Distributed science and development: the ETSF and all its little packages

M Verstraete
University of Liege, Belgium & European Theoretical Spectroscopy Facility
Experimentalists need us

Physics = explaining experiments using modeling

Materials properties = electronic structure:

• ground state
• equilibrium structure
• thermodynamics and...
• excited states

almost all experiments probe excitations
Why we need large collaborations

Intrinsically ever more complex processes for interesting systems

• different codes & types of expertise
• levels of precision (CI, CAS, DFT, TB, MM...)

Need to:

• exchange data
• benchmark
• compare results between levels of accuracy
From the top down

Wide scientific collaborations
- combine codes and talk to experimentalists
- common libraries and software infrastructure
- distribution of packages

Code level collaborations
- distributed development
- testing, cross validation, reproducibility

Library level collaborations
The European Theoretical Spectroscopy Facility
ETSF – interfacing users

Distributed facility for theoretical spectroscopy

Users:
• industry
• experimentalists
• theoreticians in training

Members:
• 10 theoretical groups with 20 years’ experience in excited state research and applications
• Associated members and nodes

http://www.etsf.eu
What is theoretical spectroscopy?

Simulation and prediction of spectroscopic quantities:
• electrons: PES, ARPES, EELS, X-Ray
• phonons: Raman, IR, Non-linear
• transport: STM, time resolved

What it’s not: ground states, structures, catalysis, thermodynamics...

The buzzword has caught on very well in Europe!
Related initiatives

molecular foundry  http://foundry.lbl.gov/

• Not just theory
• Structural continual funding
• Wider user base and project variety

NNIN/C CNF

• Repositories, specific theoretical tools

Some of you in the audience!
User needs in Spectroscopy

Main strands identified, but evolving structure:
• demand changes
• capabilities change
• user communities change

Synchrotron-like structure: beamlines according to different types of spectroscopies

Steering committee + external evaluation board
Beamlines

- Optics
- Energy Loss Spectroscopy
- Quantum Transport
- Time-resolved Spectroscopy
- Photo-emission Spectroscopy
- Vibrational Spectroscopy
- X-Ray Spectroscopy

Coordinator to orient users and asses “doability”
Life and times of an ETSF project

• Biannual call for proposals with simple submission process
• Ideally contact beamline coordinator before
• Few page application, just the beef
• Approval by beamline coordinators
• Selection by external evaluation panel
• Designation of researcher for execution among associate members and nodes
Example: Au$_2$Gd surface alloys

Experimental characterization of alloy on Au(111) moiré: decorate with Co. STM, diffraction, ARPES

- Structure of alloy?
- Explain STM inversion
- Magnetism of alloy? Bulk is AFM
- # of ML = 1 or 2? Layer stacking?
Au$_2$Gd surface alloys

Definitive alloy structure

Analyze ARPES bands

Explain STM contrast inversion

M. Corso et al. PRL 105 016101 (2010)
Training projects

Can train experimental user in basics, or DFT-savvy user in spectroscopic methods

Both for electronic and phononic spectroscopy

Training proposals have no panel:
• grouped and carried out in workshops
• on-site
• visit to ETSF node
Funding science and coding

Hard to get recognition for development work
Hard to get funding too: it’s real work, but not the substance of the physics (John Pople?)
Funding HPC centers is “ok”, with technical personnel, but rarely software development
Development is now beginning to be recognized:
• permanent technicians
• software engineers, etc...
• EU and national sources
  (Spain, Belgium, France, )
Funding the ETSF

- EU FP7 infrastructure grant: e-I3
- national
- regional
- university
- co-funding by industry
- user funding of projects, PhD students...

