An Overview of the BerkeleyGW Software Package

Mauro Del Ben
Applied Mathematics & Computational Research Division (LBNL)

2022 School on Electron-Phonon Physics From First Principle
June 13 - 19, 2022, University of Texas, Austin
Outline

1. Introduction
2. Overview of the BerkeleyGW software package
3. The GW+BSE workflow in BerkeleyGW
4. Summary
Introduction
Materials Science/Chemistry and HPC

Grätzel cells: Oxide/Organic Interfaces

Defects in crystals: qubits/quantum computers
https://www.nist.gov/programs-projects/diamond-nv-center-magnetometry

Chemical reaction at interfaces: Catalysis

Cheap, reliable and sustainable photovoltaics

Mat. Sci & Chem apps, such as VASP, Quantum ESPRESSO, QMCPACK, NWchem, BerkeleyGW, CP2K, etc... **heavily use HPC facilities**
Used to **study and understand** the fundamental electronic properties of **materials**: *necessary to design the components of novel devices*

- Applications: Quantum Computers, Batteries, Photovoltaics, Catalysis, etc...
Materials Science/Chemistry and HPC

Grätzel cells: Oxide/Organic Interfaces

Cheap, reliable and sustainable photovoltaics

Defects in crystals: qubits/quantum computers
https://www.nist.gov/programs-projects/diamond-nv-center-magnetometry

Chemical reaction at interfaces: Catalysis

Density Functional Theory (DFT) the workhorse for over three decades
• Excellent compromise between accuracy and computational efficiency
• Ground state theory: often problematic for excited state phenomena
Excited State Properties of Complex Materials

Focus shift from ground to excited state properties

Example: Divacancy point defect in crystalline silicon, prototype of a solid-state Qubit

Accuracy beyond DFT: GW and GW+BSE
The GW+BSE: State of the Art

The GW+BSE method represents one of the most effective and accurate approach to predict excited-state properties in a wide range of materials.

The application of GW+BSE to routine calculations is often perceived as prohibitive due to higher computational complexity.

Pushing GW+BSE Forward:

• Develop new and state of the art methods
  • Improve accuracy: explore new physics
  • Reduce time to solution / scaling wrt system size: tackle larger applications
• Improve implementation’s performance: from desktop to leadership class HPC systems
• Maintain a well-tested, documented and production quality software package
The BerkeleyGW Software Package
BerkeleyGW compute the electronic excited-state properties of materials via GW, Bethe-Salpeter equation (BSE) and beyond

BerkeleyGW: GW+BSE methods and beyond
Utilities: post-processing, analysis, visualization, verification

Modular structure, common file formats, input style, output etc.

QE Octopus SIESTA ABINIT EPM RMDFT PARATEC PARSEC JDFTx

Interface & Parabands

https://berkeleygw.org/
BerkeleyGW compute the electronic excited-state properties of materials via GW, Bethe-Salpeter equation (BSE) and beyond.

- QE
- Octopus
- SIESTA
- ABINIT
- EPM
- RMDFT
- PARATEC
- PARSEC
- JDFTx

BerkeleyGW: GW+BSE methods and beyond

Utilities: post-processing, analysis, visualization, verification

Modular structure, common file formats, input style, output etc.

https://berkeleygw.org/
BerkeleyGW compute the electronic excited-state properties of materials via GW, Bethe-Salpeter equation (BSE) and beyond

BerkeleyGW: GW+BSE methods and beyond

Utilities: post-processing, analysis, visualization, verification

Modular structure, common file formats, input style, output etc.

https://berkeleygw.org/
BerkeleyGW: developments are focuses on GW+BSE methodologies
BerkeleyGW Highlights

- Supports a large set of Mean-Field codes: PARATEC, Quantum ESPRESSO, PARSEC, SIESTA, Octopus, ABINIT, RMGDFT
- Supports 3D, 2D, 1D and Molecular Systems. Coulomb Truncation
- Support for Semiconductor, Metallic and Semi-Metallic Systems
- Efficient Algorithms and Use of Libraries for (Pre-) Exascale HPC systems.
- Massively Parallel. Scales to 100,000 CPUs, and recently up to 10,000 of GPUs.
New in Version 3.0

- Full, 2-component spinor support enabling spin-orbit coupling (SOC)
- Exciton finite momentum Q for exciton band structures
- Broader DFT starting-point support including hybrid, meta-GGA, DFT+U, etc...
- First public release of GPU acceleration for Epsilon and Sigma
- I/O performance improvements: full HDF5 workflow support of wavefunctions
- New tools for wavefunction self-consistent calculations
- Improved performance, tools and documentation for new and existing features
Four major modules: epsilon, sigma, kernel and absorption; *why*?

