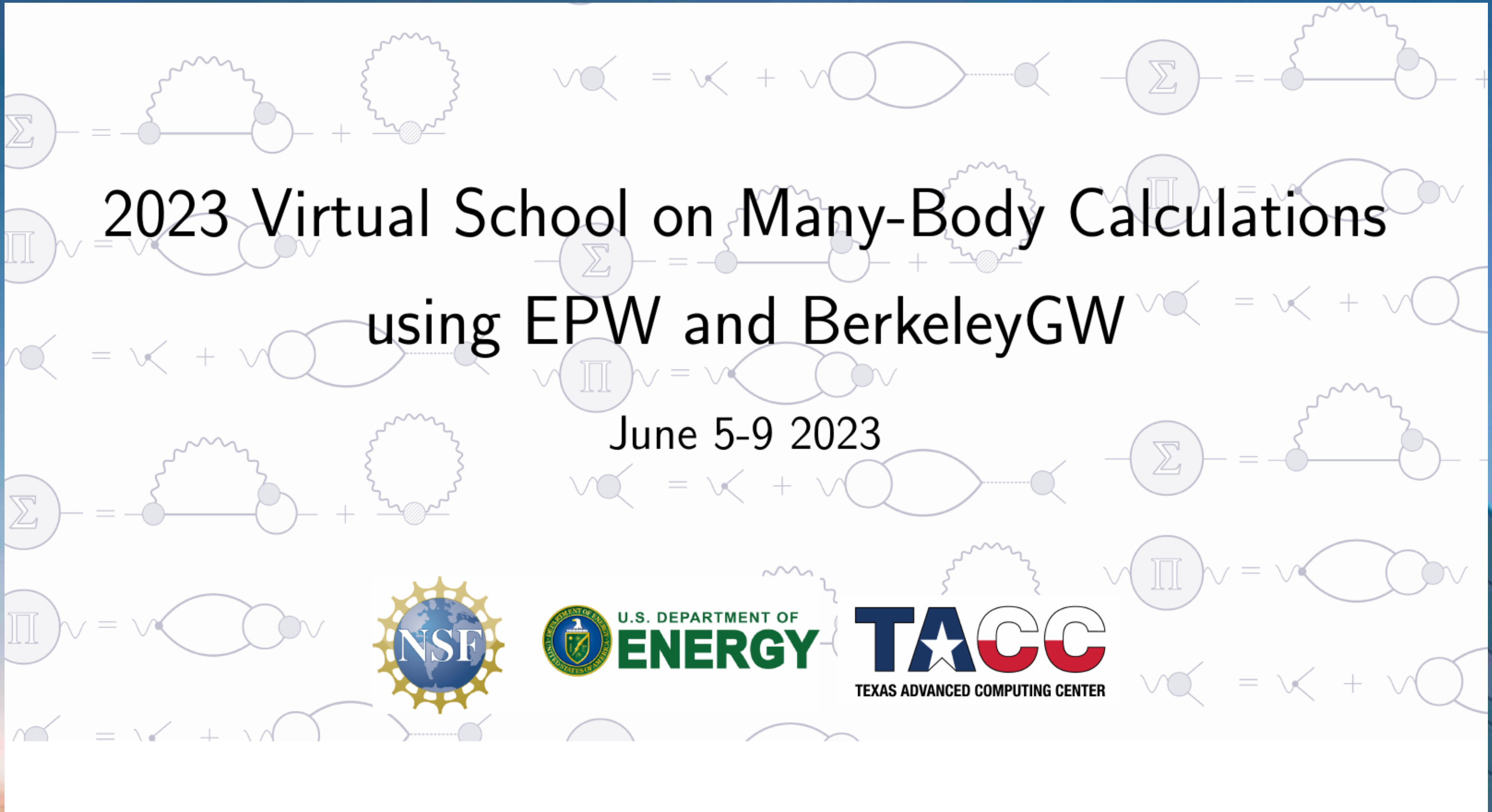
2023 Virtual School on Many-Body Calculations using EPW and BerkeleyGW

June 5-9 2023

NSF

U.S. DEPARTMENT OF ENERGY

TACC
TEXAS ADVANCED COMPUTING CENTER



2023 Virtual School on Many-Body Calculations using EPW and BerkeleyGW

June 5-9 2023

An Overview of the BerkeleyGW Software Package

Mauro Del Ben

Applied Mathematics & Computational Research Division (LBNL)

Outline

1. Introduction
2. Overview of the BerkeleyGW software package
3. The GW+BSE workflow in BerkeleyGW
4. Summary

Introduction

NSF

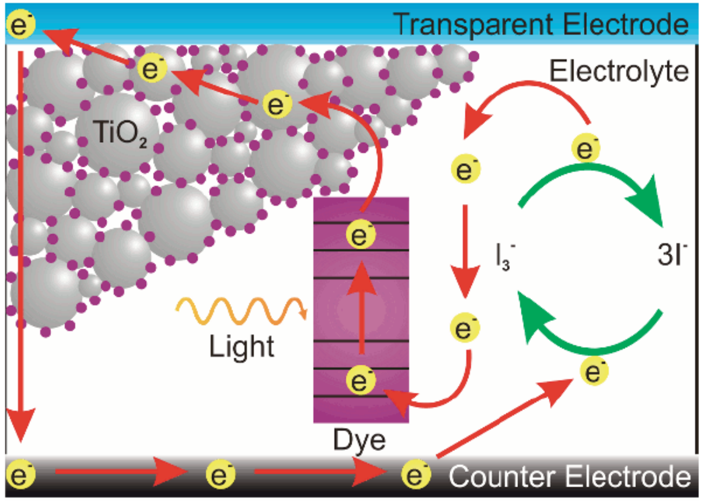
U.S. DEPARTMENT OF ENERGY

TACC

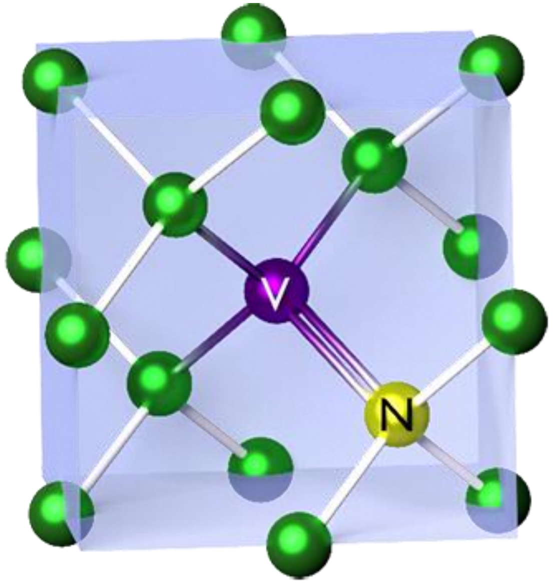
TEXAS ADVANCED COMPUTING CENTER

Materials Science/Chemistry and HPC

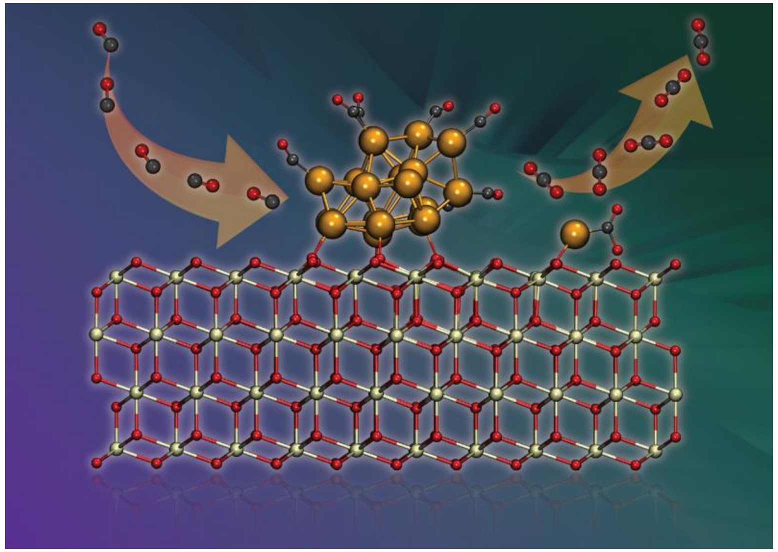
Grätzel cells: Oxide/Organic Interfaces



Cheap, reliable and sustainable **photovoltaics**



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

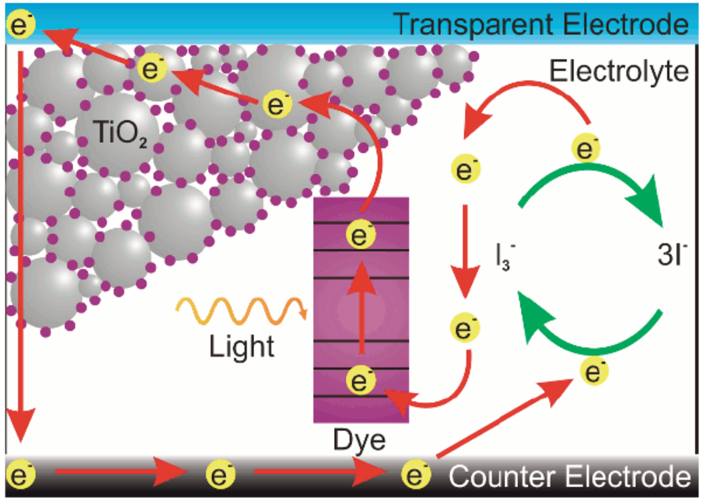


Chemical reaction at interfaces: **Catalysis**

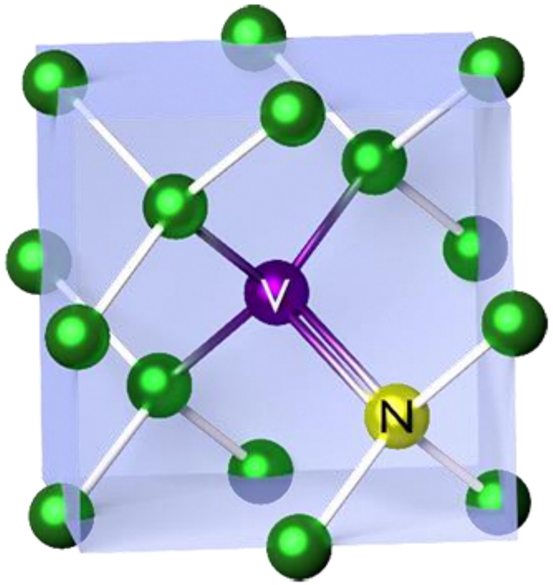
Mat. Sci & Chem apps, such as VASP, Quantum ESPRESSO, QMCPACK, NWchem, BerkeleyGW, CP2K, etc... **heavily use HPC facilities**

Materials Science/Chemistry and HPC

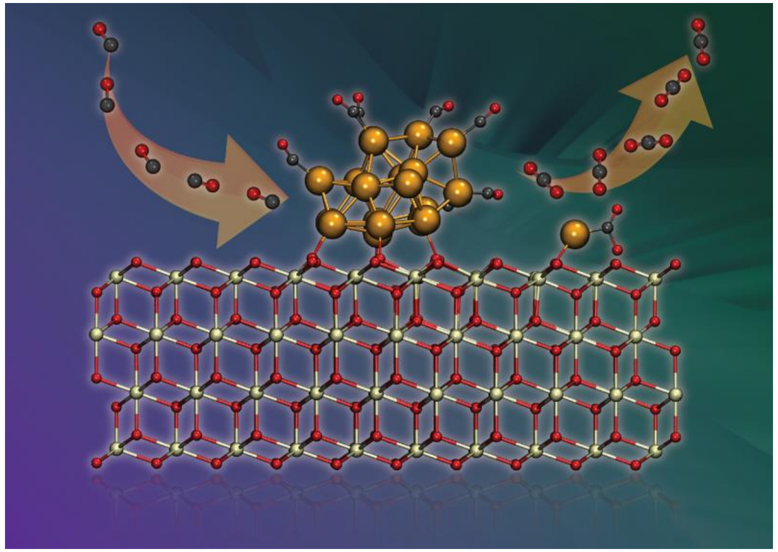
Grätzel cells: Oxide/Organic Interfaces



Cheap, reliable and sustainable **photovoltaics**



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

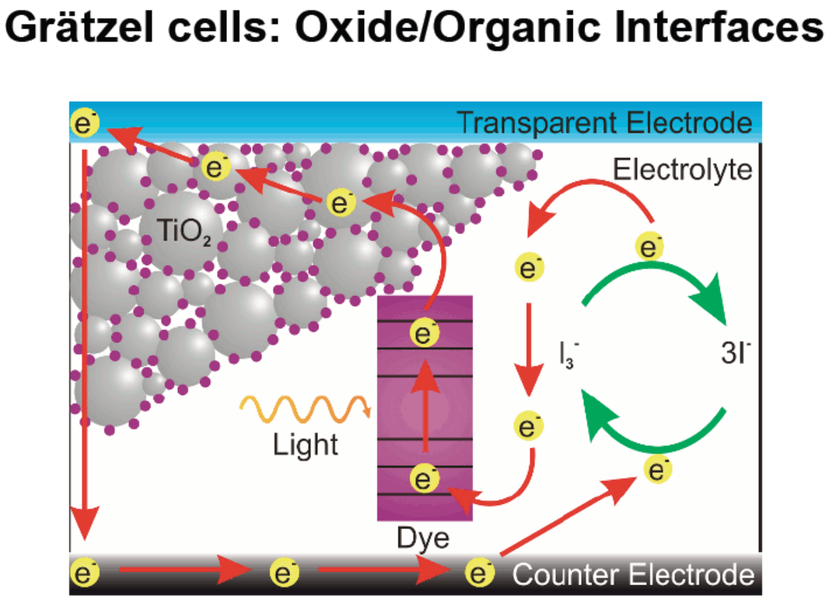


Chemical reaction at interfaces: **Catalysis**

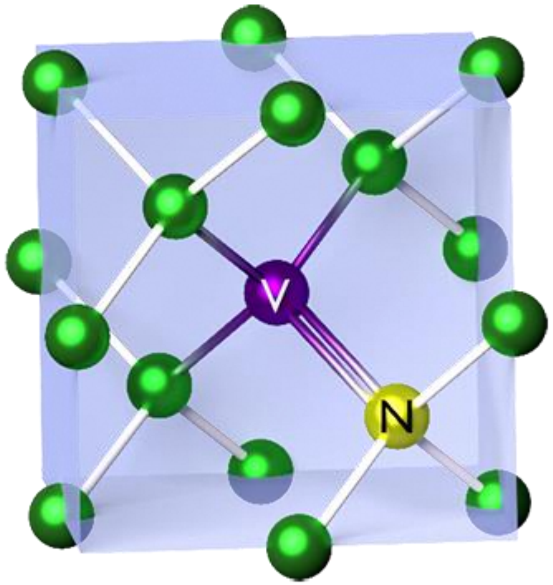
Used to **study and understand the fundamental electronic properties of materials**: *necessary to design the components of novel devices*

- Applications: Quantum Computers, Batteries, Photovoltaics, Catalysis, etc...

