Yambo©

Cheatsheet

For GPL v4.1.2

Input file generation and command line interface **(yambo -H)**

-J <opt>	:Job string identifier
-V <opt>	:Input file verbosity [opt=RL,kpt,sc,qp,io,gen,resp,all,par]
-F <opt>	:Input file
-I <opt>	:Core I/O directory
-O <opt>	:Additional I/O directory
-C <opt>	:Communications I/O directory
-D	:DataBases properties
-W <opt>	:Wall Time limitation (1d2h30m format)
-Q	:Don't launch the text editor
-M	:Switch-off MPI support (serial run)
-N	:Switch-off OpenMP support (one thread)

Combination of options

Examples: input file generation/runlevel selection:

$ yambo -o c -k hartree -V RL	Optics, LFE
$ yambo -x -g n -p p -V qp	GW with PPA
$ yambo -o b -k sex -y h -b	Optics, BSE

-i	:Initialization
-o <opt>	:Optics [opt=(c)hi is (G)-space (b)se is (eh)-space]
-k <opt>	:Kernel [opt=hartree/alda/lrc/hf/sex] hf/sex only eh-space; lrc only G-space
-y <opt>	:BSE solver [opt=h/d/(p/f)i] (h)aydock/ (d)iagonalization (i)nversion
-r	:Coulomb potential
-x	:Hartree-Fock Self-energy and local XC
-d	:Dynamical Inverse Dielectric Matrix
-b	:Static Inverse Dielectric Matrix
-p <opt>	:GW approximations [opt=(p)PA/(c)HOSEX]
-g <opt>	:Dyson Equation solver [opt=(n)ewton/(s)ecant/(g)reen]
-l	:GoWo Quasiparticle lifetimes
-a	:ACFDT Total Energy

(1) FFT and oscillators: `yambo -V RL (several runlevels)` v4.1.2

Dipole/momentum matrix elements (*q*→*0*) $\langle n\mathbf{k}|\mathbf{p} + i[V^{\mathrm{NL}}, \mathbf{r}]|m\mathbf{k}\rangle$

Screening matrix elements (FFT) $\rho_{nm}(\boldsymbol{k},\boldsymbol{q},\boldsymbol{G}) = \langle n\boldsymbol{k}|e^{i(\boldsymbol{q}+\boldsymbol{G})\cdot\boldsymbol{r}}|m\boldsymbol{k}-\boldsymbol{q}\rangle$

where the wavefunction is expanded over reciprocal lattice (G) vectors:
$$\phi_{n\mathbf{k}}(\mathbf{r}) = \frac{1}{\sqrt{V}}\sum_{\mathbf{G}} e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} c_{n\mathbf{k}}(\mathbf{G})$$

`FFTGvecs = 1 RL`

Number of G-vectors (or energy cutoff) for expanding wavefunctions in transition matrix elements and FFT operations

Units: number of G-vectors (RL), or energy unit (Ry, mHa, eV)

Determines size (memory) of calculation. Corresponds to cutoff in DFT calculation; can be much less than geometry cutoff

Non-local commutator term

Can greatly increase CPU time. Remove by hiding the `SAVE/ns.kb_pp_pwscf` file.

(2) Linear response (IP/RPA-NLF): `yambo -o c` v4.1.2

See (1)

$$\epsilon_{\alpha,\alpha}(\omega) = 1 + \frac{16\pi}{\Omega}\sum_{c,v}\sum_{\mathbf{k}} \frac{1}{E_{c\mathbf{k}} - E_{v\mathbf{k}}} \frac{|\langle v\mathbf{k}|\mathbf{p}_\alpha + i[V^{\mathrm{NL}}, \mathbf{r}_\alpha]|c\mathbf{k}\rangle|^2}{(E_{c\mathbf{k}} - E_{v\mathbf{k}})^2 - (\omega + i\gamma)^2}$$

```
% LongDrXd
 1.000 | 0.000 | 0.000 |
%
```

E-field direction (for q=0)
Vector (cartesian coordinate)
Refers to first q-point (`QpntsRXd`)

DFT k-grid

```
% EnRngeXd
 0.000 | 10.000 | eV
%
ETStpsXd = 100
```

Energy grid in output
Range from 0 to 10 in 100 steps

```
% BndsRnXd
 1 | 100 |
```

