Practical BSE Calculations with BerkeleyGW + Octopus

Felipe H. da Jornada
Department of Physics – University of California, Berkeley
and Lawrence Berkeley National Laboratory

David A. Strubbe
Department of Material Science and Engineering – Massachusetts Institute of Technology

TDDFT Benasque Workshop – Jan.12th, 2014
Summary

#1 – Theory and Algorithms

#2 – Typical BSE Workflow in BerkeleyGW

#3 – Issues Unique to the BSE Code
Theory and Algorithms

\[ \text{GW-BSE} = \quad + \quad + \quad + \quad + \]

\[ + \quad + \quad + \quad + \quad \cdots \]

\[ = \quad + \quad + \quad + \quad + \quad + \quad \cdots \]

\[ \cdots = \quad + \quad + \quad + \quad + \quad \cdots \]
Theory Review: Optical Absorption

No electron-hole interactions

Quasi-electron: $|ck\rangle$

$H_{int} \sim A \cdot \hat{v}$

$\varepsilon_2(-q, \omega) \propto \sum_{vck} |\langle vk + q | \hat{v} | ck \rangle|^2 \delta[\omega - (E_{ck} - E_{vk+q})]$

Quasi-hole: $|vk + q\rangle$

With electron-hole interactions

Correlated electron-hole pair: $|S\rangle$

$\varepsilon_2(-q, \omega) \propto \sum_{S} |\langle 0 | \hat{v} | S \rangle|^2 \delta[\omega - \Omega_S]$

$|S\rangle = \sum_{vck} A_{vck}^S |vk + q\rangle \otimes |ck\rangle$

Solutions of the Bethe-Salpeter equation (BSE)
Bethe Salpeter Equation (BSE)

- From Prof. Tokalty’s talk to BerkeleyGW:

\[ L = (1 - L_0 K)^{-1} L_0 \]
\[ L^{-1} = L_0^{-1} - K \]

Spectral Representation:
\[ -L_0^{-1} = E_C - E_V - \omega \]
\[ -L^{-1} = \left( E_C - E_V + K \right) - \omega \]

- Absorption spectrum with excitonic effects → diagonalize BSE Hamiltonian:

\[ [H]_{(vck),(v'c'k')} \]
\[ [H] = [E_C - E_V] + [K] \]

\[ \text{dense “kernel”} \]
\[ \sim \text{potential term} \]
\[ \text{diagonal} \]
\[ \sim \text{kinetic term} \]

**Challenge**: compute quasiparticle corrections and kernel matrix elements on a **very fine k-grid**!
BerkeleyGW Interpolation Scheme

BerkeleyGW solution:
Interpolate QP energies and BSE kernel

- Step 1: Expand fine WFNs in terms of coarse WFNs
  \[ u_{n\mathbf{k}_f} = \sum_{n'} C^\mathbf{k}_{n',n} u_{n'}_{\mathbf{k}_c} \]

- Step 2: Interpolate QP energies and matrix elements
  \[ \langle v_{c\mathbf{k}_f} | K | v'_{c'\mathbf{k}_f'} \rangle = \sum_{n_1,n_2,n_3,n_4} C^{\mathbf{k}_c}_{c,n_1} C^{*\mathbf{k}_c}_{v,n_2} C^{*\mathbf{k}_c'}_{c',n_3} C^{\mathbf{k}_c'}_{v',n_4} \langle n_2 n_1 \mathbf{k}_c | K | n_4 n_3 \mathbf{k}_c' \rangle \]
BerkeleyGW Interpolation Scheme

- In practice: trading bands for k-points

How to get a good interpolation?
- Include a large number of bands from the coarse grid!
BerkeleyGW QP Interpolation

- BerkeleyGW also performs a linear interpolation for QP corrections.

- Linear interpolation + expansion over bands:
  - Captures \((nk)\)-dependent QP correction and band crossing
  - Very smooth interpolation of band structure
  - Robust scheme, and very few parameters

- This is how inteqp.x works!