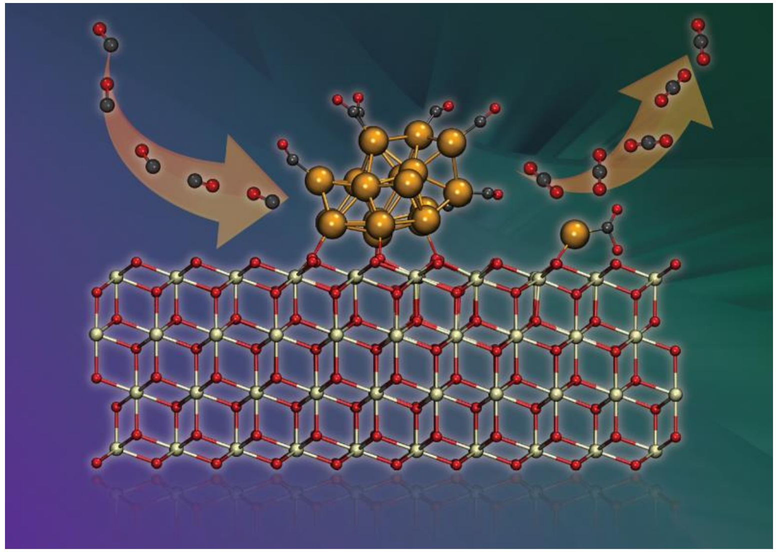
Materials Science/Chemistry and HPC



Cheap, reliable and sustainable **photovoltaics**



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



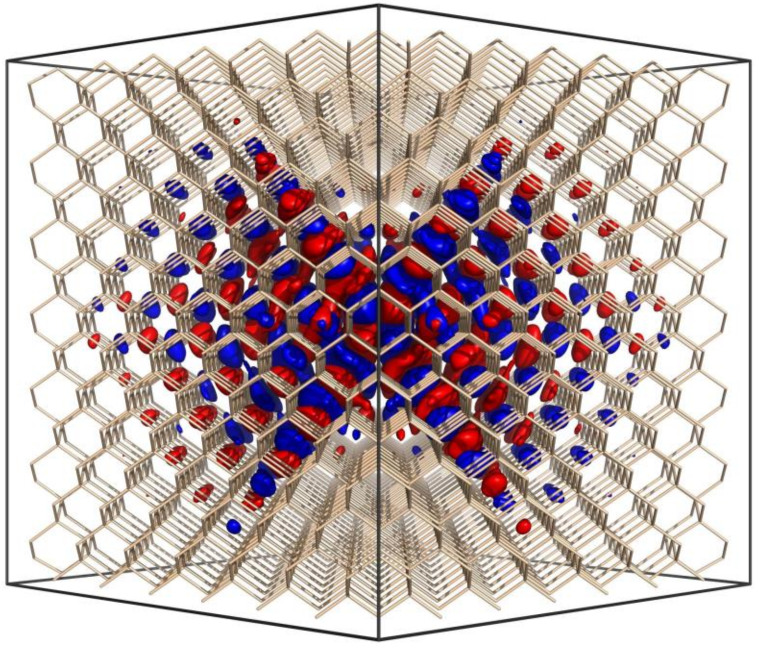
Chemical reaction at interfaces: **Catalysis**

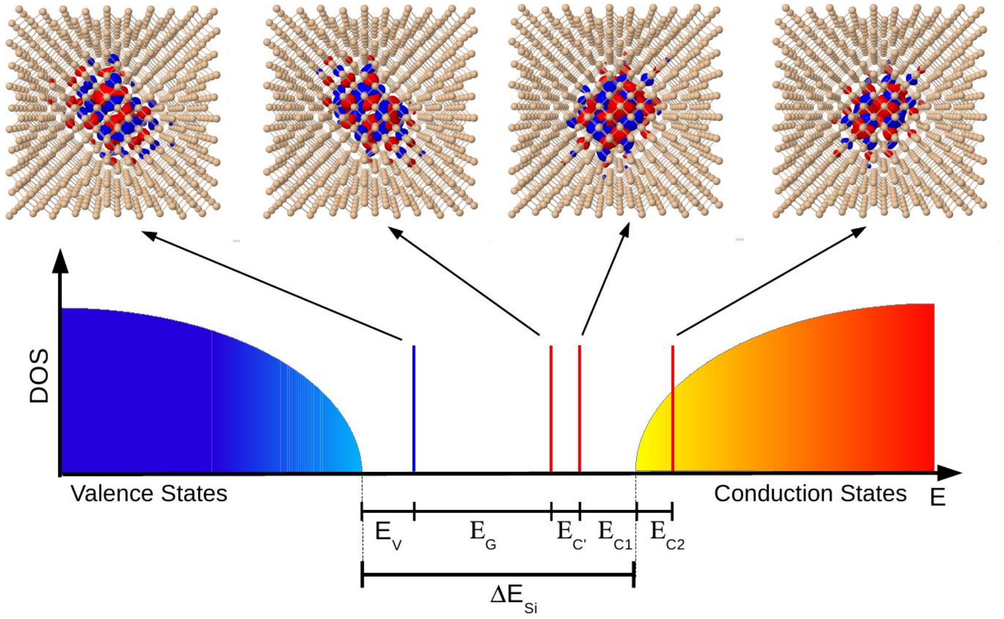
Density Functional Theory (DFT) the workhorse for over three decades

- 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

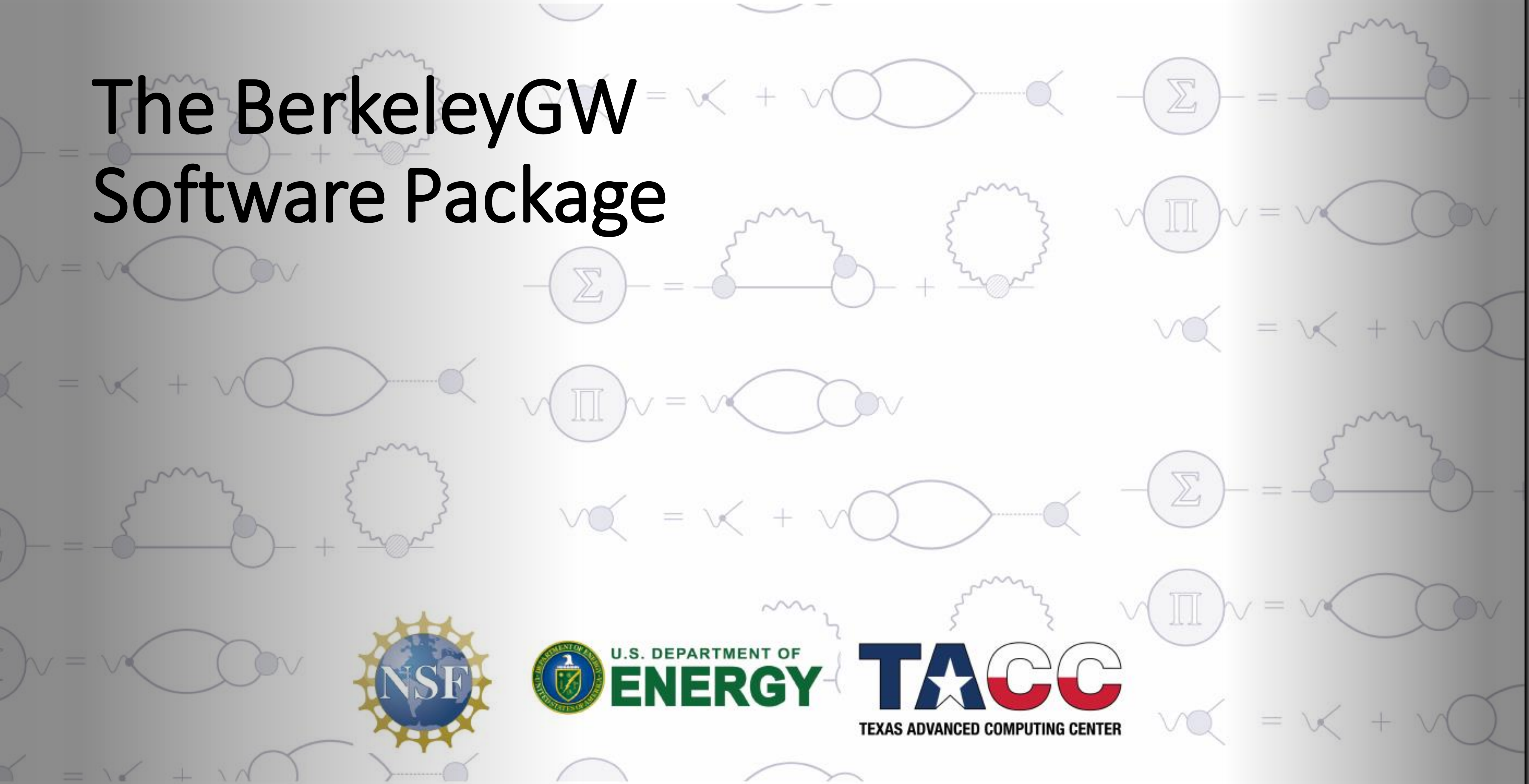
The GW+BSE method represents one of the most effective and accurate approach to predict excited-state properties in a wide range of materials

