NanoGW Tutorial

Weiwei Gao

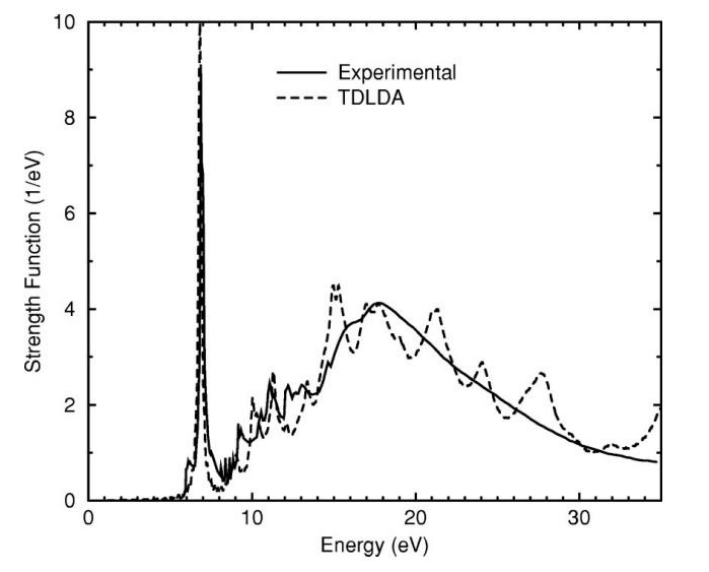
Postdoc, UT Austin

2020 Electronic Structure Workshop

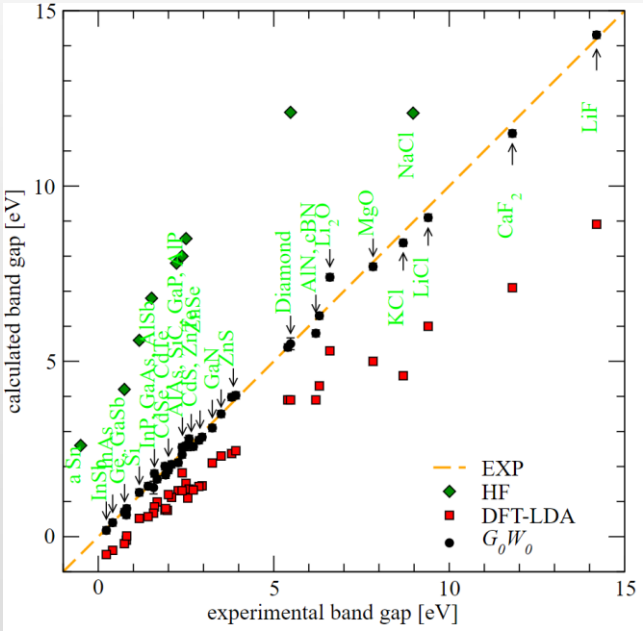
First-principles Method for Excited-states Properties

Time-dependent density functional theory (TDDFT) and GW approximation

- Widely used methods for calculating quasiparticle energies and optical properties

