

gPI code

C. J. F. Solano

April 6, 2017

The generalized perturbed ion (gPI) method [1] has been developed in the quantum chemistry group of Oviedo University (Spain). This quantum local method, which is based on the theory of separability of many-electrons system, allows for a description of ionic systems such as crystals, finite clusters and crystal defects or impurities.

A gPI run is driven through a set of commands which are specified in an input file. This file is composed by several lines of characters which are internally divided into keywords and variables. The input file is not subjected to any format, except that: (a) punctuation marks are not allowable so that different words should be separated at least by one blank space; and (b) each command should be written in a single line. Uppercase and lowercase letters can be used interchangeably, except when filenames are specified. Atomic units are used in both input and output files.

1 gPI commands

Some conventions are used in the description on the different commands. On the one hand, keywords are written in bold characters. On the other hand, data provided by the user are written in italic characters. Optional data are encapsulated between square brackets: [*option*]. Those data to be selected among several options are separated by vertical lines and encapsulated between parenthesis: (*option1|option2|...*).

1.1 General commands

gPI code has been implemented based on the earlier pi7r15 code. Thereby, gPI includes the functionalities and modules previously implemented in pi7r15, but many of them have been improved. In particular, several general commands have been preserved despite their loss of relevance. These commands, which are indicated by an asterisk (*), enable one to control one or several modules in an independent manner.

- **title** *runtitle*: Input title.
- **help**: Provide information about how run the code and most relevant commands.
- **rem** or **# (first column)**: Comment line.
- **print** *nimpre*: Input print level ($nimpre \geq -1$).
- **active** *ni* (*): Input ion type.
- **cell** *a b c α β γ* (*): Input cell distances (i.e., *a*, *b* and *c* in bohr) and angles (i.e., α , β and γ in degrees).

- **copy** *file1 file2*: Copy data from *file1* file into *file2* file.
- **end**: End of a gPI run.
- **hf**: Select HF level (default).
- **rchf**: Select rCHF level.
- **uchf**: Select uCHF level.
- **ion \sharp** *ni fileion fileulm filecoef* (*): The files *fileion*, *fileulm* and *filecoef* provide the input of the *ni* ion type (see Section 2).
- **int**> *fileint* (*): Store monocentric integrals for *ni* ion type in the *fileint* binary file.
- **int**< *fileint* (*): Read monocentric integrals for *ni* ion type from *fileint* binary file.
- **lat**< *fileint* (*): Read bicentric integrals for *ni* ion type from *fileint* binary file.
- **uint** *fileint filecint* (*): Store uncorrelated and correlated monocentric integrals for *ni* ion type in the *fileint* and *filecint* files, respectively.
- **mp**> *fileion* (*): Store \mathbf{A}^R and $P_{\Gamma pq}^R$ matrices [1] for *ni* ion type in the *fileion.vef* file.
- **vef** *fileion fileulm filecoef* (*): Combine **ion \sharp** and **mp**> commands into a single one.
- **scf**> *fileion* (*): Perform a SCF process for *ni* ion type and write final results in the *fileion* file.
- **edef** *fion fulm fcoef fionvac fulmvac fcoefvac*: Calculate the deformation energy [1]. The files *fionvac*, *fulmvac* and *fcoefvac* provide the ion description in vacuum, whereby the files *fion*, *fulm* and *fcoef* provide the ion description in the system (see Section 2).
- **set** *VAR value1 [value2]*: Modify value(s) for a control variable (see Table 1).
- **set** *PROJ value*: Input a projection operator. Options for *value*: (a) *TMSRH*; (b) *TOTHC*; (c) *GRPHC* (Höjer and Chung with a projection factor of 2); (d) *ABSHC* (Höjer and Chung with a projection factor of -2 and absolute value for orbital energy); and (e) *NOPROJ* for not using projection operators.
- **show**: Show values for internal parameters and control variables.
- **vacuo** *fileion fileulm filecoef (fileint|filecint|fileint filecint)*: Perform an exhaustive run for a single ion. To this end, it reads the files *fileion*, *fileulm* and *filecoef* which provide the input file of the ion type (see Section 2) and writes the final results in the *filecoef* file. HF and rCHF levels only require a single file for storing the monocentric integrals, i.e., *fileint* file for HF level and *filecint* file for rCHF level. Both files are required in the uCHF level.

Table 1: Control variables. Type of variable and default value(s) are given in the *value* column.

