

# The DP Code

Documentation Generator

May 2, 2013

## Contents

<b>1</b>	<b>DPtuto</b>	<b>2</b>
1.1	alpha_model_kernel_for_Si . . . . .	2
1.2	rpa_absorption_of_Si . . . . .	4
1.3	tdlda_absorption_and_EELS_of_Si . . . . .	7
<b>2</b>	<b>list_of_variables</b>	<b>10</b>

# 1 DPtuto

[ DP Tutorials ]

**NAME:** DP Tutorials

**FUNCTION:** The following tutorials are available at present:

\* Tutorial 1 - RPA calculation with and without local fields.

This tutorial, whose details can be found at `rpa_Si.tuto`, teaches the difference between a simple sum over states calculation (where only the knowledge of  $\chi_0$  is required) and a full RPA calculation where also local field effects (i.e. variations of the classical Hartree potential) are considered.

\* Tutorial 2 - TDLDA absorption and electron energy loss (EEL) spectrum of silicon.

This tutorial, whose details can be found at `tdlda_Si.tuto`, shows how to perform calculations using the adiabatic local density approximation for the exchange-correlation (xc) kernel (i.e. variations of the exchange and correlation potential at the adiabatic LDA level). Both the optical absorption and the electron energy loss function are calculated.

\* Tutorial 3 - Absorption spectrum of silicon using a long-range model kernel.

This tutorial, whose details can be found at `alpha_Si.tuto`, shows how to use the simple model 1 xc kernel  $f_{xc} = \alpha/q^2$  to account for excitonic effects in the absorption spectra of semiconductors.

## 1.1 alpha\_model\_kernel\_for\_Si

[ DPtuto ] [ DP Tutorials ]

**NAME:** DP Tutorial 3 - Absorption spectrum of silicon using a long-range model kernel.

**FUNCTION:**  $\alpha/q^2$  calculation.

$$\epsilon^{-1} = 1 + v\chi$$

$$\chi = \chi^0 + \chi^0(v + f_{xc})\chi$$

$f_{xc}$  is  $\alpha/q^2$ , where  $\alpha$  is an adjustable parameter.

**OBJECTIVES:** 1) Calculate optical spectra with the  $\alpha/q^2$  xc-kernel.

2) Observe the effects of a change of  $\alpha$  on the absorption spectrum.

**INPUT FILE:**

```

alpha      ! calculation using the model kernel  $\alpha/q^2$ 
shiftk     ! symmetries are not used for shifted k-point sets
           ! inversion is not used

! The choice of G-shells is still possible both using npwmat or matsh
!npwmat 59
matsh 6     ! number of shells of G-vectors for the  $\epsilon$  matrix
!npwfn 283
wfsh 15     ! number of shells of plane waves to describe the wavefunctions

! Number of bands to include in the construction of  $\chi_0$ 
nbands 8    ! number of the last band
lomo 2      ! number of the first band

! In the case of optical absorption the value of the
! perturbation wavevector q is 0
! DP considers automatically 6 different polarization directions
! (3 Cartesian axis and 3 reciprocal lattice vector directions)
q
0.0 0.0 0.0 ! q [reduced coordinates] (if q is exactly 0 -> 6 pol)

! Frequency range of interest in eV
omegai 0.0  ! initial frequency
omegae 7.0  ! last frequency
domega 0.01 ! delta  $\omega$ 

! Broadening of the curve: it is usually advised to use a small
! broadening here (0.01 eV) and add more broadening in a second time
! using the tool "broad"
broad 0.01 ! Lorentzian broadening

! Scissor operator in eV
! soenergy 0.65

! Value of the parameter alpha
! alpha -0.2

```

**TASKS:** 1) Calculation of optical spectra with the  $\alpha/q^2$  kernel.

Move to the directory `~/spectroscopy_lectures/dp/inputs/ALPHA/`

Link the KSS file created by ABINIT to a file "si.kss" in this directory.

Create an input file for Dp. You can use as an example the input file for the RPA calculation.

Remind that the quasiparticle eigenenergies are needed (either in.gw file or a scissor operator must be provided). For simple semiconductor materials the application of a scissor operator is usually sufficient. For Silicon soenergy must be set to 0.65 eV, for GaAs to 0.8 eV, for AlAs to 0.9 eV.

