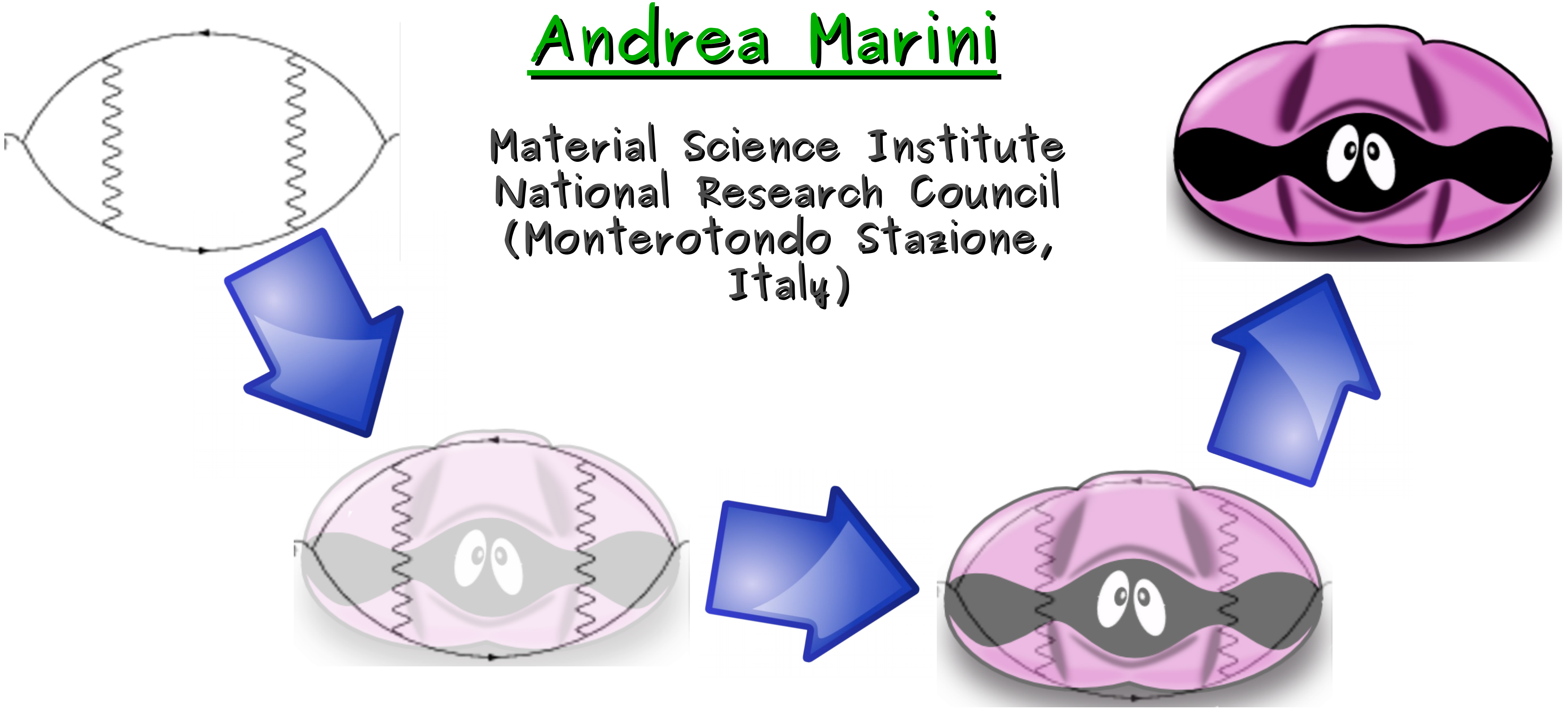
The Yambo project:
a broad and european-based environment for atomistic simulations

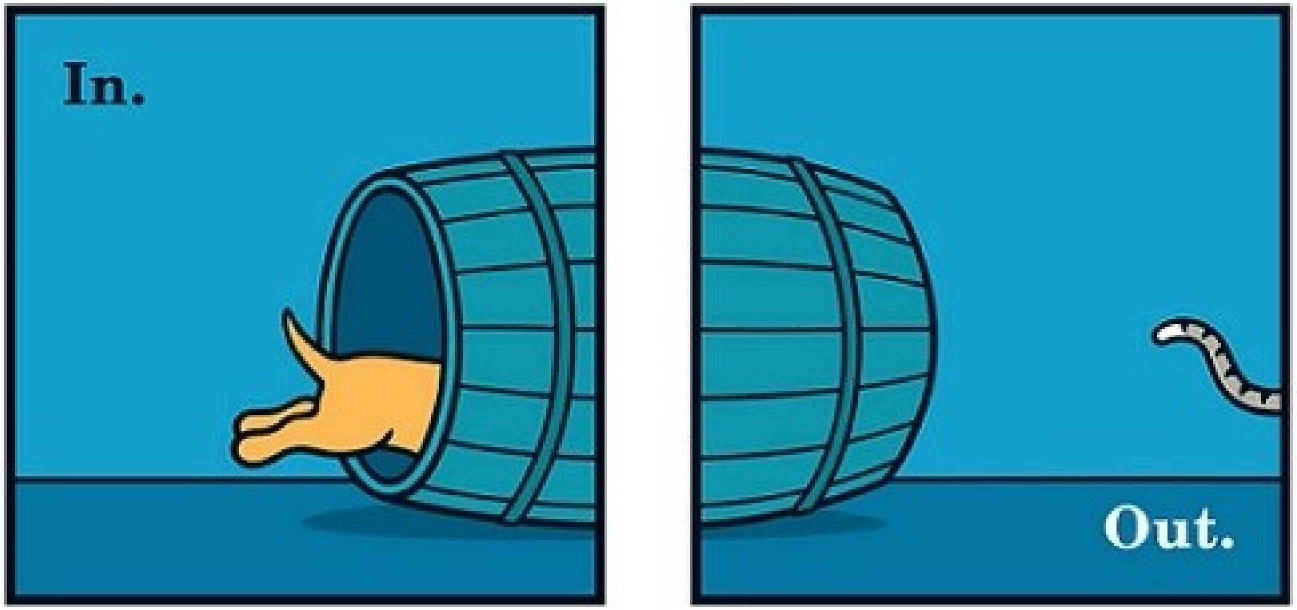
Trieste, January 2017

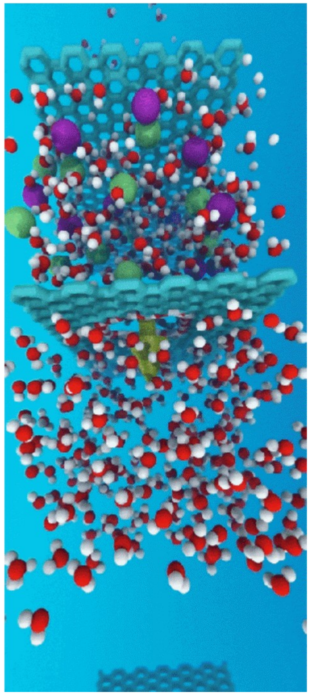
Andrea Marini

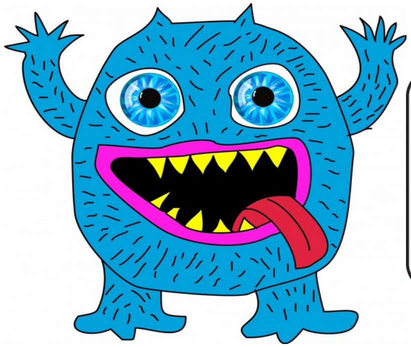
Material Science Institute
National Research Council
(Monterotondo Stazione, Italy)

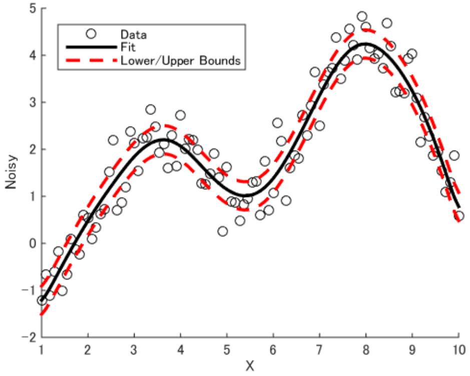


The Ab-Initio "Way"