Variable (<i>VAR</i>)	<i>value</i>	description
MXXTRP	integer,100	Maximum number of SCF iterations
BIAS	real,-2	Initial threshold value for SCF convergence= $2^{BIAS} \times THREM$
THREM	real, 10^{-8}	Initial threshold value for SCF convergence= $2^{BIAS} \times THREM$
NXTRP	integer,15	Increase by a factor of 2 the threshold value for SCF convergence each NXTRP iterations
SCFDIV	real, 10^{-2}	Divergence criteria for a SCF process
RLASTSHELL	real,30.	Maximum distance in the last neighbors shell
PICONV	real, $3 \cdot 10^{-8}$	Convergence required in the gPI cycles for energy
MAXPICYCLES	integer,100	Maximum number of gPI cycles
PIDIV	real,10.	Stop a gPI run if energy variation is greater than PIDIV in a single cycle
CON1 0 – 86	real,0.028816	c_1 value of the Coulomb hole for a given atom number (or all of them if omitted)
CON2 0 – 86	real,1.1	c_1 value of the Coulomb hole for a given atom number (or all of them if omitted)
EPSDIS	real, 10^{-6}	Maximum error allowable in the distances (in a_0)
EPSFRAC	real, 10^{-7}	Maximum error allowable in the crystallographic coordinates (in a_0)
DAMP	real,0.	Orbital coefficients are given by DAMP-times the initial coefficients plus (1-DAMP)-times the convergent coefficients ($0 \leq DAMP \leq 1$)
EWCONV	real, 10^{-9}	Convergence required in the Ewald summation
COVNSR	real, 10^{-7}	Convergence required in the lattice summation for the short-range nuclear potential
COVNL	real, 10^{-7}	Convergence required in the lattice summation for the short-range local potential
CONVESR	real, 10^{-7}	Convergence required in the lattice summation for the short-range Coulomb potential
CONVXP	real, 10^{-7}	Convergence required in the lattice summation for exchange and projection matrices
CONVINT	real, 10^{-7}	Convergence required in the lattice summation for all short-range terms
SYMMETRIZE	boolean,.false.	Symmetrize the interaction energies if this variable is .true.

1.2 Crystal commands

Module specialized in gPI calculations of pure 3D crystals.

- **crystal**: Start of the crystal module.
- **title** *runtitle*: Input title.
- **rem** or **#** (**first column**): Comment line.
- **cell** *a b c α β γ* : Input cell distances (i.e., *a*, *b* and *c* in bohr) and angles (i.e., α , β and γ in degrees).
- **spg** *spacegroup*: Input space group symbol according to the international Hermann–Mauguin notation. Some conventions should be used: (a) each different element in the symbol must be separated by at least one blank character; (b) characters in a symmetry element shouldn't be separated by blanks; (c) subindices are input with no blanks separating them from the indexed character; (d) upperscore is represented by $-$ preceding to the element; (e) letter case is irrelevant; and (f) a trailing R means use rhombohedral unit cell for trigonal rhombohedral space groups (those with R Bravais lattice). Examples: (i) enter *Fm3m* as “f m 3 m”; (ii) enter *P2₁2₁2₂* as “p 21 21 21”; (iii) enter *P4/mmm* as “p 4/m m m”; and (iv) enter *R3c* as “r -3 c” or “r -3 c r” (rhombohedral cell).
- **neq** *x y z fileion fileulm filecoef (fileint|filecint|fileint filecint) filelint*: Input with the description of every symmetry different ion contained in the primitive cell. Use a different **neq** command for each crystallographic different ion. $0 \leq x, y, z < 1$ are the crystallographic coordinates of the ion. The files *fileion*, *fileulm* and *filecoef* provide the input of the ion (see Section 2). The *fileint* file store the uncorrelated monocentric integrals for HF calculation. The *filecint* file stores the correlated monocentric integrals for a rCHF calculation. A uCHF calculation uses both *fileint* and *filecint* files. The *filelint* file stores the bicentric integrals.
- **fneq** *x y z xocup fileion fileulm filecoef (fileint|filecint|fileint filecint) filelint*: Analogous to the **neq** command except that fractional occupancies *xocup* of the crystallographic positions are allowed in order to treat crystal structures only known as an statistical average.
- **endcrystal**: End of crystal module. Do the calculation and return to the general shell.

1.3 Cluster commands

Module specialized in gPI calculations of finite size periodic clusters.

- **cluster**: Start of the cluster module.
- **title** *runtitle*: Input title.
- **rem** or **#** (**first column**): Comment line.
- **cell** *a b c α β γ* : Input cell distances (i.e., *a*, *b* and *c* in bohr) and angles (i.e., α , β and γ in degrees).
- **spg** *spacegroup*: Input space group symbol according to the international Hermann–Mauguin notation. The same conventions as in **crystal** module. This command is used to generate the point symmetry operations in combination with **center** command.