Note that the parameters which give a converged RPA spectrum are good also for this calculation. Calculate various spectra for different negative values of alpha and compare the imaginary part of the dielectric function to the experimental absorption spectrum.

**RUN IT:**

```
dp -i dp.in -k si.kss > dp.log &
```

! If you have a file with the GW corrections ("si.gw") it can be used  
! instead of the scissor operator in the following way:

```
dp -i dp.in -k si.kss -g si.gw > dp.log &
```

**RESULTS:** In the file dp.log you can find a summary of the input information (from dp.in and si.kss) and a description of the main steps of the calculation.

The dielectric function (real and imaginary parts) is in the files "ou\*.mdf".

Analyze attentively all the output files.

**QUESTIONS:** Can you answer these questions:

Q1. What can you observe when the absolute value of alpha increases from 0 to 2?

Q2. What is the value which gives the best agreement with the experiment?

Q3. How does the real part of the dielectric function compare to the experiment for this optimal value of  $\alpha$ ?

**TASKS:** 2) Scaling of alpha with respect to the dielectric constant.

You can play with other materials by calculating the values of alpha which gives the best agreement with experiments.

Using the table of dielectric constants below it is possible to establish the dependence of alpha on the dielectric constant.

Experimental dielectric constant of some semiconductors:

Si: 11.4, GaAs: 10.6, AlAs: 8.2, Diamond: 5.65, SiC: 6.5.

**QUESTIONS:** Q4. Which is the relation between alpha and the dielectric constant?

## 1.2 rpa\_absorption\_of\_Si

[ DPTuto ] [ DP Tutorials ]

**NAME:** DP Tutorial 1 - RPA absorption spectrum of silicon with and without local fields.

**FUNCTION:** Random Phase Approximation (RPA) calculation.

$$\varepsilon = 1 - v\chi_0$$

- OBJECTIVES:**
- 1) Prepare an input file for DP defining all the necessary parameters.
  - 2) Output and plot the absorption spectrum.
  - 3) Study the convergence of the spectrum with respect to:
    - a. the number of bands;
    - b. the number of plane waves in the wavefunction expansion;
    - c. the dimension of the G-G' dielectric matrix;
    - d. (the set of k points has already been tested).

**INPUT FILE:**

```

rpa          ! RPA calculation
shiftk       ! symmetries are not used for shifted k-point sets
              ! inversion is not used

! The choice of G-shells is still possible both using npwmat or matsh
!npwmat 59
matsh 6      ! number of shells of G-vectors for the \epsilon matrix
!npwfn 283
wfsh 15     ! number of shells of plane waves to describe the wavefunctions

! Another parameter to converge is the number of bands to include in the construction of \chi_0
nbands 8     ! number of the last band
lomo 2       ! number of the first band

! In the case of optical absorption the value of the
! perturbation wavevector q is 0
! DP considers automatically 6 different polarization directions
! (3 Cartesian axis and 3 reciprocal lattice vector directions)
q
0.0 0.0 0.0 ! q [reduced coordinates] (if q is exactly 0 -> 6 pol)

! Frequency range of interest in eV
omegai 0.0   ! initial frequency
omegae 7.0   ! last frequency
domega 0.01  ! delta \omega

! Broadening of the curve: it is usually advised to use a small
! broadening here (0.01 eV) and add more broadening in a second time
! using the tool "broad"
broad 0.01   ! Lorentzian broadening

verboselevel 3 ! this puts the verbosity level to 'high' (lowest is 0)

```

**TASKS:** 1) Calculation of a simple RPA spectrum.

Check the page `list_of_variables` (2) to get all the information you need about the variables to define to create an input file.

Now move in the subdirectory `~/spectroscopy_lectures/dp/inputs/RPA` and open the file `"dp-start.in"`.

Check if you understand the meaning of all variables.

Link the KSS file created by ABINIT to a file `"si.kss"` in the directory where you are working.

Then run the program

**RUN IT:**

```
dp -i dp-start.in -k si.kss > dp.log
```

**RESULTS:** In the file `dp.log` you can find a summary of the input information (from `dp.in` and `si.kss`) and a description of the main steps of the calculation.

The dielectric function (real and imaginary parts) is in the files `"ou*.mdf"`.

Analyze attentively all the output files.

**QUESTIONS:** Can you answer these questions:

Q1. What is the difference between the variables `npwmat` and `matsh`? And the variables `npwfn` and `npwmat`?