Bands used (empty & filled)
Range from 1 to nbnd
Reduce range to lower memory. In metals, includes partially filled bands. See also `EhEngyXd (-V all)`

```
% DmRngeXd
 0.1000 | 0.100 | eV
%
```

Broadening of spectra
Either a fixed value, or linearly changing between 2 values

(3) Linear response with local fields (RPA-LFE): `yambo -o c -k hartree`

$$\chi_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega) = \chi^0_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega) + \sum_{\mathbf{G}_1,\mathbf{G}_2} \chi^0_{\mathbf{G},\mathbf{G}_1}(\mathbf{q},\omega)\left[v_{\mathbf{G}_1}(\mathbf{q})\delta_{\mathbf{G}_1,\mathbf{G}_2} + f^{xc}_{\mathbf{G}_1,\mathbf{G}_2}\right]\chi_{\mathbf{G}_2,\mathbf{G}'}(\mathbf{q},\omega)$$

Hartree ALDA/LRC

$$\chi^0_{\mathbf{G}\mathbf{G}'}(\mathbf{q},\omega) = 2\sum_{c,v}\int_{BZ}\frac{d\mathbf{k}}{(2\pi)^3}\rho^*_{cv\mathbf{k}}(\mathbf{q},\mathbf{G})\rho_{cv\mathbf{k}}(\mathbf{q},\mathbf{G}')f_{v\mathbf{k}-\mathbf{q}}(1-f_{c\mathbf{k}}) \times$$

See (1)

$$\times\left[\frac{1}{\omega + \epsilon_{v\mathbf{k}-\mathbf{q}} - \epsilon_{c\mathbf{k}} + i0^+} - \frac{1}{\omega + \epsilon_{c\mathbf{k}} - \epsilon_{v\mathbf{k}-\mathbf{q}} - i0^+}\right]$$

```
% QpntsRXd
 1 | 6 |
%
```
Transferred momenta
Range from 1:nqpt (1 = q→0)

`NGsBlkXd=1`
Response block size
Depends on inhomogeneity; needs convergence study, but much less than FFTGvecs

```
% EhEngyXd (-V resp)
-1.000 | -1.000 | eV
```
Electron-hole energy range *An alternative way to BndsRnXd to restrict transitions, but will not reduce memory*

`Chimod="Hartree"`
Kernel: set with -k <opt>
Hartree/ALDA/LRC

`GrFnTpXd="T" (-V resp)`
Green's function: T/R/r/Ta/Ra
"R" = resonant gives the causal response function

(4) Linear response within TDDFT: `yambo -o c -k ALDA/LRC`

$$\chi_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega) = \chi^0_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega) + \sum_{\mathbf{G}_1,\mathbf{G}_2} \chi^0_{\mathbf{G},\mathbf{G}_1}(\mathbf{q},\omega)\left[v_{\mathbf{G}_1}(\mathbf{q})\delta_{\mathbf{G}_1,\mathbf{G}_2} + f^{xc}_{\mathbf{G}_1,\mathbf{G}_2}\right]\chi_{\mathbf{G}_2,\mathbf{G}'}(\mathbf{q},\omega)$$

Hartree ALDA/LRC

$$\chi^0_{\mathbf{G}\mathbf{G}'}(\mathbf{q},\omega) = 2\sum_{c,v}\int_{BZ}\frac{d\mathbf{k}}{(2\pi)^3}\rho^*_{cv\mathbf{k}}(\mathbf{q},\mathbf{G})\rho_{cv\mathbf{k}}(\mathbf{q},\mathbf{G}')f_{v\mathbf{k}-\mathbf{q}}(1-f_{c\mathbf{k}}) \times$$
$$\times\left[\frac{1}{\omega+\epsilon_{v\mathbf{k}-\mathbf{q}}-\epsilon_{c\mathbf{k}}+i0^+} - \frac{1}{\omega+\epsilon_{c\mathbf{k}}-\epsilon_{v\mathbf{k}-\mathbf{q}}-i0^+}\right]$$

See (1)

`NGsBlkXd=1`
Response block size
Depends on inhomogeneity; needs convergence study, but much less than FFTGvecs

`FxcGRLc=1`
XC-kernel size
Needs convergence study. Much less than FFTGvecs

`LRC_alpha=1`
LRC fitting parameter
Long-range tail of the f_{xc} kernel. Depends on the system: the larger the screening the smaller this parameter.