- **center** $x y z$: Input crystallographic coordinates (x, y, z) for the invariant point under symmetry operations. The point symmetry group is composed by all those operations from the spatial symmetry group (see **spg** command) that do not modify the position of the invariant point.
- **neqclus** $x y z fileion fileulm filecoef (fileint|filecint|fileint filecint) filelint$: Input the position (x, y, z) of a non-equivalent ion that forms the cluster. Use a different **neqclus** command for each different non-equivalent ion. The position is not restricted to be inside the unit cell. The files *fileion*, *fileulm* and *filecoef* provide the input of the ion (see Section 2). The *fileint* file stores the uncorrelated monocentric integrals for a HF calculation. The *filecint* file stores the correlated monocentric integrals for a rCHF calculation. A uCHF calculation uses both *fileint* and *filecint* files. The *filelint* file stores the bicentric integrals.
- **endcluster**: End of cluster module. Do the calculation and return to the general shell.

1.4 Impurity commands

Module specialized in gPI calculations of local defects and impurities within pure 3D crystals.

- **impurity**: Start of the impurity module.
- **title** *runtime*: Input title.
- **rem** or **#** (**first column**): Comment line.
- **cell** $a b c \alpha \beta \gamma$: Input cell distances (i.e., a , b and c in bohr) and angles (i.e., α , β and γ in degrees).
- **spg** *spacegroup*: Input space group symbol according to the international Hermann–Mauguin notation. The same conventions as in **crystal** module.
- **neq** $x y z fileion fileulm filecoef (fileint|filecint|fileint filecint) filelint$: Input with the description of every symmetry different ion contained in the primitive cell. Use a different **neq** command for each crystallographic different ion. $0 \leq x, y, z < 1$ are the crystallographic coordinates of the ion. The files *fileion*, *fileulm* and *filecoef* provide the input of the ion (see Section 2). The *fileint* file store the uncorrelated monocentric integrals for HF calculation. The *filecint* file stores the correlated monocentric integrals for a rCHF calculation. A uCHF calculation use both *fileint* and *filecint* files. The *filelint* file stores the bicentric integrals.
- **fneq** $x y z xocup fileion fileulm filecoef (fileint|filecint|fileint filecint) filelint$: Analogous to the **neq** command except that fractional occupancies *xocup* of the crystallographic positions are allowed in order to treat crystal structures only known as an statistical average.
- **neqimp** $npos fileion fileulm filecoef (fileint|filecint|fileint filecint) filelint$ followed by *npos* with format $x y z$: Input the position of a set of the equivalent ions that form the impurity cluster. The first position is used for the representant of this set. The files *fileion*, *fileulm* and *filecoef* provide the input of the ion (see Section 2). The *fileint* file stores the uncorrelated monocentric integrals for a HF calculation. The *filecint* file stores the correlated monocentric integrals for a rCHF calculation. A uCHF calculation uses both *fileint* and *filecint* files. The *filelint* file stores the bicentric integrals.
- **hole** $x y z$: Remove the ions of the pure crystal from this position.
- **endimpurity**: End of impurity module. Do the calculation and return to the general shell.

2 Input files for ion description

The input files for ion description furnish information about ions that is required in a gPI run. Thereby, user should provide three different input files for describing each non-equivalent ion of the system. In this section, one describes the format required for these files. It should be noted that a specific format is described between parenthesis, where c , i and r indicate character, integer and real data, respectively. Free format is indicated by an asterisk $*$.

2.1 *fileion* input file

The *fileion* input file provides orbital coefficients and energies as well as the electron configuration for ions in vacuum (i.e., ions with spherical symmetry).

(A) General data

- Record A1 (c,c): *piversion*, *basistype*
piversion should be either GPI or PI7. *basistype* should be STO which means Slater type orbitals.
- Record A2 (c): *titio*
File title.
- Record A3 (c,r): *name*, *zn*
Ion name and atomic number.
- Record A4 (*): *nsym*
 $nsym = l + 1$, where l is the angular quantum number.

(B) STO basis functions (omitted if $nsym \leq 0$)

- Record B1 (*): $nsto(i)$, $i = 1, nsym$
Number of STOs for each symmetry. From this information, gPI obtains: (a) a vector $nasto(i)$ that stores the number of STOs for symmetries $nsym = 1, \dots, i - 1$; and (b) the total number $ntsto$ of STOs.
- Record B2 (*): $n(k)$, $k = 1, ntsto$
Principal quantum numbers of STOs.
- Record B3 (*): $\zeta(k)$, $k = 1, ntsto$
Orbital exponents of STOs.