Q2. What is the dimension of the dielectric matrix? What happens if you run again the program after setting the dimension of the dielectric matrix in the input file to 1?

Q3. Is a scissor operator used?

Q4. Which xc kernel is used?

Q5. How many k-points are used?

Consider the two files `"outlf.mdf"` and `"outnlf.mdf"`. `"lf"` indicates that LFE are included (according to the value of `matsh`), while `"nlf"` means that the result is without local fields.

Q6. Can you understand by reading the summary included at the beginning of the file what they contain? A little hint: `"r"1` means "quantity averaged over the 3 reciprocal lattice vector directions" while `"c"` means "quantity averaged over the 3 Cartesian directions". Every couple of columns contain the real and the imaginary part of a dielectric function. Looking back at question 2: which is the difference between the content of `"outlf.mdf"` and `"outnlf.mdf"` when `npwmat=1`?

The two files `"ouclf.mdf"` and `"oucunlf.mdf"` contain only the dielectric functions averaged over the 3 Cartesian directions.

**TASKS:** 2) Plot the spectra.

Plot the imaginary part and the real part of the dielectric function of silicon and compare with the experimental spectrum (you can find the experimental curves in `~/spectroscopy_lectures/experiments`).

Try to change the artificial broadening of the spectrum using the utility `"broad"`:

**RUN IT:**

```

broad outlf.mdf
!
! A reasonable broadening is usually about 0.1-0.2 eV.

```

**TASKS:** 3) Convergence check.

The set of shifted k-points that we are using have already been tested to assure the convergence of the spectra.

Check the convergence with the number of bands and the number of planewaves in the wave-function starting from the absorption spectrum without local fields, in the energy range up to 6 eV.

Once you have found the converged values for nbands and npwfn/wfnsh, then check the convergence of the local fields, by changing the dimension of the dielectric matrix through npwmat/matsh.

**QUESTIONS:** Q7. Were the variables in "dp-start.in" large enough to obtain a converged absorption spectrum?**TASKS:** 4) From silicon to other semiconductors and insulators.

In the directory spectroscopy\_lectures/inputs/other\_materials input files for other simple semiconductors are available.

Choose another material and calculate its RPA optical spectra.

**1.3 tdllda\_absorption\_and\_EELS\_of\_Si**

[ DPtuto ] [ DP Tutorials ]

**NAME:** DP Tutorial 2 - TDLDA absorption and electron energy loss (EEL) spectrum of silicon.**FUNCTION:** Adiabatic local density approximation ( TDLDA, also called ALDA) calculation.

$$\varepsilon^{-1} = 1 + v\chi$$

$$\chi = \chi^0 + \chi^0(v + f_{xc})\chi$$

$f_{xc}$  is the frequency-independent (i.e. adiabatic) functional derivative of the static LDA exchange-correlation (xc) potential

**OBJECTIVES:** 1) Compare the spectra obtained within RPA and with a TDLDA xc kernel.

2) Understand the origin of the peaks through the comparison with the bandstructure and density of state plots.

3) Calculate an EELS spectrum within the TDLDA.

**INPUT FILE:**

```

alda          ! TDLDA calculation

```

```

shiftk      ! symmetries are not used for shifted k-point sets
            ! inversion is not used

!
! The choice of G-shells is still possible both using npwmat or matsh
!npwmat 59
matsh 6      ! number of shells of G-vectors for the \epsilon matrix
!npwfn 283
wfsh 15      ! number of shells of plane waves to describe the wavefunctions

! Number of bands to include in the construction of \chi_0
nbands 8     ! number of the last band
lomo 2       ! number of the first band

! Note that the nbands and wfsh which give a converged RPA spectrum
! should also give a converged TDLDA spectrum.
! On the other hand, the convergence with matsh must be checked.

! In the case of optical absorption the value of the
! perturbation wavevector q is 0
! DP considers automatically 6 different polarization directions
! (3 Cartesian axis and 3 reciprocal lattice vector directions)
q
0.0 0.0 0.0 ! q [reduced coordinates] (if q is exactly 0 -> 6 pol)

! Frequency range of interest in eV
omegai 0.0   ! initial frequency
omegae 7.0   ! last frequency
domega 0.01  ! delta \omega

! Broadening of the curve: it is usually advised to use a small
! broadening here (0.01 eV) and add more broadening in a second time
! using the tool "broad"
broad 0.01   ! Lorentzian broadening

```

**TASKS:** 1) Calculation of a TDLDA absorption spectrum.

Move to the directory `~/spectroscopy_lectures/dp/inputs/ALDA/`.

