Are gradients accurately computed for the small positronic density ?

TC-DFT: validation

Results on bulk systems

Species	Lifetime (ps)		
	FPLAPW[1]	PAW	Exp.[2]
Li	298	297	291
Al	166	166	163
Fe	100	100	106
Mo	104	106	103
Ag	124	125	133

[1] Takenaka et Singh, PRB **77** (2008) 155132
[2] K.O. Jensen, J. Phys.: Condens. Matter **1** (1989) 10595

Results on point defects: silicon mono-vacancy

$$\tau_{\text{exp.}} = 282ps$$

Vacancy without relaxation $\qquad\qquad \tau = 250ps \quad \Delta d/d_0 = 0\%$

Relaxed vacancy (effect of e⁻) $\qquad \tau = 213ps \quad \Delta d/d_0 = -13\%$

Relaxed vacancy (effect of e⁻ & p⁺) $\quad \tau = 276ps \quad \Delta d/d_0 = +7\%$

Previous calculations
Makhov, Lewis, PRB **71** (2005):
Unrelaxed vacancy : $\quad \tau = 250ps$
Full relaxation : $\qquad \tau = 278ps$

Influence of semi-core states and self-consistency

Mono-vacancy positron lifetime for Ag and Cu

	τ without self-consistency		τ with self-consistency		Exp.
	No semi-core states	Semi-core states	No semi-core states	Semi-core states	
Ag	259	192	243	220	208
Cu	218	164	204	182	180

107 atoms in simulation cell

UO$_2$: numerical details

✓ Computations are done at the experimental volume V$_0$=40,92 Å^3, at T=0K

✓ Supercell contains 96 atomes and more (needed for defect simulation)

✓ Exchange-correlation (electrons) is done in the GGA formalism.

✓ LDA and LDA+U computations have been performed
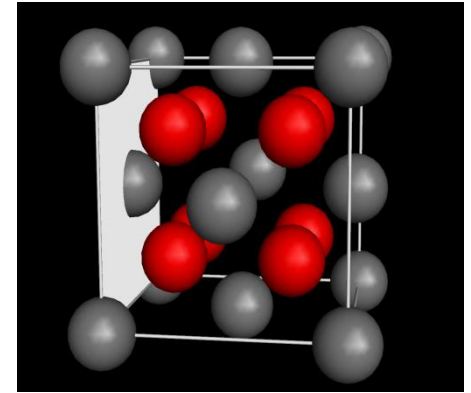
Computed bulk positron lifetime: $\tau = 166\,ps$
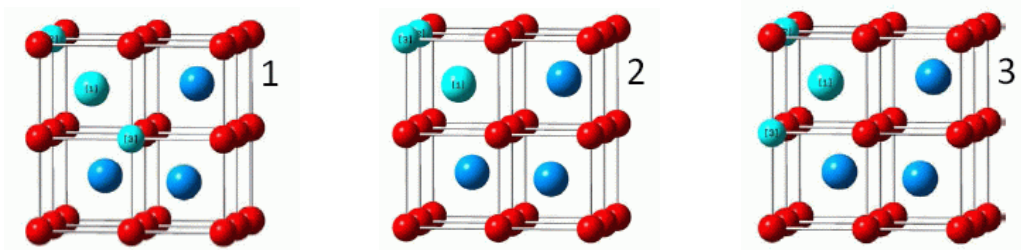
In agreement with experimental data : 170 ps.

Results on UO$_2$ point defects

Various kinds of defects have been investigated:

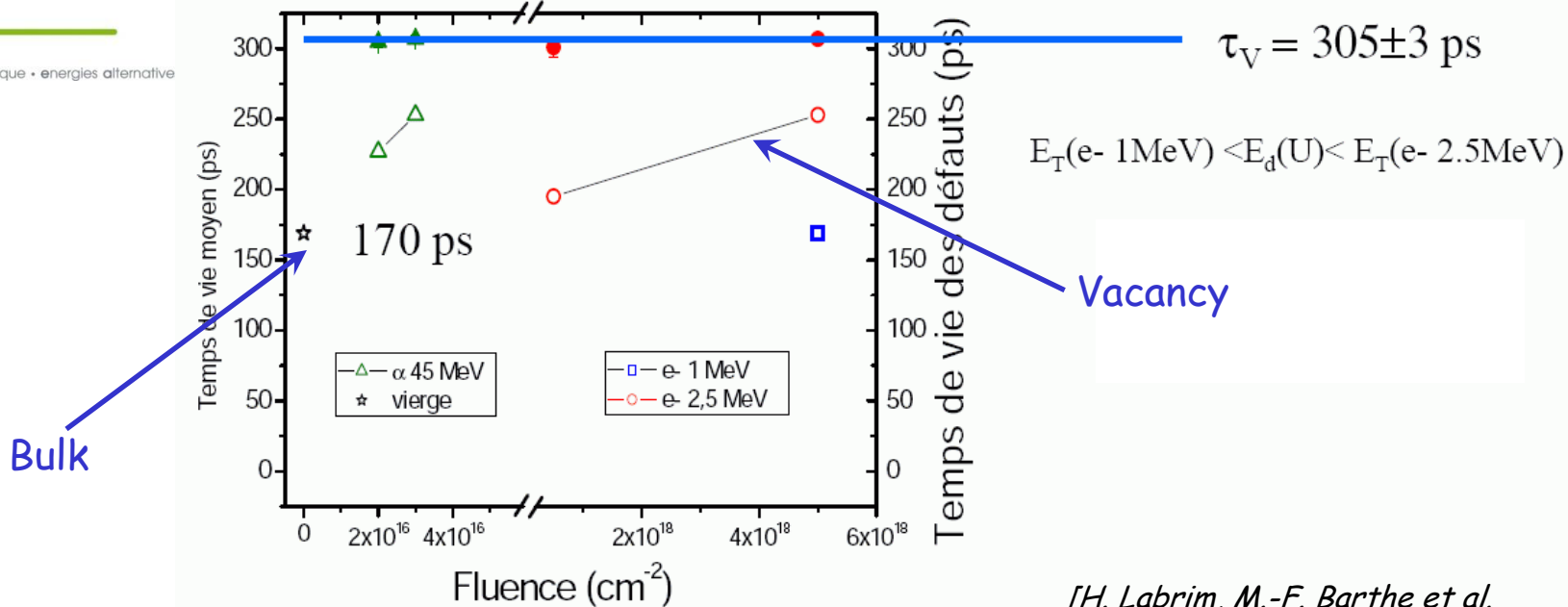*Oxygen vacancy, Oxygen interstitial, Uranium vacancy, Three Schottky defects*