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



NSF

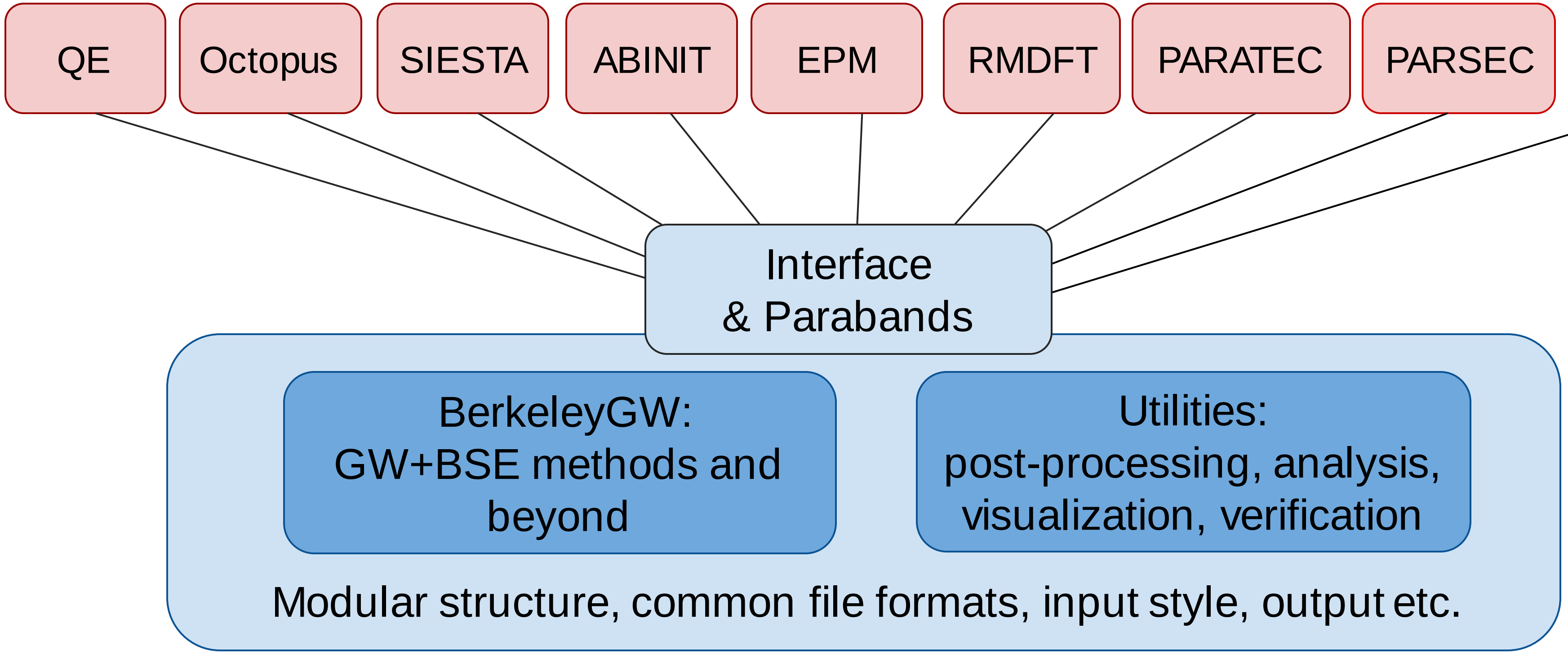
U.S. DEPARTMENT OF ENERGY

TACC

TEXAS ADVANCED COMPUTING CENTER

BerkeleyGW Software Design Vision

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

Interface
& Parabands

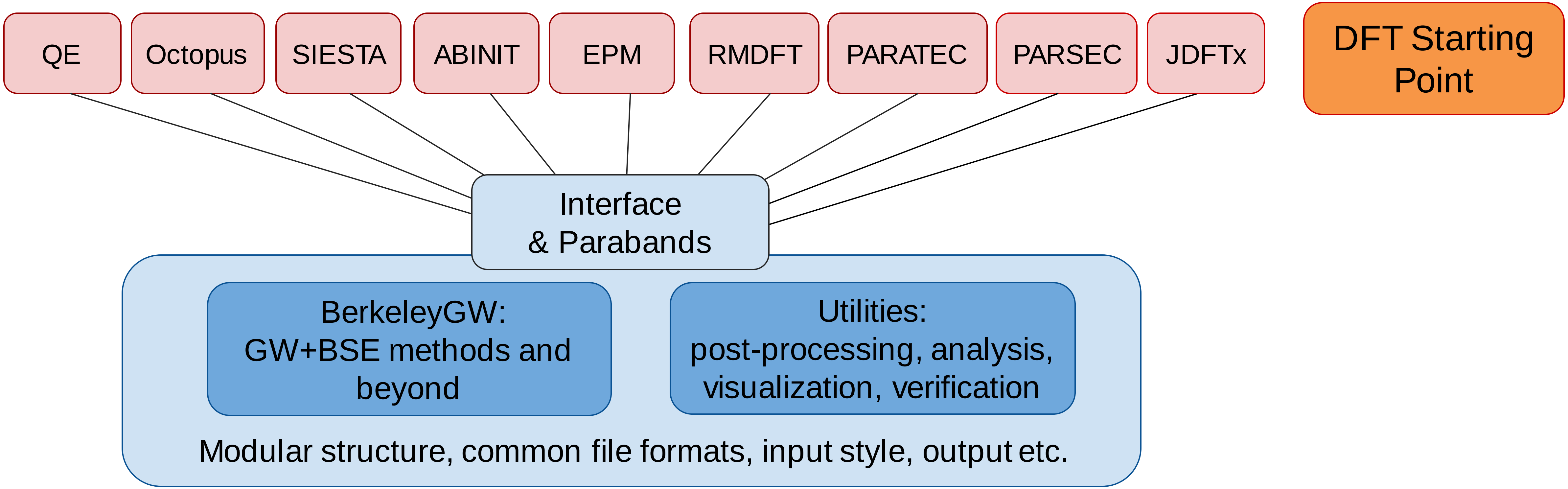
BerkeleyGW:
GW+BSE methods and beyond

Utilities:
post-processing, analysis, visualization, verification

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

BerkeleyGW Software Design Vision

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

DFT Starting Point

Interface & Parabands

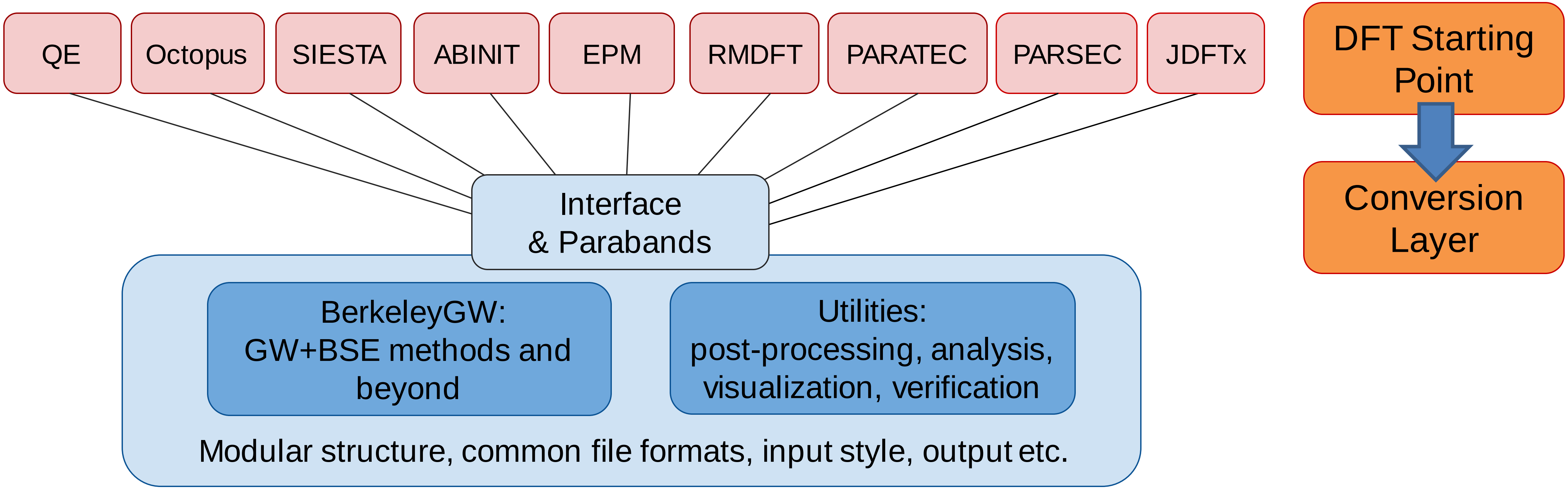
BerkeleyGW: GW+BSE methods and beyond

Utilities: post-processing, analysis, visualization, verification

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

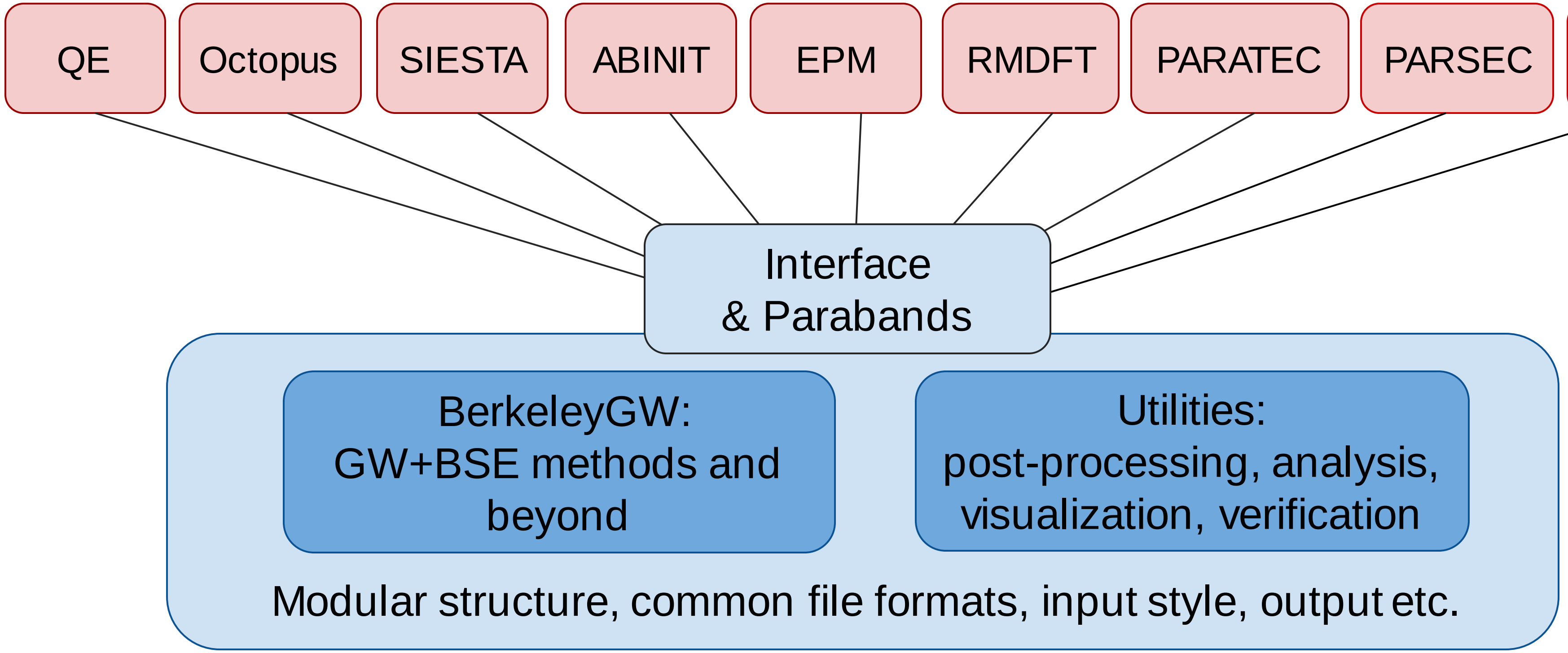
BerkeleyGW Software Design Vision

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
DFT Starting Point
Conversion Layer
Interface & Parabands
BerkeleyGW: GW+BSE methods and beyond
Utilities: post-processing, analysis, visualization, verification
Modular structure, common file formats, input style, output etc.

BerkeleyGW Software Design Vision

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

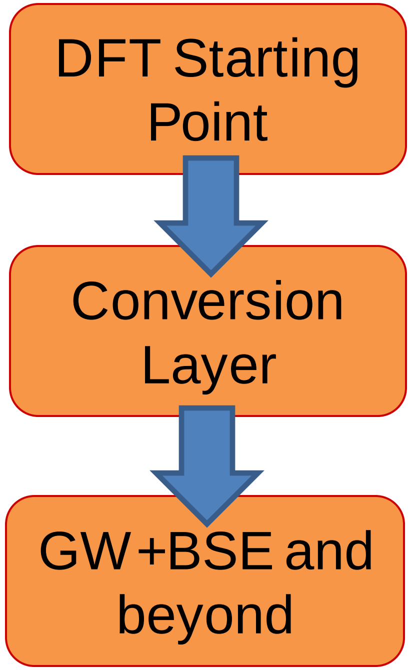
JDFTx

Interface
& Parabands

BerkeleyGW:
GW+BSE methods and beyond

Utilities:
post-processing, analysis, visualization, verification

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



DFT Starting Point

Conversion Layer

GW+BSE and beyond

BerkeleyGW: developments are focuses on GW+BSE methodologies

https://berkeleygw.org/

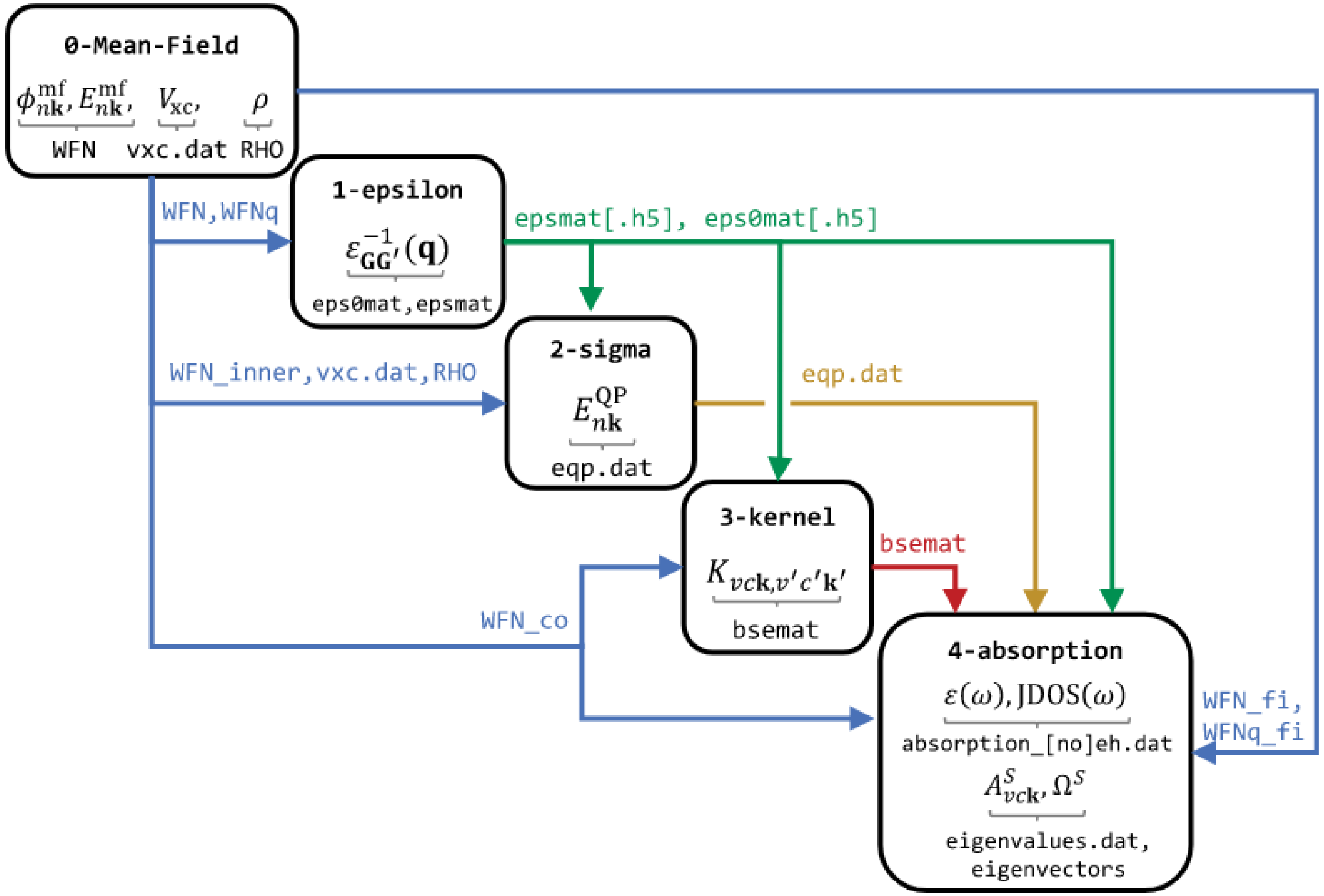
BerkeleyGW Highlights

- 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.