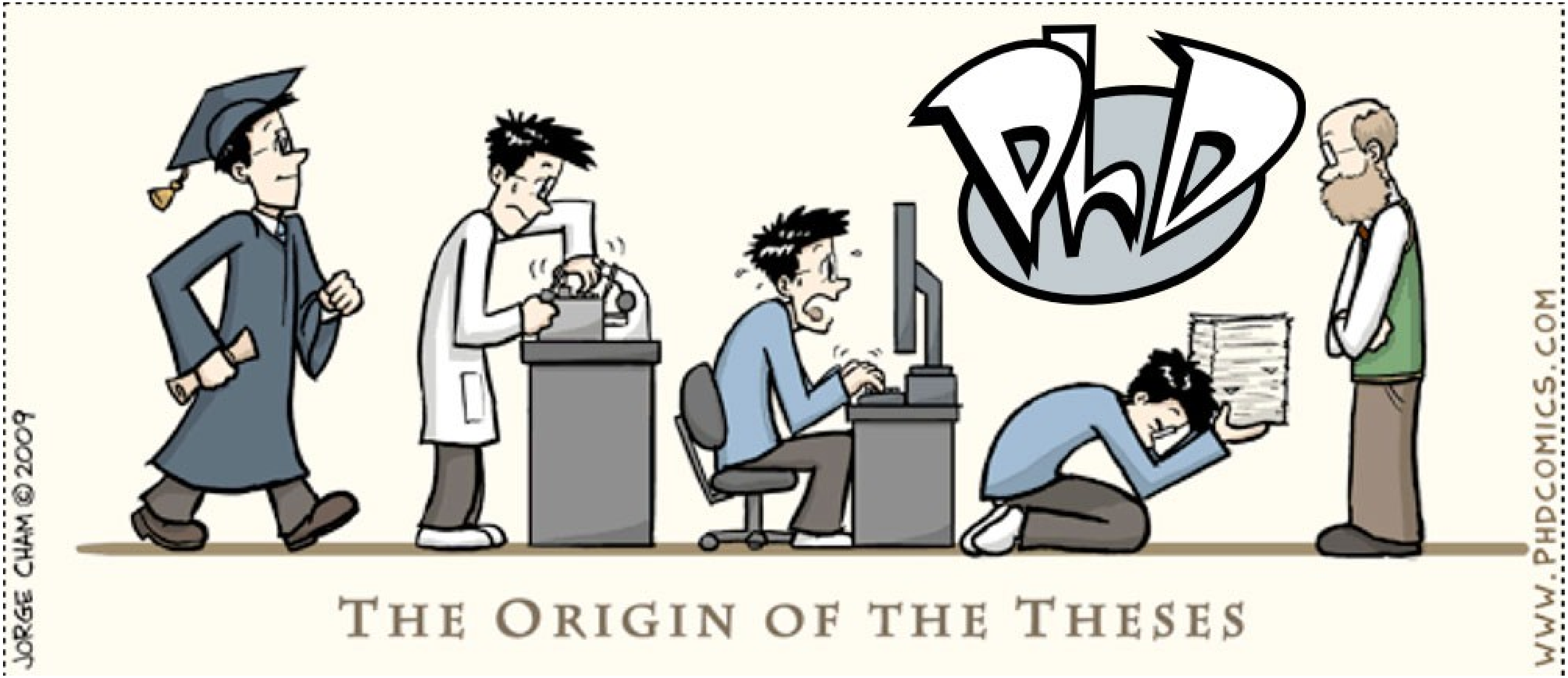


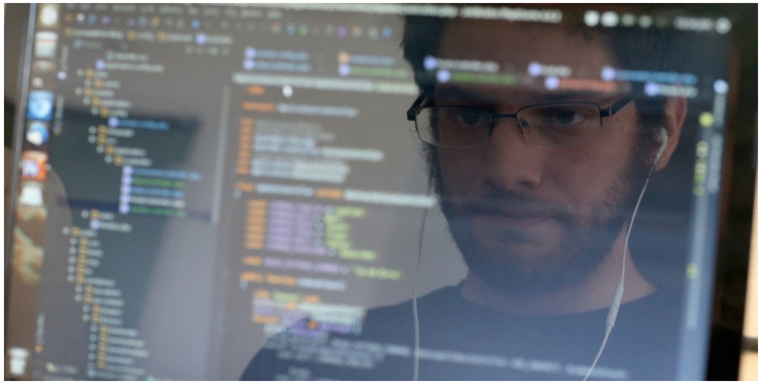






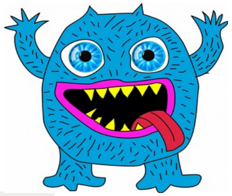
The Ab-Initio "Way": Codes… YESTERDAY!

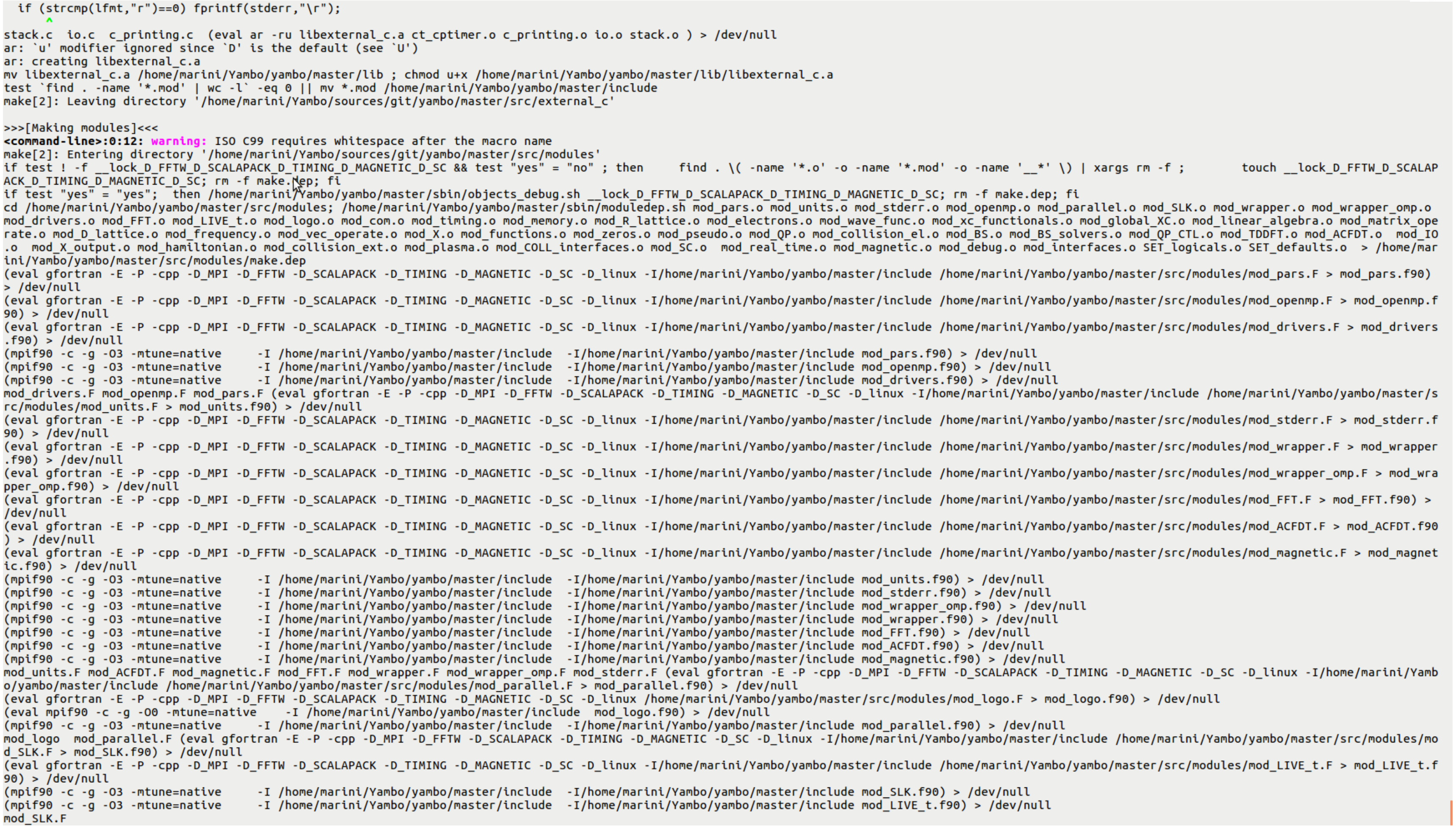




```
if (outcome≠bad && 🐱===fun) {
  thenWhyNot();
}
```

The Ab-Initio "Way": Codes… TODAY!



