PWgui tutorial: a PWscf's Graphical User Interface

PWgui == PWscf GUI

Copy me ...

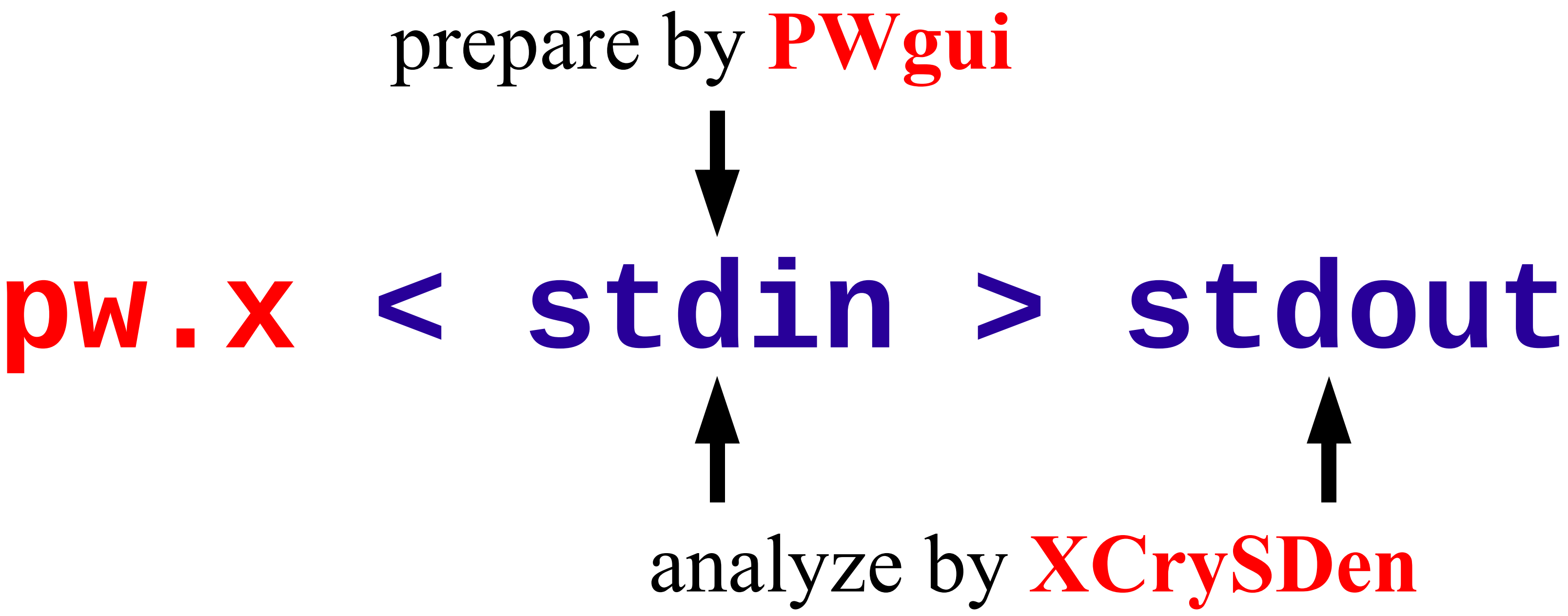
- This tutorial can be downloaded from:

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

- **PWgui** (linux binary package) can be copied from:

  `http://[ESPRESSO-TUTORIAL-SITE]/pwgui-linux-x86.zip`

Basic scheme @ PWscf

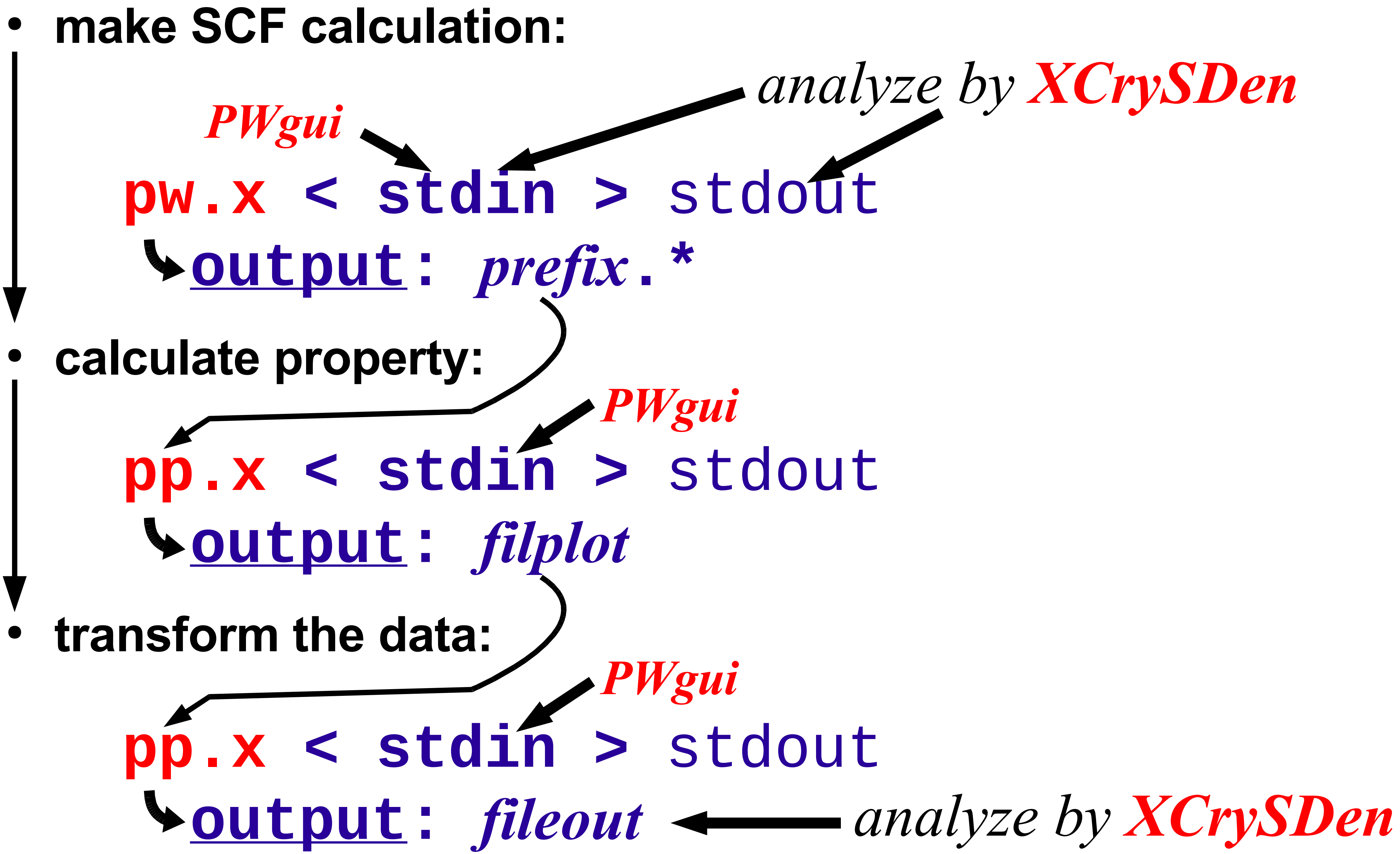


prepare by **PWgui**

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

analyze by **XCrySDen**

Basic scheme @ PWscf [extended]