BerkeleyGW New Features (v3.0)

- 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

BerkeleyGW Workflow


0-Mean-Field
$\underbrace{\phi^{\rm mf}_{n\mathbf{k}}, E^{\rm mf}_{n\mathbf{k}},}_{\texttt{WFN}}$ $\underbrace{V_{\rm xc},}_{\texttt{vxc.dat}}$ $\underbrace{\rho}_{\texttt{RHO}}$
WFN,WFNq
1-epsilon
$\underbrace{\varepsilon^{-1}_{\mathbf{GG'}}(\mathbf{q})}_{\texttt{eps0mat,epsmat}}$
epsmat[.h5], eps0mat[.h5]
WFN_inner,vxc.dat,RHO
2-sigma
$\underbrace{E^{\rm QP}_{n\mathbf{k}}}_{\texttt{eqp.dat}}$
eqp.dat
3-kernel
$\underbrace{K_{vc\mathbf{k},v'c'\mathbf{k}'}}_{\texttt{bsemat}}$
bsemat
WFN_co
4-absorption
$\underbrace{\varepsilon(\omega), \mathrm{JDOS}(\omega)}_{\texttt{absorption\_[no]eh.dat}}$
$\underbrace{A^S_{vc\mathbf{k}}, \Omega^S}_{\texttt{eigenvalues.dat, eigenvectors}}$
WFN_fi,
WFNq_fi

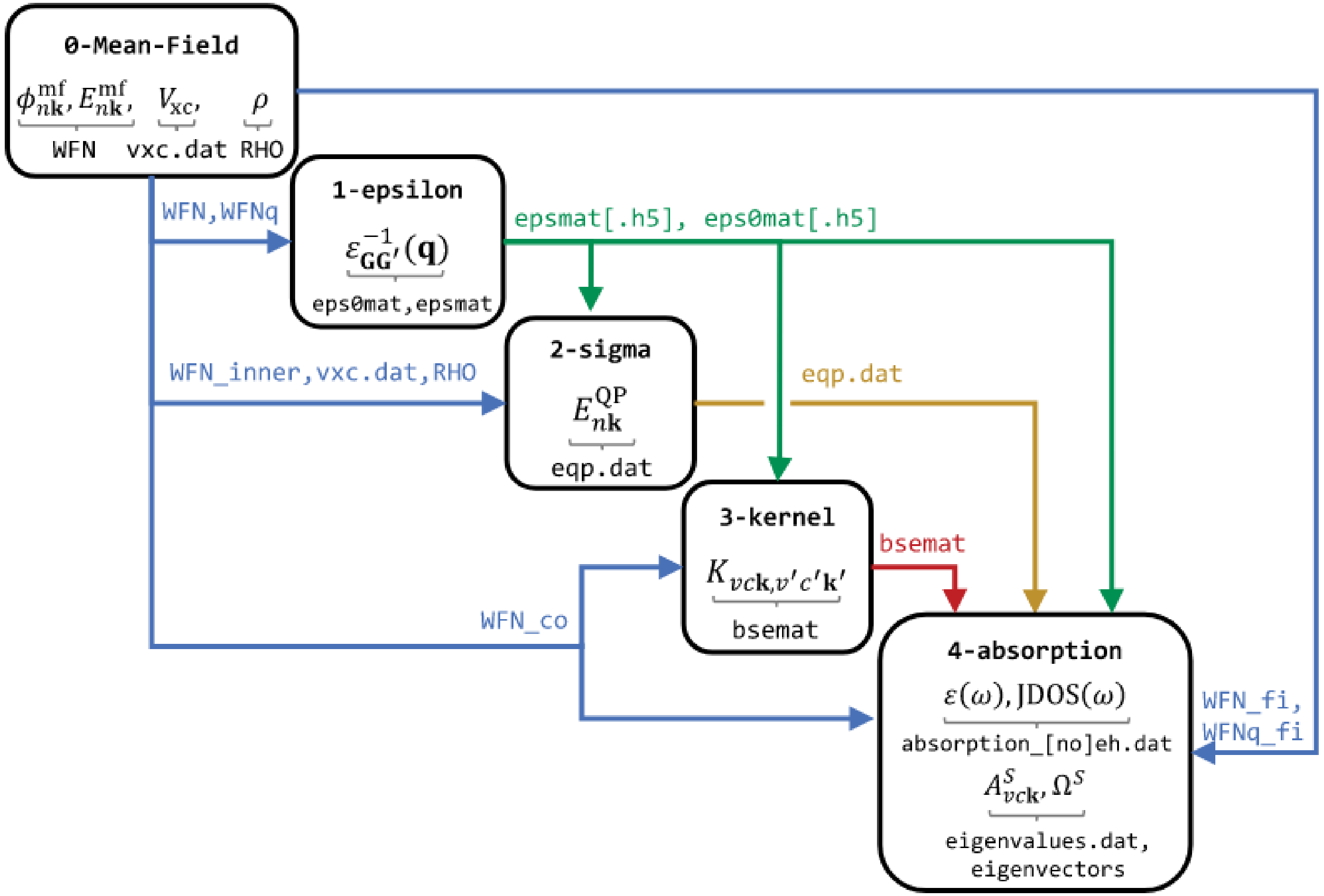
Four major modules: epsilon, sigma, kernel and absorption; *why?*

(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

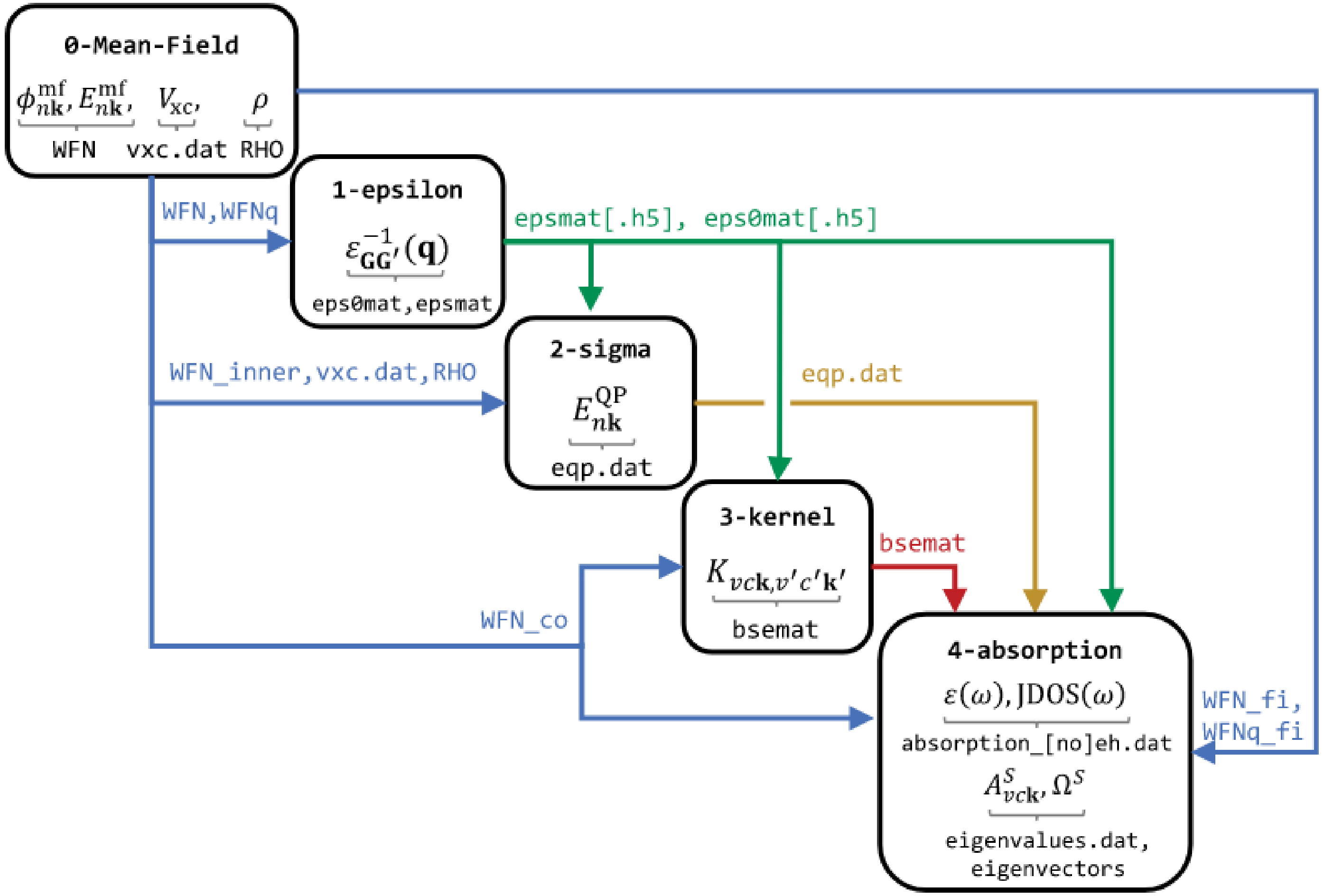

0-Mean-Field
$\underbrace{\phi_{n\mathbf{k}}^{\mathrm{mf}}, E_{n\mathbf{k}}^{\mathrm{mf}},}_{\text{WFN}} \underbrace{V_{\mathrm{xc}},}_{\text{vxc.dat}} \underbrace{\rho}_{\text{RHO}}$
WFN,WFNq
1-epsilon
$\underbrace{\varepsilon_{\mathbf{GG}'}^{-1}(\mathbf{q})}_{\text{eps0mat,epsmat}}$
epsmat[.h5], eps0mat[.h5]
WFN_inner,vxc.dat,RHO
2-sigma
$\underbrace{E_{n\mathbf{k}}^{\mathrm{QP}}}_{\text{eqp.dat}}$
eqp.dat
3-kernel
$\underbrace{K_{vc\mathbf{k},v'c'\mathbf{k}'}}_{\text{bsemat}}$
bsemat
WFN_co
4-absorption
$\underbrace{\varepsilon(\omega), \mathrm{JDOS}(\omega)}_{\text{absorption\_[no]eh.dat}}$
$\underbrace{A_{vc\mathbf{k}}^{S}, \Omega^{S}}_{\text{eigenvalues.dat, eigenvectors}}$
WFN_fi,
WFNq_fi

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

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

BerkeleyGW Workflow


0-Mean-Field
$\underbrace{\phi^{\mathrm{mf}}_{n\mathbf{k}}, E^{\mathrm{mf}}_{n\mathbf{k}},}_{\text{WFN}} \underbrace{V_{\mathrm{xc}},}_{\text{vxc.dat}} \underbrace{\rho}_{\text{RHO}}$
WFN,WFNq
1-epsilon
$\underbrace{\varepsilon^{-1}_{\mathbf{GG}'}(\mathbf{q})}_{\text{eps0mat,epsmat}}$
epsmat[.h5], eps0mat[.h5]
WFN_inner,vxc.dat,RHO
2-sigma
$\underbrace{E^{\mathrm{QP}}_{n\mathbf{k}}}_{\text{eqp.dat}}$
eqp.dat
3-kernel
$\underbrace{K_{vc\mathbf{k},v'c'\mathbf{k}'}}_{\text{bsemat}}$
bsemat
WFN_co
4-absorption
$\underbrace{\varepsilon(\omega), \mathrm{JDOS}(\omega)}_{\text{absorption\_[no]eh.dat}}$
$\underbrace{A^S_{vc\mathbf{k}}, \Omega^S}_{\text{eigenvalues.dat, eigenvectors}}$
WFN_fi,
WFNq_fi

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

Synopsis

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

BerkeleyGW Basic Computational Motifs

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/HIP dedicated branches and OpenACC/OpenMP-target mainline)*

BerkeleyGW Overview Summary

The goals of the BerkeleyGW developers community are:

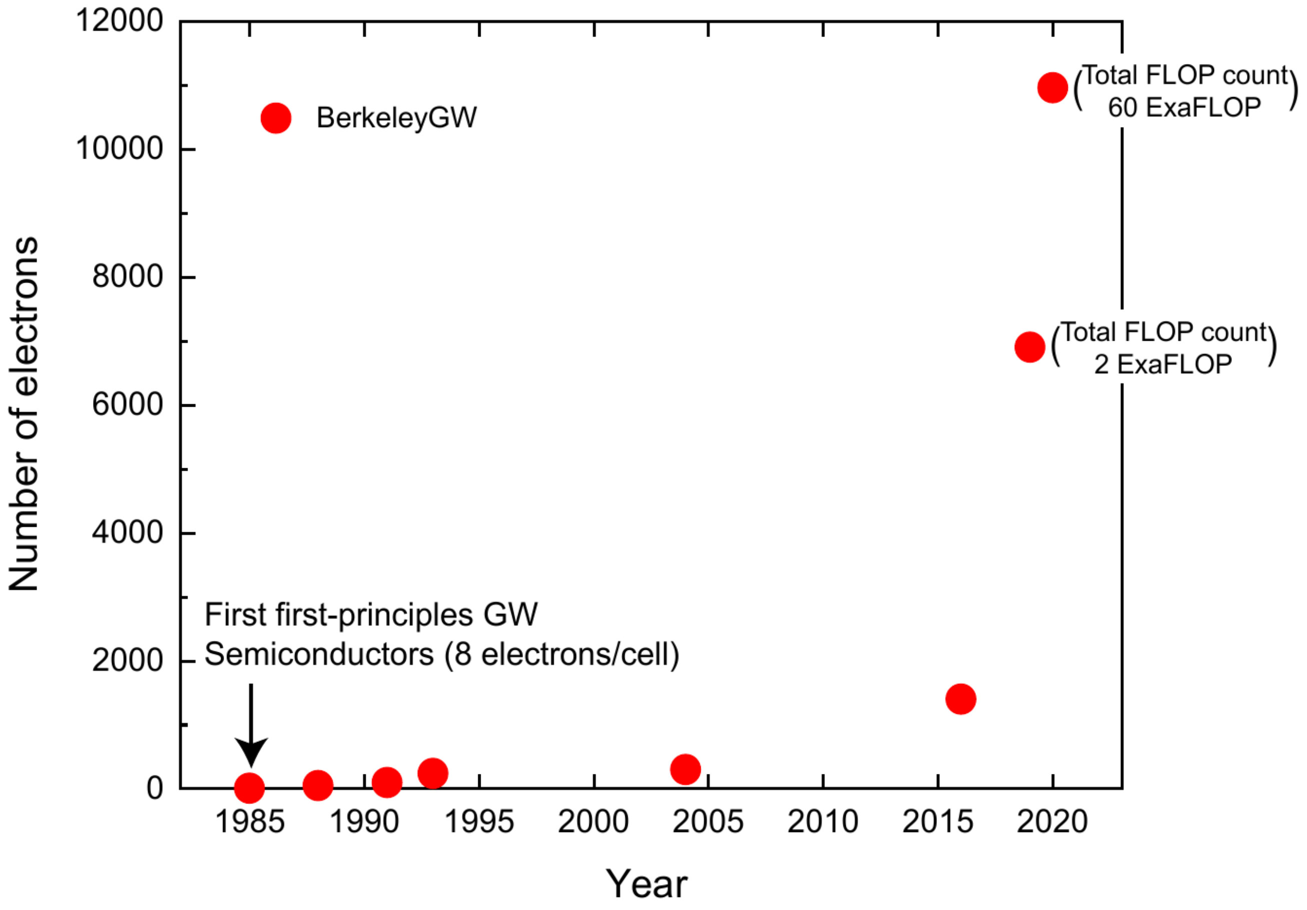
1. Distribute a highly performant and production quality set of computational tools

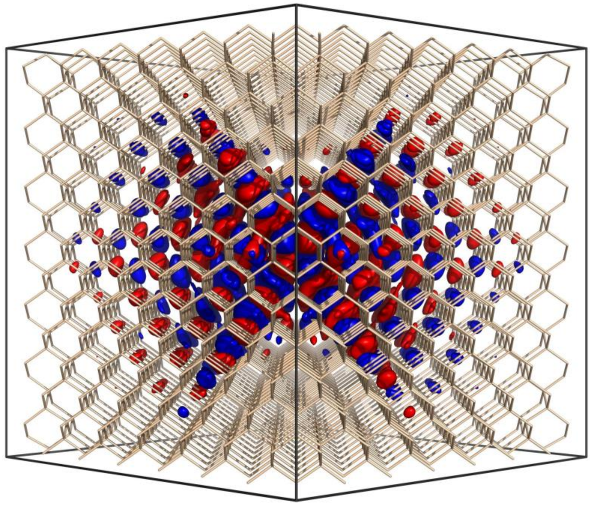
2. Optimize components for pre-exascale and exascale HPC systems to enable the study of systems with increasing complexity

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

BerkeleyGW Overview Summary





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

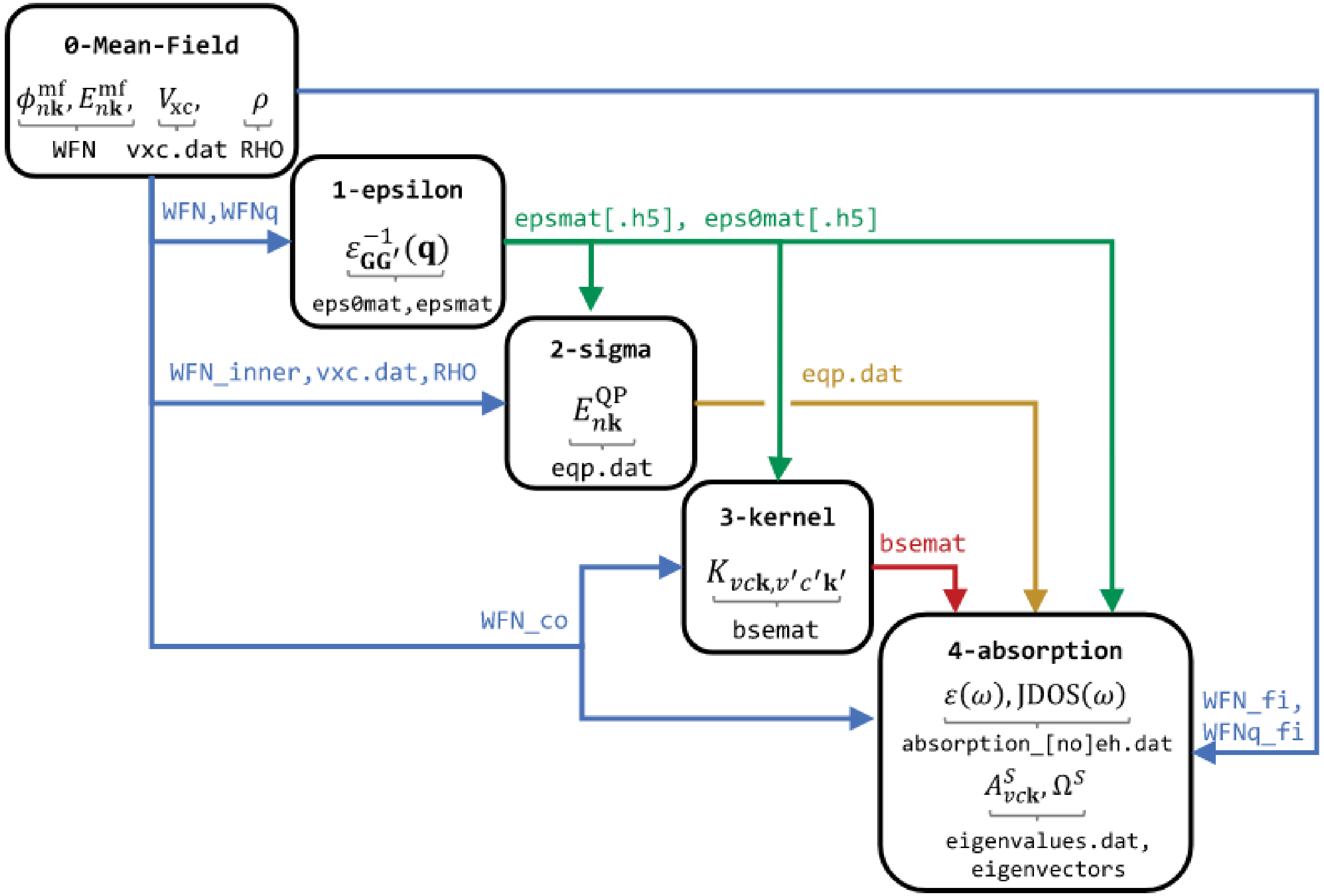
2020 ACM Gordon-Bell finalist: https://dl.acm.org/doi/abs/10.5555/3433701.3433706

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

The BerkeleyGW Modules

NSF
U.S. DEPARTMENT OF ENERGY
TACC
TEXAS ADVANCED COMPUTING CENTER

BerkeleyGW Workflow

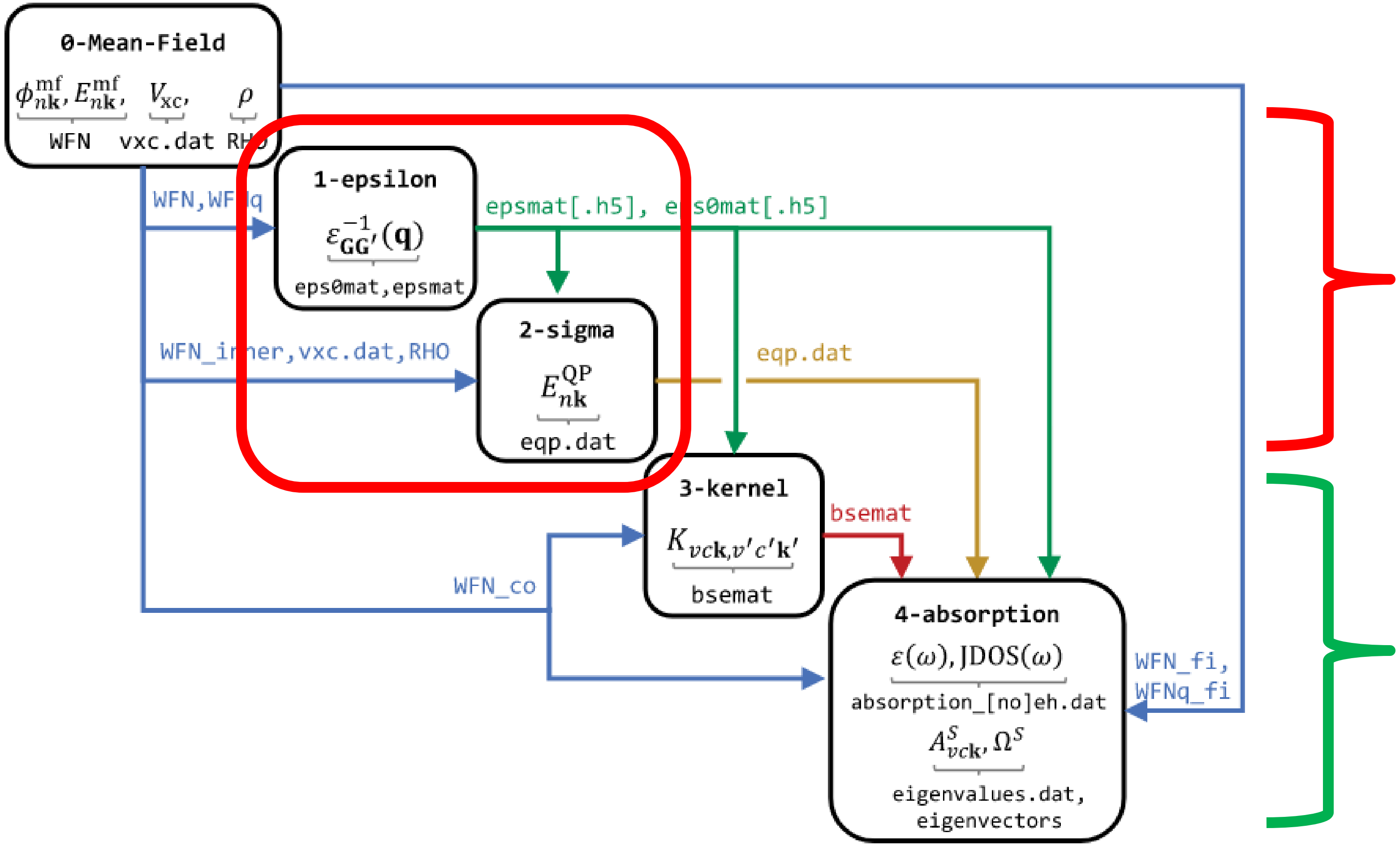

0-Mean-Field
$\underbrace{\phi^{\text{mf}}_{n\mathbf{k}}, E^{\text{mf}}_{n\mathbf{k}}}_{\text{WFN}}, \underbrace{V_{\text{xc}}}_{\text{vxc.dat}}, \underbrace{\rho}_{\text{RHO}}$
WFN,WFNq
1-epsilon
$\underbrace{\varepsilon^{-1}_{\mathbf{GG'}}(\mathbf{q})}_{\text{eps0mat,epsmat}}$
epsmat[.h5], eps0mat[.h5]
WFN_inner,vxc.dat,RHO
2-sigma
$\underbrace{E^{\text{QP}}_{n\mathbf{k}}}_{\text{eqp.dat}}$
eqp.dat
3-kernel
$\underbrace{K_{vc\mathbf{k},v'c'\mathbf{k}'}}_{\text{bsemat}}$
bsemat
WFN_co
4-absorption
$\underbrace{\varepsilon(\omega), \text{JDOS}(\omega)}_{\text{absorption\_[no]eh.dat}}$
$\underbrace{A^S_{vc\mathbf{k}}, \Omega^S}_{\text{eigenvalues.dat, eigenvectors}}$
WFN_fi,
WFNq_fi