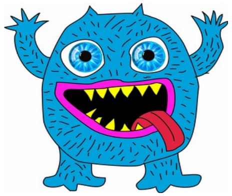


```
 if (strcmp(lfmt,"r")==0) fprintf(stderr,"\r");
   ^
stack.c  io.c  c_printing.c  (eval ar -ru libexternal_c.a ct_cptimer.o c_printing.o io.o stack.o ) > /dev/null
ar: `u' modifier ignored since `D' is the default (see `U')
ar: creating libexternal_c.a
mv libexternal_c.a /home/marini/Yambo/yambo/master/lib ; chmod u+x /home/marini/Yambo/yambo/master/lib/libexternal_c.a
test `find . -name '*.mod' | wc -l` -eq 0 || mv *.mod /home/marini/Yambo/yambo/master/include
make[2]: Leaving directory '/home/marini/Yambo/sources/git/yambo/master/src/external_c'

>>>[Making modules]<<<
<command-line>:0:12: warning: ISO C99 requires whitespace after the macro name
make[2]: Entering directory '/home/marini/Yambo/sources/git/yambo/master/src/modules'
if test ! -f __lock_D_FFTW_D_SCALAPACK_D_TIMING_D_MAGNETIC_D_SC && test "yes" = "no" ; then     find . \( -name '*.o' -o -name '*.mod' -o -name '__*' \) | xargs rm -f ;        touch __lock_D_FFTW_D_SCALAP
ACK_D_TIMING_D_MAGNETIC_D_SC; rm -f make.dep; fi
if test "yes" = "yes";  then /home/marini/Yambo/yambo/master/sbin/objects_debug.sh __lock_D_FFTW_D_SCALAPACK_D_TIMING_D_MAGNETIC_D_SC; rm -f make.dep; fi
cd /home/marini/Yambo/yambo/master/src/modules; /home/marini/Yambo/yambo/master/sbin/moduledep.sh mod_pars.o mod_units.o mod_stderr.o mod_openmp.o mod_parallel.o mod_SLK.o mod_wrapper.o mod_wrapper_omp.o
mod_drivers.o mod_FFT.o mod_LIVE_t.o mod_logo.o mod_com.o mod_timing.o mod_memory.o mod_R_lattice.o mod_electrons.o mod_wave_func.o mod_xc_functionals.o mod_global_XC.o mod_linear_algebra.o mod_matrix_ope
rate.o mod_D_lattice.o mod_frequency.o mod_vec_operate.o mod_X.o mod_functions.o mod_zeros.o mod_pseudo.o mod_QP.o mod_collision_el.o mod_BS.o mod_BS_solvers.o mod_QP_CTL.o mod_TDDFT.o mod_ACFDT.o  mod_IO
.o  mod_X_output.o mod_hamiltonian.o mod_collision_ext.o mod_plasma.o mod_COLL_interfaces.o mod_SC.o  mod_real_time.o mod_magnetic.o mod_debug.o mod_interfaces.o SET_logicals.o SET_defaults.o  > /home/mar
ini/Yambo/yambo/master/src/modules/make.dep
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_pars.F > mod_pars.f90)
> /dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_openmp.F > mod_openmp.f
90) > /dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_drivers.F > mod_drivers
.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_pars.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_openmp.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_drivers.f90) > /dev/null
mod_drivers.F mod_openmp.F mod_pars.F (eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/s
rc/modules/mod_units.F > mod_units.f90) > /dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_stderr.F > mod_stderr.f
90) > /dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_wrapper.F > mod_wrapper
.f90) > /dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_wrapper_omp.F > mod_wra
pper_omp.f90) > /dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_FFT.F > mod_FFT.f90) >
/dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_ACFDT.F > mod_ACFDT.f90
) > /dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_magnetic.F > mod_magnet
ic.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_units.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_stderr.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_wrapper_omp.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_wrapper.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_FFT.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_ACFDT.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_magnetic.f90) > /dev/null
mod_units.F mod_ACFDT.F mod_magnetic.F mod_FFT.F mod_wrapper.F mod_wrapper_omp.F mod_stderr.F (eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yamb
o/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_parallel.F > mod_parallel.f90) > /dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux /home/marini/Yambo/yambo/master/src/modules/mod_logo.F > mod_logo.f90) > /dev/null
(eval mpif90 -c -g -O0 -mtune=native    -I /home/marini/Yambo/yambo/master/include  mod_logo.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_parallel.f90) > /dev/null
mod_logo  mod_parallel.F (eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mo
d_SLK.F > mod_SLK.f90) > /dev/null
(eval gfortran -E -P -cpp -D_MPI -D_FFTW -D_SCALAPACK -D_TIMING -D_MAGNETIC -D_SC -D_linux -I/home/marini/Yambo/yambo/master/include /home/marini/Yambo/yambo/master/src/modules/mod_LIVE_t.F > mod_LIVE_t.f
90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_SLK.f90) > /dev/null
(mpif90 -c -g -O3 -mtune=native     -I /home/marini/Yambo/yambo/master/include  -I/home/marini/Yambo/yambo/master/include mod_LIVE_t.f90) > /dev/null
mod_SLK.F
```

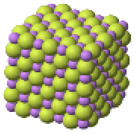
- Hundreds of thousand of lines of code (C, C++, Fortran…)
- Written in several versions (4 for Yambo) along many years (16 for Yambo)
- Tens of developers (Students, Researchers, HPC experts…) involved
- Growing User community (and NEEDS!)

The Ab-Initio "Way": Density-Functional Theory



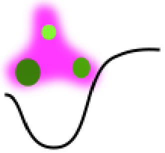
IN: System:

unit cell
lattice vectors
basis



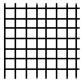
Hamiltonian:

(physical approx.)

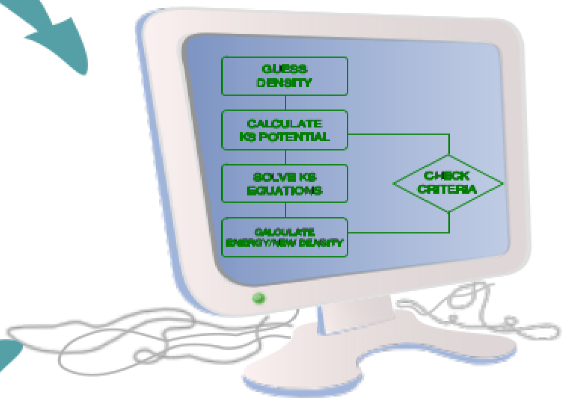


xc-approximation
(relativistic effects)

Numerical approx:



energy cut-off
k-points grid
pseudopotentials
SCF procedure/threshold



RUN:

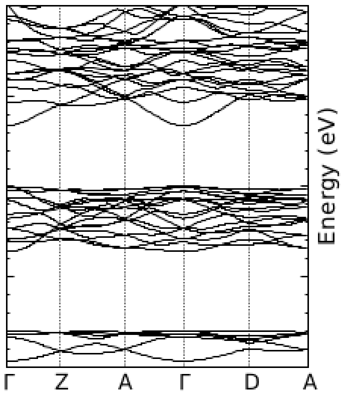
Solve
KS equations

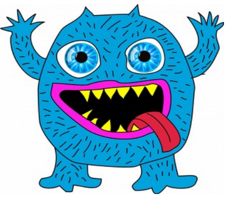
OUT: Physical quantities

density and related quantities
total energy and components
any GS observable (in principle)