Point defect type	Lifetime (ps)
Oxygen mono-vacancy	177
Oxygen interstitial	169
Uranium mono-vacancy	270
Schottky defects	310

Calculated positron lifetimes for various point defects in UO$_2$.

Results on UO$_2$ point defects

$\tau_V = 305 \pm 3$ ps

$E_T(e- 1MeV) < E_d(U) < E_T(e- 2.5MeV)$

Vacancy

Bulk

170 ps

[H. Labrim, M.-F. Barthe et al.
PhD thesis, Université d'Orléans]

Calculations validate the assumption that
irradiated UO$_2$ samples contain Shottky defects

TC-DFT in ABINIT : HOWTO

positron

Mnemonics: POSITRON calculation
Characteristic:
Variable type: integer parameter
Default is 0

This input parameter can be positive or negative.
Negative values for **positron** are only relevant for PAW calculations.
Electron-positron correlation functional is defined by ixcpositron.
Other relevant input parameter: posocc (occupation number for the positron).

Positive values for **positron**:
For **positron=1 or 2**, will perform the calculation of positron lifetime (and annihilation rate).

- **positron=1**:
Starting from a previous electronic GS density (with **positron=0**), a positronic ground-state calculation is performed, considering that the electrons are not perturbed by the presence of the positron.
This is almost correct for a positron in a perfect bulk material. But this approximation fails when defects exist in the material (for instance: the positron might be trapped by a vacancy).
The electronic density will be automatically read from a _DEN file (with or without getden/irdden keyword).
At the end of the SCF cycle, the positron lifetime and annihilation rate are printed out.

Additional information for the use of pseudopotentials:

- PAW datasets: nothing to do; simply use usual electronic PAW datasets

- Norm-conserving pseudopotentials: One has to use specific pseudopotentials for the positron calculation. They must be of the FHI type (pspcod=6), and must contain at their end, the all-electrons core density generated with FHI98PP. They must have lmax=lloc=0 (check that this works for the electronic GS !! No ghost, etc ...). Otherwise, their are similar to an usual FHI pseudopotential.

- **positron=2**:
Starting from a previous positronic GS density (with **positron=1**), an electronic ground-state calculation is performed, keeping the positronic density constant.
The positronic density will be automatically read from a _DEN file (with or without getden/irdden keyword).
At the end of the SCF cycle, the positron lifetime and annihilation rate are printed out.

Additional information for the use of pseudopotentials:

- PAW datasets: nothing to do; simply use usual electronic PAW datasets

- Norm-conserving pseudopotentials: One has to use specific pseudopotentials for the electron calculation. They must be of the FHI type (pspcod=6), and must contain at their end, the all-electrons core density generated with FHI98PP.

TC-DFT in ABINIT : HOWTO

Advice for use:

There are two typical cases which have to be differently treated:

- **A positron in a perfect bulk system:**

  In that case, the positron is delocalized in the whole crystal. Its density is almost zero.

  Thus, the "zero density positron limit" has to be used. ixcpositron has to be choosen accordingly.

  In order to have the zero density positron limit it is adviced to follow these points:

  1- Put a small positronic charge (by setting a posocc to a small value) **OR** use a big supercell.

  2- Use only k=gamma wave vector for the positronic calculation.

  3- Use the manual procedure in 2 steps: first **positron**=0 and then **positron**=1 ; avoid the **positron=2** step and the automatic procedure (**positron**<0).

  In principle, the positron lifetime should converge with the value of posocc or the size of the supercell.