`Chimod="ALDA"`
Kernel: set with -k <opt>
<opt>=ALDA
<opt>=LRC: semi-empirical kernel with proper long-range behaviour. It needs a fitting parameter!

(5) Screening (RPA): `yambo -d`

See sheet (3): this runlevel computes the inverse dielectric matrix from X(G,G')

$$\epsilon^{-1}_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega) = \delta_{\mathbf{G},\mathbf{G}'} + v_{\mathbf{G}}(\mathbf{q})\chi_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega)$$

Connection with experiment:

$$\epsilon_M(\omega) = \lim_{\mathbf{q}\to 0} \frac{1}{\epsilon^{-1}_{\mathbf{G}=\mathbf{0},\mathbf{G}'=\mathbf{0}}(\mathbf{q},\omega)}$$

$$\mathrm{Abs}(\omega) = \mathrm{Im}\,\epsilon_M(\omega) \qquad \mathrm{EELS}(\omega) = -\mathrm{Im}\,\frac{1}{\epsilon_M(\omega)}$$

(6a) Coulomb integrals RIM (Random Integration Method): `yambo -r`

$$v(\boldsymbol{q}+\boldsymbol{G}) = \frac{4\pi}{|\boldsymbol{q}+\boldsymbol{G}|^2}$$

$$\int_{Bz} \frac{d^3q}{2\pi^3} f(q,G) v(q+G) \approx \sum_{q_i} f(q_i,G) v(q_i+G) \Omega_{q_i}$$

Discretization of Bz for integrals

A better approximation is given by:

$$\int_{Bz} \frac{d^3q}{2\pi^3} f(q,G) v(q+G) \approx \sum_{q_i} f(q_i,G) I_{q_i}(G)$$

Monte Carlo integral of the Coulomb potential in each region the Bz has been dissected by the q point sampling

$$I_{q_i}(G) = \int_{R_\Gamma} \frac{d^3q'}{(2\pi)^3} v(q+q'+G)$$

Tip: 1. Needed for non 3D system to avoid divergences for small q
2. Needed to build cutoff potential with box shape

Random Integration Method
RandQpts=1000000
Number of q points to perform Monte Carlo Integration,
RandGvec= 1 RL
Number of G vectors the RIM is calculated
Tip: RandGvec=1 (gamma) is usually enough. 1 Million q points is usually accurate.

(6b) Coulomb cutoff : `yambo -r`

$$v(\boldsymbol{q}+\boldsymbol{G})=\frac{4\pi}{|\boldsymbol{q}+\boldsymbol{G}|^2}$$

Truncation of the Coulomb potential for non 3D system
to speed up convergence with respect the vacuum

$$V_c(\mathbf{r}) = \begin{cases} \frac{1}{|\mathbf{r}|}, & \text{if } \mathbf{r} \in S. \\ 0, & \text{otherwise.} \end{cases}$$

S: interactting region:

CUTGeo= "box Z"   X/Y/Z or XY/XZ/YZ or XYZ

- *Possible region:*
- *sphere (0D for molecules),*
- *cylinder (1D for polymers, tubes, etc),*
- *box (0D, 1D, 2D).*

XYZ: cut in all directions
Box: XY: cut in XY only, etc…
Cylinder X/Y/Z indicates cylinder axis

- Sphere XYZ: assign: CUTRadius= 10.0 a.u
- Cylinder Z: assign CUTRadius and CUTCylLen (CUTCylLen=0 indicates infinite cylinder)
- Box Z: assign CUTBox

```
% CUTBox
 0.00     | 0.00     | 32.00     |        # [CUT] [au] Box sides
%
```

Box side=0 means do not cut in that direction

Tip: When using Box shapes, the RIM is also needed to calculate the potential. In Box for large enough boxes assigns Box side slighlty smaller than the cell box