Building a Collaborative Framework for Nanoscale Simulations
IP issues

Adaptable model for intellectual property:
1. Normal scientific collaboration: publications in concert with users
2. Patents: depends on contract with ETSF
3. For some users (in particular industrial) all IP owned by user. This is not compatible with standard public funding, though
Career issues

What do ETSF scientists get out of it?
1. Publications – calculations are usually run of the mill, but focus and thematic are cutting edge thanks to users
2. Jobs – part time user projects and part time normal development
3. Jobs – some PhD students continue on with industrial users by contract
ETSF codes

Distribute a suite of spectroscopy codes:

GPL (or academic) license

Set of coding (& doc) standards

- GS codes
- GW, TDDFT (real time and Casida), BSE
- Utility codes

Building a Collaborative Framework for Nanoscale Simulations
Distributions of ETSF codes

Many avenues for distribution
The simpler for the user/sysadmin the better

- live cd (mainly for demos)
- deb, rpm, tgz
- get into linux distributions!
- Debian science blend «nanoscale-physics»
  gives meta packages needed for related projects and codes, and ETSF in particular (2011)
ETSF – interfacing codes
ETSF – interfacing codes

Projects need different codes according to required spectroscopy, precision, system size...
Typically, at least, GS code, excited state, post-processing + visualization: 4 or 5 codes!

Codes interface for:
• basic electronic structure (etsf-io)
• crystal structure (etsf-io)
• pseudopotentials (pspconvert, libpssp)
• exchange correlation functionals (libxc)
ETSF-io: Problems and solutions

Large volume of data (MB to > GB): binary
Cross-platform portability: netcdf
Easy i-o and interfacing: wrap it in a library (+add bindings for other programming languages)
Open GPL format and lib (of course)
Self-describing netcdf with clear variable names

File format specs involve all ETSF code teams
The ETSF file format

Basics:
- Crystal structure
- Density, potential (x,c,H,...)
- Wavefunctions: basis dependent
  - PW
  - real space
  - wavelets (BigDFT)
  - LAPW planned
- Electronic bands, k-points...

Images: v_sim (http://inac.cea.fr/L_Sim/V_Sim/)
The ETSF file format

Less basic:

• GW corrections $\Delta \epsilon_{nk}$
• response function (or dielectric matrix) $\chi_{k\omega}(G, G')$
• electron-phonon $\langle \psi_{nk} | \delta V | \psi_{n'k'} \rangle$
• anything you want: open structure of netcdf

Limitations:

• not hierarchical
• only one “unlimited” array
The ETSF-io library

Full spec of ETSF file format
F90 (C coming) from templates, python bindings
Written by Damien Caliste (CEA Grenoble)
Extensive use of pointers (yes, in Fortran!) for transparent inclusion in existing codes
Netcdf calls hidden in low level access layer
Porting to Netcdf v4 (HDF superset w/parallel io)
Interfacing codes: an example

- Na under high pressure
- Becomes insulating
- GS + GW: abinit, EXC
- BSE: DP+EXC
- Excitonic weight below the band gap
- Charge transfer into void in DHCP structure
- Anisotropic optical properties

PRL 104, 216404 (2010)
ETSF knowledge base

**Idea:** database and datamine input files from user (and other) projects

Input from expert users:
- converged choices of cutoffs
- pseudopotentials
- example structures
- strategies for typical calculations

Make the collection process automatic: extract and send data to central db
ETSF knowledge base

Very important to collect qualitative experience too

Infrastructure is up and running: MySQL, HTML, Python and PHP

Several trial users from octopus
Building a Collaborative Framework for Nanoscale Simulations
ETSF call for applications: [http://www.etsf.eu/](http://www.etsf.eu/)

- Next deadline: April 2011
The package level
Abinit

General purpose ab-initio code
• plane wave code, PAW/NC
• parallelized over k-points, bands, G vectors

Focus on:
• DFPT: phonons, NL, electric fields (also finite)
• PAW-GW/BSE: MBPT spectroscopy
• robust execution, parallelization, portability
• extensive documentation, also for developers, both inline and in howto/wiki/manual
Abinit

• First written in Cornell in the 80s
• Sold to biosym then finally given to the public domain in 1997-2000: GPL release of v3
• > 40 active developers (80 total for v5)
• world wide: Belgium, France, USA, Canada, Italy, Spain, Japan...