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

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

The GW Workflow: Epsilon + Sigma


0-Mean-Field
$\underbrace{\phi^{\mathrm{mf}}_{n\mathbf{k}}, E^{\mathrm{mf}}_{n\mathbf{k}}}_{\text{WFN}}, \underbrace{V_{\mathrm{xc}}}_{\text{vxc.dat}}, \underbrace{\rho}_{\text{RHO}}$
1-epsilon
$\underbrace{\varepsilon^{-1}_{\mathbf{GG'}}(\mathbf{q})}_{\text{eps0mat,epsmat}}$
2-sigma
$\underbrace{E^{\mathrm{QP}}_{n\mathbf{k}}}_{\text{eqp.dat}}$
3-kernel
$\underbrace{K_{vc\mathbf{k},v'c'\mathbf{k}'}}_{\text{bsemat}}$
4-absorption
$\underbrace{\varepsilon(\omega), \mathrm{JDOS}(\omega)}_{\text{absorption\_[no]eh.dat}}$
$\underbrace{A^{S}_{vc\mathbf{k}}, \Omega^{S}}_{\text{eigenvalues.dat, eigenvectors}}$
WFN,WFNq
epsmat[.h5], eps0mat[.h5]
WFN_inner,vxc.dat,RHO
eqp.dat
bsemat
WFN_co
WFN_fi,
WFNq_fi

GW Workflow
epsilon + sigma

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:

$$h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r},\mathbf{r}';E_n)\phi_n(\mathbf{r}')\mathrm{d}\mathbf{r}' = E_n\phi_n(\mathbf{r})$$

The GW Workflow: Epsilon + Sigma

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

$$\boxed{h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r},\mathbf{r}';E_n)\phi_n(\mathbf{r}')\mathrm{d}\mathbf{r}' = E_n\phi_n(\mathbf{r})}$$

The GW Workflow: Epsilon + Sigma

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

$$h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \boxed{\Sigma(\mathbf{r},\mathbf{r}';E_n)}\phi_n(\mathbf{r}')\mathrm{d}\mathbf{r}' = E_n\phi_n(\mathbf{r})$$

GW Self-Energy Operator Σ : non-Hermitian, non-local, frequency dependent

(Note: In DFT, the role of self-energy is replaced by static and local $V_{\mathbf{xc}}(\mathbf{r})$)

The GW Workflow: Epsilon + Sigma

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

$$h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r},\mathbf{r}';E_n)\phi_n(\mathbf{r}')\mathrm{d}\mathbf{r}' = E_n\phi_n(\mathbf{r})$$

GW Self-Energy Operator Σ : non-Hermitian, non-local, frequency dependent

(Note: In DFT, the role of self-energy is replaced by static and local $V_{\mathbf{xc}}(\mathbf{r})$)

High GW Computational Cost in Two Major Bottlenecks:

The GW Workflow: Epsilon + Sigma

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

$$h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r},\mathbf{r}';E_n)\phi_n(\mathbf{r}')\mathrm{d}\mathbf{r}' = E_n\phi_n(\mathbf{r})$$

GW Self-Energy Operator Σ : non-Hermitian, non-local, frequency dependent

(Note: In DFT, the role of self-energy is replaced by static and local $V_{\mathbf{xc}}(\mathbf{r})$)

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:

$$h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r},\mathbf{r}';E_n)\phi_n(\mathbf{r}')\mathrm{d}\mathbf{r}' = E_n\phi_n(\mathbf{r})$$

GW Self-Energy Operator Σ : non-Hermitian, non-local, frequency dependent

(Note: In DFT, the role of self-energy is replaced by static and local $V_{\mathbf{xc}}(\mathbf{r})$)

High GW Computational Cost in Two Major Bottlenecks:

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

ϵ^{-1} matrix

Epsilon: Inverse Dielectric Function (Matrix)

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

Epsilon: Inverse Dielectric Function (Matrix)

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

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

$$M^{G}_{ja\mathbf{k}}(\mathbf{q}) = \langle\psi_{j\mathbf{k}+\mathbf{q}}|\, e^{i(\mathbf{G}+\mathbf{q})\cdot\mathbf{r}}\, |\psi_{a\mathbf{k}}\rangle$$

Epsilon: Inverse Dielectric Function (Matrix)

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

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

$$M^{G}_{ja\mathbf{k}}(\mathbf{q}) = \langle \psi_{j\mathbf{k}+\mathbf{q}} | \, e^{i(\mathbf{G}+\mathbf{q})\cdot\mathbf{r}} \, | \psi_{a\mathbf{k}} \rangle$$

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

$$\chi(\mathbf{q}, \omega_i) = \mathbf{M}(\mathbf{q})^{\dagger} \mathbf{\Delta}_{ja\mathbf{k}}(\epsilon_{j\mathbf{k}}, \epsilon_{a\mathbf{k}}, \mathbf{q}, \omega) \mathbf{M}(\mathbf{q})$$

Epsilon: Inverse Dielectric Function (Matrix)

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

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

$$M^G_{ja\mathbf{k}}(\mathbf{q}) = \langle\psi_{j\mathbf{k}+\mathbf{q}}|\, e^{i(\mathbf{G}+\mathbf{q})\cdot\mathbf{r}}\, |\psi_{a\mathbf{k}}\rangle$$

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

$$\chi(\mathbf{q},\omega_i) = \mathbf{M}(\mathbf{q})^\dagger \mathbf{\Delta}_{ja\mathbf{k}}(\epsilon_{j\mathbf{k}},\epsilon_{a\mathbf{k}},\mathbf{q},\omega)\mathbf{M}(\mathbf{q})$$

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

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

Epsilon: Inverse Dielectric Function (Matrix)

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

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

$$M^{G}_{ja\mathbf{k}}(\mathbf{q}) = \langle \psi_{j\mathbf{k}+\mathbf{q}} |\, e^{i(\mathbf{G}+\mathbf{q})\cdot\mathbf{r}} \,| \psi_{a\mathbf{k}} \rangle$$

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

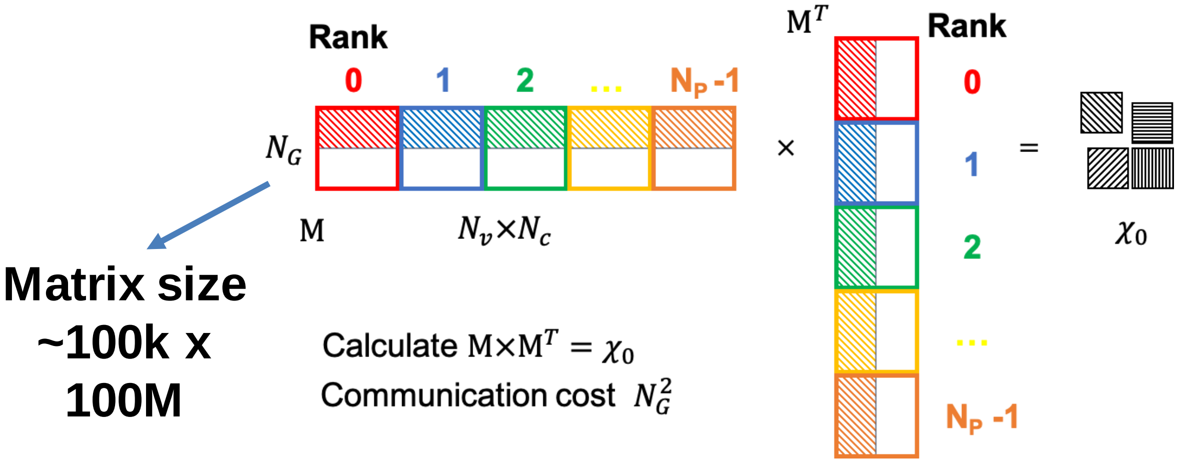
$$\chi(\mathbf{q},\omega_i) = \mathbf{M}(\mathbf{q})^{\dagger} \boldsymbol{\Delta}_{ja\mathbf{k}}(\epsilon_{j\mathbf{k}}, \epsilon_{a\mathbf{k}}, \mathbf{q}, \omega) \mathbf{M}(\mathbf{q})$$

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

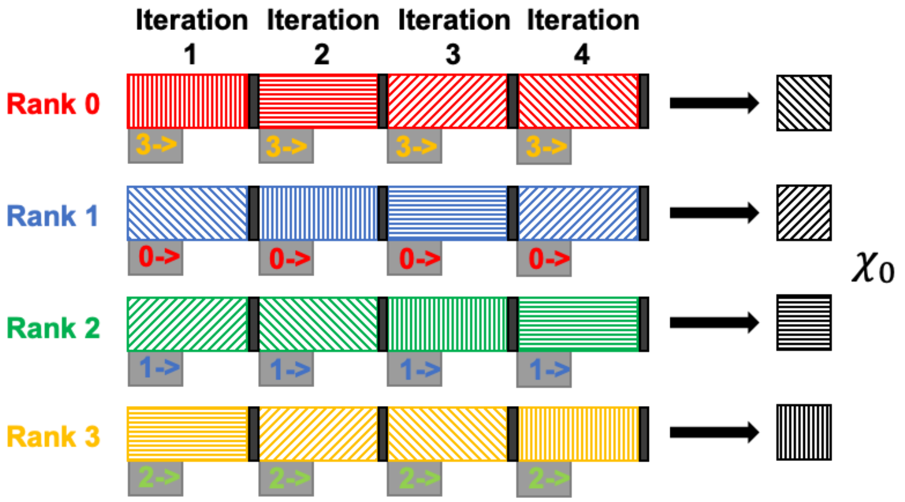
$$\epsilon^{-1}(\mathbf{q},\omega_i) = (I - v\chi(\mathbf{q},\omega_i))^{-1}$$

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)



Data layout for **M** matrix in CHI-0 kernel



Comput./Commun. pattern for non-blocking cyclic scheme

- Memory demanding O(N^3)

  Number of columns N_v x N_c, number of rows N_G

- Computationally intensive O(N^4)

  Implemented as ZGEMM operation (small prefactor)