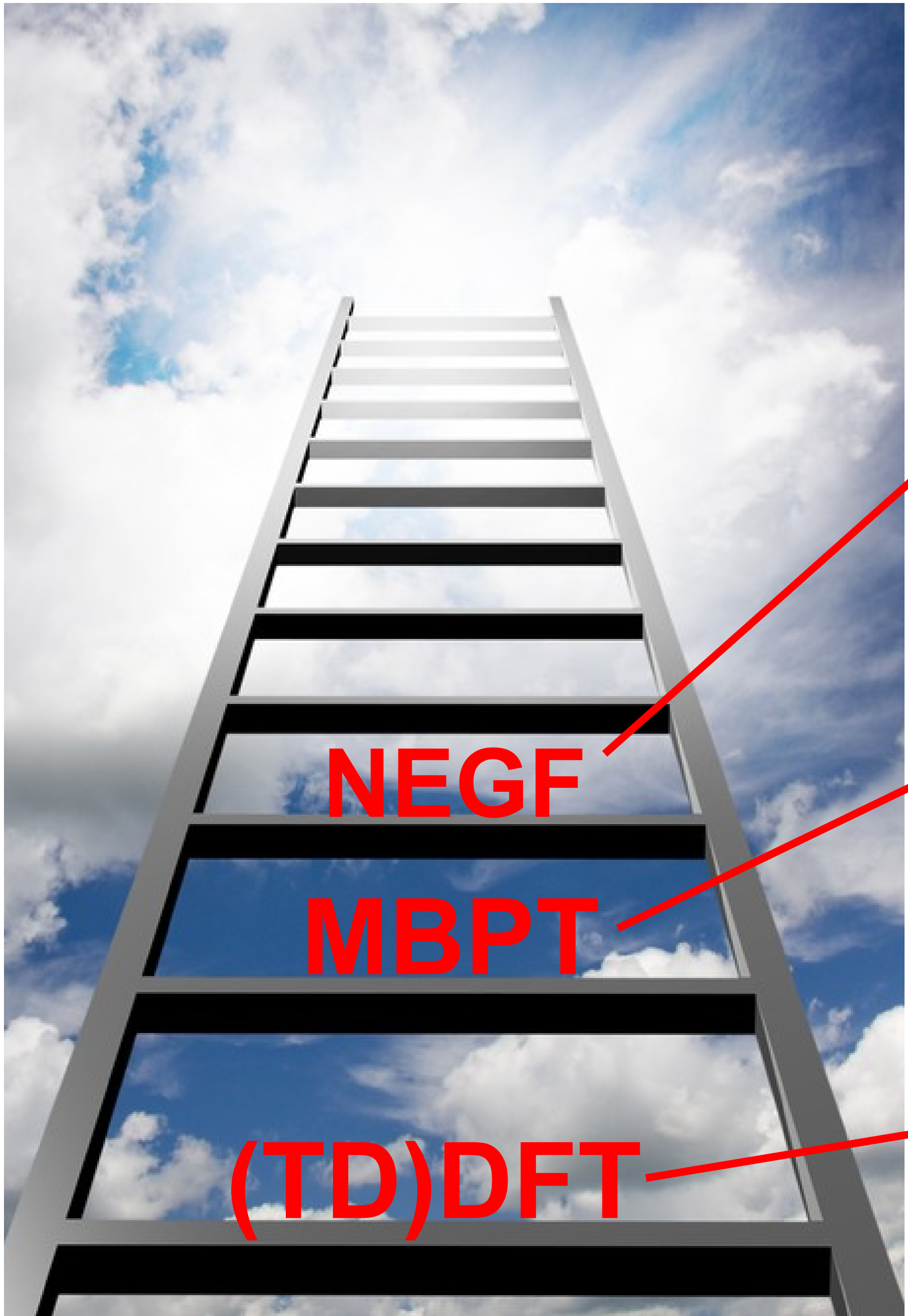
1-particle quantities

Kohn-Sham 1-p wavefunctions
Kohn-Sham 1-p energies





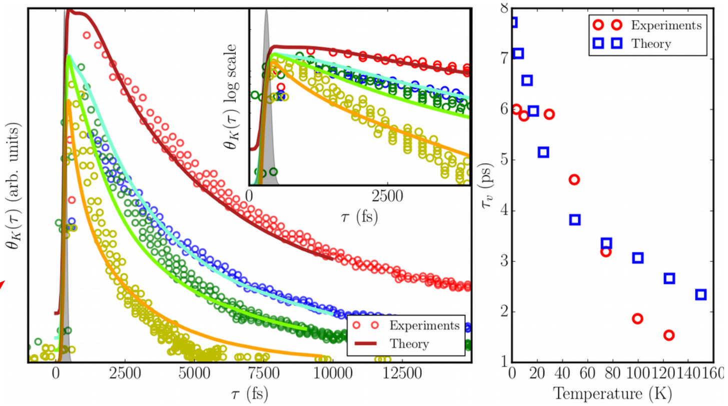
The Ab-Initio "Way": Codes & Physics...

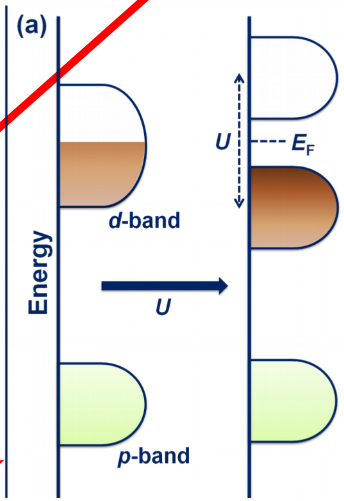


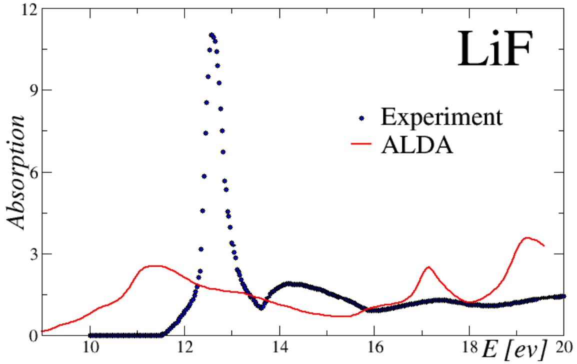
NEGF

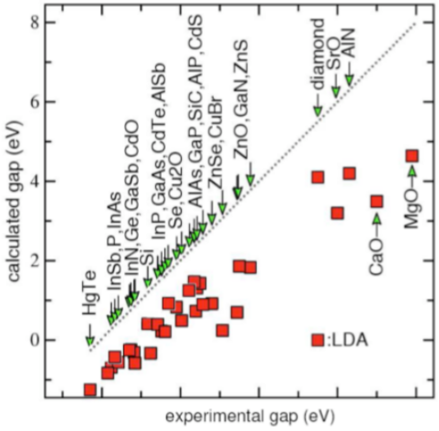
MBPT

(TD)DFT







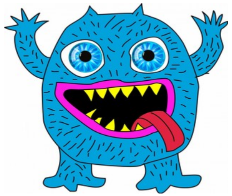


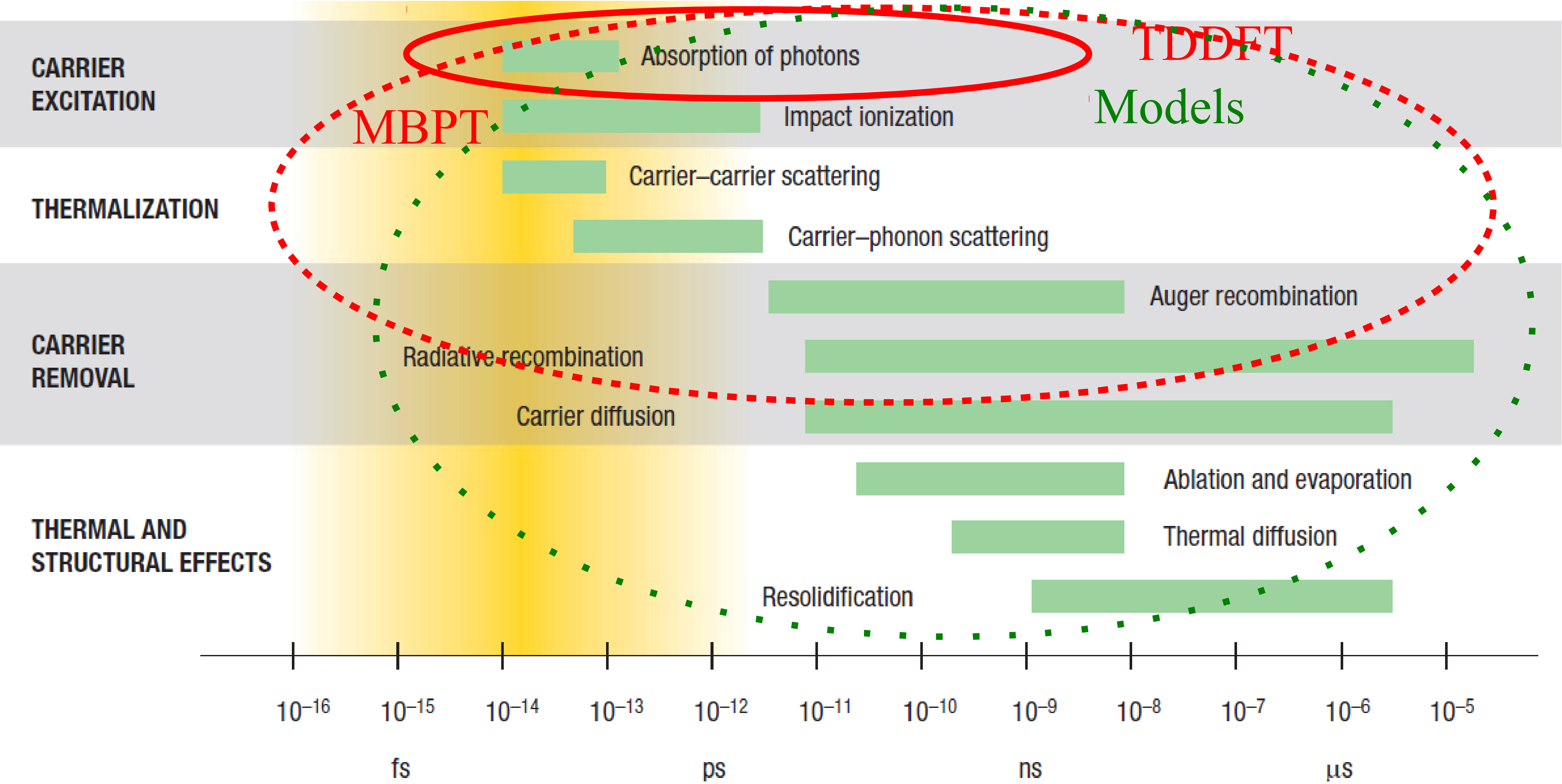
Si:
0.47 eV (LDA) vs 1.1 eV (expt)

GaAs:
0.30 eV (LDA) vs 1.4 eV (expt)