- make SCF calculation:

PWgui

analyze by XCrySDen

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

output: *prefix*.*

- calculate property:

PWgui

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

output: *filplot*

- transform the data:

PWgui

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

output: *fileout* ← *analyze by XCrySDen*

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

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/`

What is GUIB

***GUIB* is an engine for building GUIs for numerical simulation software:**

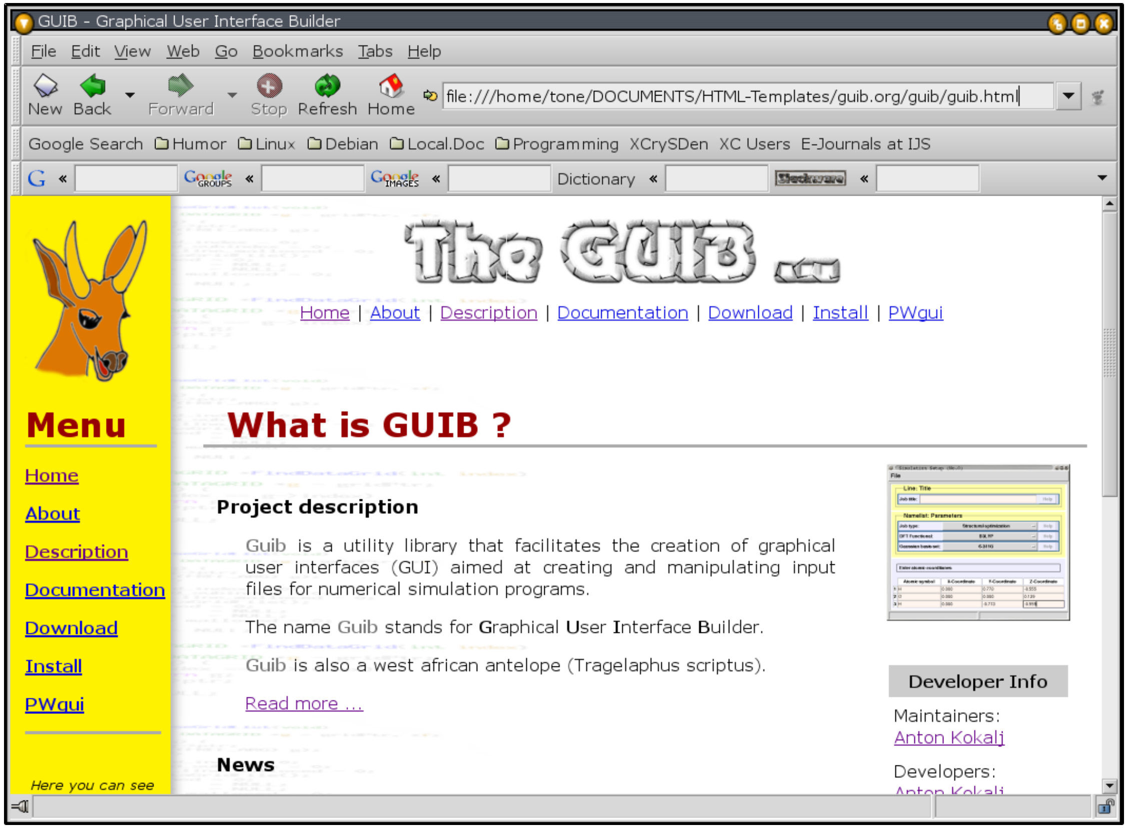
- ✓ *define the input syntax of your input*
- ✓ *automatic GUI construction*

GUIB *is free software (GPL)*

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

GUIB *engine is used for*

PWgui* – *a GUI for the PWscf set of programs


GUIB - Graphical User Interface Builder
File Edit View Web Go Bookmarks Tabs Help
New Back Forward Stop Refresh Home
file:///home/tone/DOCUMENTS/HTML-Templates/guib.org/guib/guib.html
Google Search Humor Linux Debian Local.Doc Programming XCrySDen XC Users E-Journals at IJS
Google Groups Google Images Dictionary
The GUIB
Home | About | Description | Documentation | Download | Install | PWgui
Menu
Home
About
Description
Documentation
Download
Install
PWgui
Here you can see
Developer Info

What is GUIB ?

Project description

Guib is a utility library that facilitates the creation of graphical user interfaces (GUI) aimed at creating and manipulating input files for numerical simulation programs.

The name Guib stands for Graphical User Interface Builder.

Guib is also a west african antelope (Tragelaphus scriptus).

Read more ...

News

Maintainers:
Anton Kokalj

Developers:
Anton Kokalj

How GUIB works

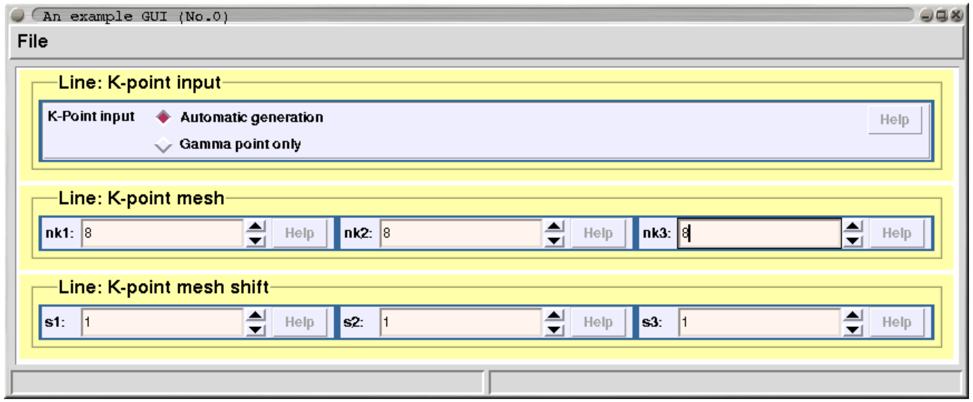
(1) *Consider the following input:*

```
K_POINTS automatic
8 8 8
1 1 1
```