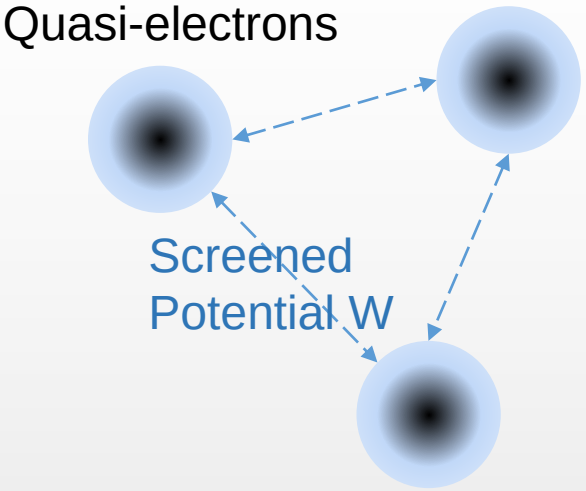


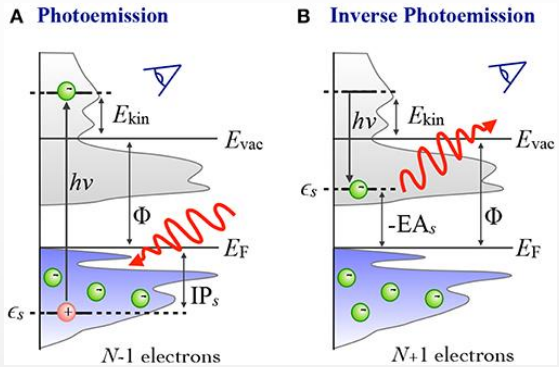
G.Onida et al, Review of Modern Physics (2002)



L. Hedin, Journal of Physics: condensed matter (1999)

Quasiparticle energies with GW approximation





Golze Dorothea et al, frontiers in chemistry, 2019

Kohn-Sham equation: $\left[-\frac{1}{2}\nabla^2 + V_H(r) + V_{ext}(r) + V_{XC}(r)\right]\psi^{KS}(r) = E^{KS}(r)$

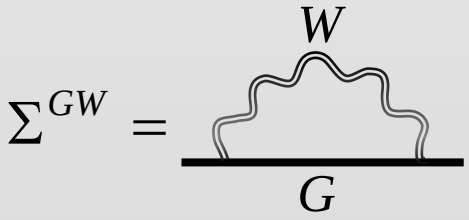
Quasiparticle: $\left[-\frac{1}{2}\nabla^2 + V_H(r) + V_{ext}(r)\right]\psi^{QP}(r) + \int dr'\Sigma(r,r';E_{nk})\psi^{QP}(r') = E_{nk}\psi^{QP}(r)$

$$E^{QP}_{n\vec{k}} = \varepsilon^{KS}_{n\vec{k}} + \langle n\vec{k}\,|\,\Sigma - V_{xc}\,|\,n\vec{k}\rangle$$

$$\Sigma^{GW} = iGW$$

G : electron Green function

W : screened Coulomb interaction



Important references

- Review of GW approximation, TDLDA, and Bethe-Salpeter equations:

L. Hedin, Journal of Physics: condensed matter (1999)

G. Onida, L. Rening, A. Rubio, Review of Modern Physics (2002)

- First-principles GW/BSE calculations:

M. Hybertsen, S. Louie, Physical Review B (1986)

M. Rohlfing and S. Louie, Physical Review B, 62, 4927 (2000)

- TDLDA,GW and BSE calculations with real-space formalism (NanoGW):

M. Tiago, J. Chelikowsky, Physical Review B **73**, 205334 (2006)

Full-frequency GW: orbital-based formulation

- Input: Kohn-sham wave functions calculated with real-space based code PARSEC
- NanoGW: efficient for small-medium sized mol and clusters
- Solving Casida's equation:

$$R^{\frac{1}{2}}\,[R + 4(K^{x} + K^{LDA})]R^{\frac{1}{2}}\;\; X = \omega_s^2 \;\; X$$