(7) Exchange self energy: `yambo -x`

$$\Sigma^x_{n\mathbf{k}} = \langle n\mathbf{k}|\Sigma^x(\mathbf{r}_1,\mathbf{r}_2)|n\mathbf{k}\rangle = -\sum_m \int_{BZ} \frac{d\mathbf{q}}{(2\pi)^3} \sum_{\mathbf{G}} v(\mathbf{q}+\mathbf{G}) |\rho_{nm}(\mathbf{k},\mathbf{q},\mathbf{G})|^2 f_{m(\mathbf{k}-\mathbf{q})}$$

See (6) See (1)

occupied bands only

DFT k-grid
{q} = {k-k'}

```
EXXRLvcs= 2487001   RL
```

G-vectors in the exchange
Number of RL vectors, or energy in Ry / mHa / etc
Tip: set to less than FFTGvecs

```
%QPkrange
  1|   5|  20|   59|
  4|   8|  60|   80|
%
%QPerange (-V qp)
  1|  32|  0.0|-1.0|
%
```

nk, n'k' ranges where GW/Σ_x elements are calculated
first k-point | last k-point | lower band | upper band
This can be split over several lines for multiple groups
Tip: careful use of fewer k-points and bands reduces the calculation time; yambo will interpolate the rest
nk,nk' ranges (alternative method)
first k-point | last k-point | lower energy | upper energy

(8) Correlation part of self energy: `yambo -g n`

$$\Sigma^c_{n\mathbf{k}}(\omega) = \langle n\mathbf{k}|\Sigma^c|n\mathbf{k}\rangle = i\sum_m \int_{BZ} \frac{d\mathbf{q}}{(2\pi)^3} \sum_{\mathbf{G},\mathbf{G}'} \frac{4\pi}{|\mathbf{q}+\mathbf{G}|^2} \rho_{nm}(\mathbf{k},\mathbf{q},\mathbf{G})\rho^*_{nm}(\mathbf{k},\mathbf{q},\mathbf{G}') \times \int d\omega' G^0_{m\mathbf{k}-\mathbf{q}}(\omega-\omega')\epsilon^{-1}_{\mathbf{G}\mathbf{G}'}(\mathbf{q},\omega')$$

```
% GbndRnge
  1 | 50 |
%
```

Bands used in the GW summation
QP energies usually shows slow convergence
Tip: If you are interested in gaps, energy differences converge faster

```
NGsBlkXp= 100   RL
```

Response block size
See (9)

```
%QPkrange
  1|   5| 20|   59|
  4|   8| 60|   80|
%
%QPerange (-V qp)
  1| 32| 0.0|-1.0|
%
```

nk, n'k' ranges where GW/Σ_c elements are calculated
first k-point | last k-point | lower band | upper band
This can be split over several lines for multiple groups
Tip: careful use of fewer k-points and bands reduces the calculation time; yambo will interpolate the rest

(8a) Dyson Solver: `yambo -g n/s`

$$E_{nk}^{QP} = \epsilon_{nk} + \langle \psi_{nk} | \Sigma(E_{nk}^{QP}) - V_{xc} | \psi_{nk} \rangle$$

DysSolver= "n" First order expansion around KS eigenvalue

$$E_{nk}^{QP} = \epsilon_{nk} + Z_{nk} \langle \psi_{nk} | \Sigma(\epsilon_{nk}) - V_{xc} | \psi_{nk} \rangle$$

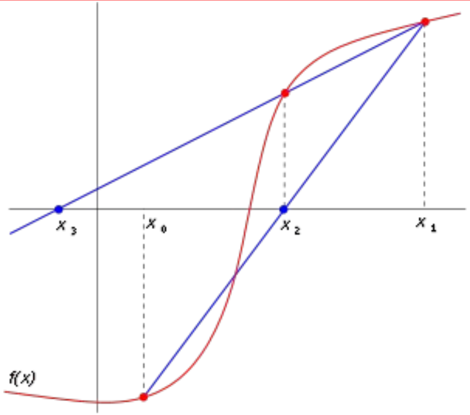
$$Z_{nk} = \left[1 - \frac{d\Sigma_{nk}(\omega)}{d\omega} \Big|_{\omega=\epsilon_{nk}} \right]^{-1}$$

dScStep= 0.10000   eV   # [GW] Energy step to evaluate Z

DysSolver= "s" Secant iterative method

https://en.wikipedia.org/wiki/Secant_method

$$x_n = x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})} = \frac{x_{n-2} f(x_{n-1}) - x_{n-1} f(x_{n-2})}{f(x_{n-1}) - f(x_{n-2})}$$