(1) Each have different data layout, computational cost and memory requirements

(2) Intermediates from each modules reused by others in multiple runs

http://manual.berkeleygw.org/3.0/overview-workflow/
BerkeleyGW Workflow

1-epsilon

2-sigma

3-kernel

4-absorption

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

http://manual.berkeleygw.org/3.0/overview-workflow/
**Epsilon**: Generate the dielectric function and its frequency dependence

**Sigma**: Solve Dyson's equation for quasiparticle energies

**Kernel**: Compute BSE kernel matrix elements on a coarse k-point grid

**Absorption**: Interpolate BSE kernel matrix elements onto a fine k-point grid, diagonalize the BSE Hamiltonian, and compute optical absorption spectrum

http://manual.berkeleygw.org/3.0/overview-workflow/
Knowing the basic computational kernels is key to understand how BerkeleyGW works

Basic computational motifs implemented in BerkeleyGW:
- Large distributed matrix-multiplication over short and fat matrices
- Large distributed linear algebra: LU decomposition, matrix inversion, eigen-decomposition, etc...
- Many, non-distributed fast Fourier transformations (FFT)
- Dimensionality reduction and low-rank approximations
- Parallel I/O of rank-2 -3 and -4 tensors

BerkeleyGW implements these kernels exploiting an hybrid parallelization strategy:
- multi-node (MPI)
- multi-core (OpenMP)
- multi-GPU (CUDA/OpenACC/OpenMP-target)
The goals of the BerkeleyGW developers community are:

1. Distribute a highly performant and production quality set of computational tools

2. Optimize components for pre-exascale and exascale HPC systems to enable the study of systems with increasing complexity

3. Integrate new features from developers all around the world into the package in a well-tested / production-quality way

4. Maintain a vibrant user and developer community that helps drive further developments
Overview Summary

Divacancy defect in silicon ~11,000 electrons and over 2,700 atoms: time to solution of the ~10s of mins.


By software optimization of BerkeleyGW on leadership class HPC systems the application of GW to systems of increasing size closely follow the Moore's law.
BerkeleyGW Workflow

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

http://manual.berkeleygw.org/3.0/overview-workflow/
The GW Workflow: Epsilon + Sigma

The BSE Workflow: Kernel + Absorption

http://manual.berkeleygw.org/3.0/overview-workflow/
The GW Workflow: Epsilon + Sigma

Dynamical properties of electrons as solution of Dyson's equation:

\[ h_0(r) \phi_n(r) + \int \Sigma(r, r'; E_n) \phi_n(r') dr' = E_n \phi_n(r) \]
The GW Workflow: Epsilon + Sigma

Dynamical properties of electrons as solution of Dyson's equation:

\[ h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r}, \mathbf{r}'; E_n)\phi_n(\mathbf{r}')d\mathbf{r}' = E_n \phi_n(\mathbf{r}) \]
The GW Workflow: Epsilon + Sigma

Dynamical properties of electrons as solution of Dyson's equation:

\[ h_0(r)\phi_n(r) + \int \Sigma(r, r'; E_n)\phi_n(r')dr' = E_n\phi_n(r) \]

GW Self-Energy Operator \( \Sigma \): non-Hermitian, non-local, frequency dependent

(Note: In DFT, the role of self-energy is replaced by static and local \( V_{xc}(r) \))
The GW Workflow: Epsilon + Sigma

Dynamical properties of electrons as solution of Dyson's equation:

\[ h_0(r)\phi_n(r) + \int \Sigma(r, r'; E_n)\phi_n(r')dr' = E_n\phi_n(r) \]

GW Self-Energy Operator \( \Sigma \): non-Hermitian, non-local, frequency dependent
(Note: In DFT, the role of self-energy is replaced by static and local \( V_{xc}(r) \))

High GW Computational Cost in Two Major Bottlenecks:
The GW Workflow: Epsilon + Sigma

Dynamical properties of electrons as solution of Dyson's equation:

\[ h_0(r) \phi_n(r) + \int \Sigma(r, r'; E_n) \phi_n(r') dr' = E_n \phi_n(r) \]