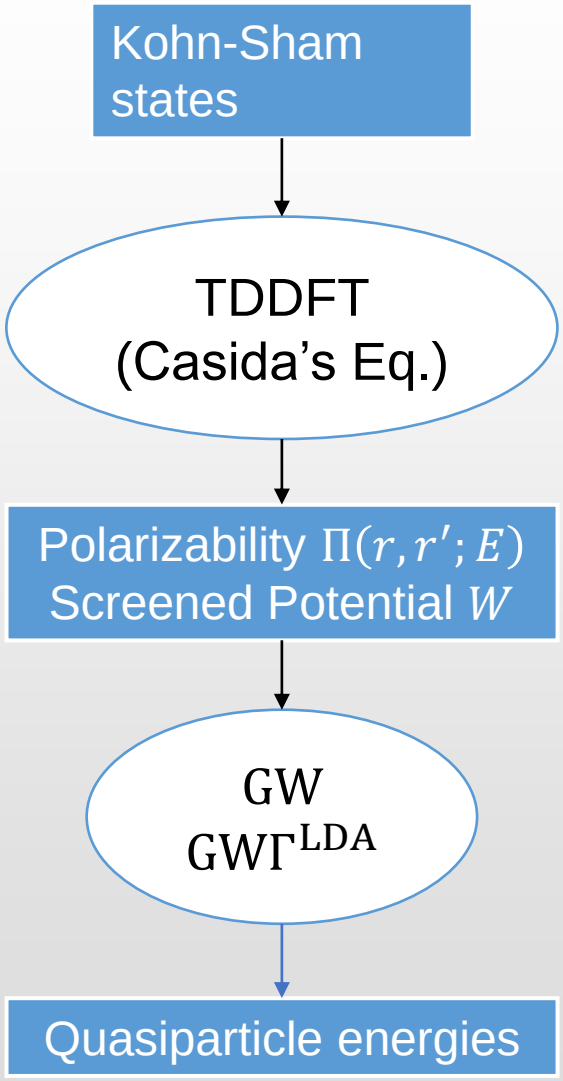
$$\Downarrow$$

$$\Pi(r, r'; E) = 2\sum_s \rho_s(r)\rho_s(r')\left[\frac{1}{E - \omega_s + i0^+} - \frac{1}{E + \omega_s - i0^+}\right]$$

- Full-frequency GW

  (analytic, no frequency-grids)

$$\langle j|\Sigma(E)|j'\rangle = -\sum_v^{occ} K^{x}_{vj,vj'} + 2\sum_n^{all}\sum_s \frac{V^s_{nj}V^s_{nj'}}{E - \epsilon_n - \omega_s\eta_n}$$



M. Tiago and J. Chelikowsky,
PHYSICAL REVIEW B **73**, 205334 (2006)

Directories of source files and examples

- Source files:

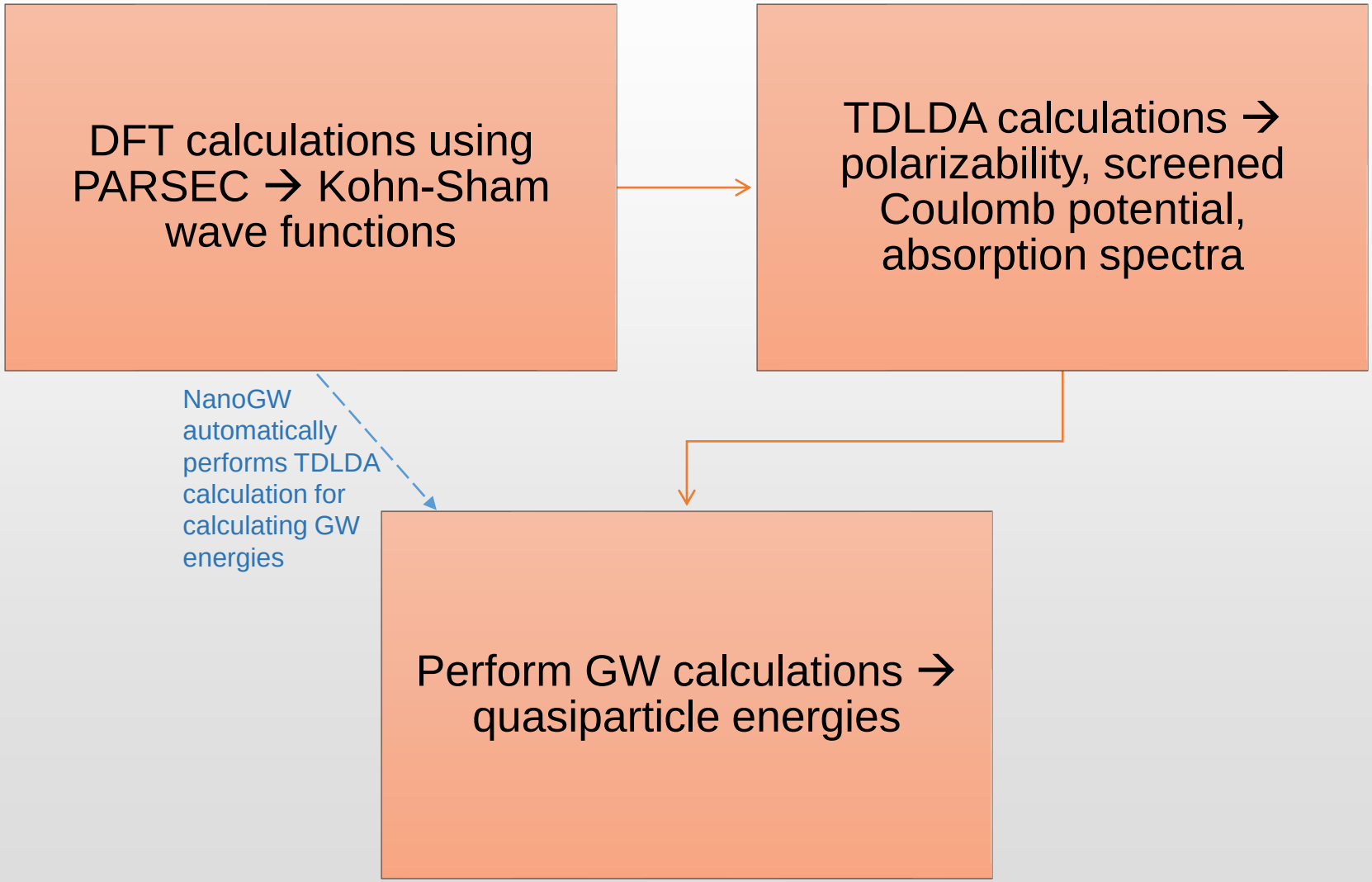
/project/projectdirs/m3034/vESW_2020/NanoGW/src/nanogw

- Examples:

/project/projectdirs/m3034/vESW_2020/NanoGW/ESW_tutorial/Benzene

/project/projectdirs/m3034/vESW_2020/NanoGW/ESW_tutorial/si29h36

General steps



Benzene (C_6H_6): DFT calculation

STEPS:

- Copy “/project/projectdirs/m3034/vESW_2020/NanoGW/ESW_tutorial/Benzene” to your own directory

- Perform DFT calculations with PARSEC to generate Kohn-Sham wave functions under directory “Benzene/”
  - ✓ run “sbatch job.cori”
  - ✓ Estimated running time: 1 minutes with 1 Haswell nodes

Benzene (C_6H_6): DFT calculation

- Result for the DFT calculation (See parsec.out file)

Fermi level at -0.2150 [Ry]

State	Eigenvalue [Ry]	Eigenvalue [eV]	Occup.	Repr.
1	-1.5579705737	-21.1974360311	1.0000	1
2	-1.3526423851	-18.4037817634	1.0000	7
3	-1.3521449985	-18.3970144201	1.0000	6
4	-1.0883868748	-14.8083741415	1.0000	1
5	-1.0868427780	-14.7873654684	1.0000	4
6	-0.9468613331	-12.8828059257	1.0000	1
7	-0.8205311232	-11.1639823561	1.0000	7
8	-0.8041143068	-10.9406184357	1.0000	6
9	-0.7544169286	-10.2644458477	1.0000	7
10	-0.7521038344	-10.2329743500	1.0000	6
11	-0.6681156301	-9.0902476394	1.0000	8
12	-0.6088613618	-8.2840459161	1.0000	1
13	-0.6063784954	-8.2502645333	1.0000	4
14	-0.4679289528	-6.3665477457	1.0000	3
15	-0.4677973842	-6.3647576502	1.0000	2
16	-0.0926176054	-1.2601366157	0.0000	5
17	-0.0920888711	-1.2529427626	0.0000	8
18	-0.0323023328	-0.4394990797	0.0000	1
19	0.0128987905	0.1754983637	0.0000	6