Theory and Algorithms

Key Points

- Interpolation scheme: trading bands for k-points
- Interpolation of kernel and QP corrections
Typical BSE Workflow in BerkeleyGW
Goal

Goal: Diagonalize BSE Hamiltonian on a fine grid

\[ [H]_{\text{fi}} = [E_c - E_v]_{\text{fi}} + [K]_{\text{fi}} \]
BerkeleyGW Workflow

**Step 1:** Calculate QP–corrected band structure on a coarse grid

\[ \{E_c\}_\text{co}, \{E_v\}_\text{co}, \]

**Step 2:** Calculate BSE kernel on the same coarse grid

\[ [K]_\text{co} \]

**Step 3:** Interpolate to a fine k-grid and build BSE Hamiltonian…

\[ [H]_\text{co} \Rightarrow [H]_\text{fi} \]

… and diagonalize BSE Hamiltonian

\[ \text{evals } [H]_\text{fi} \Rightarrow \varepsilon_2 \]

(Not shown: mean-field, epsilon, convergence)
1. Sigma

**Step 1:** Calculate QP–corrected band structure on a coarse grid

\[ \{E_c\}_\text{co}, \{E_v\}_\text{co}, \]

- Same procedure done in previous sessions.

- Recommended: eqp.dat
  - Calculate QP energies on all k-points from WFN inner.
  - Use the script eqp.py to generate eqp.dat file → no human intervention!

- Also possible: scissors operators, less reliable (not covered here)
1. Sigma

Sample sigma.inp (assuming we are using eqp.dat)

```plaintext
screened_coulomb_cutoff <>
bare_coulomb_cutoff <>

number_bands <>
band_occupation <>

band_index_min <>
band_index_max <>

screening_semiconductor
number_kpoints <>
begin kpoints
  <put all k-points from WFN_INNER here>
end
```

Note the two different # of bands:

\[ E_{nk}^{QP} = \langle n_k | \Sigma | n_k \rangle \]

Remember to calculate Sigma on more bands because of the interpolation!
2. Kernel

**Step 2:** Calculate BSE kernel on the same coarse grid

\[
[K]_{co}
\]

- Time consuming!
  - Computes \((n_v n_c n_k)^2\) matrix elements

- Recommended:
  - Use same WFN\_co as in Sigma (WFN\_inner)
2. Kernel

Sample kernel.inp

```plaintext
number_val_bands
number_cond_bands
screened_coulomb_cutoff
use_symmetries_coarse_grid
screening
```

Remember to calculate Kernel on more bands because of the interpolation!

Must be ≤ than the # of bands used in Sigma.

You’ll typically want to use symmetries here, so put:

`use_symmetries_coarse_grid`
Absorption needs same coarse WFN_co from Kernel/Sigma

Absorption also need two fine WFN files:
- WFN_fi: for conduction states
- WFNq_fi: for valence states

Good practice: use randomly-shifted k-grids
- This maximizes the number of inequivalent transitions you capture.

Step 3: Interpolate to a fine k-grid and build BSE Hamiltonian...

\[ [H]_{co} \Rightarrow [H]_{fi} \]

... and diagonalize BSE Hamiltonian

evals \([H]_{fi} \Rightarrow \varepsilon_2\)
3. Absorption

Sample absorption.inp

```plaintext
diagonalization

number_val_bands_coarse
number_cond_bands_coarse
number_val_bands_fine
number_cond_bands_fine

coarse_grid_points

use_symmetries_coarse_grid
no_symmetries_fine_grid
no_symmetries_shifted_grid

screening_semiconductor

use_velocity
q_shift 0.0 0.0 0.0 0.001

gaussian_broadening
energy_resolution 0.15

eqp_co_corrections
```

Same as used in kernel.

We interpolate to these bands!

How many k-points in the coarse grid after unfolding BZ?

Typical values.

Recommended!

\[ \mathbf{k}_{\text{WFNf}} + \mathbf{q}_{\text{shift}} = \mathbf{k}_{\text{WFNd}} \]

Broaden each delta function.

Interpolate eqp_co.dat to eqp.dat
3. Absorption – Workflow

- absorption.inp
- epsmat
- eps0mat
- eqp*.dat
- bsgedmat
- bsexmat
- WFN_co
- WFN_fin
- absorption_noeh.dat
- absorption_eh.dat
- eigenvalues.dat
- eigenvectors.dat
- d?mat_norm.dat
Typical BSE Workflow in BerkeleyGW

Key Points

- BSE codes separated into two parts:
  - Kernel.x: calculates kernel on coarse grid
  - Absorption.x: interpolates and diagonalizes \([H]\)