(2) *Define the input syntax:*

```
module example1\#auto -title "An example GUI" -script {

  # 1st-line of input

  line ktype -name "K-point input" {

      keyword kpoints K_POINTS

      var kpoint_type {
          -label       "K-Point input"
          -textvalue { "Automatic generation" "Gamma point only" }
          -value     { automatic gamma }
          -widget     radiobox
      }
  }

  # 2nd-line of input

  line kmesh -name "K-point mesh" {

      packwidgets left

      var nk1  -label "nk1:"  -widget entry  -validate posint  -default 1

      var nk2  -label "nk2:"  -widget entry  -validate posint  -default 1

      var nk3  -label "nk3:"  -widget entry  -validate posint  -default 1
  }

  # 3rd-line of input

  line kshift -name "K-point mesh shift" {

      packwidgets left

      var s1 -label "s1:"  -widget entry  -validate posint  -default 1

      var s2 -label "s2:"  -widget entry  -validate posint  -default 1

      var s3 -label "s3:"  -widget entry  -validate posint  -default 1
  }
}
```

(3) *Corresponding GUI [automatically constructed on the basis of definition (2)]*



An example GUI (No.0)

Line: K-point input

K-Point input ◆ Automatic generation ◇ Gamma point only

Line: K-point mesh

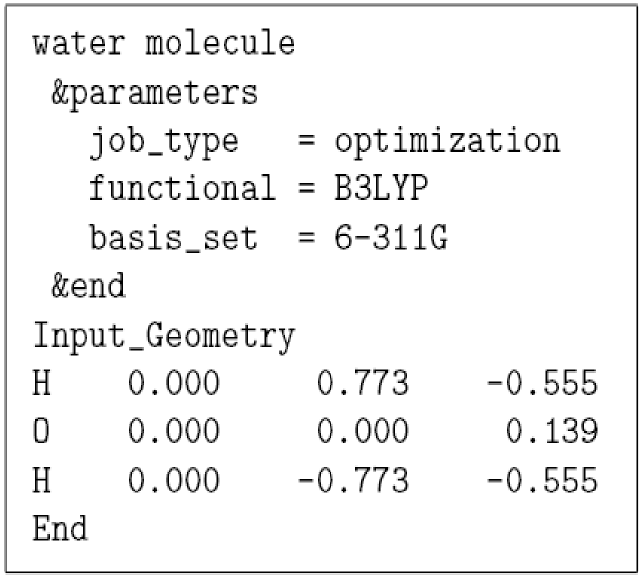
nk1: 8 nk2: 8 nk3: 8

Line: K-point mesh shift

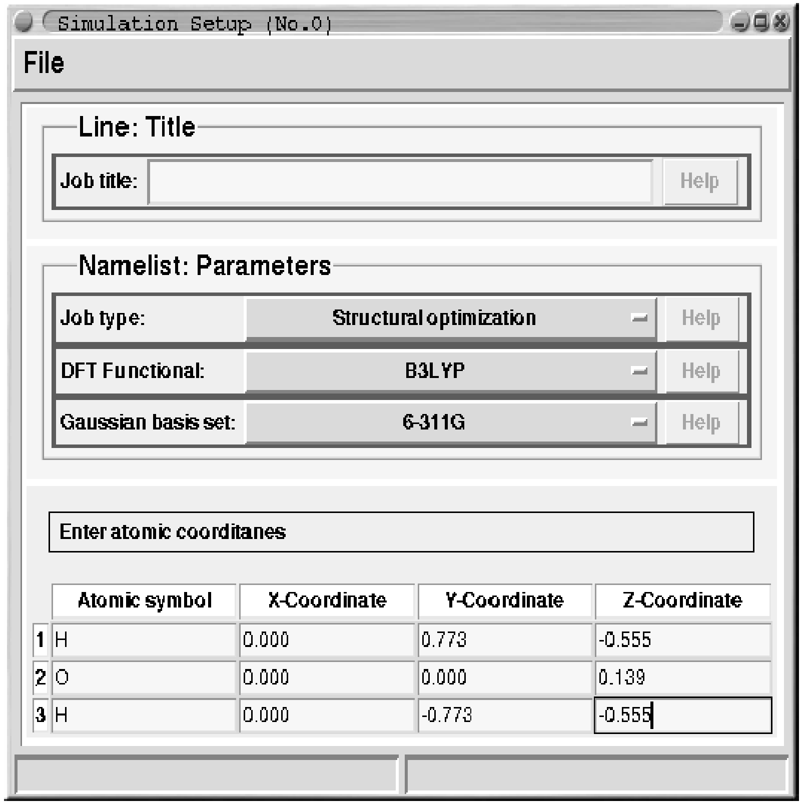
s1: 1 s2: 1 s3: 1

GUI based on GUIB

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



```
water molecule
 &parameters
   job_type    = optimization
   functional = B3LYP
   basis_set   = 6-311G
 &end
Input_Geometry
H      0.000      0.773     -0.555
O      0.000      0.000      0.139
H      0.000     -0.773     -0.555
End
```



Simulation Setup (No.0)

Line: Title

Job title:

Namelist: Parameters

Job type: Structural optimization

DFT Functional: B3LYP

Gaussian basis set: 6-311G

Enter atomic cooditanes

	Atomic symbol	X-Coordinate	Y-Coordinate	Z-Coordinate
1	H	0.000	0.773	-0.555
2	O	0.000	0.000	0.139
3	H	0.000	-0.773	-0.555

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

PWgui installation

- ***PWgui** comes in two flavors:*
  - binary standalone package (for Linux / MAC OSX)
  - source package
- *Installation of binary (standalone) package:*
  - download PWgui from **`http://www.pwscf.org/`** or **`http://www-k3.ijs.si/kokalj/pwgui/`**
  - unzip (or untar) the package
  - execute: **`./pwgui`**
- *Installation of source package:*
  - define environmental variables
    - **`export PWGUI=/path/to/PWgui-3.1`** (syntax for bash shell)
    - add **`$PWGUI`** to path: **`PATH=$PATH:$PWGUI`**
    - execute: **`./pwgui`**

PWgui: what it provides?