Benzene (C_6H_6): TDLDA calculation

Perform linear-response TDLDA calculations in “Benzene/tdlda”

✓ Link *POTRE.DAT, parsec.in and parsec.dat files

✓ Run “sbatch job.cori”

✓ Running time: 2 minutes with 1 haswell nodes

```
# C6H6
distribute_representations     4
buffer_size 2.0
cache_size  30000
distribute_wavefunctions   16

lessmemory
fastselect
# --- For ISDF method ----
#doisdf
#num_isdf_points    3050
#intp_type 3
#no_lda_kernel
begin tdlda_valence
range 1 15
end tdlda_valence
begin tdlda_conduction
range 16 550
end tdlda_conduction
```

“distribute_representations” should be divided by the total number of symmetry

distribute_representation*distribute_wavefunction = # of mpi_tasks

Specify the range of valence and conduction states

Lines start with “#” are ignored by the code

Benzene (C_6H_6): TDLDA calculation

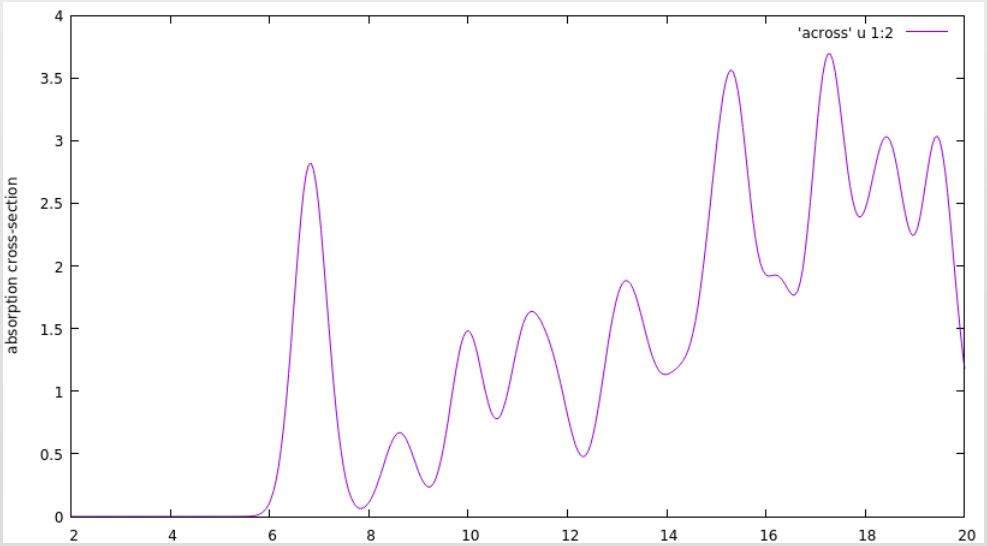
✓ Use post-processing tool “absp” to calculate absorption cross-section:

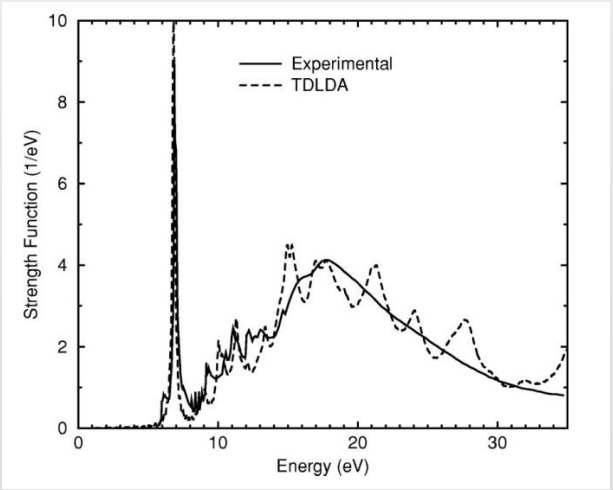
> cp eigenvalue_lda eigenvalue

> add smearing parameter “0.3” at the end of the first row of “eigenvalue” file

> run “./absp” and get output file “across”