GW Self-Energy Operator \( \Sigma \): non-Hermitian, non-local, frequency dependent
(Note: In DFT, the role of self-energy is replaced by static and local \( V_{xc}(r) \))

High GW Computational Cost in Two Major Bottlenecks:

- **Epsilon**: Inverse Dielectric Matrix \( O(N^4) \)
The GW Workflow: Epsilon + Sigma

Dynamical properties of electrons as solution of Dyson's equation:

\[ h_0(r)\phi_n(r) + \int \Sigma(r, r'; E_n)\phi_n(r')dr' = E_n\phi_n(r) \]

GW Self-Energy Operator \( \Sigma \): non-Hermitian, non-local, frequency dependent
(Note: In DFT, the role of self-energy is replaced by static and local \( V_{xc}(r) \))

High GW Computational Cost in Two Major Bottlenecks:

- **Epsilon**: Inverse Dielectric Matrix \( O(N^4) \)
- **Sigma**: Self-Energy Matrix Elements \( O(N^4) \)
Epsilon: Inverse Dielectric Function (Matrix)

Three major computational steps: input $\psi_{mk}$, $\epsilon_{mk}$, \{q-points\}, \{\omega_i\}
Epsilon: Inverse Dielectric Function (Matrix)

Three major computational steps: input $\psi_{mk}$, $\epsilon_{mk}$, \{q-points\}, $\{\omega_i\}$

1. Calculate plane-waves matrix elements (FFT's) $O(N^3)$

$$M_{jak}^G(q) = \langle \psi_{j-k+q}\mid e^{i(G+q)\cdot r}\mid \psi_{ak} \rangle$$
Epsilon: Inverse Dielectric Function (Matrix)

Three major computational steps: input $\psi_{mk}, \epsilon_{mk}, \{q\text{-points}\}, \{\omega_{i}\}$

1. Calculate plane-waves matrix elements (FFT’s) $O(N^3)$

$$M_{jak}^{G}(q) = \langle \psi_{j}^{k+q} | e^{i(G+q) \cdot r} | \psi_{ak} \rangle$$

2. Calculate RPA polarizability (Matrix-Multiplication/ZGEMM) $O(N^4)$

$$\chi(q, \omega_{i}) = M(q)^{\dagger} \Delta_{jak}(\epsilon_{jk}, \epsilon_{ak}, q, \omega) M(q)$$
Epsilon: Inverse Dielectric Function (Matrix)

Three major computational steps: input $\psi_{m_k}$, $\epsilon_{m_k}$, {q-points}, {$\omega_i$}

1. Calculate plane-waves matrix elements (FFT’s) $O(N^3)$
   
   $M^G_{jak}(q) = \langle \psi_{j_k+q} | e^{i(G+q) \cdot r} | \psi_{a_k} \rangle$

2. Calculate RPA polarizability (Matrix-Multiplication/ZGEMM) $O(N^4)$
   
   $\chi(q, \omega_i) = M(q)^\dagger \Delta_{jak}(\epsilon_{j_k}, \epsilon_{a_k}, q, \omega) M(q)$

3. Compute dielectric matrix and its inverse (ScalaPACK) $O(N^3)$
   
   $\epsilon^{-1}(q, \omega_i) = (I - \nu \chi(q, \omega_i))^{-1}$
Epsilon: Inverse Dielectric Function (Matrix)

Three major computational steps: input $\psi_{mk}$, $\epsilon_{mk}$, $\{q\text{-points}\}$, $\{\omega_i\}$

1. Calculate plane-waves matrix elements (FFT’s) $O(N^3)$

$$M_{jak}^G(q) = \langle \psi_{jk+q} | e^{i(G+q) \cdot r} | \psi_{ak} \rangle$$

2. Calculate RPA polarizability (Matrix-Multiplication/ZGEMM) $O(N^4)$

$$\chi(q, \omega_i) = M(q)^\dagger \Delta_{jak}(\epsilon_{jk}, \epsilon_{ak}, q, \omega) M(q)$$

3. Compute dielectric matrix and its inverse (ScalaPACK) $O(N^3)$

$$\epsilon^{-1}(q, \omega_i) = (I - \nu \chi(q, \omega_i))^{-1}$$

For large scale applications the evaluation of the polarizability (CHI-0) is by far the most computationally intensive part of the calculation: large distributed matrix-multiplication over fat and short matrices
Epsilon: Polarizability Matrix (CHI-0)