(C) Atomic Orbitals (omitted if $nsym \leq 0$)

- Record C1 (*): $naos(i)$, $i = 1, nsym$
Number of atomic orbitals for each symmetry. From this information, gPI obtains: (a) a vector $naaos(i)$ that stores the number of atomic orbitals for symmetries $nsym = 1, \dots, i - 1$; and (b) the total number $ntaos$ of atomic orbitals.
- Record C2 (*): $nelec(k)$, $k = 1, ntaos$
Electron population of the atomic orbitals. Only one open shell is allowed by symmetry.

- Record C3 (*): $eorb(k), k = 1, ntaos$
Orbital energies.
- Record C4 (*): $((c(k, j + naaos(i)), k = 1, nsto(i)), j = 1, naos(i)), i = 1, nsym$
Orbital coefficients.

(D) *Coupling constants*

There are not coupling constants for closed-shell ions. **gPI** ignores these constants even when there are open shells.

- Record D1 (*): nj, nk
Number of coupling constants $J(lmn)$ y $K(lmn)$ (see Eq. 3.210 in [1]).
- Record D2 (*): $l m n xnum den$ (nj times)
 $J(lmn)$ is given by $xnum/den$.
- Record D3 (*): $l m n xnum den$ (nk times)
 $K(lmn)$ is given by $xnum/den$.

(E) *Energies*

- Record E1 (*): $eeff, eadd, ecorr, ekin, epot$
Effective, additive, correlation, kinetic and potential energies.

An example for *fileion* input file is shown in Fig. 1.

2.2 *fileulm* input file

The *fileulm* input file provides the symmetry coefficients (see Section 3.2.4 in [1]) that enable one to build the symmetry basis functions.

(A) *Irreducible representations (irreps) and symmetry subspecies*

- Record A1 (*): $nirrep$
Number of *irreps*.
- Record A2 (*): $nsub(i), i = 1, nirrep$
Number of symmetry subspecies for each *irrep*.
- Record A3 (c): $label(i), i = 1, nirrep$
Labels for each *irrep*.

(B) *Symmetry coefficients*

Next records are given by:

```

for  $l = 0, nsym - 1$ 
  for  $k = 1, 2l + 1$ 
    Record (a, *):  $label, u(l, m), m = -l, l$ 
  end-for
end-for

```

Figure 1: *fileion* input file for Zn(+2) ion.

```

GPI STO
Zn(+2) VACUO
Zn(+2)      30.000000
  3
    8      6      5
  1  1  2  2  3  3  3  3  2  2  3  3  3  3  3
  3  3  3  3
29.297200000  43.941000000  24.393000000  13.034400000
11.657200000  7.074610000  4.603680000  4.153600000
12.351300000  20.284500000  10.106400000  6.938750000
 4.155380000  1.888010000  4.921590000  13.925700000
 7.927710000  2.898990000  1.910890000
 3  2  1
 2.000000  2.000000  2.000000  6.000000  6.000000  10.000000
-353.2770013589 -44.3217200133 -5.6063026539 -38.8884822254
-3.8106777158 -0.7576816374
-0.960477027258 -0.032044019866 -0.008910299441 -0.007346617335 0.004988253985
-0.002896173181 0.004438211812 -0.003034355164
0.324798062193 -0.005598819199 0.183552491483 -1.038596700406 -0.152194562400
-0.017694434308 0.011261763025 -0.008040542783
0.121879889673 -0.001813032400 0.078732095273 -0.434384091894 -0.243378248663
0.604825413306 0.568839310316 0.064796726298
0.853244829871 0.114016647254 0.052673191292 0.011239890934 -0.000815266300
0.000334670073
-0.345768763310 -0.034670339178 -0.094563979403 0.562136417551 0.620018633172
0.010436255029
0.408286184377 0.024019048566 0.233135810532 0.374814319096 0.119121543951
-1.780267516E+03 -1.779460525E+03 -1.695369458E+00 1.776890906E+03
-3.557158422E+03

```

Figure 2: *fileulm* input file for Zn(+2) ion with point symmetry O_h .

```

4
1 3 3 2
A1g T1u T2g Eg
A1g 1
T1u 0 0 1
T1u 1 0 0
T1u 0 1 0
T2g 1 0 0 0 0
T2g 0 0 0 1 0
T2g 0 1 0 0 0
Eg 0 0 1 0 0
Eg 0 0 0 0 1

```

The variable *nsym* is defined in the *fileion* input file and *label* is defined in record A3. The variables $u(l, m)$ are the symmetry coefficients. For a given l , the symmetry adapted functions have to be orthogonal among them. The symmetry adapted functions are internally normalized if necessary. According to the Wigner–Eckardt theorem, the election of the symmetry subspecies for a given *irrep* if irrelevant whenever they are independent among them.