Founding papers cited > 1500 times
Estimated several thousand users from forum activity and citations
Developer’s workshop

Every 2 years, alternate with user workshop
Learn about concurrent projects
Avoid double implementations, reuse code

Increases collaboration
• on package
• many joint applications
Licensing

GNU Public License (v3) distribution
• Free and open software
• Enlarges user/tester base
• Good for the science above all
• Enable result reproduction!

Constraints:
• Must link only to other GPL/LGPL
• Return on investment? Contacts+citations
Versioning system - bzr

We use bazaar (bzr - http://bazaar.canonical.com/)

• most (svn, cvs...) have similar user front ends
• bzr is very efficient for merging
• nice tools and interfaces (html, php, python)
• robust for network failures
• rename files and keep history
Flexible archive distributions

Local archives easy: push when you’re online

local archive 1

commit

local copy 1

merge

archives.abinit.org

push

local archive 2

commit

local copy 2

pull
Loggerhead

web view of code diffs, trace modifications

<table>
<thead>
<tr>
<th>Rev</th>
<th>Summary</th>
<th>Authors</th>
<th>Date</th>
<th>Diff Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>405</td>
<td>abirules (note that pspso is erroneously reported to be in keyhr.html)</td>
<td>Matteo Giantomassi</td>
<td>2010-11-07</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>files modified:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tests/abirules/Refs/t01.out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>404</td>
<td>Merge with 551/trunk-private/402</td>
<td>Matteo Giantomassi</td>
<td>2010-11-07</td>
<td></td>
</tr>
<tr>
<td>403</td>
<td>Imported 651/gmatteo-training/410 (contains Stankovski's branch)</td>
<td>Matteo Giantomassi</td>
<td>2010-11-07</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>* tU2 should succeed on inca since gbound is not passed as assumed shape array anymore.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Portability problem with v6/t72-t76 should be solved thanks to the use of a stricter tolerance.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$(399.1.1$ gmatteo-training)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Matteo Giantomassi</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>files added:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>doc/tutorial/lesson_paw3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>doc/tutorial/lesson_paw3.html</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>doc/tutorial/lesson_paw3/codes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>doc/tutorial/lesson_paw3/codes/atomsaw2abinit.paw3 tutorial.tar.gz</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Testing

Users have very diverse platforms, compilers...

Ensure temporal stability of code and features

Backward compatibility: formats, inputs, binaries

• Automatic tests:
  – ca. 100 per version (>600)
  – all variables must be tested (and documented)
  – special suites for physics and fast checks
  – also for certain plugins

• Buildbot farm: nightly builds of modified branches. Also used for other ETSF codes
Test farm

Many representative platforms & compilers (intel, ibm, gcc, g95, pathscale)

Test many configurations: mpi/seq, all plugins

Check fortran quality (vars, check bounds...)

Check documentation
Pseudopotentials

Extensive online NC psp database
Several hundred LDA/GGA in many different schemes (HGH, TM, GTH, Teter)
PAW starting (most elements represented)

**Big issue:** Testing and validation
GS, GW (semi-core states), high P (small core)
Planned test farm for psp and PAW datasets
Package level

Systematic and clean use of autotools

Many plugins:

• netcdf + etsf-io
• Wannier90
• FoX XML parser
• LibXC
• BigDFT: use wavelet basis transparently!
Package level

Progressive librarification:
• Symmetries
• Pseudopotentials: read all formats
• AtomPAW: check PAW datasets, relax core
• Input reading: for v_sim and python
• Geometry relaxation
• ...
Conclusions

Novel techniques are needed for ever more complex challenges

Both for development and collaboration among theoreticians

And we have to find new ways of interfacing with experimentalists

ETSF is one player, ideally in a network of facilities (joint applications w/ synchrotrons...)
Advertising

• EU master’s : FAME master in functional materials http://www.fame-master.com/
• EU Erasmus mundus PhD program: http://www.idsfunmat.u-bordeaux1.fr/
• ETSF call for applications: http://www.etsf.eu/
• Next deadline: April 2011