Matrix size
~100k x 100M

Data layout for \( M \) matrix in CHI-0 kernel

- Memory demanding \( O(N^3) \)
  Number of columns \( N_v \times N_c \), number of rows \( N_G \)
- Computationally intensive \( O(N^4) \)
  Implemented as ZGEMM operation (small prefactor)
- Non-blocking cyclic communication
  Overlap Computation and MPI communication
- Repeated at multiple frequencies for full-frequency calculations
  Can be accelerated using low-rank approximation techniques
Epsilon: Performance

Strong Scaling of $\epsilon$ measured on Edison@NERSC (Cray XC30, IvyBridge processors)
Epsilon: Performance on Summit (GPU)

Epsilon scales linearly well to thousands of GPUs

Strong Scaling of epsilon measured on Summit@OLCF (Node: 2 IBM POWER9 CPUs and 6 NVIDIA V100 GPUs)
Sigma: Quasiparticle Properties

Compute a set (100-1000) of Self-Energy matrix elements to solve the Dyson equation

\[ h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r}, \mathbf{r}'; E_n)\phi_n(\mathbf{r}')d\mathbf{r}' = E_n \phi_n(\mathbf{r}) \]
Compute a set (100-1000) of Self-Energy matrix elements to solve the Dyson equation

\[ h_0(r)\phi_n(r) + \int \Sigma(r, r'; E_n)\phi_n(r')dr' = E_n\phi_n(r) \]

Each Self-Energy matrix element:

\[
\Sigma_{lm}(E) = \frac{i}{2\pi} \int_0^\infty d\omega \sum_n \sum_{GG'} M_{nl}^{-G} \frac{1}{E - E_n - \omega} \frac{\epsilon_{GG'}^{-1}(\omega) \cdot \nu(G')}{E - E_n - \omega} M_{nm}^{-G'}
\]
Sigma: Quasiparticle Properties

Compute a set (100-1000) of Self-Energy matrix elements to solve the Dyson equation

\[ h_0(r)\phi_n(r) + \int \Sigma(r, r'; E_n)\phi_n(r')dr' = E_n\phi_n(r) \]

Each Self-Energy matrix element:

\[ \Sigma_{lm}(E) = \frac{i}{2\pi} \int_0^\infty d\omega \sum_n \sum_{GG'} M_{nl}^{-G} \frac{\epsilon_{GG'}^{-1}(\omega) \cdot v(G')}{E - E_n - \omega} M_{nm'}^{-G'} \]

Frequency treatment

- **Generalized Plasmon Pole (GPP) model**
  - Analytical approximation to the frequency dependence
  - Require only the static dielectric matrix

- **Full-Frequency (FF) model**
  - Analytical integration over frequency (Contor-Deformation)
  - Require frequency dependent dielectric matrix
Sigma: Two Level Parallelization

Inter-Pool Parallelization
16 MPI tasks, 2 pools, 4 Self-Energies: \(\{\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4\}\) → Nearly independent calculation

Two Level Parallelization Strategy:
- Inter-Pool parallelization (independent self-energy matrix elements)
- Intra-Pool parallelization (same self-energy matrix elements)
Sigma: The GPP Kernel

Inter-Pool Parallelization
16 MPI tasks, 2 pools, 4 Self-Energies: \(\{\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4\}\)

Two Level Parallelization Strategy:

- **Inter-Pool parallelization** *(independent self-energy matrix elements)*
- Intra-Pool parallelization *(same self-energy matrix elements)*
Sigma: The GPP Kernel

Two Level Parallelization Strategy:

- Inter-Pool parallelization (independent self-energy matrix elements)
- Intra-Pool parallelization (same self-energy matrix elements)

Intra-Pool data layout

Large data reduction across different matrices with a complex matrix-vector interdependence.