Create an input file (dp.in) to run the TDLDA calculation of the optical response. You can use as an example the input file for the RPA calculation. Check the page list\_of\_variables (2) to get all the information you need. Remind that the value of matsh needs a convergence check.

Link the KSS file created by ABINIT to a file "si.kss" in the directory where you want to run dp.

Then run the program

**RUN IT:**



```
dp -i dp.in -k si.kss > dp-abs-TDLDA.log
```

**RESULTS:** In the file dp.log you can find a summary of the input information (from dp.in and si.kss) and a description of the main steps of the calculation. The dielectric function is in the files "ou\*.mdf".

Analyze attentively all the output files and plot the real and imaginary part of the dielectric function.

**QUESTIONS:** Can you answer these questions:

Q1. Is it necessary to make the dielectric matrix larger to get a converged spectrum?

Compare the TDLDA absorption spectrum with the RPA one.

Q2. What can you conclude about the effect of the TDLDA xc kernel for the spectrum of silicon?

Q3. Can you understand which transitions contribute to the peaks by inspecting the plots of the bandstructure and the density of states?

**TASKS:** 2) Calculation of the EEL spectrum at  $q=0$ .

Try to make a fully converged calculation for the EELS (pay particular attention to the number of bands).

Perform both a RPA and a TDLDA calculation.

**RUN IT:**

```
! Run DP as usual
dp -i dp-eels.in -k si.kss > dp-eels.log &
```

**RESULTS:** All the information necessary to calculate the EELS is inside the complex dielectric function.

To transform the dielectric function contained in "outlf.mdf" to the EELS function, use the utility "mdf2eel".

**RUN IT:**

```
mdf2eel outlf.mdf
```

**RESULTS:** The new file "outlf.eel" contains the EELS. You can broaden the curve with the utility "broad" as usual.

**RUN IT:**

```
broad outlf.eel
```

**QUESTIONS:** Q4. How do the RPA and TDLDA EELS spectra compare to the experiment? (You can find the experimental curves in ~/spectroscopy\_lectures/experiments).

**TASKS:** 3) From silicon to other semiconductors and insulators.

Choose another material and calculate its TDLDA absorption spectrum.

## 2 list\_of\_variables

[ DP Tutorials ]

**NAME:** DP List of Input variables

**SYNOPSIS:**

```

verboselevel <num> ! <num>=0,1,2,3
                    !      0) minimum amount of info written down
                    !      1) default values. essential things are written down
                    !      2) Verbose mode: many more things are written down
                    !      3) Debug mode: to use only in debugging case,
                    !                  all the loop indices are written
                    !                  (tree and mem files created)

!----- Type of calculation (default: linear response:tddft) -----
linear          ! default: linear response in G space.
                !      solving the tddft equation for \chi

exciton         ! excitonic calculation (the EXC code) in transition space

photoemission   ! photoemission spectrum. implies the variable ephoton (see below)
ephoton <num>   ! value of the incoming photon for the photoemission spectrum

nonlinear       ! nonlinear tddft equation: to be merged from the nonlinear code

current         ! time dependent current DFT: to be implemented
!-----

!----- Dielectric function (default: testparticle) -----
testparticle    ! default: normal equation for the inverse dielectric function
testelectron    ! to be used only for testing or for particular purposes
                ! (vertex corrections)
!-----

!----- Energies for the \chi^0 (default:enks) -----
enks            ! default: use the Kohn-Sham eigenvalues found in the ground-state file
                !      to construct the denominator of \chi^0
gw              ! perturbative GW energies: this variable implies the existence of a gwfile
qpsenergy       ! self-consistent quasiparticle energies.
                !      this variable implies the existence of a qps file
so              ! scissor operator correction. see below soenergy, stretch_valence,
                !                  stretch_conduction
soenergy <num>  ! value of the scissor correction. default: 0.0

