The PSPIO library

Micael Oliveira

Center for Computational Physics - Universidade de Coimbra

Octopus Developers Meeting 2012

Lyon, October 23, 2012

CENTRO DE FÍSICA COMPUTACIONAL

Why a pseudopotential I/O library?

- Routines to parse pseudopotential data in DFT codes are a classic example of code duplication.
- Most codes are only able to read or write a small number of file formats.
- Some file formats specifications are ambiguous or ill-defined, i.e., they are code dependent.

CENTRO DE FÍSICA COMPUTACIONAL

What should PSPIO do?

Input:

- Parse a pseudopotential file
- Store the pseudopotential data internally
- Provide routines to access specific chunks of the psp data

Output:

- Provide routines to set specific chunks of the psp data
- Store the pseudopotential data internally
- Write data to pseudopotential file

CENTRO DE FÍSICA COMPUTACIONAL

Design

- Written in C
- Autotools
- Error handling: always return control to main program if there is a problem with the file
- Doxygen documentation
- Fortran interface
- Testsuite
- Use atomic units
- Use Libxc id for internal representation of xc
- Return data either on the original grid or interpolated

CENTRO DE FÍSICA COMPUTACIONAL

Dependencies

- GSL
- Libxc (optional)

CENTRO DE FÍSICA COMPUTACIONAL

Coding Party

Initial implementation done in September 2011 in Coimbra:

- Four participants (J. Alberdi, M. Oliveira, Y. Pouillon, and M. Verstraete)
- More than 5000 lines of code written in 5 days
- Substantial amount of food and drinks consumed

CENTRO DE FÍSICA COMPUTACIONAL

What is working

- Data structures and corresponding methods.
- Error handling system.
- Autotools.
- Reading and writing of UPF files.
- Fortran interface.

- Interfaced with Octopus (pseudopotentials in non-local form only).

CENTRO DE FÍSICA COMPUTACIONAL

Near future

- Finish implementation of Abinit format 6.
- Implement Siesta format (psf).
- Octopus interface for pseudopotentials in semi-local form.
- Abinit interface to PSPIO.
- Several small improvements.

CENTRO DE FÍSICA COMPUTACIONAL

Not so near future

- Implement remaining formats.
- Pseudopotential output.
- Pseudopotential format conversion using APE + PSPIO.
- PAW datasets.

CENTRO DE FÍSICA COMPUTACIONAL

Main data structure

```
typedef struct{
  // General data
  char *info;          /**< Descriptive string for content of file read in.
                            Nothing should ever be assumed about its content. */
  char *symbol;        /**< Atomic symbol */
  double z;            /**< Atomic number */
  double zvalence;     /**< Charge of pseudopotential ion - valence electrons */
  double nelvalence;   /**< Number of electrons - normally equal to zion,
                            except for special cases for ions */
  int l_max;           /**< Maximal angular momentum channel */
  int wave_eq;         /**< Type of wave equation which was solved:
                            Dirac, Scalar Relativistic, or Schroedinger */
  double total_energy; /**< the total energy of the pseudo atom */

  // The radial mesh.
  pspio_mesh_t *mesh; /**< Radial mesh - all functions should be discretized on this mesh */

  // The states
  int **qn_to_istate;      /**< lookup table giving the position of the state
                                from the quantum numbers */
  int n_states;            /**< number of electronic states */
  pspio_state_t **states; /**< struct with electronic states */

  ...
```

CENTRO DE FÍSICA COMPUTACIONAL

Main data structure (cont.)

```
  ...

  // The pseudopotentials
  int scheme;                     /**< scheme used to generate the pseudopotentials */
  int n_potentials;               /**< number of pseudopotentials */
  pspio_potential_t **potentials; /**< struc with pseudopotentials */

  // Kleinman and Bylander non-local projectors
  int n_kbproj;                          /**< number of Kleinman and Bylander projectors */
  pspio_projector_t **kb_projectors; /**< Kleinman and Bylander projectors */
  int l_local;                           /**< angular momentum channel of local potential */
  int kb_l_max;                          /**< maximum angular momentum of KB projectors*/
  pspio_potential_t *vlocal;             /**< local potential for the KB form of the pseudopoten

  // Exchange and correlation data, including non-linear core corrections
  pspio_xc_t *xc; /**< xc structure */

  // Valence density
  pspio_meshfunc_t *rho_valence; /**< valence density */

} pspio_pspdata_t;
```

Example

```
program example_fortran
 use pspio_f90_types_m
 use pspio_f90_lib_m
 implicit none

 character(len=256) :: filename
 character(len=3) :: symbol
 integer :: ierr, l_max, np, ir
 real(8), allocatable :: r(:), mesh_func(:)
 type(pspio_f90_pspdata_t)    :: pspdata
 type(pspio_f90_mesh_t)       :: mesh
 type(pspio_f90_potential_t) :: vlocal

 ! Init pspio data structure and parse file
 ierr = pspio_f90_pspdata_init(pspdata)

 read(*,'(A)') filename
 ierr = pspio_f90_pspdata_read(pspdata, PSPIO_UNKNOWN, filename)
 if (ierr /= 0) then
   ierr = pspio_f90_error_flush()
   stop
 end if

 ! General info
 ierr = pspio_f90_pspdata_get_symbol(pspdata, symbol)
 ierr = pspio_f90_pspdata_get_l_max(pspdata, l_max)

 ...
```

CENTRO DE FÍSICA COMPUTACIONAL

Example (cont.)

```
  ...

  ! Mesh
  ierr = pspio_f90_pspdata_get_mesh(pspdata, mesh)
  ierr = pspio_f90_mesh_get_np(mesh, np)
  allocate(r(np))
  ierr = pspio_f90_mesh_get_r(mesh, r(1))

  ! Local potential
  ierr = pspio_f90_pspdata_get_vlocal(pspdata, vlocal)
  allocate(mesh_func(np))
  do ir = 1, np
    ierr = pspio_f90_potential_eval(vlocal, r(ir), mesh_func(ir))
  end do

  ierr = pspio_f90_pspdata_free(pspdata)

end program example_fortran
```

CENTRO DE FÍSICA COMPUTACIONAL

How to use PSPIO in Octopus

- Compile with PSPIO support
- New species type:

```
%Species
 "Ti" | 47.867 | spec_pspio | 22 | "Ti.UPF" | 2 | 0
%
```

CENTRO DE FÍSICA COMPUTACIONAL

Acknowledgments

- Joseba Alberdi
- Yann Pouillon
- Matthieu Verstraete

- Center for Computational Physics

CENTRO DE FÍSICA COMPUTACIONAL

Suggestions for other useful libraries

- Christallographic data
- Geometry optimization
- Units
- SCF mixing
- ?

CENTRO DE FÍSICA COMPUTACIONAL