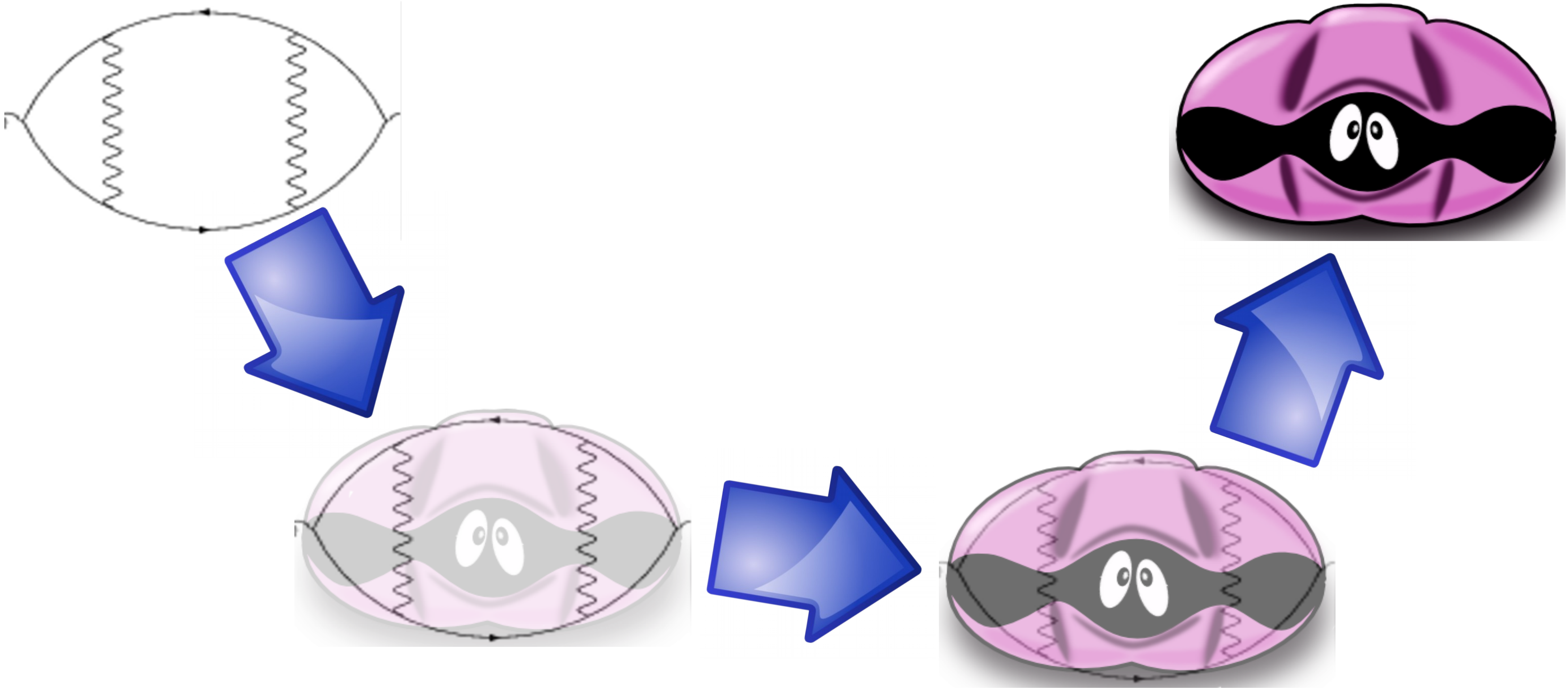
Adapted from M. van Schilfgaarde *et al.* PRL **96** (2006)

The Ab-Initio "Way":
Codes & Physics…





The Yambo project

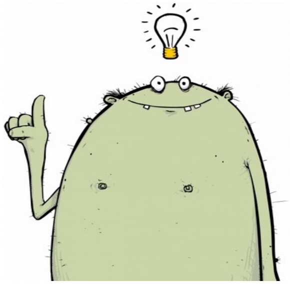


Yambo©

IS

NOT

QUANTUMESPRESSO



Theory(/ies)

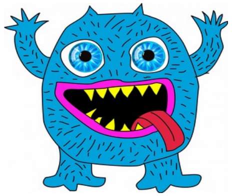
ONE "code"

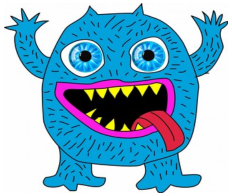
Modular structure and layers of coding

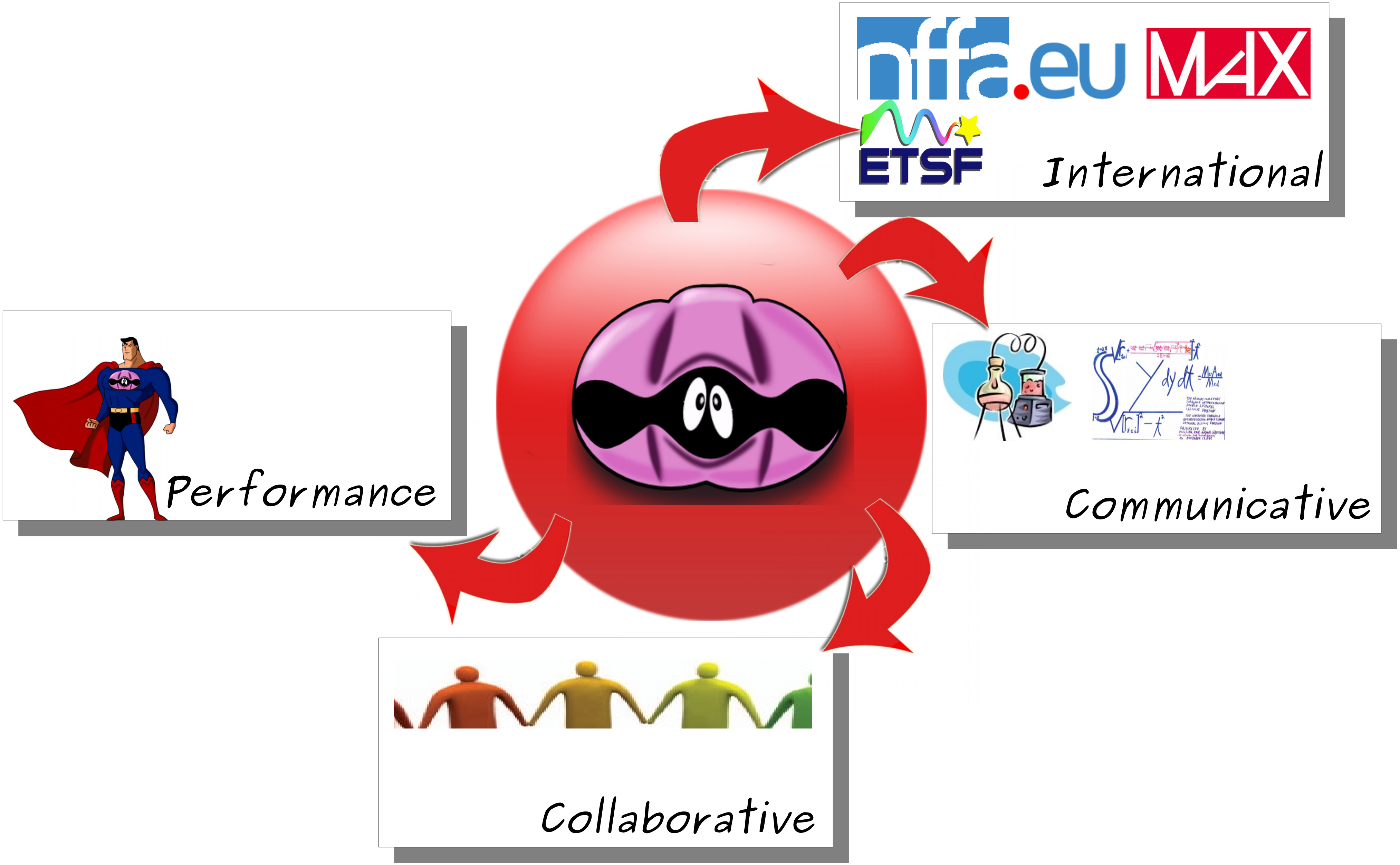
Interface with
- Users
- Developers
- Students
- Communities

Interface with other codes

Performance







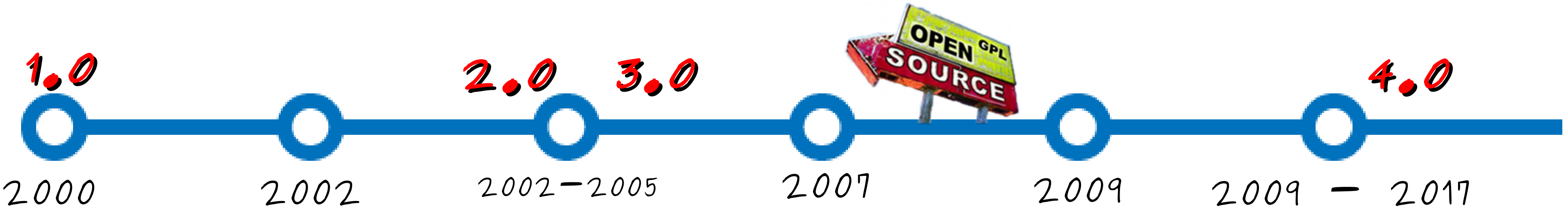
International

Communicative

Collaborative

Performance

Be collaborative



GW

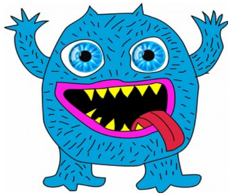
Excitons

Lifetimes
TDDFT
DFT
(Orbital
functionals)

Electron-Phonon

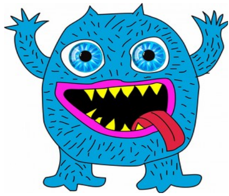
Iterative
solvers

Magnetic properties
Time-Dependent
phenomena
Total Energy
Collinear and non
collinear spin support

