PWscf's Graphical User Interface

PWgui == PWscf GUI

Cagliari, September 2005

Guido Fratesi

(mostly based on Anton Kokalj's work)

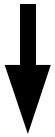
Copy me ...

- this tutorial can be downloaded from:

http://[ESPRESSO-TUTORIAL-SITE]/tutorial_pwgui.pdf

Basic scheme ...

prepare by **PWgui**



```
pw.x < stdin > stdout
```

analyze by **XCrySDen**

- perform SCF calculation:

PWgui

analyze by **XCrySDen**

```
pw.x < stdin > stdout
```

output: *prefix*.*

- calculate property:

PWgui

```
pp.x < stdin > stdout
```

output: *filplot*

- filter (transform) the data:

PWgui

analyze by **XCrySDen**

```
pp.x < stdin > stdout
```

output: *fileout*

PWgui == PWscf GUI

- is free software (GNU General Public License)
- WEB page:

  http://www.pwscf.org/, there is link to:

  http://www-k3.ijs.si/kokalj/pwgui/

PWgui and GUIB project

CONSIDER: inputs for numerical simulation software are simple from computer perspective

IDEA: construct a two-purpose meta-language:

- define the input syntax

  provide automatic GUI construction }

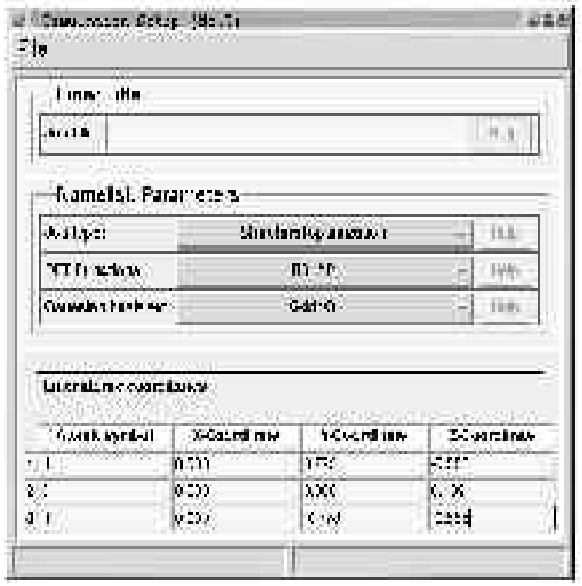
GUIB: simple **G**raphical **U**ser **I**nterface **B**uilder

`http://www-k3.ijs.si/kokalj/guib/`

GUI based on GUIB

- a GUI based on **GUIB** closely follows the structure of the input
- example:

```
Water molecule
 &parameters
   job_type    = optimization
   functional  = BLYP
   basis_set   = 6-31 G
 &end
Input_Geometry
H     0.000     0.774    -0.588
O     0.000     0.000     0.149
H     0.000    -0.774    -0.588
End
```



- the function of such GUI is to manage input files (creation + editing)

PWgui Installation

- **download `PWgui`** (stand-alone executable) from:

`http://www-k3.ijs.si/kokalj/pwgui/`

- **unzip or untar the file**
- **execute: `./pwgui`**

CVS version

- **define environment variables (for the bash shell):**
  - `export PWGUI=ESPRESSODIR/GUI/PWgui`
  - `export GUIB=ESPRESSODIR/GUI/Guib`
- **add to path:**
  - `export PATH=$PATH:$PWGUI`
- **execute: `pwgui`**

PWgui: what it provides?

- manage (create and edit) inputs for the following modules:
  - **pw.x**
  - **ph.x**
  - **pp.x**
  - **projwfc.x**
  - **ld1.x**
  - **d3.x**
- contain help:
  - User's manual
  - INPUT_* files
  - description of individual variables (**Help** buttons)
- visualization of structure: **PWgui** uses **XCRYSDEN**

Back to PWgui

- **let us start using PWgui !!!**
- launch: **`pwgui`**

PWgui: create a new pw.x input

- select menu: `File->New Input ...->New PW.X Input`

- **PWgui** opens a new page and this page holds several pages:

  + one page per each ***namelist***

  + one page for CELL_PARAMETERS/ATOMIC_SPECIES/ATOMIC_POSITIONS cards

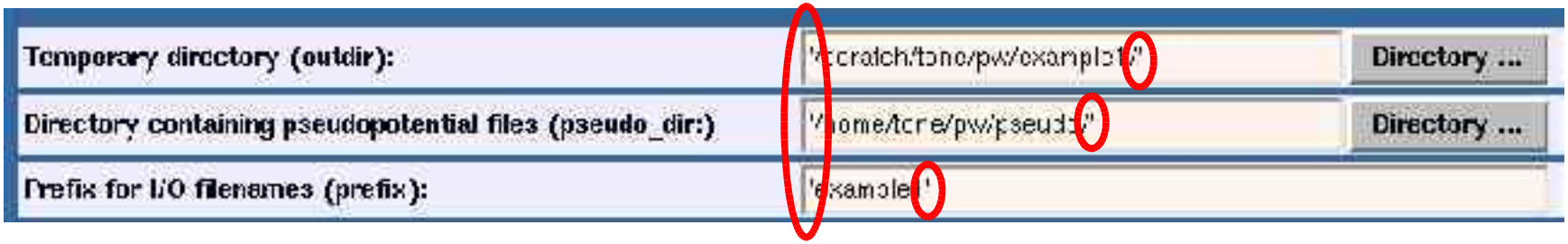
  + one page for K_POINTS card

  + one page for CLIMBING_IMAGES/OCCUPATIONS cards

PWgui: create a new pw.x input

- the input constructed by **PWgui** is syntactically correct, **BUT** the **STRINGS MUST BE QUOTED**. Example:

pseudo_dir = '/home/tone/pw/pseudo/'



PWgui: display help for pw.x

- description of variables:

  use **`Help`** buttons on the right

- description of whole pw.x input:

  select menu: **`Help->PW.X Input Syntax`**

- PWscf User's manual:

  select menu: **`Help->PWSCF User's guide`**

PWgui: event driven mechanism

- on ***Control*** page select:
  `Type of calculation` = **`Self-Consistent-field`**

- goto ***Ions*** page: all items are disabled

- on ***Control*** page select:
  `Type of calculation` = **`Ionic-relaxation`**

- goto ***Ions*** page: some items are enabled now.

- select a given type of ionic dynamics (first item): more items get enabled ...

PWgui: viewing input in text-mode

- on ***Control*** page select:
  `Type of calculation` = **`Self-Consistent-field`**
- now try the following menu items:
  - **`View->Input file`**

  (text layout of appropriate pw.x's input is displayed)

- now try input-layouts for other types of calculations ...

PWgui: editor vs. GUI mode

- from menu: **`File->Open Input ...->Open PW.X Input`**
- select file:
  `YOUR_ESPRESSO_DIR``/examples/example03/results/al001.rx.in`

  (NOTE: the input files are there after example03 has been run)
- now try the following menu items:
  - **`Edit->Input with editor`**
  - **`Edit->Input's copy with editor`**

PWgui & XCRYSDEN: visualization

- try menu: **`View->Structure with XCRYSDEN`**
  (XCRYSDEN will display structure)
- **Note: XCrySDen should be installed first. See the following tutorial**

- then select the **PWgui** menu: **`File->Settings`**
  - on ***PWgui settings*** page select:
    - **`launch XCRYSDEN`** = **`in notebook page`**

- retry the menu: **`View->Structure with XCRYSDEN`**
  (XCRYSDEN will appear inside PWgui as a new notebook page)

PWgui: input error checking

- select menu: **Edit->Input with editor**

- do an error on purpose, for example, add an undefined variable:

```
&CONTROL
   my_var = 'my_value',
   calculation = 'relax',
   ...
```

- save the file and exit from editor: PWgui will complain **!!!**

- **Message:** when PWgui complains about input, then the input probably contains syntax errors !!!

More info on Installation

- **PWgui** is written in [incr Tcl], which is a scripting language:
  - **ADVANTAGE:** no compilation
  - **DISADVATAGE:** **source**-package requires [incr Tcl] and related software

- How to install [incr Tcl] and related software:
  - Compile sources:
    - For Tcl/Tk see: http://www.scriptics.com/
    - For ITcl/Itk/Iwidgets see: http://incrtcl.sourceforge.net/
  - Binaries (for a few platforms only)
    - ActiveTcl (contains everything): http://aspn.activestate.com/ASPN/Tcl

PWgui: about modules

- modules are defined in `$PWGUI/modules` directory
- each module is in its own directory
- example: **pw.x** module:
  - located in `pw/` subdirectory
  - files:
    - » `pw.tcl` — definition of input syntax and GUI
    - » `pw-event.tcl` — event driven mechanism
    - » `pw-help.tcl` — help for variables
    - » `commands.tcl` — various functions for GUI

1st PWscf(pw.x) example

- create a simple **pw.x's** input for a **SCF** calculation (use **PWgui**)
- ***structure:*** **`Si bulk`**
  - *lattice parameter:* **`10.2 Bohr`**
  - *Braivas lattice:* **`fcc-cubic`**
  - *cutoff energy:* **`18.0 ryd`**
  - *pseudopotential:* **`Si.vbc.UPF`**
  - *atomic positions:*
    `Si 0.00 0.00 0.00`
    `Si 0.25 0.25 0.25`
  - *k-point mesh:* **`4x4x4`**

1st PWscf(pw.x) example

- then ... run a calculation via Run-->... menu(s).
  [*use the menu* File-->Settings (*tab:* PWscf settings) *to define the PWscf executables*]

- Run also from command-line (say we saved the file as **`si.scf.in`**):
  - **`pw.x < si.scf.in > si.scf.out`**