- `number_*_bands_coarse, number_*_bands_fine`. 
Issues Unique to the BSE Code

1. Velocity Operator

2. Finite Systems + Octopus

3. Estimating the Quality of the Interpolation

4. Analyzing Exciton Files

5. Convergence!
1. Velocity Operator

\[ \varepsilon_2(-q, \omega) \propto \sum_S |\langle 0| \hat{v}|S \rangle|^2 \delta[\omega - \Omega_S] \]  
\[ \langle 0| \hat{v}|S \rangle = \sum_{vck} A^S_{vck} \langle v\mathbf{k} + q| \hat{v}|c\mathbf{k} \rangle \]

- Because of non-local pseudopotential and QP corrections, the velocity operator is not the same as the momentum!

\[ \hat{v} = i[H, \hat{r}] = \hat{p} + i[V, \hat{r}] \]

---

**use_velocity**

- Recommended option!
- Needs WFN\_fi and WFNq\_fi.
- Specify q-shift: \( \mathbf{k}_{WFNfi} + \mathbf{q}_{\text{shift}} = \mathbf{k}_{WFNqfi} \)

\[ \langle 0| \hat{v}|S \rangle \approx \frac{\Omega_S}{q} \sum_{vck} A^S_{vck} \langle v\mathbf{k} + q| e^{-i\mathbf{q}\cdot\mathbf{r}}|c\mathbf{k} \rangle \]

---

**use_momentum**

- Not recommended!
- Needs only WFN.
- Specify polarization \( \mathbf{e}_\lambda \) of \( \hat{v} \).

\[ \langle v\mathbf{k} + q| \hat{v}|c\mathbf{k} \rangle \approx \langle v\mathbf{k}| \hat{p}_\lambda |c\mathbf{k} \rangle \]
2. Finite Systems + Octopus

- For finite systems, Octopus can calculate the single-particle velocity matrix elements directly!
  - No need for two different WFN files and a q-shift in the absorption code.
  - BerkeleyGW can calculate $\varepsilon_2(q = 0)$ “exactly”.

- In BerkeleyGW:
  
  ```
  ...  
  read_vmtxel
  use_momentum
  polarization <...
  skip_interpolation
  eqp_corrections
  ...  
  ```

  Read velocity matrix elm from file
  Tell BerkleyGW that there’s no q-shift.
  **Note:** this doesn’t affect calculation of matrix elms because of the read_vmtxel flag.
  There are no k-points, so there’s no interpolation! Use directly the eqp.dat file
  (don’t interpolate eqp_co.dat → eqp.dat)
3. Quality of the Interpolation

- How to measure the quality of WFN expansion?
- If we include $\infty$ bands:

$$\sum_{n'} |C_{n,n'}^{k_{co}}|^2 = 1$$

- Finite basis set – normalization is reported in files d?mat_norm.dat:

| k-point | ik_co | v | dist | $|dvv|^2$ |
|---------|-------|---|------|----------|
| ( 0.059, 0.046, 0.039 ) | 1 | 1 | 0.054 | 0.987006 |
| ( 0.059, 0.046, 0.039 ) | 1 | 2 | 0.054 | 0.953488 |
| ( 0.059, 0.046, 0.039 ) | 1 | 3 | 0.054 | **0.892665** |
| ( 0.059, 0.046, 0.164 ) | 2 | 1 | 0.139 | 0.923182 |

- How to get a good interpolation?
  - Include a large number of bands from the coarse grid!
  - Start from a fine enough grid

Before renormalization of coefficients.
4. Analyzing Excitons

- Optical spectrum $\varepsilon_2(\omega), \varepsilon_1(\omega)$:
  - `absorption_noeh.dat`: GW-RPA without local fields
  - `absorption_eh.dat`: GW-BSE with local fields

- Eigenvalues of the BSE equation $\Omega_S$:
  - `eigenvalues.dat`: useful to see if there are degeneracies, splitting, etc.

- Where the exciton is coming from:
  - `summarize_eigenvectors.x`
  - Need to set the flag `write_eigenvalues` in `absorption.inp`
5. Convergence!

There are 4 convergence parameters in a typical BSE calculation:

- # of k-points in the fine grid
- # of bands in the fine grid
- # of k-points in the coarse grid
- # of bands in the coarse grid

Make sure you converge your calculations!
Issues Unique to the BSE Code

1. Velocity operator

Key Points

- Why we need 2 WFN files: velocity operator
- **Convergence!**

5. Convergence!
Let’s Put the Donkey to Work!

Predicting quasiparticle band structures since 1985.