- manages (create and edit) inputs for the following modules:

  `pw.x`

  `ph.x`

  `pp.x`

  `projwfc.x`

  `ld1.x (atomic)`

  `d3.x`

- contains help:
  - User's manual
  - `INPUT_*` files
  - description of individual variables (**Help** buttons)
- visualization of structure: **PWgui** uses **XCRYSDEN**

Let's start using PWgui

- launch: `./pwgui`

create **new** input

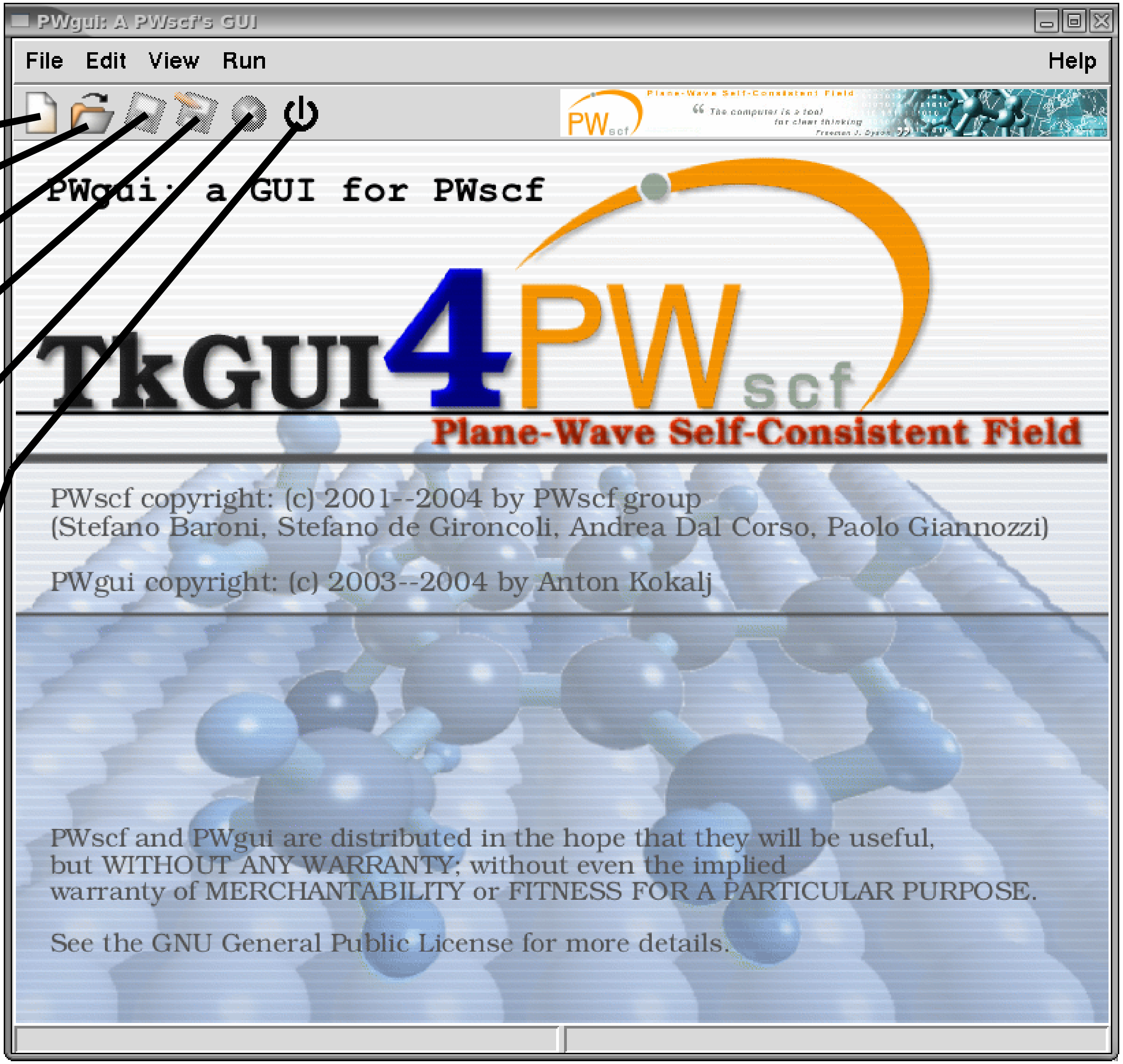
open existing input

save current input

save current input **as**

close current page

exit


PWgui: A PWscf's GUI
File
Edit
View
Run
Help
PW scf
Plane-Wave Self-Consistent Field
The computer is a tool for clear thinking
Freeman J. Dyson
TkGUI4PWscf
Plane-Wave Self-Consistent Field

PWgui: a GUI for PWscf

PWscf copyright: (c) 2001--2004 by PWscf group
(Stefano Baroni, Stefano de Gironcoli, Andrea Dal Corso, Paolo Giannozzi)

PWgui copyright: (c) 2003--2004 by Anton Kokalj

PWscf and PWgui are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details.

Create a new pw.x input file

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

- **PWgui** opens a new page, which contains several pages:

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

  + one page for **`CELL_PARAMETERS/ATOMIC_SPECIES/ATOMIC_POSITIONS`** cards

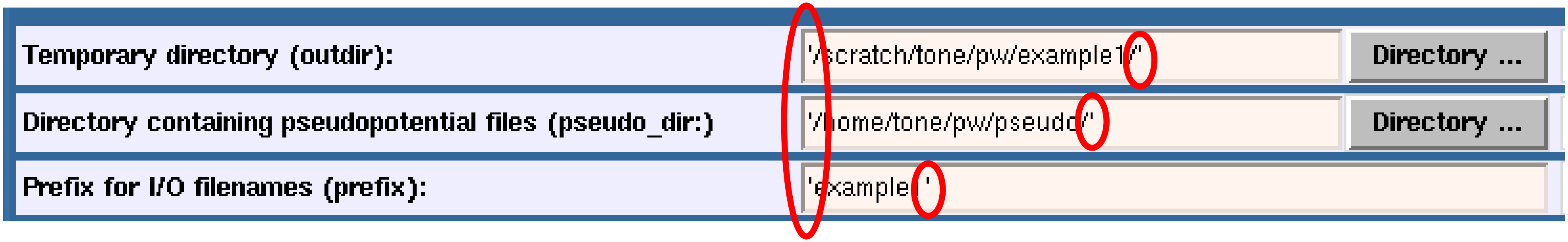
  + one page for **`K_POINTS`** card

  + one page for **`CLIMBING_IMAGES/OCCUPATIONS`** cards

Create a new pw.x input file

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

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



Temporary directory (outdir):	'/scratch/tone/pw/example1/'	Directory ...
Directory containing pseudopotential files (pseudo_dir:)	'/home/tone/pw/pseudo/'	Directory ...
Prefix for I/O filenames (prefix):	'example1'	

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`

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 ...

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 ...