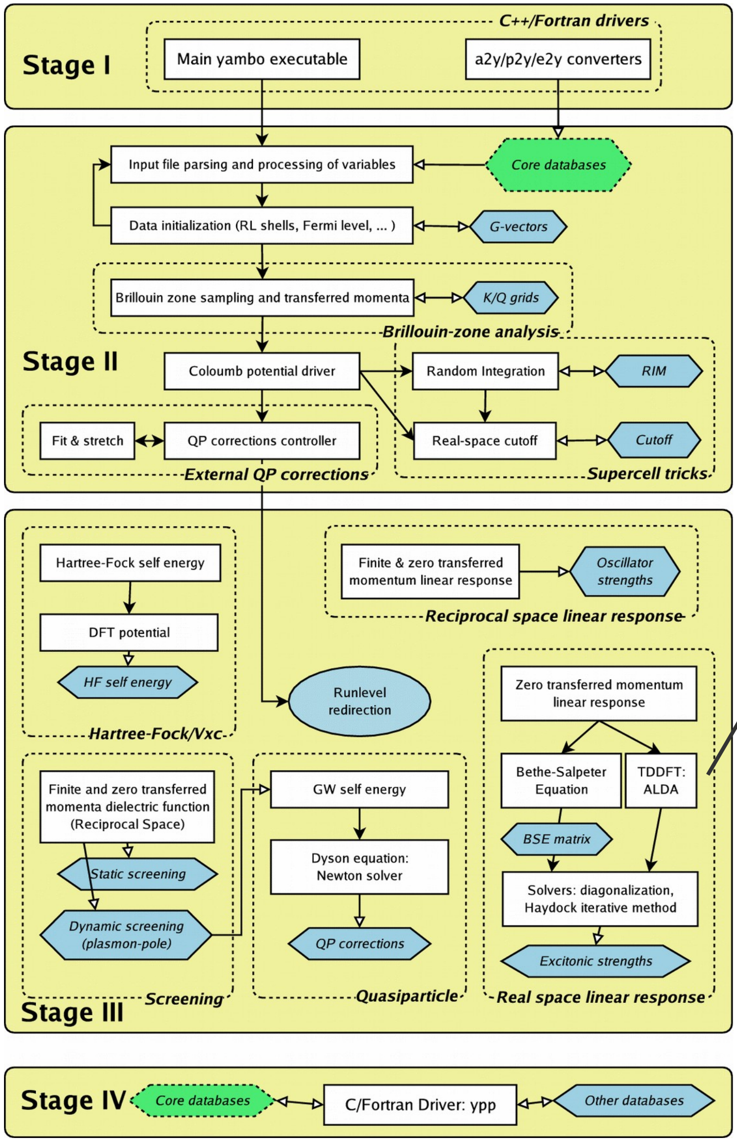


Yambo: an ab initio tool for excited state calculations, Andrea Marini, Conor Hogan, Myrta Grüning, Daniele Varsano, Comp. Phys. Comm. 180, 1392 (2009).

Be collaborative

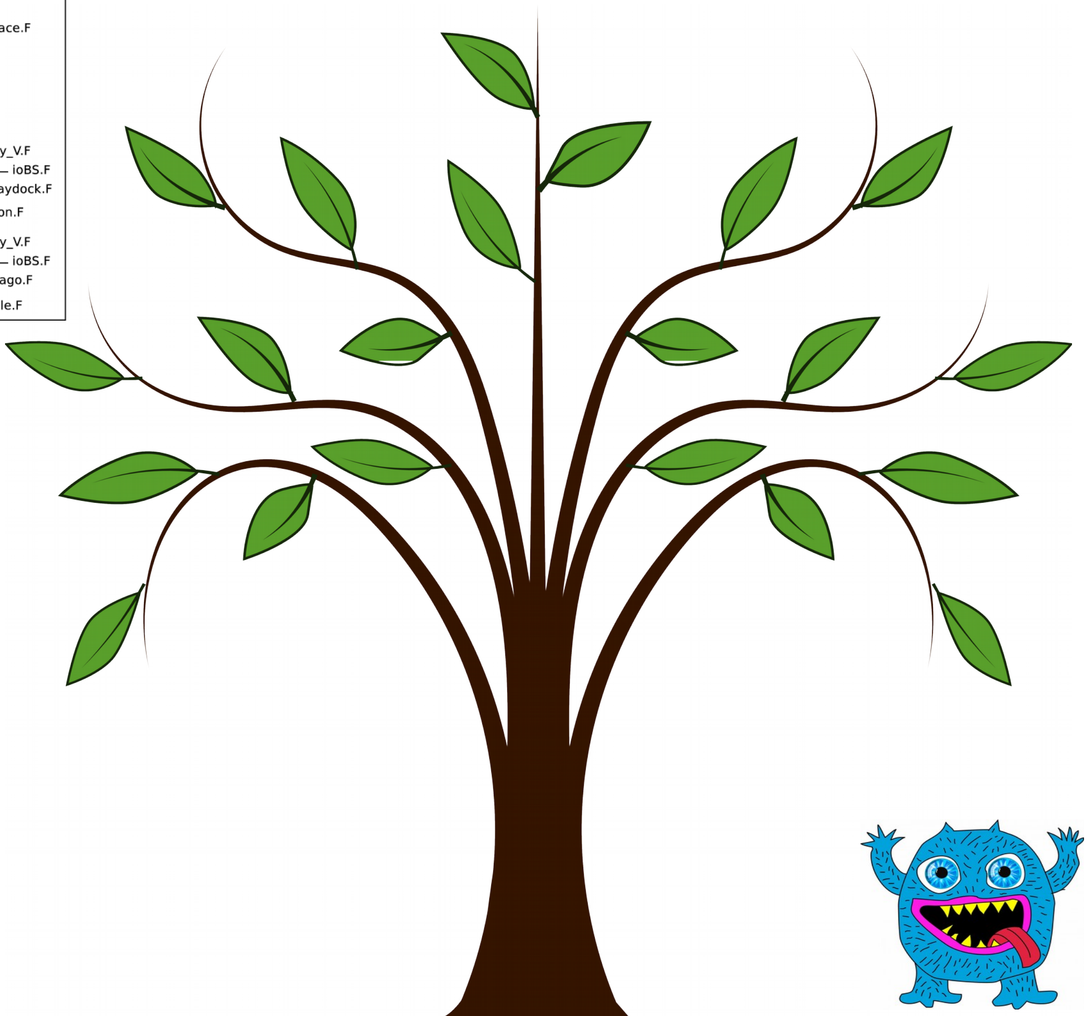


Be collaborative = Modular structure

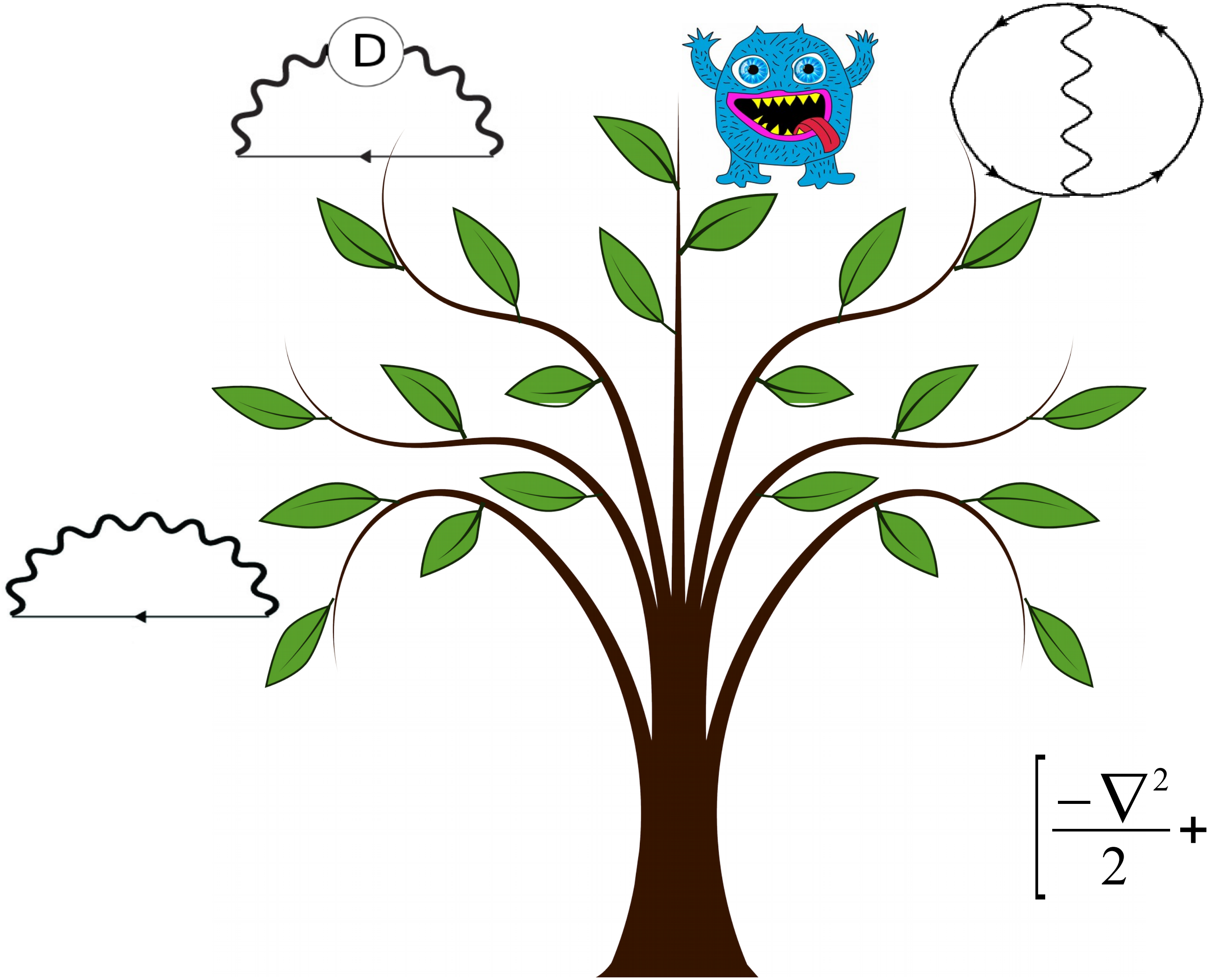


Objective:	Real Space Linear response
Driver:	K_driver.F
Directory:	src/bse/
Databases:	(n)db.BS_Q1 (n)db.BS_diago_Q1 (n)db.Haydock_restart
Runlevels:	-o b / -t a / -y dh
Restart:	YES

```
K_driver.F
 ├── [QP_apply (mod_QP_CTL.F)]
 ├── K
 │    ├── K_eh_setup.F
 │    ├── K_filling.F
 │    ├── K_exchange.F
 │    ├── tddft_alda_r_space.F
 │    ├── K_scattering.F
 │    └── ioBS.F
 └── K_solvers.F
      ├── K_Haydock.F
      │    ├── Kernel_by_V.F
      │    │    └── ioBS.F
      │    └── ioBSS_Haydock.F
      ├── K_diagonalization.F
      │    ├── Kernel_by_V.F
      │    │    └── ioBS.F
      │    └── ioBSS_diago.F
      └── K_dump_to_o_file.F
```