(9) Plasmon Pole approximation (PPA): `yambo -p p`

Components of the Dielectric matrix approximated has a single pole functions:

$$\epsilon^{-1}_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega) \sim \delta_{\mathbf{G},\mathbf{G}'} + \mathbf{R}_{\mathbf{G},\mathbf{G}'}(\mathbf{q})[(\omega - \Omega_{\mathbf{G},\mathbf{G}'}(\mathbf{q}) + i0^+)^{-1} - (\omega + \Omega_{\mathbf{G},\mathbf{G}'}(\mathbf{q}) - i0^+)^{-1}]$$

Residuals $R_{\mathbf{G},\mathbf{G}'}(\mathbf{q})$ and energies $\Omega_{\mathbf{G},\mathbf{G}'}(\mathbf{q})$ are found by imposing the PPA to reproduce the exact ϵ^{-1} function at ω = 0 and ω = iE_{PPA} with E_{PPA} being a suitable user-defined parameter.

$$R_{\mathbf{G},\mathbf{G}'}(\mathbf{q}) = -\frac{\epsilon^{-1}_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega=0)\Omega_{\mathbf{G},\mathbf{G}'}}{2}$$

$$\Omega_{\mathbf{G},\mathbf{G}'} = E_{PPA}\sqrt{\frac{\epsilon^{-1}_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega=\mathbf{E_{PPA}})}{\epsilon^{-1}_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega=0)-\epsilon^{-1}_{\mathbf{G},\mathbf{G}'}(\mathbf{q},\omega=\mathbf{E_{PPA}})}}$$

```
% BndsRnXp
 1 | 100 |
```

Bands used (empty & filled)
Range from 1 to nbnd
Reduce range to lower memory.

```
NGsBlkXp= 100          RL
```

Response block size

The QP energies should not depend too much on the choice of imaginary plasmon frequency. **Tip:** Choose a value higher in energy than the plasmon peak (EELS spectrum)

```
PPAPntXp= 27.21138    eV
```

PPA imaginary energy

(10a) Construction of the BSE Hamiltonian: `yambo -o b -k sex -b`

BSE is rewritten as an eigenvalue problem for the 2 particle Hamiltonian:
size of matrix $[N_v \times N_c \times K_{BZ}] \times [N_v \times N_c \times K_{BZ}]$

$$H^{exc}_{\substack{vc\mathbf{k}\\ v'c'\mathbf{k}'}} = (\epsilon_{c\mathbf{k}} - \epsilon_{v\mathbf{k}})\delta_{c,c'}\delta_{v,v'}\delta_{\mathbf{k}\mathbf{k}'} + (f_{c\mathbf{k}} - f_{v\mathbf{k}})[2\bar{V}_{\substack{vc\mathbf{k}\\ v'c'\mathbf{k}'}} - W_{\substack{vc\mathbf{k}\\ v'c'\mathbf{k}'}}]$$

Difference of quasiparticle energies:
From DFT + QP corrections:

Kernel part: see next slide

KfnQPdb= " E < ./SAVE/ndb.QP"
Location of QP corrections database
From previous GW calculation

OR

% KfnQP_E
1.4000 | 1.200 | 0.900 |
QP corrections parameters
scissor | stretch conduction | stretch valence

% BSEBands
2 | 8 |
Bands Range
lower band | upper band |

(10b) Construction of the BSE kernel: `yambo -o b -k sex -b`

Electron-hole exchange part (from Hartree potential - local field effects):

$$K^{x}_{vc\mathbf{k},v'c'\mathbf{k}'} = \bar{V}_{vc\mathbf{k},v'c'\mathbf{k}'} = \frac{1}{\Omega}\sum_{\mathbf{G}\neq 0} v(\mathbf{G})\langle c\mathbf{k}|e^{i\mathbf{G}\mathbf{r}}|v\mathbf{k}\rangle\langle v'\mathbf{k}'|e^{-i\mathbf{G}'\mathbf{r}}|c'\mathbf{k}'\rangle$$