```

```

stretch_valence <num>    ! stretch = GW bandwidth / DFT bandwidth. default: 1.0
stretch_conduction <num> ! modify energy in a linear way in order to reproduce the GW bandwidth
                        !                                     like:  $e_{\text{GW}} = e_{\text{DFT}} * \text{stretch}$ 
somult                ! multi-scissor operator correction. implies the existence of a file
                        !                                     called somult.in
lifetime              ! use the  $\text{Im}(\Sigma)$  as lifetime (broadening) in the denominator of  $\chi^0$ .
                        !                                     this implies the presence in the gw file of the  $\text{ImIm}(\Sigma)$ 
!-----

!----- Wavefunctions for the  $\chi^0$  (default:kss) -----
kss                   ! default. use the Kohn-Sham wavefunctions found in the ground-state file
                        !                                     to construct the numerator of  $\chi^0$ 
qps                   ! self-consistent quasiparticle wavefunctions. this variable implies the existence
                        !                                     of a qps file
!-----

!----- analysis calculation -----
jdos                  ! joint density of states = only KS energies taken into account
printmatrixelement   ! only matrix elements (rhotw) taken into account, not energies
matelanalysis        ! do a normal calculation, but in addition it creates a file, called
                        !                                     matelanalysis.out containing the matrix elements for a further analysis
!-----

!----- kernel (tddft default:rpa; exciton default:exc) -----
rpa                   ! default: random phase approximation ==> fxc = 0
alda                 ! adiabatic local density approximation ==>
                        !                                      $\text{fxc} = \Delta V_{\text{xc}}^{\text{lda}} / \Delta n$ 
longrange            ! longrange alpha kernel ==>  $\text{fxc} = \alpha / q^2$ . see below for the value of alpha
dyn_longrange        ! dynamical longrange alpha kernel ==>  $\text{fxc} = (\alpha + \beta \omega) / q^2$ .
                        ! see below for the value of alpha and beta.
alpha <num>           ! value of alpha in the longrange or dyn_longrange kernel. default: 0.0
beta <num>            ! value of beta in the dyn_longrange kernel. default: 0.0
fxcmt                ! mapping theory kernel. implies the existence of a screening file (in.em1)
exc                  ! full exciton calculation. implies the existence of a screening file.
                        ! this is the default for EXC.
!-----

!----- Tamm-Dancoff Approximation (default only in the excitonic case) -----
tammdancoff          ! if this flag occurs the calculation will use the so-called Tamm-Dancoff
                        ! approximation, i.e. only the resonant part of  $\chi_0$  will be taken into account
resonant              ! equivalent of tammdancoff but in the exciton framework
!-----

!----- Exciton solver (default:fulldiago) -----
fulldiago            ! default. diagonalization of the excitonic hamiltonian.

```

```

haydock      ! iterative inversion of the excitonic hamiltonian. implies the variable niter.
              ! only the final spectrum is obtained but no excitonic eigenvalues and
              ! eigenvectors, preventing further analysis.
niter <num>   ! number of iterations for the haydock iterative procedure. default:100
haydock_restart ! write restart file. default false
!-----

!----- Variables for the EXC code -----
wdiag      ! default: only the diagonal (in G) of Wgg' is retained. it turns out to be a good
              ! approximation for solids
wfull      ! the whole gg' structure of W is used. important for clusters.

lf          ! default: calculate also the exchange term <<v>>.
nlf         ! avoid the calculation of the exchange term <<v>>. this corresponds in an rpa
              ! calculation to do a calculation without local fields.
!-----

!----- Use of shifted k-points -----
shiftk      ! if the ground state structure has been calculated using shifted
              ! k-points, we have to tell dp about it, otherwise it will try to use
              ! the symmetries of the system to sample the whole Brillouin zone.
              ! Alternatively (and deprecatively) you can still use the old way ...
nsym 1      ! force DP to consider only the first Symmetry (the identity)
noinv       ! so not use inversion
              ! Remember: either one uses the 'shiftk' flag OR 'nsym 1' + 'noinv'.
              ! TODO: this will be evaluated automatically by DP, from version 5.3
!-----

!----- convergency parameters -----
npwwfn <num> ! number of plane waves to describe the wavefunctions
              ! default: all plane waves found on the _KSS file will be used
wfnsn <num>  ! number of shells of plane waves to describe the wavefunctions
              ! default:all
              ! Remark: only one of the above parameters has to be specified

npwmat <num> ! number of G-vectors for the \epsilon matrix (related to local
              ! fields) ! default:all
matsh <num>  ! number of shells of G-vectors for the \epsilon matrix (related
              ! to local fields) ! default:all
              ! Remark: only one of the above parameters has to be specified

nbands <num> ! number of the last band (highest energy) to be included in the
              ! calculation ! default:all
lomo <num>   ! number of the first band (lowest energy) to be included in the
              ! calculation ! default:1