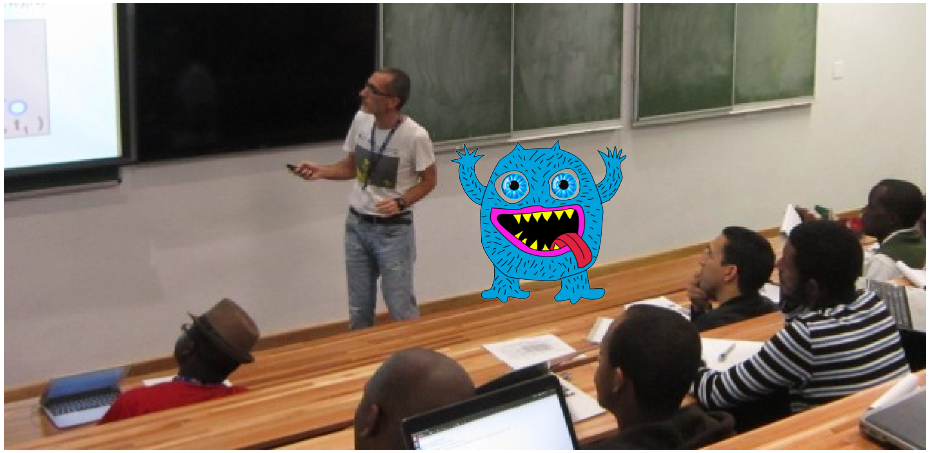
Be collaborative = Modular structure

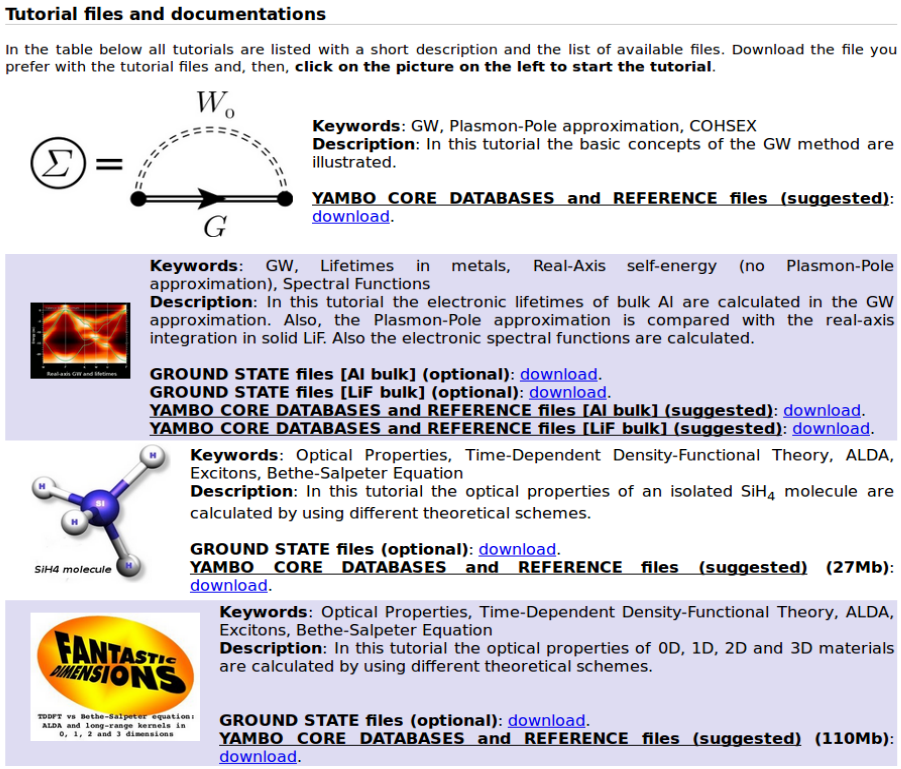


$$\left[\frac{-\nabla^2}{2}+v_{DFT}(r)\right]\psi_{nk}(r)=\epsilon_{nk}\psi_{nk}(r)$$

FFTW OpenMP MPI NetCDF PETSc ScaLAPACK

Dissemination





Tutorial files and documentations

In the table below all tutorials are listed with a short description and the list of available files. Download the file you prefer with the tutorial files and, then, **click on the picture on the left to start the tutorial**.

Keywords: GW, Plasmon-Pole approximation, COHSEX
Description: In this tutorial the basic concepts of the GW method are illustrated.

YAMBO CORE DATABASES and REFERENCE files (suggested): download.

Keywords: GW, Lifetimes in metals, Real-Axis self-energy (no Plasmon-Pole approximation), Spectral Functions
Description: In this tutorial the electronic lifetimes of bulk Al are calculated in the GW approximation. Also, the Plasmon-Pole approximation is compared with the real-axis integration in solid LiF. Also the electronic spectral functions are calculated.

GROUND STATE files [Al bulk] (optional): download.
GROUND STATE files [LiF bulk] (optional): download.
YAMBO CORE DATABASES and REFERENCE files [Al bulk] (suggested): download.
YAMBO CORE DATABASES and REFERENCE files [LiF bulk] (suggested): download.

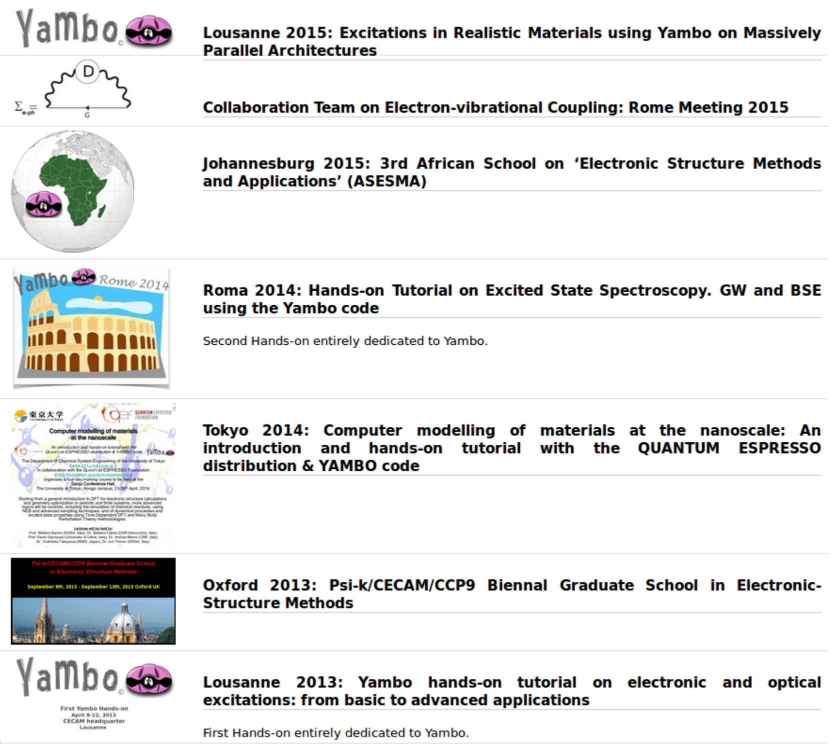
Keywords: Optical Properties, Time-Dependent Density-Functional Theory, ALDA, Excitons, Bethe-Salpeter Equation
Description: In this tutorial the optical properties of an isolated SiH_4 molecule are calculated by using different theoretical schemes.

GROUND STATE files (optional): download.
YAMBO CORE DATABASES and REFERENCE files (suggested) (27Mb): download.

FANTASTIC DIMENSIONS
TDDFT vs Bethe-Salpeter equation:
ALDA and long-range kernels in
0, 1, 2 and 3 dimensions

Keywords: Optical Properties, Time-Dependent Density-Functional Theory, ALDA, Excitons, Bethe-Salpeter Equation
Description: In this tutorial the optical properties of 0D, 1D, 2D and 3D materials are calculated by using different theoretical schemes.

GROUND STATE files (optional): download.
YAMBO CORE DATABASES and REFERENCE files (suggested) (110Mb): download.