For each $E_{qp}$

- GPP -> $O(N_G^2 N_b)$
- FF -> $O(N_{freq} N_G^2 N_b)$
Divacancy in Silicon

- Large reconstruction
- Supercell containing 998/1726 atoms
- Calculation parameters:
  \( N_G = 100k \); \( N_c = 37k \); \( N_b = 12k \); \( Nfreq = 35 \)

Sigma: Performance on Perlmutter (GPU)

Strong Scaling for Sigma measured on Perlmutter@NERSC (Cray Shasta, Node: 2 AMD Milan + 4 A100 GPUs)

- Left: Strong scaling to (almost) entire Perlmutter
- Right: Comparison between CPU (Cori-Haswell) and GPU (Perlmutter)
Sigma: Performance on Perlmutter (GPU)

Strong Scaling for Sigma measured on Perlmutter@NERSC (Cray Shasta, Node: 2 AMD Milan + 4-A100 GPUs)

- **Left:** Strong scaling to (almost) entire Perlmutter
- **Right:** Comparison between CPU (Cori-Haswell) and GPU (Perlmutter)
Sigma: Performance on Perlmutter (GPU)

Strong Scaling for Sigma measured on Perlmutter@NERSC (Cray Shasta, Node: 2 AMD Milan + 4-A100 GPUs)

- Left: Strong scaling to (almost) entire Perlmutter
- Right: Comparison between CPU (Cori-Haswell) and GPU (Perlmutter)
Speed-Up Full-Frequency GW Calculations

- Low-Rank Approximation for the static polarizability
- Select the Static-Subspace (truncation threshold)
- Use eigenvectors as new basis for the frequency dependent part

Silicon Carbide (β-SiC) band structure as obtained with and without approximation (0.01 threshold, 20x speed-up)

Order of magnitude speed-up of FF calculations, time to solution comparable with GPP calculations

The BSE Workflow: Kernel + Absorption

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

http://manual.berkeleygw.org/3.0/overview-workflow/
The BSE Workflow: Kernel + Absorption

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

http://manual.berkeleygw.org/3.0/overview-workflow/
The BSE Workflow: Kernel + Absorption

Calculate the electron-hole excitation states for each exciton state $S$:

$$
(E_{ck}^{QP} - E_{vk}^{QP}) A_{vck}^S + \sum_{v'c'k'} \langle vck | K^{eh} | v'c'k' \rangle A_{v'c'k'}^S = \Omega^S A_{vck}^S
$$
The BSE Workflow: Kernel + Absorption

Calculate the electron-hole excitation states for each exciton state $S$:

\[
\left( E_{ck}^{QP} - E_{vk}^{QP} \right) A_{vck}^S + \sum_{v'c'k'} \langle vck | K_{eh}^{v'c'k'} | v'c'k' \rangle A_{v'c'k'}^S = \Omega^S A_{vck}^S
\]
The BSE Workflow: Kernel + Absorption

Calculate the electron-hole excitation states for each exciton state $S$:

$$\left( E_{ck}^{\text{QP}} - E_{vk}^{\text{QP}} \right) A_{vck}^S + \sum_{v'c'k'} \langle vck | K^{\text{eh}} | v'c'k' \rangle A_{v'c'k'}^S = \Omega^S A_{vck}^S$$

GW quasiparticle energies (diagonal matrix)  
Electron-Hole interaction kernel (dense matrix)
The BSE Workflow: Kernel + Absorption

Calculate the electron-hole excitation states for each exciton state $S$:

\[
\left( E_{ck}^{QP} - E_{vk}^{QP} \right) A_{vck}^S + \sum_{v'c'k'} \langle vck | K^{eh} | v'c'k' \rangle A_{v'c'k'}^S = \Omega^S A_{vck}^S
\]

GW quasiparticle energies (diagonal matrix)  
Electron-Hole interaction kernel (dense matrix)

The solution gives eigenvalues and eigenvectors:

- Excitation energy $\Omega^S$
- Exciton wavefunction $A_{vck}^S$

Calculate: exciton WFN in real space, optical response etc...
The BSE Workflow: Kernel + Absorption

Calculate the electron-hole excitation states for each exciton state $S$:

$$
\left( E_{ck}^{QP} - E_{vk}^{QP} \right) A_{vck}^S + \sum_{v'c'k'} \langle vck | K_{\rm eh}^{\rm QP} | v'c'k' \rangle A_{v'c'k'}^S = \Omega^S A_{vck}^S
$$

GW quasiparticle energies (diagonal matrix) \hspace{2cm} Electron-Hole interaction kernel (dense matrix)

The solution gives eigenvalues and eigenvectors:
- Excitation energy $\Omega^S \\
- Exciton wavefunction $A_{vck}^S$

Calculate: exciton WFN in real space, optical response etc...