> plot absorption cross-section (filename “across”) with GNUPLOT





G.Onida et al, Review of Modern Physics (2002)

Benzene (C_6H_6): GW calculation

- Perform GW calculation with NanoGW in “Benzene/sigma”

✓ Link *POTRE.DAT, parsec.in and parsec.dat files

✓ Run “sbatch job.cori”

✓ Running time: 4 minutes with 1 knl node

✓ GW energies are output “sigma_001_0000”

OUTPUT SELF-ENERGY DIAGONAL MATRIX ELEMENTS (eV)

n	occ	E_dft	V_xc	Sigma_x	Sigma_sx	Sigma_c	Sigma_g	Sigma-V_xc	zfac	Im[Sigma]	E_0	E_qp
15	1.000	-6.365	-13.046	-15.614	5.919	-0.385	0.000	-2.953	0.816	0.008	-6.365	-8.821
16	0.000	-1.260	-12.427	-7.493	7.391	-2.034	0.000	2.901	0.842	-0.007	-1.260	1.217

Benzene (C_6H_6): GW calculation

```
# C6H6
distribute_representations     4
buffer_size 2.0
cache_size  30000
distribute_wavefunctions   16

lessmemory
fastselect
# --- For ISDF method ----
#doisdf
#num_isdf_points    3050
#intp_type 3
no_lda_kernel
begin tdlda_valence
range 1 15
end tdlda_valence
begin tdlda_conduction
range 16 550
end tdlda_conduction

max_number_states 550
dynamic_energy_resolution 0.005
energy_data_points 301
energy_range 30
begin diag
15
16
end diag
only_diagonal
```

Smearing parameter for energy denominator

We calculate $\Sigma(E)$ for a range of E
The number of sampled energy points
The range of energy

The range of states for which you want to calculate quasiparticle energies

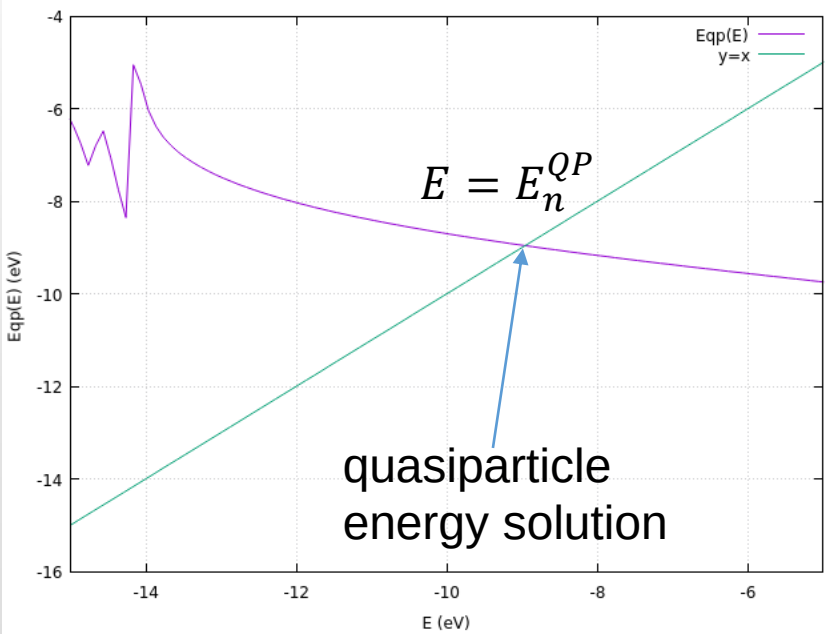
Benzene (C_6H_6): GW calculation

- We can also plot the quasiparticle energies $\mathrm{E}_{\mathrm{n}}^{\mathrm{QP}}(E) = E_n^{KS} + \langle n|\Sigma^{GW}(E) - V_{xc}|n\rangle$ with respect to energy E
- The quasiparticle energy $E_n^{QP}(E)$ need to be calculated at $E = E_n^{QP}$!!