```

```

selected_bands ! DP: enable the selection of the specific bands to be used in
                ! the calculation. this implies the existence of a file called
                ! bands.in where the selected bands are listed (one per line).

selected_kpts   ! DP: enable the selection of the specific k-points to be used in
                ! the calculation. this implies the existence of a file called
                ! kpts.in where the kpoints are listed (like: -0.5 0.33 0) one
                ! per line
!-----

!-- Pseudopotential -----
novkb ! exclude the nonlocal part of the pseudopotential, which is unfortunately
      ! the leading term for a big number of transitions
!-----

!--- Hilbert transform -----
hilbert ! if this flag occurs, the Hilbert Transform will be used to
        ! construct the \chi_0. This is particularly convenient when a big
        ! range of energies is requested. See details at
        ! http://link.aps.org/abstract/PRB/v75/e205405
!-----

!---- metallic and DRUDE (EXPERIMENTAL)-----
metal ! if this flag occurs, a metallic behaviour is considered, the
      ! occupation numbers will be treated as fractional.
      ! The temperature is set by kBT (see below).
kBT <num> ! electronic temperature (eV). Default: 0.001 (315 K)
occopt 99 ! debugging option: take the occupation values as written in the kss

drude ! calculate also the intraband contribution using the drude formula
onlydrude ! calculate only the intraband contribution using the drude formula
!-----

!---- Energy Range (all values in eV) -----
omegai <num> ! initial frequency
omegae <num> ! ending frequency
domega <num> ! frequency Mesh (delta omega). Default: 0.1
            ! the choice of the above 3 values uniquely determines the number
            ! of frequencies, via nomega=(omegae-omegai)/domega + 0.5 +1
broad <num> ! Lorentzian broadening. It's better to put a small number here
            ! (0.001 - 0.01) and use the 'broad' utility to convolute the
            ! obtained curves afterwards. Default: domega
!-----

```

```

!----- Momentum Transfer - Polarization of light -----
q
<numx> <numy> <numz> ! This specifies the momentum transfer (vector) to the
! system. The value of q can range from 0 to very high
! value (several Brillouin zones equivalent)
! Very high q vector can be found in Electron energy loss
! or Inelastic X-ray Scattering. The case q=0 is
! particularly important for optical absorption, where the
! polarization of light can be specified giving a small
! value in one or another direction, e.g.
! q 0.0001 0.0001 0.0
! Remark: q uses the reduced coordinates here, not the
! cartesian ones.
! TODO: give the possibility to choose the coordinates
! (reduced or cartesian)
doublegrid ! Allows to read two separate KSS-files for initial and final
! states, which are shifted by q with respect to each other.
! (e.g. k.p perturbation theory will be avoided at small q)
! USAGE: dp -i <dp.in> -k <dp1.kss> -k <dp2.kss>
! dp1.kss corresponds to the final states at k,
! dp2.kss corresponds to the initial states at k-q
antieps ! calculation of eps*(-q,-w) instead of eps(q,w), which
! corresponds to exchanging the two kss files in doublegrid
!-----

```

```

!----- Handling chi0 -----
writechi0 ! write down the chi0 into a (binary) file called chi0.dat
writefullchi ! write down the chi into a (binary) file called chi.dat
! (currently only supported for RPA calculations)
readchi0 ! read the chi0 from a (binary) file called chi0.dat
!-----

```

```

!----- Optimization flags -----
savememory ! if present this flag will make the code calculating the rhoTw
! at any transition with a double Fourier transform, rather than
! storing the wave functions in real space. Advantage: important
! reduction of the memory. Disadvantage: 3 instead of 2 Fourier
! transform per wavefunction.
! TODO: making this default.
testrun ! performs testrun to estimate calculation time and memory
!-----

```

```

!----- undocumented variables (and so to be documented :) -----
pert
blockpert
encutoff

```

dielconstscreen  
coulcutoff  
ircut  
uvcut  
stripecut  
renormdielconst  
rlcut  
kernel  
x-1x

!-----