Lousanne 2015: Excitations in Realistic Materials using Yambo on Massively Parallel Architectures

Collaboration Team on Electron-vibrational Coupling: Rome Meeting 2015

Johannesburg 2015: 3rd African School on 'Electronic Structure Methods and Applications' (ASESMA)

Roma 2014: Hands-on Tutorial on Excited State Spectroscopy. GW and BSE using the Yambo code

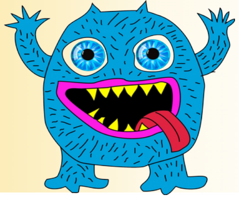
Second Hands-on entirely dedicated to Yambo.

Tokyo 2014: Computer modelling of materials at the nanoscale: An introduction and hands-on tutorial with the QUANTUM ESPRESSO distribution & YAMBO code

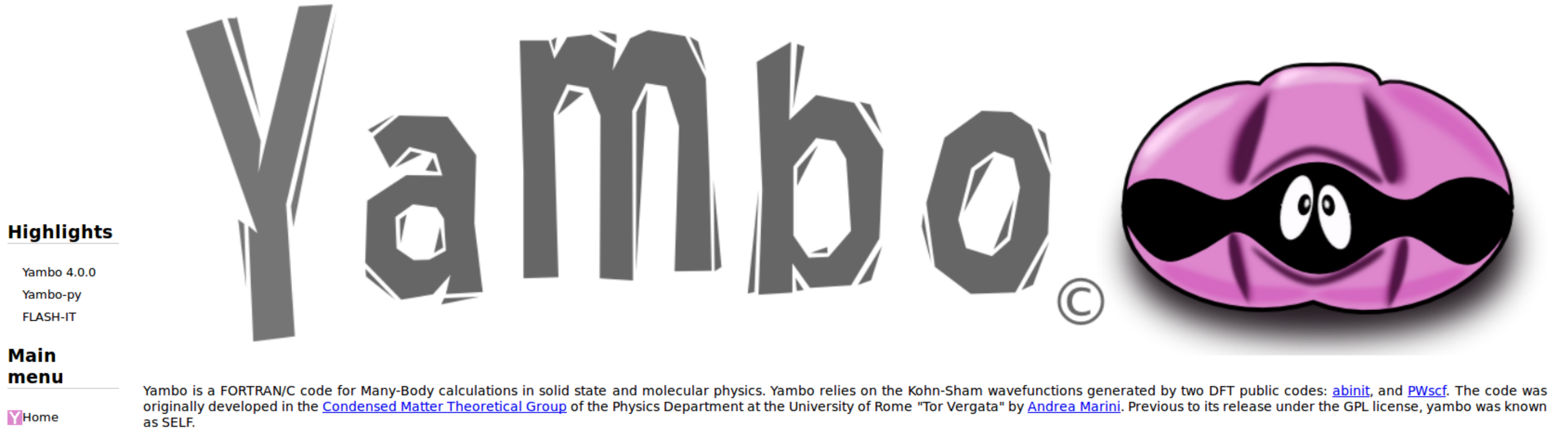
Oxford 2013: Psi-k/CECAM/CCP9 Biennal Graduate School in Electronic-Structure Methods

Lousanne 2013: Yambo hands-on tutorial on electronic and optical excitations: from basic to advanced applications

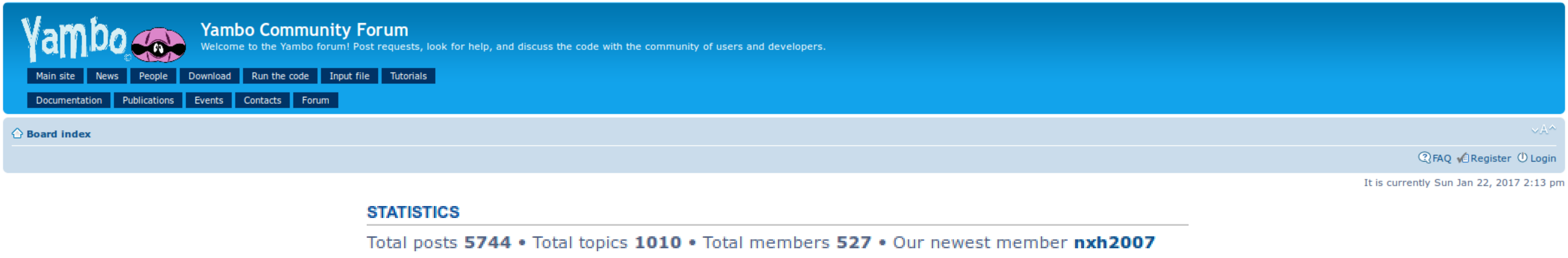
First Hands-on entirely dedicated to Yambo.



Dissemination



Yambo is a FORTRAN/C code for Many-Body calculations in solid state and molecular physics. Yambo relies on the Kohn-Sham wavefunctions generated by two DFT public codes: abinit, and PWscf. The code was originally developed in the Condensed Matter Theoretical Group of the Physics Department at the University of Rome "Tor Vergata" by Andrea Marini. Previous to its release under the GPL license, yambo was known as SELF.



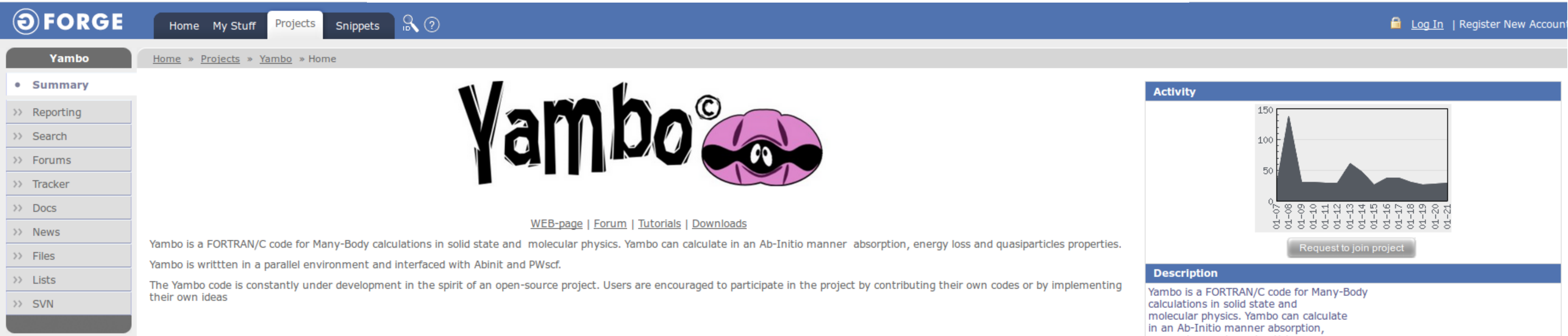
Yambo Community Forum

Welcome to the Yambo forum! Post requests, look for help, and discuss the code with the community of users and developers.

It is currently Sun Jan 22, 2017 2:13 pm

STATISTICS

Total posts **5744** • Total topics **1010** • Total members **527** • Our newest member **nxh2007**



WEB-page | Forum | Tutorials | Downloads

Yambo is a FORTRAN/C code for Many-Body calculations in solid state and molecular physics. Yambo can calculate in an Ab-Initio manner absorption, energy loss and quasiparticles properties.

Yambo is writtten in a parallel environment and interfaced with Abinit and PWscf.

The Yambo code is constantly under development in the spirit of an open-source project. Users are encouraged to participate in the project by contributing their own codes or by implementing their own ideas

Yambo is a FORTRAN/C code for Many-Body calculations in solid state and molecular physics. Yambo can calculate in an Ab-Initio manner absorption,

ONE MAN SHOW

Community development

HPC challenge

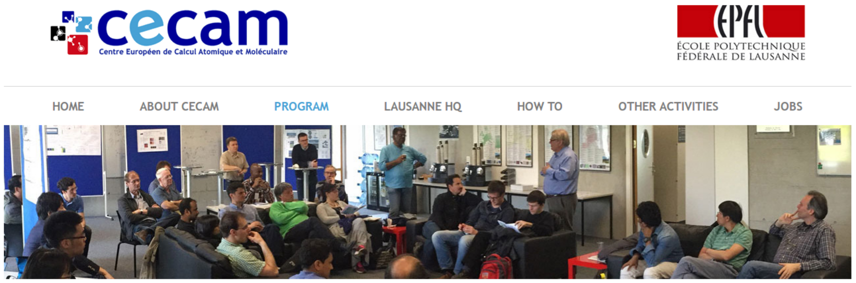
User support and feedback

http://www.yambo-code.org/

Schools

Advanced computing of excited state properties in solids and nanostructures with Yambo

April 24, 2017 to April 28, 2017
Location : CECAM-HQ-EPFL, Lausanne, Switzerland


cecam
Centre Européen de Calcul Atomique et Moléculaire
EPFL
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
HOME
ABOUT CECAM
PROGRAM
LAUSANNE HQ
HOW TO
OTHER ACTIVITIES
JOBS

Details | Participants | Apply

If you are interested in this School, please contact the organisers directly by using the form which is available in the Registered User Area.

- If you already have a CECAM account, please first login.
- If not, please sign up here.

http://www.yambo-code.org/