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 other tutorial**)

- **xcrysden** in **PWgui**'s page:
  - select the **PWgui**'s 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)

Input error checking

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

- make 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 about 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/
  - Tcl binaries (for a few platforms only)
    - ActiveTcl (contains everything):
      http://www.activestate.com/Products/Download/Download.plex?id=ActiveTcl

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 pw.x example by PWgui

- create a simple pw.x' input for an **SCF** calculation with the **PWgui**
- **structure:** **`Si bulk`**
  - *lattice parameter:* **`10.2 Bohr`**
  - *Braivas lattice:* **`fcc-cubic`**
  - *cutoff energy:* **`18.0 ryd`**
  - *pseudopotential:* **`Si.vbc.UPF`**
  - *atomic positions (crystal units):*

    ```
    Si 0.00 0.00 0.00
    Si 0.25 0.25 0.25
    ```

  - *k-point mesh:* **`4x4x4`**

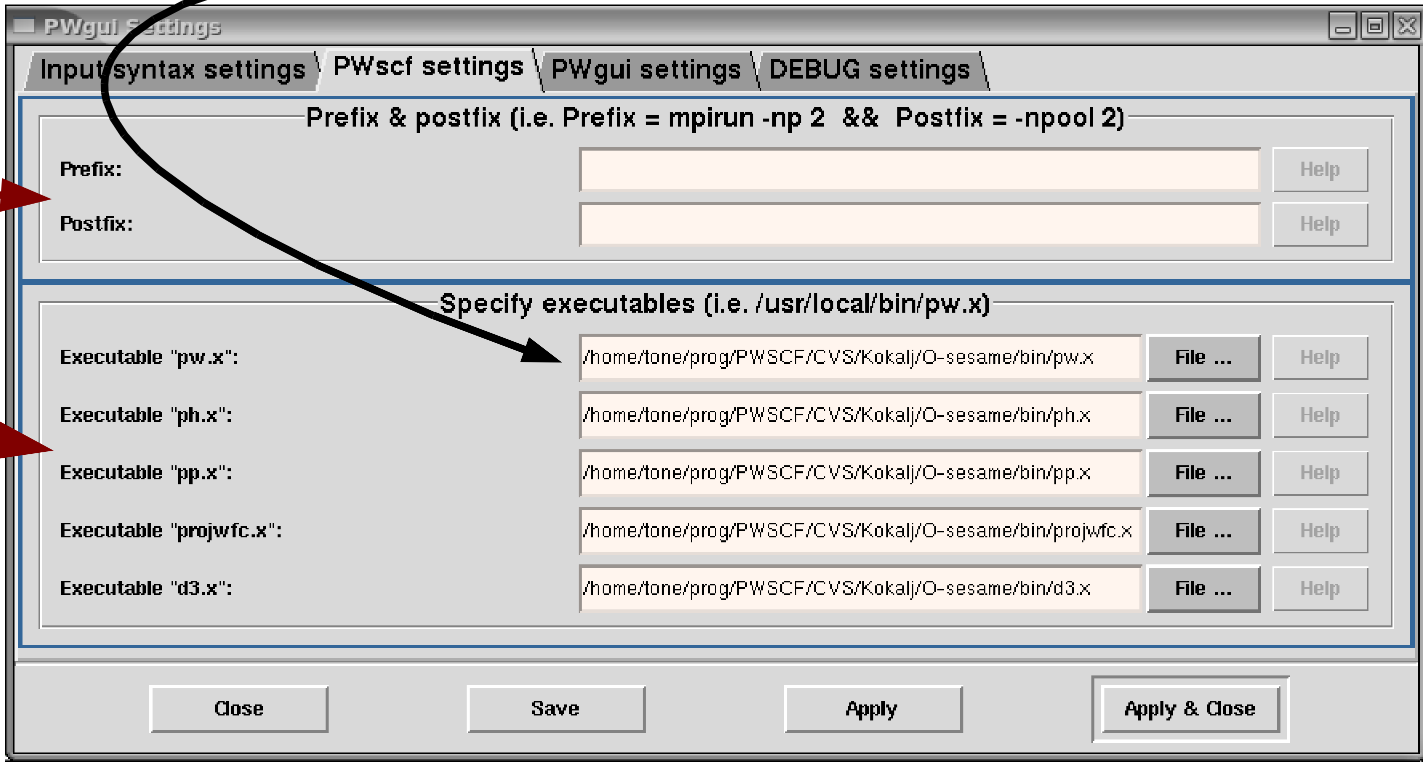
Define PWscf executables

- select the **PWgui**'s menu: **`File->Settings`**
  - a new window will appear, select page: **`PWscf settings`**

Example: **`/proper/path/to/pw.x`**

these are used for parallel execution

define PWscf executables here: set the proper path



Prefix & postfix (i.e. Prefix = mpirun -np 2 && Postfix = -npool 2)

Prefix:

Postfix:

Specify executables (i.e. /usr/local/bin/pw.x)

Executable "pw.x": /home/tone/prog/PWSCF/CVS/Kokalj/O-sesame/bin/pw.x

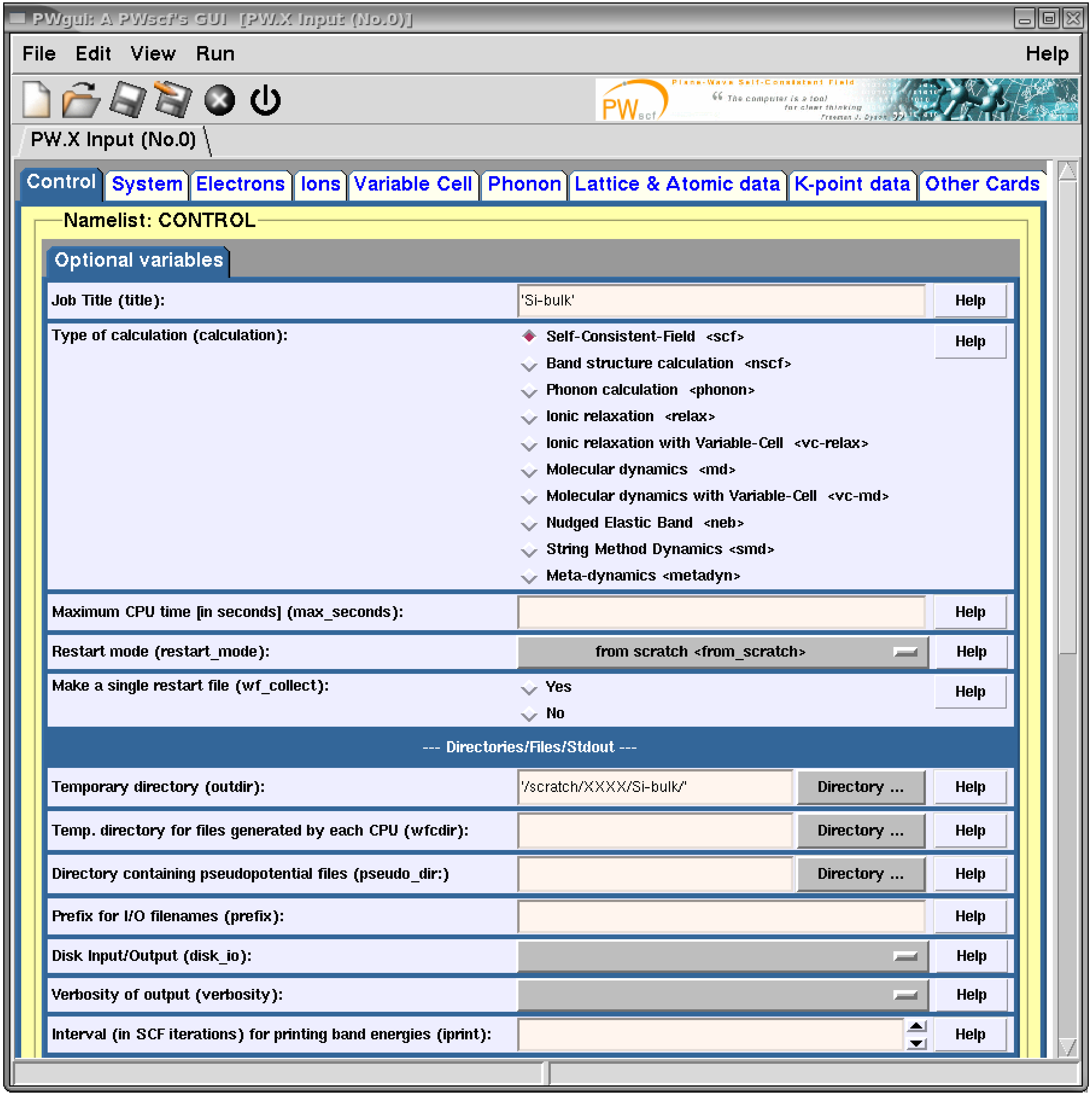
Executable "ph.x": /home/tone/prog/PWSCF/CVS/Kokalj/O-sesame/bin/ph.x