BSENGexx = 30 Ry
Components of Hartree potential

Electron-hole attraction part (from screened exchange potential - excitonic effects):

$$K^{c}_{vc\mathbf{k},v'c'\mathbf{k}'} = W_{vc\mathbf{k},v'c'\mathbf{k}'} = \frac{1}{\Omega}\sum_{\mathbf{G},\mathbf{G}'} v(\mathbf{q}+\mathbf{G})\varepsilon^{-1}_{\mathbf{G},\mathbf{G}'}(\mathbf{q})\langle c\mathbf{k}|e^{i(\mathbf{q}+\mathbf{G})\mathbf{r}}|c'\mathbf{k}'\rangle\langle v'\mathbf{k}'|e^{-i(\mathbf{q}+\mathbf{G}')\mathbf{r}}|v\mathbf{k}\rangle\delta_{\mathbf{q}\mathbf{k}-\mathbf{k}'}$$

BSENGblk = 2 Ry
Screened interaction block size

%BandsRnXs
1 | 20 |
NGsBlkXs = 2 Ry
% LongDrXS
1.000 | 1.000 | 1.000|

See (3)

(11a) BSE solver (diagonalisation): `yambo -y d`

The macroscopic dielectric function is obtained as:

$$\epsilon_M(\omega) \equiv 1 - \lim_{\mathbf{q}\to 0} \frac{8\pi}{|\mathbf{q}|^2\Omega} \sum_{vc\mathbf{k}} \sum_{v'c'\mathbf{k}'} \langle v\mathbf{k} - \mathbf{q}|e^{-i\mathbf{qr}}|c\mathbf{k}\rangle \langle c'\mathbf{k}'|e^{i\mathbf{qr}}|v'\mathbf{k}' - \mathbf{q}\rangle \sum_{\lambda} \frac{A^{\lambda}_{cv\mathbf{k}} \left(A^{\lambda}_{c'v'\mathbf{k}'}\right)^*}{\omega - E_\lambda}$$

% BEnRange
2.00000 | 8.00000 | eV
%
Energy range
min energy | max energy

BEnSteps= 200
Number of evenly spaced energy points

% BLongDir
1.000000 | 1.000000 | 0.000000 |
%
Direction of the longitudinal perturbation

% BDmRange
0.10000 | 0.10000 | eV
%
Lorentzian broadening changes linearly
broad@min energy| broad@max energy

(11b) BSE solver (Lanczos-Haydock): `yambo -y h`

The macroscopic dielectric function is obtained as:

$$\epsilon_M(\omega) \equiv 1 - \lim_{\mathbf{q}\to 0} \frac{8\pi}{|\mathbf{q}|^2\Omega} \sum_{vc\mathbf{k}} \left|\langle v\mathbf{k}-\mathbf{q}|e^{-i\mathbf{q}\mathbf{r}}|c\mathbf{k}\rangle\right|^2 \frac{1}{(\omega - a_1) - \frac{b_2^2}{(\omega - a_2) - \frac{b_3^2}{\cdots}}}.$$

Where the a's and b's are obtained iteratively from Lanczos algorithm

BSHayTrs= -0.02000
Threshold for accuracy of the iterative process
<u>Negative sign</u>: average difference, over the energy range, of two consecutive approximations to the spectrum
<u>Positive sign</u>: maximum difference, over the energy range, of two consecutive approximations to the spectrum

In addition to input parameters defined in (11a)

(12) Postprocessing - exciton plot: `ypp -e w`

```
excitons                     # [R] Excitons
wavefunction                 # [R] Wavefunction
Format= "x"                  # Output format [(c)ube/(g)nuplot/(x)crysden]
Direction= "123"   # [rlu] [1/2/3] for 1d or [12/13/23] for 2d [123] for 3D
FFTGvecs= 30 Ry              # [FFT] Plane-waves
States= "1 - 3"              # Index of the BS state(s)
Degen_Step=   0.0100   eV   # Maximum energy separation of two degenerate states
% Cells
 5 |  5 |  1 |               # Number of cell repetitions in each direction (odd or 1)
%
% Hole
 0.00 | 3.44        | 0.00 |       # [cc] Hole position in unit cell
```