An example for *fileulm* input file is shown in Fig. 2.

2.3 *filecoef* input file

The *fileion* input file provides the initial orbital coefficients and energies for the local symmetry of a given ion as well as the electron population for each orbital.

(A) General data

- Record A1 (c): *title*
File title.

(B) Orbitals

- Record B1 (*): $ngaos(i), i = 1, nirrep$
Number of orbitals for each *irrep*. The variable *nirrep* is defined in the *fileulm* input file. From this information, **gPI** obtains: (a) a vector $nagaos(i)$ that stores the number of orbitals for *irreps* $1, \dots, i - 1$; and (b) the total number *ntgaos* of orbitals.
- Record B2 (*): $ngelec(k), k = 1, ntgaos$
Electron population of the orbitals. Only a single open shell is allowed in the SCF process.
- Record B3 (*): $georb(k), k = 1, ntgaos$
Orbital energies.
- Record B4 (*): $((gc(k, j + nagaos(i)), k = 1, npgao(i)), j = 1, ngaos(i)), i = 1, nirrep)$
Orbital coefficients. It should be noted that $npgao(i)$ is the number of STOs associated with *ith irrep*.

(C) Coupling constants

There are not coupling constants for closed-shell ions.

- Record C1 (*): n_j, nk
Number of coupling constants J and K .

- Record C2 (*):

```

for  $jj = 1, n_j$ 
  for  $\Gamma_{pq} = 1, ndens$ 
    for  $\Gamma'_{rs} = 1, lpq$ 
      Record (*,*,*)  $nn, xnum, xden$ 
    end-for
  end-for
end-for
for  $kk = 1, nk$ 
  for  $\Gamma_{pq} = 1, ndens$ 
    for  $\Gamma'_{rs} = 1, lpq$ 
      Record (*,*,*)  $nn, xnum1, xden1, xnum2, den2$ 
    end-for
  end-for
end-for

```

Coupling constants $J(\Gamma_{pq}, \Gamma'_{rs}, nn) = xnum/xden$, $K(\Gamma_{pq}, \Gamma'_{rs}, nn) = xnum1/xden1$ y $K(\Gamma_{pq}, \Gamma'_{sr}, nn) = xnum2/xden2$ (see Eq. 3.213 in [1]).

(D) *Energ'ies*

- Record D1 (*): $eeff, eadd, ecorr, ekin, epot$
Effective, additive, correlation, kinetic and potential energies.

An example for *filecoef* input file is shown in Fig. 3.

References

- [1] C. J. F. Solano, *PhD Thesis* (2009).

Figure 3: *filecoef* input file for Zn(+2) ion with point symmetry O_h .

```

LOCAL
  3   2   1   1
    2.000000   2.000000   2.000000   6.000000   6.000000
6.000000
  4.000000
  -352.9100920185   -43.9542639562   -5.2300489065
-38.5212371717
  -3.4337035602   -0.3769998994   -0.3757081448
-0.960478632883E+00 -0.320429185867E-01 -0.890718859615E-02
-0.735342906678E-02  0.499774233399E-02
-0.291002039349E-02  0.447769009307E-02 -0.306586866736E-02
  0.324726910157E+00 -0.555615500715E-02  0.183683947581E+00
-0.103885855884E+01 -0.151822833746E+00
-0.182975848030E-01  0.132490452255E-01 -0.965800682260E-02
  0.122843641382E+00 -0.237372614918E-02  0.768950375465E-01
-0.430537129337E+00 -0.250039591271E+00
  0.619222355439E+00  0.516596313683E+00  0.107688496394E+00
  0.853299268850E+00  0.114004406252E+00  0.526047988810E-01
0.112807674239E-01 -0.868903942928E-03
  0.371678885071E-03
-0.344205292628E+00 -0.352581864802E-01 -0.989478883935E-01
0.569393522915E+00  0.612999869196E+00
  0.159703586036E-01
  0.418961290705E+00  0.246932532430E-01  0.229704347433E+00
0.355616425421E+00  0.132938480849E+00
  0.430546625283E+00  0.254051618248E-01  0.223501089836E+00
0.324723401183E+00  0.163885860015E+00
-1.779442759E+03 -1.778199118E+03  0.000000000E+00  1.776599957E+03
-3.556042716E+03

```