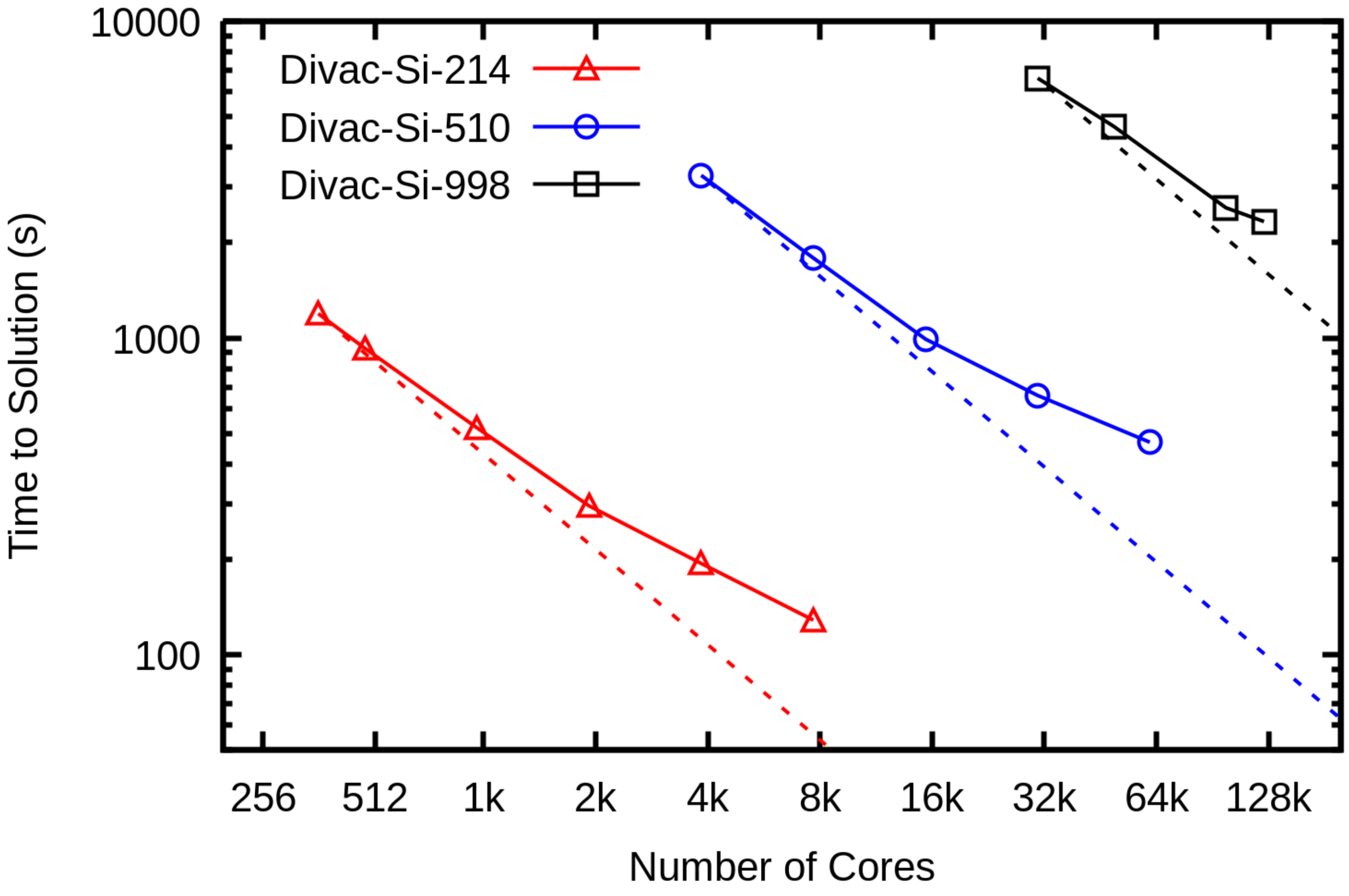
- Non-blocking cyclic communication

  Overlap Computation and MPI communication

- Repeated at multiple frequencies for full-frequency calculations

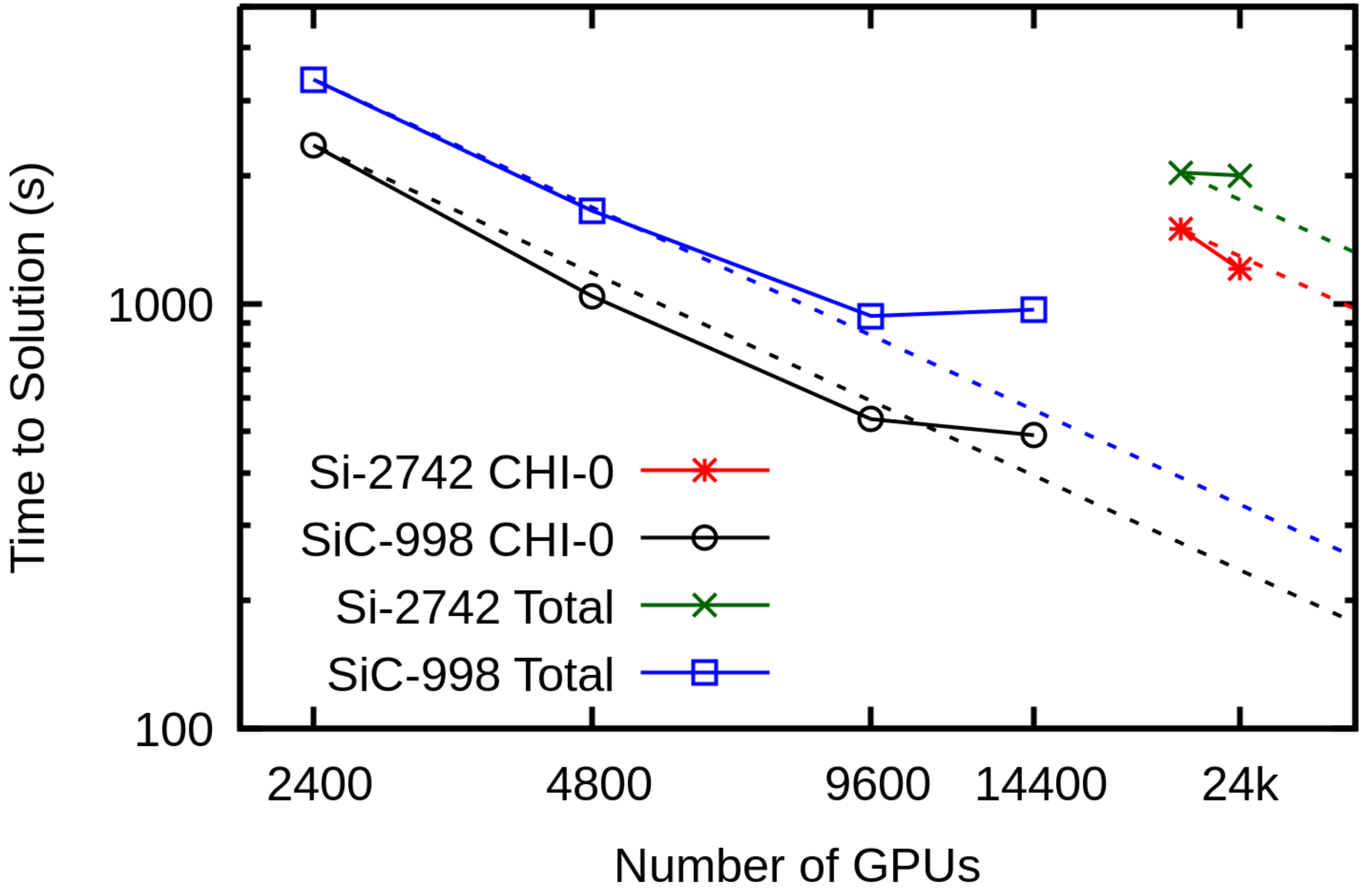
  Can be accelerated using low-rank approximation techniques

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)

Epsilon: The NVblock Algorithm

Implementation to alleviate the O(N^3) memory bottleneck in epsilon

- Computation is divided over **N** blocks of **V**alence bands (***NV block***)
- Block size depends on available memory
- Each batch repeated the same communication/computation pattern
- Particularly advantageous for GPUs were FLOPs are "for free"

Epsilon: The NVblock Algorithm

Implementation to alleviate the O(N^3) memory bottleneck in epsilon

- Computation is divided over **N** blocks of **V**alence bands (***NV block***)
- Block size depends on available memory
- Each batch repeated the same communication/computation pattern
- Particularly advantageous for GPUs were FLOPs are "for free"

Effectively memory requirements becomes O(N^2) at the expenses of little extra computation

Epsilon: The NVblock Algorithm

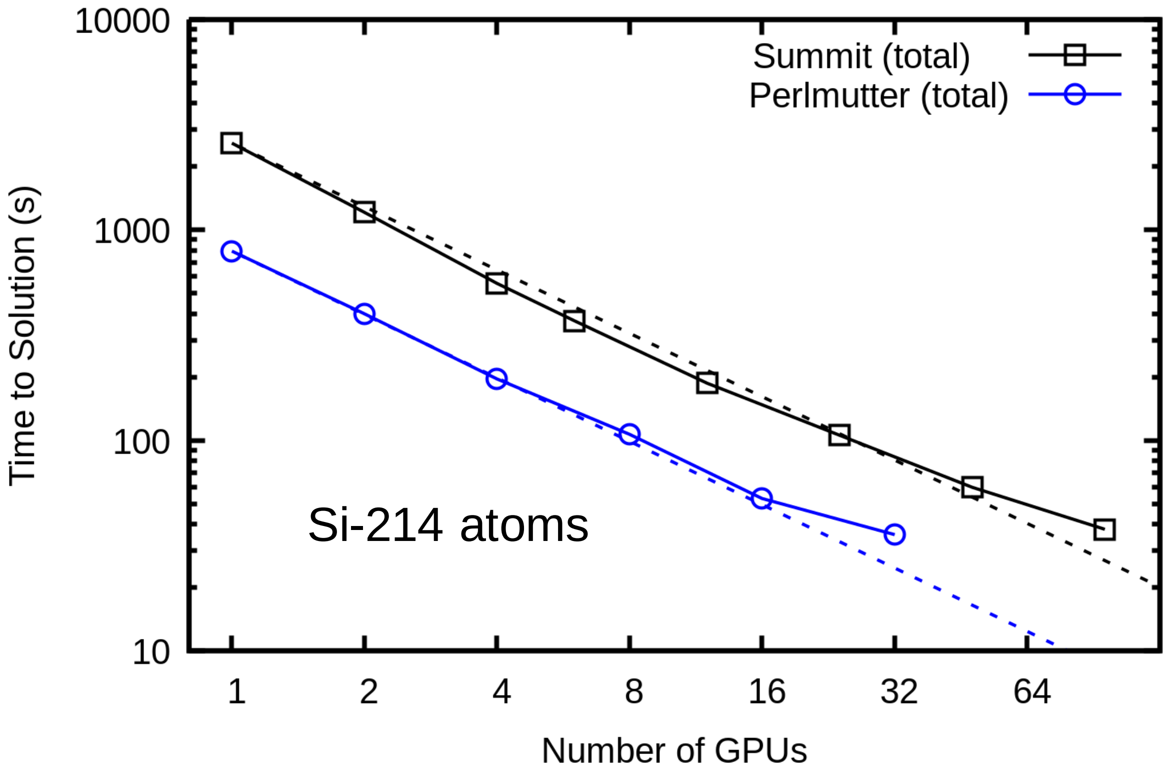
Implementation to alleviate the $O(N^3)$ memory bottleneck in epsilon

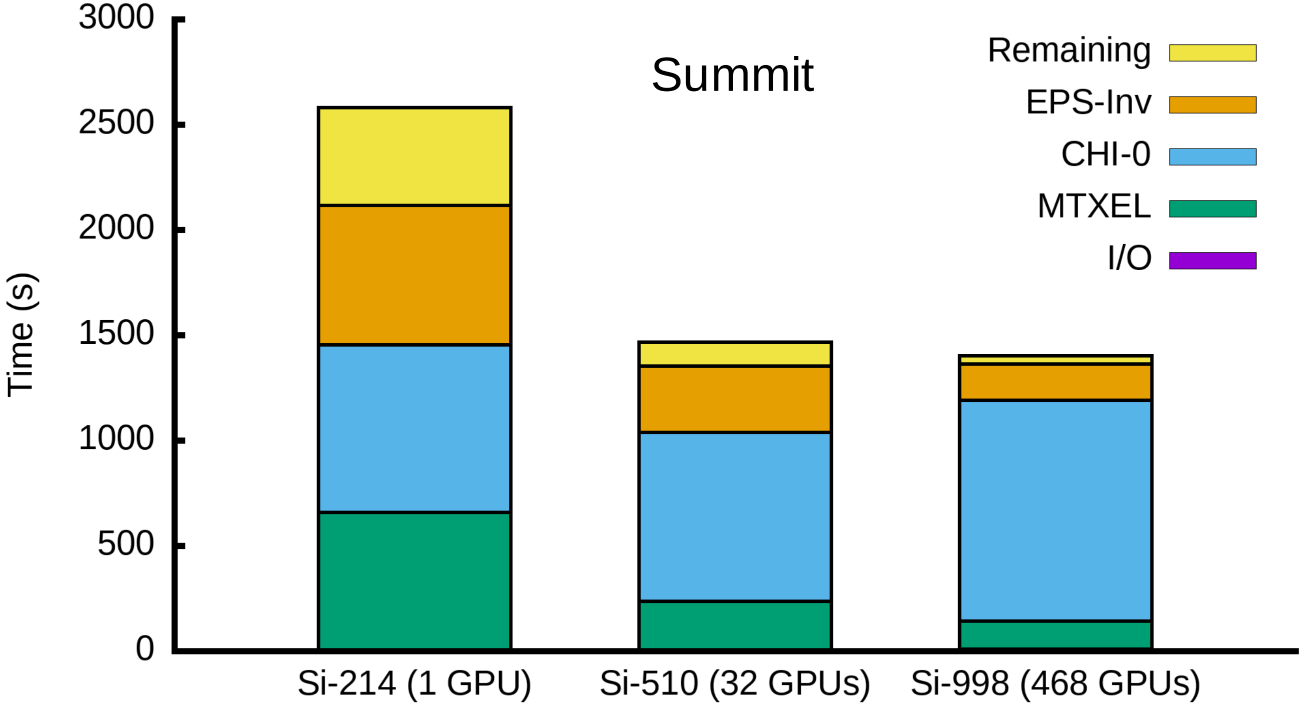
- Computation is divided over **N** blocks of **V**alence bands (***NV block***)
- Block size depends on available memory
- Each batch repeated the same communication/computation pattern
- Particularly advantageous for GPUs were FLOPs are "for free"

Effectively memory requirements becomes $O(N^2)$ at the expenses of little extra computation

Extremely large systems (>1000 atoms) became feasible with few GPU nodes

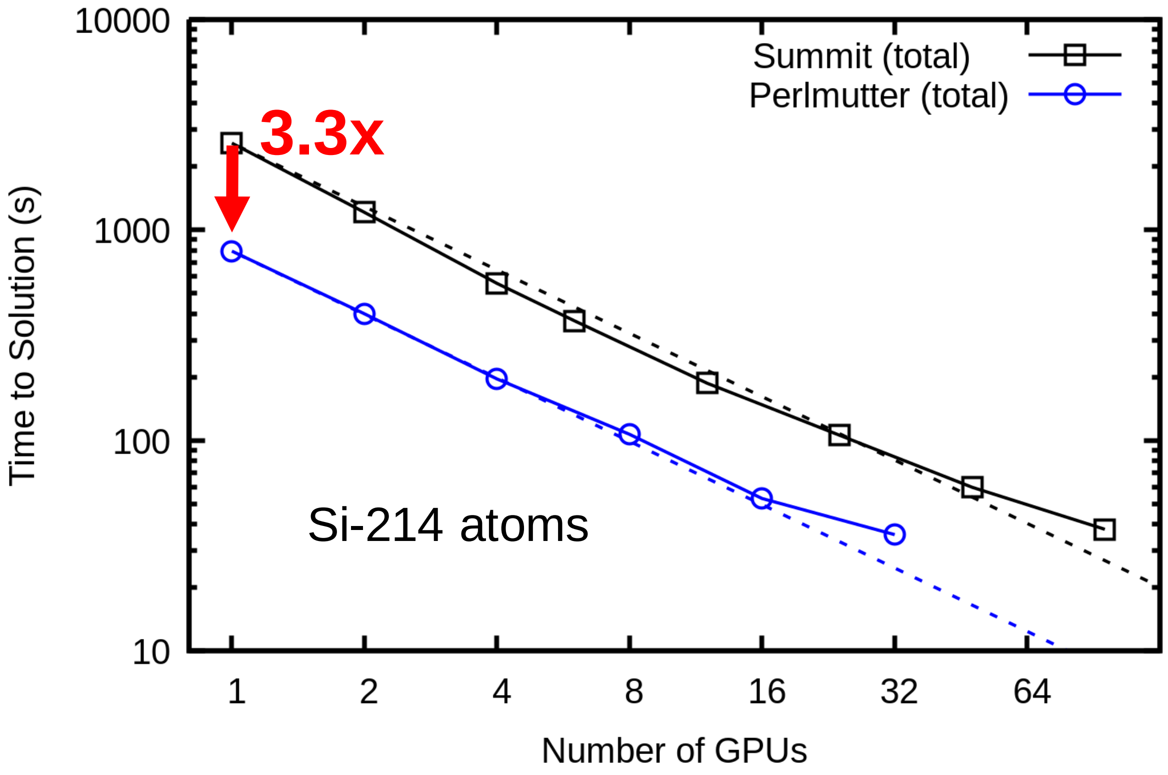
Epsilon: The NVblock Algorithm Performance