Excitonic wavefunction does not have the periodicity of the e and h wavefunctions but is generally more extended, with a fictitious periodicity due to the k-points sampling

(13a) Parallelism: Linear Response

$$\chi^0_{\mathbf{G}\mathbf{G}'}(\mathbf{q},\omega) = 2\sum_{c,v}\int_{BZ}\frac{d\mathbf{k}}{(2\pi)^3}\rho^*_{cv\mathbf{k}}(\mathbf{q},\mathbf{G})\rho_{cv\mathbf{k}}(\mathbf{q},\mathbf{G}')f_{v\mathbf{k}-\mathbf{q}}(1-f_{c\mathbf{k}}) \times \left[\frac{1}{\omega+\epsilon_{v\mathbf{k}-\mathbf{q}}-\epsilon_{c\mathbf{k}}+i0^+} - \frac{1}{\omega+\epsilon_{c\mathbf{k}}-\epsilon_{v\mathbf{k}-\mathbf{q}}-i0^+}\right]$$

Q momenta (MPI q)

Xo bands (MPI c,v)

k momenta (MPI k)

Space DoF

$$\chi(\mathbf{q},\omega) = \left[I - \chi^0(\mathbf{q},\omega)v\right]^{-1}\chi^0(\mathbf{q},\omega)$$

```
X_all_q_ROLEs= "q k c v"   # [PARALLEL] CPUs roles (q,k,c,v)
X_all_q_CPU= "1 2 4 2"     # [PARALLEL] CPUs for each role
X_Threads= 4               # [OPENMP/GW] Number of threads
                           # for  response functions
X_all_q_LinAlg_INV = 32   # [PARALLEL] CPUs for matrix inv
```

num MPI tasks = 1 * 2 * 4 * 2
num threads/MPI-tasks = 4
Total num threads = 4 * (1 * 2 *4 *2)

MPI-c,v   best memory distribution
MPI-k   efficient, some mem repl
MPI-q   may lead to load unbalance
OpenMP   efficient, need extra mem

(13b) Parallelism: Correlation part of self energy

$$\Sigma^c_{n\mathbf{k}}(\omega) = \langle n\mathbf{k}|\Sigma^c|n\mathbf{k}\rangle = i\sum_m \int_{BZ} \frac{d\mathbf{q}}{(2\pi)^3} \sum_{\mathbf{G},\mathbf{G}'} \frac{4\pi}{|\mathbf{q}+\mathbf{G}|^2} \rho_{nm}(\mathbf{k},\mathbf{q},\mathbf{G})\rho^*_{nm}(\mathbf{k},\mathbf{q},\mathbf{G}') \times \int d\omega' G^0_{m\mathbf{k}-\mathbf{q}}(\omega-\omega')\epsilon^{-1}_{\mathbf{G}\mathbf{G}'}(\mathbf{q},\omega')$$

QP states
(MPI qp)

G bands
(MPI b)

Q transferred momenta
(MPI q)

Space DoF
(OMP SE_Threads)

```
SE_ROLEs= "q qp b"    # [PARALLEL] CPUs roles (q,qp,b)
SE_CPU= "1 2 8"       # [PARALLEL] CPUs for each role
SE_Threads= 4         # [OPENMP/GW] Number of threads
                      # for  self-energy
```

num MPI tasks = 1 x 2 x 8
num threads/MPI-tasks = 4
Total num threads = 4 x (1 x 2 x 8)
MPI-b best memory distribution
MPI-qp no communication
MPI-q leads to load unbalance
OpenMP very efficient

(14) IO: `yambo -V io`

```
StdoHash=  40                 # [IO] Live-timing Hashes
DBsIOoff= "none"              # [IO] Space-separated list of DB with NO I/O.
                                DB=(DIP,X,HF,COLLs,J,GF,CARRIERs,W,SC,BS,ALL)
DBsFRAGpm= "none"             # [IO] Space-separated list of +DB to be FRAG and -DB NOT to be FRAG.
                                DB=(DIP,X,W,HF,COLLS,K,BS,QINDX,
#WFbuffIO                     # [IO] Wave-functions buffered I/O
```

Parts of the WFs are stored by the node. Nodes communicate when these elements are needed. Memory heavy.