High BSE Computational Cost in Two Major Bottlenecks:
The BSE Workflow: Kernel + Absorption

Calculate the electron-hole excitation states for each exciton state $S$:

$ \left( E^\text{QP}_{ck} - E^\text{QP}_{vk} \right) A^S_{vck} + \sum_{v'c'k'} \langle vck|K^{\text{eh}}|v'c'k'\rangle A^S_{v'c'k'} = \Omega^S A^S_{vck} \quad \text{Eigenvalue Problem}$

GW quasiparticle energies (diagonal matrix)  
Electron-Hole interaction kernel (dense matrix)

The solution gives eigenvalues and eigenvectors:

- Excitation energy $\Omega^S$
- Exciton wavefunction $A^S_{vck}$

Calculate: exciton WFN in real space, optical response etc...

High BSE Computational Cost in Two Major Bottlenecks:

- **Kernel**: calculate kernel matrix elements on a coarse grid $\mathcal{O}(N^5)$
The BSE Workflow: Kernel + Absorption

Calculate the electron-hole excitation states for each exciton state $S$:

$$
\left( E_{ck}^{\text{QP}} - E_{vk}^{\text{QP}} \right) A_{vck}^S + \sum_{v'c'k'} \langle vck | K^{\text{eh}} | v'c'k' \rangle A_{v'c'k'}^S = \Omega_{vck}^S A_{vck}^S
$$

GW quasiparticle energies (diagonal matrix) \hspace{2cm} Electron-Hole interaction kernel (dense matrix)

The solution gives eigenvalues and eigenvectors:
- Excitation energy $\Omega_{vck}^S$
- Exciton wavefunction $A_{vck}^S$

Calculate: exciton WFN in real space, optical response etc...

High BSE Computational Cost in Two Major Bottlenecks:
- **Kernel**: calculate kernel matrix elements on a coarse grid $O(N^5)$
- **Absorption**: interpolate $E^{\text{QP}}$ and kernel matrix elements onto a fine grid and diagonalize the BSE Hamiltonian $O(N^6)$
Kernel: Kernel Matrix Elements

The electron hole interaction kernel is composed of the **screened direct** interaction and a **bare exchange** interaction $K^{\text{eh}} = K^d + K^x$. 
The electron hole interaction kernel is composed of the **screened direct** interaction and a **bare exchange** interaction $K^{\text{eh}} = K^d + K^x$.

1) Compute intermediates plane-wave matrix elements (cv, cc, vv blocks)

$$M_{nn'}(k, q, G) = \langle nk+q | e^{i(q+G) \cdot r} | n'k \rangle$$
Kernel: Kernel Matrix Elements

The electron hole interaction kernel is composed of the screened direct interaction and a bare exchange interaction, \( K^{\text{eh}} = K^d + K^x \).

1) Compute intermediates plane-wave matrix elements (cv, cc, vv blocks)

\[
M_{nn'}(\mathbf{k}, \mathbf{q}, \mathbf{G}) = \langle n\mathbf{k}+\mathbf{q} | e^{i(\mathbf{q}+\mathbf{G})\cdot\mathbf{r}} | n'\mathbf{k} \rangle
\]

2) Compute screen direct terms (ZGEMM + DotProducts) \( O(N^5) \)

\[
\langle v\mathbf{c}\mathbf{k} | K^d | v'\mathbf{c}'\mathbf{k}' \rangle = \sum_{GG'} M_{cc'}(\mathbf{k}, \mathbf{q}, \mathbf{G}) W_{GG'}(\mathbf{q}; 0) M^*_{cv'}(\mathbf{k}, \mathbf{q}, \mathbf{G}')
\]

Screen Coulomb interaction \( W \) computed from the inverse dielectric function (epsilon)

3) Compute bare exchange terms (DotProducts) \( O(N^5) \)

\[
\langle v\mathbf{c}\mathbf{k} | K^x | v'\mathbf{c}'\mathbf{k}' \rangle = \sum_{GG'} M_{cv}(\mathbf{k}, \mathbf{q}, \mathbf{G}) v(\mathbf{q} + \mathbf{G}) \delta_{GG'} M^*_{cv'}(\mathbf{k}, \mathbf{q}, \mathbf{G}')
\]

Bare Coulomb interaction \( v \) (diagonal)
Kernel: Parallelization

Different parallel distribution depending on $N_p$:

• Distribute pairs of k-points: $N_p \leq N_k^2$
• Distribute pairs of $(N_k \, N_c)$: $N_p \leq (N_k \, N_c)^2$
• Distribute pairs of $(N_k \, N_c \, N_v)$: $N_p \leq (N_k \, N_c \, N_v)^2$
• **high_memory**: Save all WFNs FFTs to reuse in inner most loop
• **low_comm**: Replicate dielectric matrix among processors
Absorption: Interpolation

- Excitonic effects depend critically on k-point sampling -> **Fine k-grid required**
- Compute $E^{QP}$ and kernel matrix elements on a fine grid -> **Expensive**

Interpolate $E^{QP}$ and kernel matrix elements from a coarse onto a fine grid
Absorption: Interpolation

- Excitonic effects depend critically on k-point sampling -> **Fine k-grid required**
- Compute $E^{QP}$ and kernel matrix elements on a fine grid -> **Expensive**

Interpolate $E^{QP}$ and kernel matrix elements from a coarse onto a fine grid

1) Compute overlaps between coarse and fine wavefunctions

\[ C^{nk}_{n',n''} = \int \mathbf{r} u_{n'k'\mathbf{r}}(\mathbf{r}) u^*_{n''k''\mathbf{r}}(\mathbf{r}) \]

2) Use overlaps to interpolate Kernel to Fine Grid

\[ \langle vck_{k} | K | v'c'k'_{k'} \rangle = \sum_{n_1,n_2,n_3,n_4} C^{k}_{c_{n_1},c_{n_2}} C^{*k'_{c_{n_3},c_{n_4}}} C^{k'_{c_{n_1},c_{n_2}}} \langle n_2n_1k_{co}|K|n_4n_3k'_{co} \rangle \]

3) Use overlaps to interpolate $E^{QP}$ energies without missing band crossings etc..

\[ E^{QP}_n(k_{fi}) = E^{MF}_n(k_{fi}) + \sum_{n'} |C^{k}_{n,n'}|^2 (E^{QP}_n(k_{co}) - E^{MF}_n(k_{co}))_{k_{co}} \]

Example: interpolated $E^{QP}$ band-structure for (10,0) SWCNT
Absorption: Diagonalization

Excitation energy, exiton wavefunctions and absorption spectrum are obtained as solution of the eigenvalue problem associated to the BSE Hamiltonian.

\[
\left( E_{ck}^{QP} - E_{vk}^{QP} \right) A_{vck}^S + \sum_{v'c'k'} \langle vck | K^{eh}^{S} | v'c'k' \rangle A_{v'c'k'}^S = \Omega^S A_{vck}^S
\]

- **Direct Solver (ScalaPACK, ELPA) \(O(N^6)\)**
  Exact diagonalization, compute all exciton states

- **Iterative Solvers (PRIMME)\)**
  Exact diagonalization, compute selected lowest exciton states

- **Haydock-Recursion Method (haydock.cplx.x) \(O(N^4)\)**
  Computes only the absorption spectra
The ParaBands Tool

1-epsilon
\( \epsilon^{-1}_{\text{GW}}(q) \)
epsmat[,h5], eps0mat[,h5]

2-sigma
\( E_{\text{QP}} \)
\( E_{\text{nk}} \)
eqq.dat

3-kernel
\( K_{\mathbf{k} \mathbf{k}'} \)
bsemat

4-absorption
\( \varepsilon(\omega) \), JDOS(\( \omega \))
\( \text{absorption}_{\text{nojeh.dat}} \)
\( A_{\mathbf{k} \mathbf{k}'}^S \Omega^S \)
eigenvalues.dat, eigenvectors

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

http://manual.berkeleygw.org/3.0/overview-workflow/
The ParaBands Tool

Generate a large number of empty states parabands

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

http://manual.berkeleygw.org/3.0/overview-workflow/
The ParaBands Tool

ParaBands builds and diagonalizes the dense DFT Hamiltonian in plane wave basis using a direct solver to obtain a large number of empty bands.