  > run "./chkpt_bin_asc" to get "Eqp*.dat" and "Spectral*.dat"

  > In GNUPLOT, run script:

```
set xr [-15:-5]
set ylabel "Eqp(E) (eV)"
set xlabel "E (eV)"
f(x)=x
plot "Eqp15-spin1.dat" u 2:3 w l t 'Eqp(E)',
     f(x) w l t 'y=x'
```



Benzene (C_6H_6): use ISDF to speed up GW

- Speed up the TDLDA and GW calculation with interpolative separable density fitting (ISDF), in this example, we use around 3000 interpolation points to fit orbital pairs $\phi_i(r)\phi_j(r)$.

STEPS (similar as previous runs)

- ✓ Under “Benzene/tdlda-isdf”, run “sbatch job.cori” (around 1 minutes)
- ✓ Under “Benzene/sigma-isdf”, run “sbatch job.cori” (around 1 minutes)
- ✓ Compare the calculation results with those calculated without ISDF, how large are the differences?
- ✓ For more information about ISDF method, see:

W. Gao and J. Chelikowsky, J. Chem. Theory Comput. 2020, 16, 4, 2216–2223

Benzene (C_6H_6): use ISDF to speed up GW

- There are typically three additional parameters for ISDF approach

```
# --- For ISDF method ----
doisdf
num_isdf_points    1200
intp_type 3
```

Enable ISDF method

Number of interpolation points, $20\sqrt{N_v \cdot (N_c + N_v)}$ typically yields $\pm$ 0.03 eV error

Use 3 here for now, other options do not work

Silicon cluster $Si_{29}H_{36}$

- copy “/project/projectdirs/m3034/vESW_2020/NanoGW/ESW_tutorial/si29h36” to your own working directory

- Perform DFT calculations with PARSEC to generate Kohn-Sham wave functions under directory “si29h36/”

  > run “sbatch job.cori”

  > Estimated running time: around 30 seconds with 1 haswell nodes (we use a sparse real-space grid here, so it runs faster than the “Benzene” example)

Silicon cluster $Si_{29}H_{36}$

- Perform a GW calculation with NanoGW in “si29h36/sigma”

✓ Link *POTRE.DAT, parsec.in and parsec.dat files

✓ Run “sbatch job.cori”

✓ Running time: 11 minutes with 1 haswell node

✓ Example results:

OUTPUT SELF-ENERGY DIAGONAL MATRIX ELEMENTS (eV)

n	occ	E_dft	V_xc	Sigma_x	Sigma_sx	Sigma_c	Sigma_g	Sigma-V_xc	zfac	Im[Sigma]	E_0	E_qp
73	1.000	-6.396	-10.514	-12.882	5.722	0.557	0.000	-1.811	0.801	0.012	-6.396	-7.874
76	1.000	-6.140	-10.909	-13.306	6.006	0.581	0.000	-1.816	0.807	0.011	-6.140	-7.630
77	0.000	-2.044	-8.762	-4.899	4.074	-1.944	0.000	1.919	0.841	-0.008	-2.044	-0.406
78	0.000	-2.017	-8.442	-4.566	3.787	-1.977	0.000	1.899	0.842	-0.009	-2.017	-0.393

You may also try a GW calculation with ISDF in “si29h36/sigma”, which takes about 2 minutes with 1 haswell node.

Thank you!