Fragments the database. Smaller files (e.g. ndb.em1s_fragment_*) are created instead of a large one (e.g. ndb.em1s).

Faster read/write operations in parallel runs

No ndb.* file is written. Example: DBsIOoff= “DIP” - ndb.dip_iR_and_P_fragment_* is not written, but stored in memory if Yambo needs it.

(15) Yambo-python code structure

yambopy - python module

io
YamboIn: read, write and manipulate yambo input files
YamboOut: read yambo output files and save in .json

analyse
analyse: read .json files generated with yamboout and plot them together
recipes: user contributed scripts

bse
read and manipulate yambo databases

dbs
YamboSaveDB: read ns.db1
YamboLatticeDB: read lattice parameters, symmetries and k-points from ns.db1
YamboElectronsDB: electronic states from ns.db1
YamboDipolesDB: dipole matrix elements from ndb.dip*
YamboStaticScreeningDB: static dielectric screening from ndb.em1s*
YamboQPDB: read the quasiparticle energies db ndb.QP
YamboGreenDB: read the Green's functions calculated using yambo
(...):

qepy

PwIn, PhIn, DynmatIn, ProjwfcIn
read, write and manipulate Quantum espresso input files (pw.x, ph.x, dynmat.x, projwfc.x respectively)

PwXML, ProjwfcXMl
read output files (datafile.xml, datafile-schema.xml, projwfc.xml)

schedulerpy

scheduler
submit and run codes with the same interface for:
bash, pbs, oar

yamboparser
parsing of files, databases, etc

yambopy
analysebse, plotem1s, analysegw, mergeqp, test, plotexciton

(15) Yambo-python quick-reference

qepy

```
from qepy import *

#create input file from scratch
qe = PwIn()

#input structure
qe.atoms = [['Si',[0.125,0.125,0.125]],
            ['Si',[-.125,-.125,-.125]]]
qe.atypes = {'Si': [28.086,"Si.pbe-mt_fhi.UPF"]}

#control variables
qe.control['prefix'] = "'si'" #strings need
double "''"
qe.control['wf_collect'] = '.true.' #logicals

#system
qe.system['celldm(1)'] = 10.3
qe.system['ecutwfc'] = 30
qe.system['occupations'] = "'fixed'"
qe.system['nat'] = 2
qe.system['ntyp'] = 1
qe.system['ibrav'] = 2

#electrons
qe.electrons['conv_thr'] = 1e-8

#write file
qe.write('si.scf')
```

yambopy

```
from yambopy import *

#create input file in 'bse' folder with SAVE
y = YamboIn('yambo -b -o b -k sex -y d -V
all',folder='bse')

# define variables
y['FFTGvecs'] = [30,'Ry'] # scalar + units
y['BndsRnXs'] = [1,30] # array with integers
y['BSEBands'] = [3,6] # array with integers
y['BEnRange'] = [[0,8],'eV'] # array + units
y['BEnSteps'] = 500 # numbers
y['KfnQPdb'] = 'E < yambo/ndb.QP' #strings

#write the file
y.write('bse/yambo_run.in')

#create ypp input file
y = YamboIn('ypp -e -a -V all',filename='ypp.in')

#read local file
y = YamboIn(filename='bse/yambo_run.in')

#analyse data in the bse folder
ya = YamboAnalyser('bse')
print(ya)

# plot eel and eps from BSE
ya.plot_bse('eel')
ya.plot_bse('eps')
```

schedulerpy

```
from schedulerpy import *

# scheduler 1 node and 4 cores
shell = Scheduler.factory(nodes=1,cores=4)

# scheduler of pbs type
shell = Scheduler.factory(scheduler='pbs')

#add commands to the shell
shell.add_command("echo 'hello world'")

#view commands on the screen
print( shell )

#write to a file
shell.write("commands.sh")

#submit or run the job
shell.run()
```

yambopy (bash)

```
$ yambopy #lists all possible commands
$ yambopy plotem1s #help about this command
```