Executable "pp.x": /home/tone/prog/PWSCF/CVS/Kokalj/O-sesame/bin/pp.x

Executable "projwfc.x": /home/tone/prog/PWSCF/CVS/Kokalj/O-sesame/bin/projwfc.x

Executable "d3.x": /home/tone/prog/PWSCF/CVS/Kokalj/O-sesame/bin/d3.x

Page: Control


PWgui: A PWscf's GUI [PW.X Input (No.0)]
File
Edit
View
Run
Help
PWscf
Plane-Wave Self-Consistent Field
" The computer is a tool for clear thinking Freeman J. Dyson "
PW.X Input (No.0)
Control
System
Electrons
Ions
Variable Cell
Phonon
Lattice & Atomic data
K-point data
Other Cards

Namelist: CONTROL

Optional variables

Job Title (title): 'Si-bulk'

Type of calculation (calculation):
- ● Self-Consistent-Field <scf>
- ○ Band structure calculation <nscf>
- ○ Phonon calculation <phonon>
- ○ Ionic relaxation <relax>
- ○ Ionic relaxation with Variable-Cell <vc-relax>
- ○ Molecular dynamics <md>
- ○ Molecular dynamics with Variable-Cell <vc-md>
- ○ Nudged Elastic Band <neb>
- ○ String Method Dynamics <smd>
- ○ Meta-dynamics <metadyn>

Maximum CPU time [in seconds] (max_seconds):

Restart mode (restart_mode): from scratch <from_scratch>

Make a single restart file (wf_collect):
- ○ Yes
- ○ No

--- Directories/Files/Stdout ---

Temporary directory (outdir): '/scratch/XXXX/Si-bulk/'

Temp. directory for files generated by each CPU (wfcdir):

Directory containing pseudopotential files (pseudo_dir:)

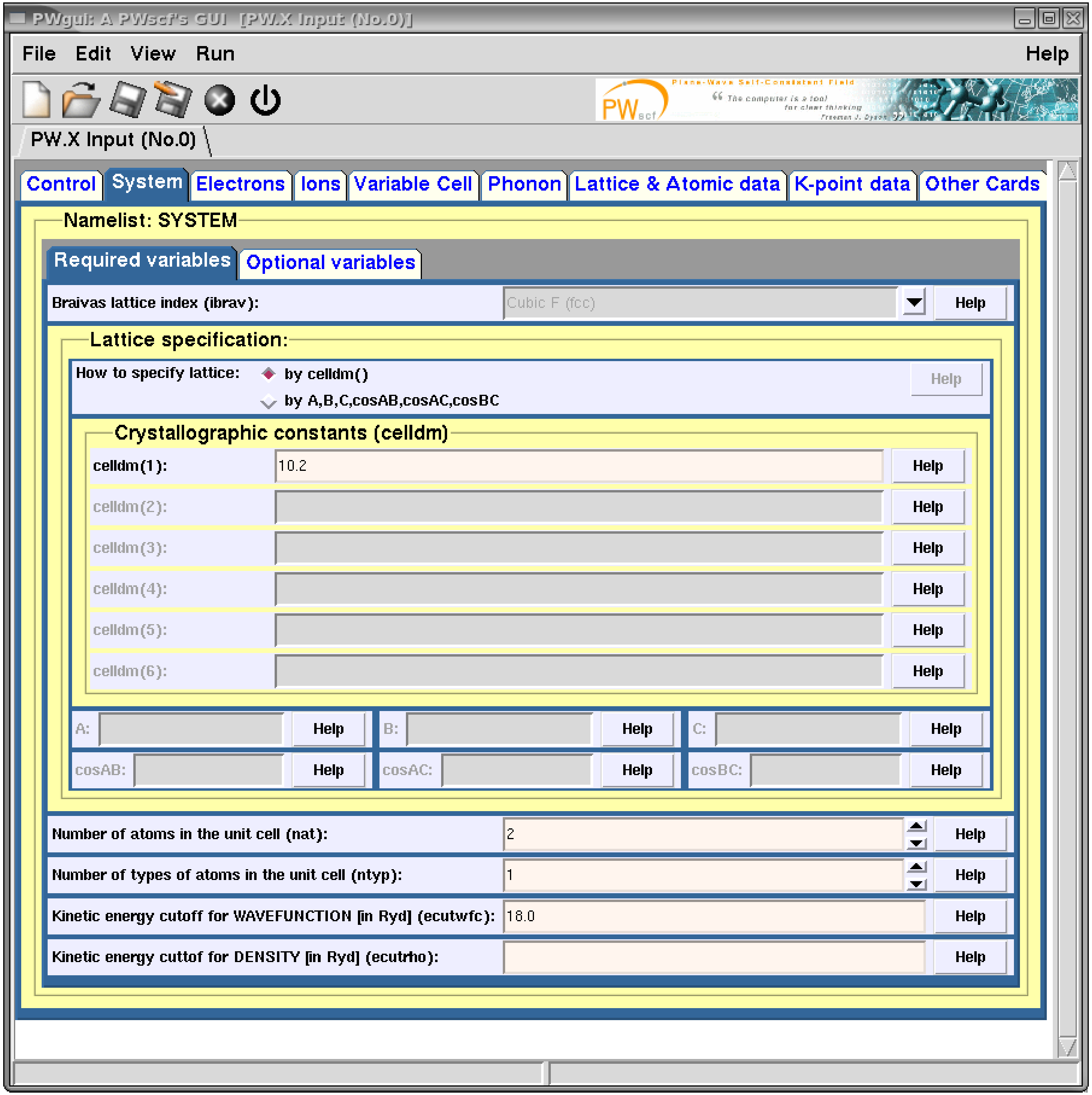
Prefix for I/O filenames (prefix):