- **A positron trapped in a default (vacancy...):**

  In that case, the positron is localized in the default. Its density can be localized in the simulation cell (provided that the cell is sufficiently large) and influences the electronic density.

  So, it is adviced to use the automatic procedure (**positron**<0) or the manual procedure with several **positron**=0,1,2,1,... steps.

  K-points can be used as in usual electronic calculations.

  Also note that it is possible to use forces and stresses to perform structural minimization.

References:

[1] J. Arponen and E. Pajanne, Ann. Phys. (N.Y.) 121, 343 (1979).

[2] Boronski and R.M. Nieminen, Phys. Rev. B 34, 3820 (1986).

[3] P.A. Sterne and J.H. Kaiser, Phys. Rev. B 43, 13892 (1991).

[4] M.J. Puska, A.P. Seitsonen and R.M. Nieminen, Phys. Rev. B 52, 10947 (1994).

[5] B. Barbiellini, M.J. Puska, T. Torsti and R.M.Nieminen, Phys. Rev. B 51, 7341 (1994)

posnstep

Mnemonics: POSitron calculation: max. Number of STEPs for the two-component DFT

Characteristic:

Variable type: integer parameter

Default is 50

Relevant only when positron<0.

Sets the maximum number of electronic/positronic iterations that, when reached, will cause the two-component DFT SCF cycle to stop.

The code will first compute the electronic ground-state, then the positronic ground state in the electronic density, then the electronic ground-state in the positronic density,

...

...until diff_Etotal<postoldfe or diff_Forces<postoldff or the number of electronic/positronic steps is **posnstep**.

---

postoldfe

Mnemonics: POSitron calculation: TOLerance on the DiFference of total Energy

Characteristic: ENERGY

Variable type: real parameter

Default is 1.e-6 if postoldff=0, otherwise 0

Relevant only when positron<0.

Sets a tolerance for absolute difference of total energy (of *ions+electrons+positron* system) that, when reached, will cause the SCF cycle to stop before the number of SCF steps is nstep or the number of electronic/positronic steps is posnstep.

Can be specified in Ha (the default), Ry, eV or Kelvin, since **toldfe** has the 'ENERGY' characteristics.

Only one and only one of postoldfe or postoldff can be set.

---

postoldff

Mnemonics: POSitron calculation: TOLerance on the DiFference of Forces

Characteristic:

Variable type: real parameter

Default is 0

Relevant only when positron<0.

Sets a tolerance for absolute difference of maximum force acting on ions (due to *ions+electrons+positron* system) that, when reached, will cause the SCF cycle to stop before the number of SCF steps is nstep or the number of electronic/positronic steps is posnstep.

Only one and only one of postoldfe or postoldff can be set.

TC-DFT in ABINIT : HOWTO

ixcpositron
Mnemonics: Integer for the eXchange-Correlation applied to the electron-POSITRON interaction
Characteristic:
Variable type: integer parameter
Default is 0

Relevant only when positron/=0.
Define the type of electron-positron correlation that is used in case of a electron-positron two-component DFT calculation.
Define also the analytical formula of the enhancement factor used to compute the electron-positron annhilation rate:

Electron-positron correlation functional:

ixcpositron=1: LDA zero positron density limit parametrized by Arponen & Pajanne and provided by Boronski & Nieminen [1,2]
ixcpositron=11: LDA zero positron density limit parametrized by Arponen & Pajanne and fitted by Sterne & Kaiser [1,3]
ixcpositron=2: LDA electron-positron correlation provided by Puska, Seitsonen, and Nieminen [1,4]
ixcpositron=3: GGA zero positron density limit parametrized by Arponen & Pajanne and provided by Boronski & Nieminen [1,2,5]
ixcpositron=31: GGA zero positron density limit parametrized by Arponen & Pajanne and fitted by Sterne & Kaiser [1,3,5]
Annihilation rate enhancement factor:

ixcpositron=1: Boronski and Nieminen full modelisation and RPA limit [1]
ixcpositron=11: Sterne and Kaiser [2]
ixcpositron=2: Puska, Seitsonen and Nieminen [3]
ixcpositron=3: Boronski and Nieminen full modelisation and RPA limit [1], with GGA corrections
ixcpositron=31: Sterne and Kaiser [2], with GGA corrections
References:

[1] J. Arponen and E. Pajanne. Ann. Phys. (N.Y.) 121, 343 (1979).

posocc
Mnemonics: POSitron calculation: OCCupation number for the positron
Characteristic:
Variable type: real parameter
Default is 1.

Relevant only when positron/=0.
Sets the occupation number for the positron. Has to be <=1.
Changing **posocc** is only usefull for bulk calculation when one wants to perform lifetime computations using a small simulation cell (can avoid the use of a supercell). It simulates the dispersion of the positron in the whole crystal.

Conclusion

- This is an implementation of a fully self-consistent TC-DFT code with all-electron accuracy.

- It is easy to use : no specific PAW atomic data, few additional keywords, …

- Completeness of PAW basis for positronic wave-function has to be checked

- This work contributes to the interpretation of PAS experiments on UO2.
  Computational results show that *Shottky* defects are present in irradiated UO_2.