- Input (from QE using pw2bgw.x):
  - WFN_in: the wave function file for reference (few bands)
  - VKB: the Kleinman-Bylander non-local projectors in G-space
  - VSC: the self-consistent potential in G-space

- Support different solvers libraries: Scalapack, ELPA, PRIMME
- Embarrassingly parallel over k-points
- High performance I/O routine via HDF5
- Fully support for many core architectures
### The ParaBands Tool: Reference Timings

<table>
<thead>
<tr>
<th>N</th>
<th>Nodes</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>17000</td>
<td>4</td>
<td>10 s</td>
</tr>
<tr>
<td>60000</td>
<td>16</td>
<td>10 min</td>
</tr>
<tr>
<td>100000</td>
<td>64</td>
<td>15 min</td>
</tr>
<tr>
<td>150000</td>
<td>256</td>
<td>17 min</td>
</tr>
<tr>
<td>200000</td>
<td>512</td>
<td>25 min</td>
</tr>
<tr>
<td>250000</td>
<td>512</td>
<td>38 min</td>
</tr>
<tr>
<td>320000</td>
<td>512</td>
<td>68 min</td>
</tr>
<tr>
<td>416000</td>
<td>2560</td>
<td>80 min</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Matrix Size</th>
<th>Number on Nodes</th>
<th>GPU Support</th>
<th>Time for Diagonalization (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>19,381</td>
<td>1</td>
<td>No</td>
<td>215</td>
</tr>
<tr>
<td>19,381</td>
<td>1</td>
<td>Yes</td>
<td>83.2</td>
</tr>
<tr>
<td>65,117</td>
<td>16</td>
<td>Yes</td>
<td>135</td>
</tr>
<tr>
<td>155,331</td>
<td>128</td>
<td>Yes</td>
<td>279</td>
</tr>
</tbody>
</table>

Timing for the diagonalization of a matrix with different sizes as measured on Cori-KNL (left) and Perlmutter (top) at NERSC.
Summary

• Overview of the BerkeleyGW software package, software vision design, structure and main workflow

• General algorithms, parallelization strategies and computational motifs

• More specific details about the structures of the four major modules
  • Epsilon: Generate the dielectric function and its frequency dependence
  • Sigma: Solve Dyson's equation for quasiparticle energies
  • Kernel: Compute BSE kernel matrix elements on a coarse k-point grid
  • Absorption: Interpolate BSE kernel matrix elements on a fine k-point grid, diagonalize the BSE Hamiltonian, and compute optical absorption spectrum
BerkeleyGW overview

BerkeleyGW is a free, open source, and massively parallel computational package for electron excited-state properties that is based on the many-body perturbation theory employing the ab initio GW and plus Bethe-Salpeter equation methodology.

It is able to calculate accurate electronic and optical properties in materials of different dimensionalities and complexity, from bulk semiconductors and metals to nanostructured materials and molecules.

It can be used in conjunction with many external and well established density-functional theory codes for ground-state properties, including PARATEC, Abinit, PARSEC, Quantum ESPRESSO, OCTOPUS and SIESTA. These codes are used to generate initial files, containing the ground-state density and wavefunctions from density-functional theory. In addition, BerkeleyGW also ships with two codes to generate a large number of empty states for GW calculations: SAPO and ParaBands. See the page on mean-field calculations for further information.

After you compile and test BerkeleyGW, we suggest you follow the following tutorials on how to run calculations with BerkeleyGW:

1. GW calculation:
   a. epsilon: evaluating the dielectric screening
   b. sigma: calculating the electronic self-energy

2. Bethe-Salpeter equation (BSE) calculation:
   a. kernel: calculating the electron-hole interaction kernel
   b. absorption: computing neutral optical excitation properties, such as optical absorption

• The BerkeleyGW online manual

• BerkeleyGW-Help mailing list:
  [help@berkeleygw.org](mailto:help@berkeleygw.org)

Papers about implementation:

• **MPI and overview**: J Deslippe et al, *Computer Physics Communications* 183 (6), 1269-1289

• **Multi-core/OpenMP**: M Del Ben et al, *Computer Physics Communications* 235, 187-195

Foreseen exascale HPC systems will be GPU accelerated architectures

Optimized version of BerkeleyGW on GPU accelerated systems:

- Scale up to the full Summit machine at OLCF: >27k GPUs
- Reach nearly 53% of the peak performance at 106 PFLOP/s
- Time to solution of ~10 mins for 11k electrons system

Divacancy defect in semiconductor (such as Si and SiC) are proxy for solid state Qubits. For silicon shown is the 2742-atoms Si supercell, 10,968 electrons.