Disk Input/Output (disk_io):

Verbosity of output (verbosity):

Interval (in SCF iterations) for printing band energies (iprint):

Page: System



Namelist: SYSTEM

Braivas lattice index (ibrav): Cubic F (fcc)

Lattice specification:

How to specify lattice: ● by celldm() ○ by A,B,C,cosAB,cosAC,cosBC

Crystallographic constants (celldm)

celldm(1): 10.2
celldm(2):
celldm(3):
celldm(4):
celldm(5):
celldm(6):

A:
B:
C:
cosAB:
cosAC:
cosBC:

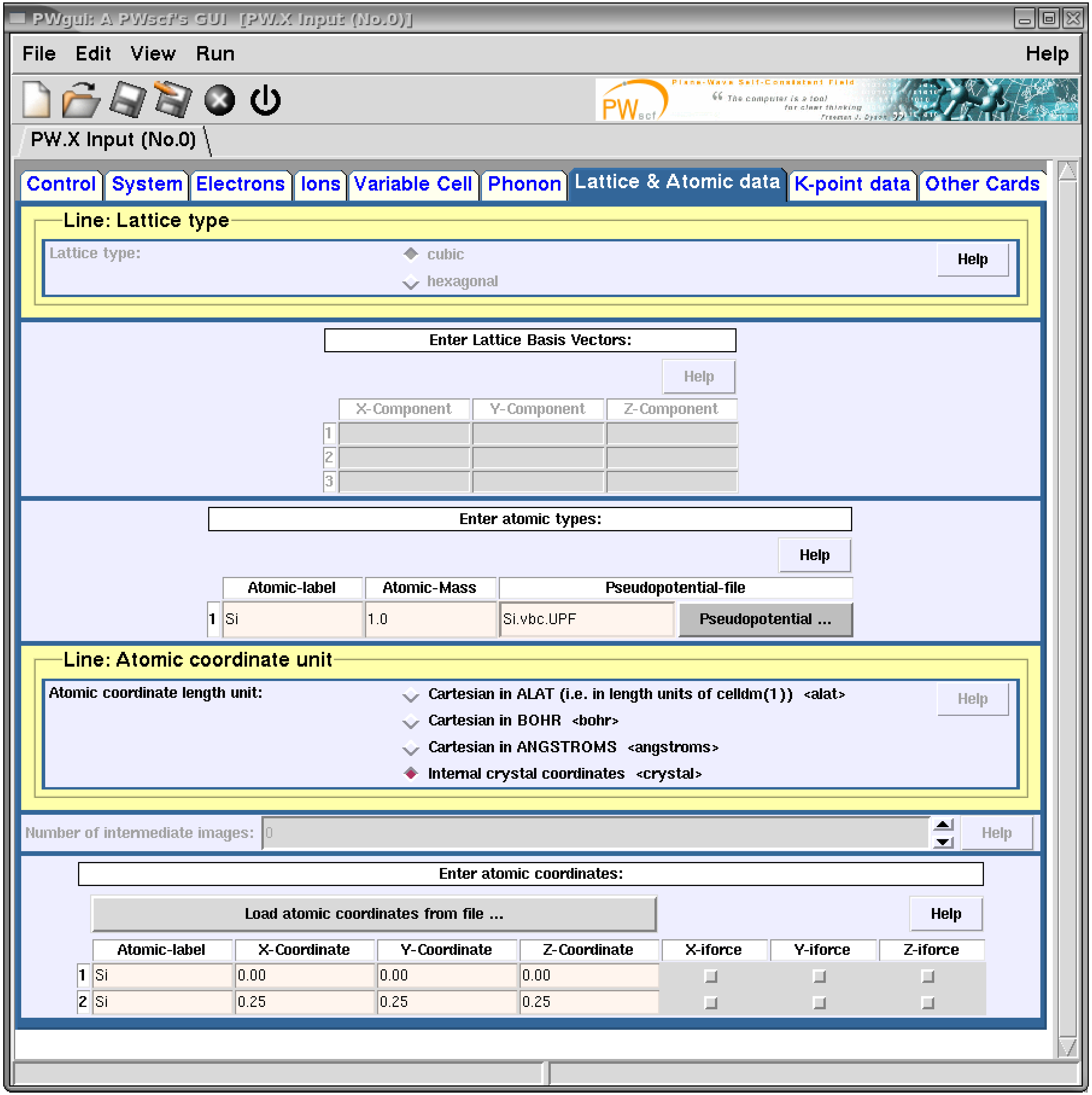
Number of atoms in the unit cell (nat): 2

Number of types of atoms in the unit cell (ntyp): 1

Kinetic energy cutoff for WAVEFUNCTION [in Ryd] (ecutwfc): 18.0

Kinetic energy cuttof for DENSITY [in Ryd] (ecutrho):

Page: Lattice & Atomic data


PWgui: A PWscf's GUI [PW.X Input (No.0)]
File
Edit
View
Run
Help
PWscf Plane-Wave Self-Consistent Field
"The computer is a tool for clear thinking" Freeman J. Dyson
PW.X Input (No.0)
Control
System
Electrons
Ions
Variable Cell
Phonon
Lattice & Atomic data
K-point data
Other Cards

Line: Lattice type

Lattice type: ◆ cubic ◇ hexagonal

Enter Lattice Basis Vectors:

	X-Component	Y-Component	Z-Component
1			
2			
3			

Enter atomic types:

	Atomic-label	Atomic-Mass	Pseudopotential-file	
1	Si	1.0	Si.vbc.UPF	Pseudopotential ...

Line: Atomic coordinate unit

Atomic coordinate length unit:

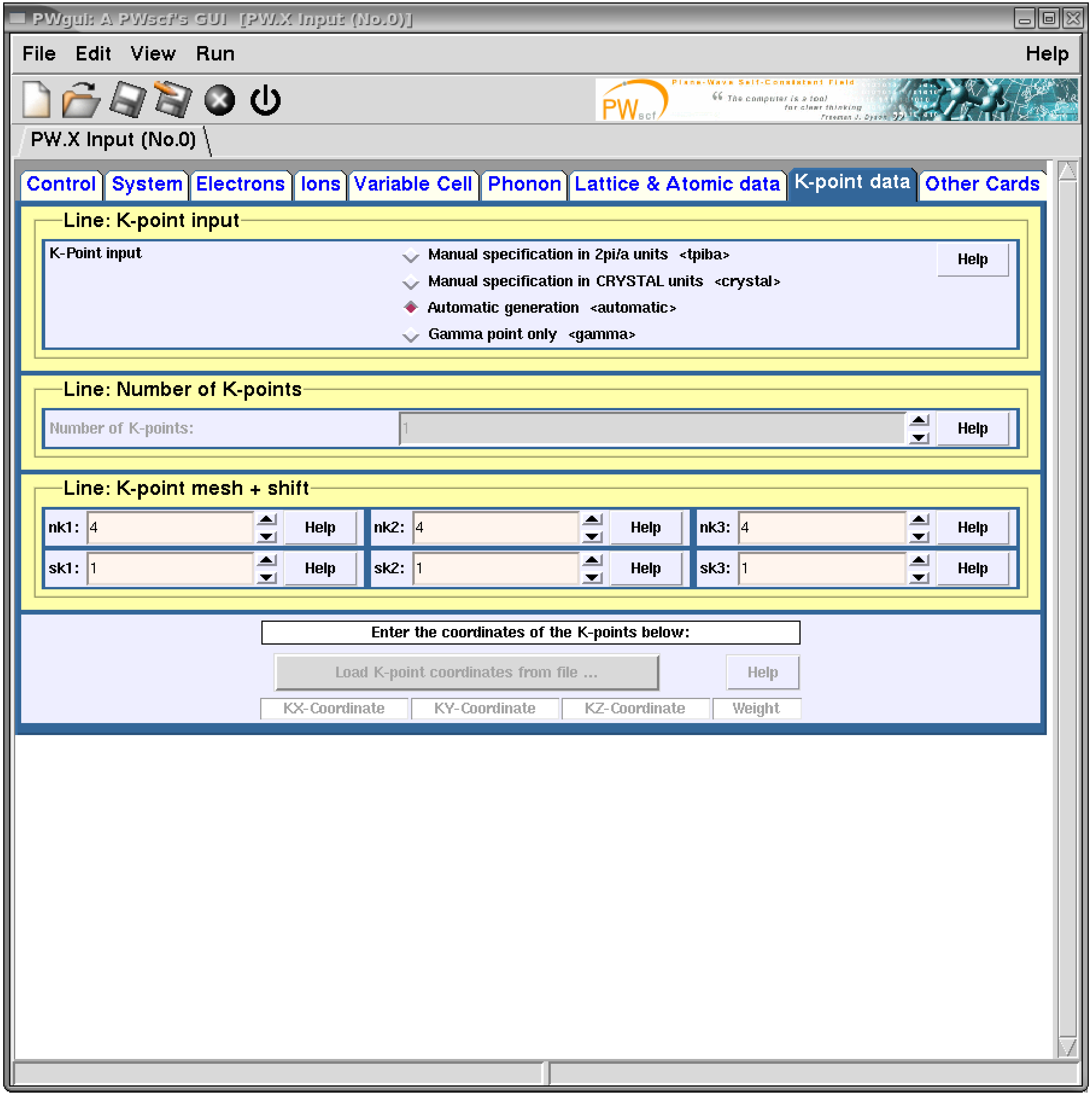
- ◇ Cartesian in ALAT (i.e. in length units of celldm(1)) <alat>
- ◇ Cartesian in BOHR <bohr>
- ◇ Cartesian in ANGSTROMS <angstroms>
- ◆ Internal crystal coordinates <crystal>

Number of intermediate images: 0

Enter atomic coordinates:

	Atomic-label	X-Coordinate	Y-Coordinate	Z-Coordinate	X-iforce	Y-iforce	Z-iforce
1	Si	0.00	0.00	0.00	□	□	□
2	Si	0.25	0.25	0.25	□	□	□

Page: K-point data



Line: K-point input

K-Point input
○ Manual specification in 2pi/a units <tpiba>
○ Manual specification in CRYSTAL units <crystal>
● Automatic generation <automatic>
○ Gamma point only <gamma>

Line: Number of K-points

Number of K-points: 1

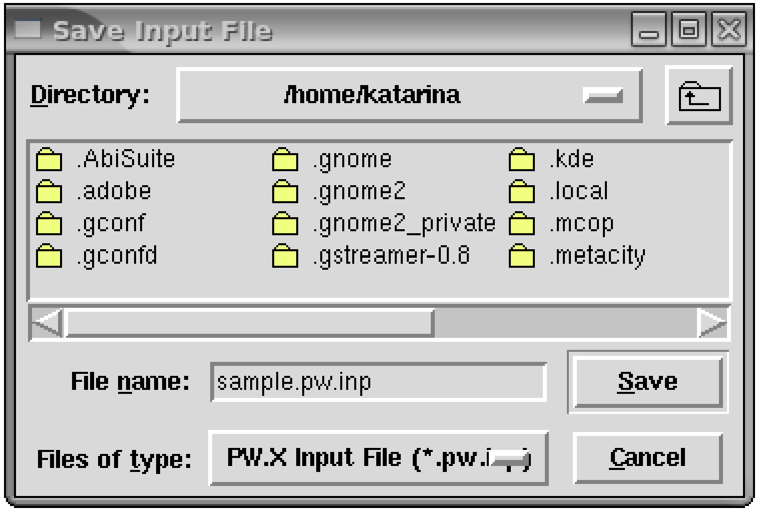
Line: K-point mesh + shift

nk1: 4	nk2: 4	nk3: 4
sk1: 1	sk2: 1	sk3: 1

Enter the coordinates of the K-points below:

Visualize and run

- to see the input file select menu item:
  View -> Input File
- to visualize the structure select menu item:
  View -> Structure with XCrySDen
- to run calculation use, for example, menu item:
  Run -> Run calculation
  - **PWgui** will ask you to save the input:



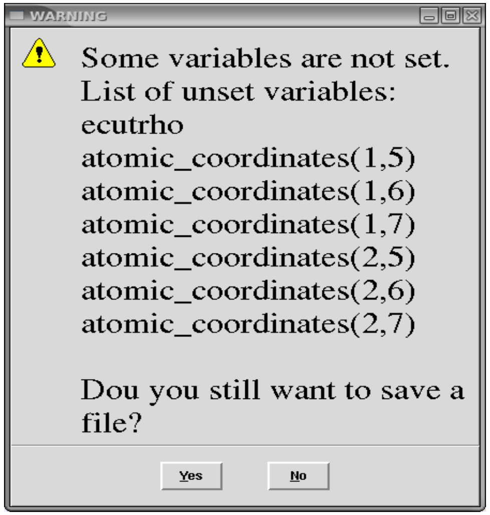
Directory: /home/katarina

.AbiSuite .adobe .gconf .gconfd .gnome .gnome2 .gnome2_private .gstreamer-0.8 .kde .local .mcop .metacity

File name: sample.pw.inp

Files of type: PW.X Input File (*.pw.i

  - ignore this message:



Some variables are not set.
List of unset variables:
ecutrho
atomic_coordinates(1,5)
atomic_coordinates(1,6)
atomic_coordinates(1,7)
atomic_coordinates(2,5)
atomic_coordinates(2,6)
atomic_coordinates(2,7)

Dou you still want to save a file?