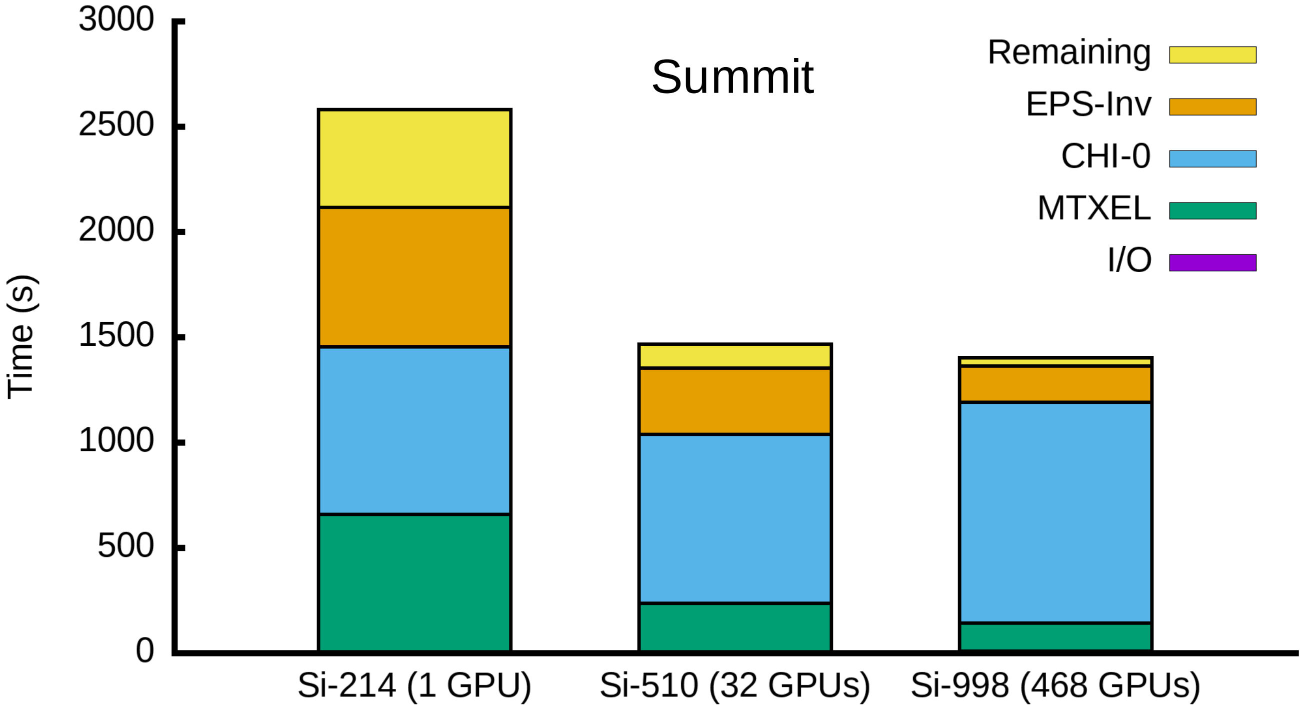




Strong/Weak Scaling of epsilon measured on Summit@OLCF and Perlmutter@NERSC
Summit Node: 2 IBM POWER9 CPUs and 6 NVIDIA V100 GPUs
Perlmutter Node: 1 AMD EPYC "Milan" CPU + 4 NVIDIA A100 GPUs

Epsilon: The NVblock Algorithm Performance





Strong/Weak Scaling of epsilon measured on Summit@OLCF and Perlmutter@NERSC
Summit Node: 2 IBM POWER9 CPUs and 6 NVIDIA V100 GPUs
Perlmutter Node: 1 AMD EPYC "Milan" CPU + 4 NVIDIA A100 GPUs

Sigma: Quasiparticle Properties

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

$$h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r},\mathbf{r}';E_n)\phi_n(\mathbf{r}')\mathrm{d}\mathbf{r}' = E_n\phi_n(\mathbf{r})$$

Sigma: Quasiparticle Properties

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

$$h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r},\mathbf{r}';E_n)\phi_n(\mathbf{r}')\mathrm{d}\mathbf{r}' = E_n\phi_n(\mathbf{r})$$

Each Self-Energy matrix element:

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

Sigma: Quasiparticle Properties

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

$$h_0(\mathbf{r})\phi_n(\mathbf{r}) + \int \Sigma(\mathbf{r},\mathbf{r}';E_n)\phi_n(\mathbf{r}')\mathrm{d}\mathbf{r}' = E_n\phi_n(\mathbf{r})$$

Each Self-Energy matrix element:

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

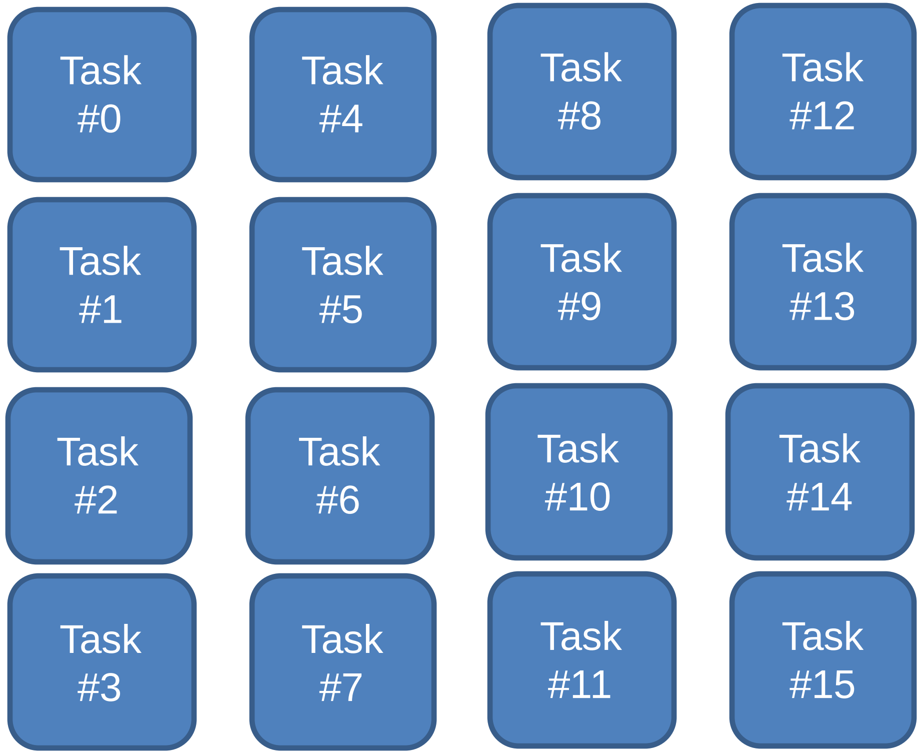
Frequency treatment

- 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*



Task #0	Task #4	Task #8	Task #12
Task #1	Task #5	Task #9	Task #13
Task #2	Task #6	Task #10	Task #14
Task #3	Task #7	Task #11	Task #15

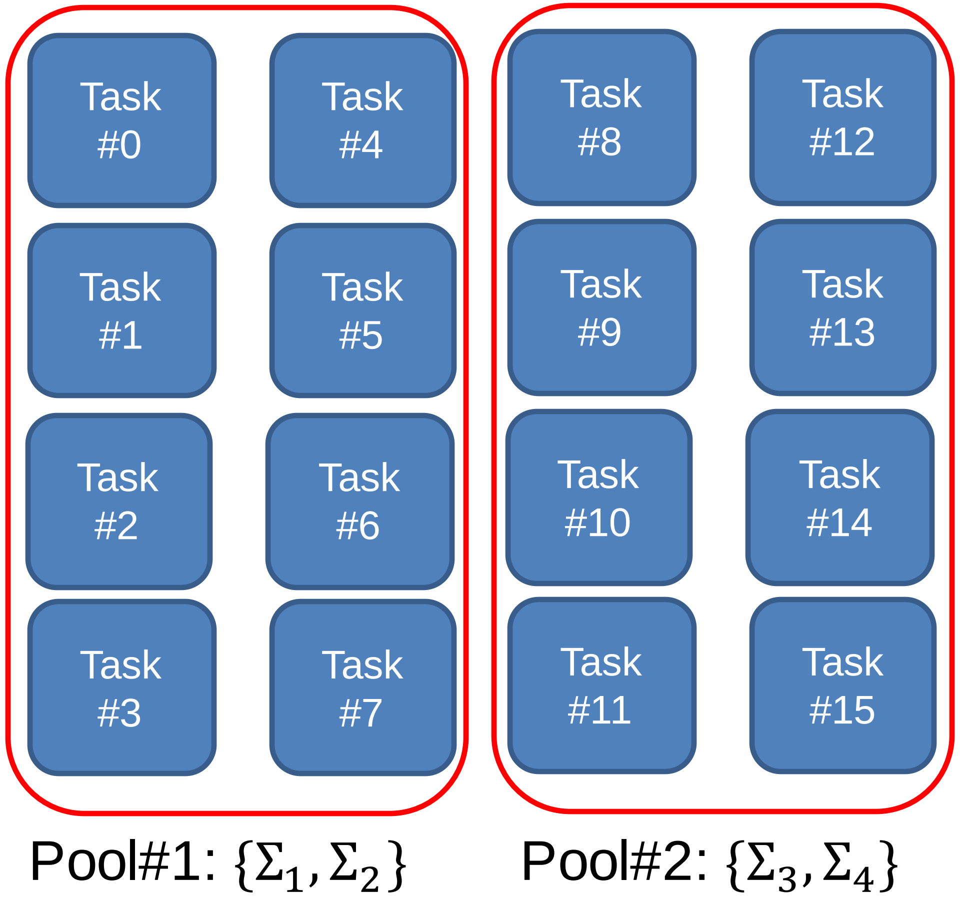
Two Level Parallelization Strategy:

- 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: $\{\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4\}$



Task #0
Task #4
Task #1
Task #5
Task #2
Task #6
Task #3
Task #7

Task #8
Task #12
Task #9
Task #13
Task #10
Task #14
Task #11
Task #15

Pool#1: $\{\Sigma_1, \Sigma_2\}$

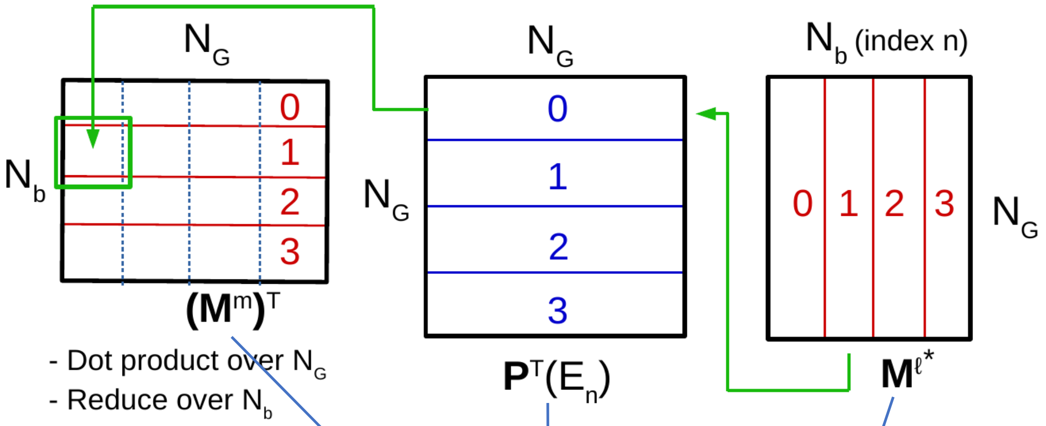
Pool#2: $\{\Sigma_3, \Sigma_4\}$

Two Level Parallelization Strategy:

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

Sigma: The GPP Kernel

Intra-Pool data layout



- Dot product over N_G
- Reduce over N_b

$$\Sigma_{lm} = \sum_{n}^{N_b} \sum_{GG'}^{N_G} M^{m}_{G'n} [P^{\mathsf{T}}(E_n)]_{G'G} M^{l*}_{Gn}$$

Two Level Parallelization Strategy:

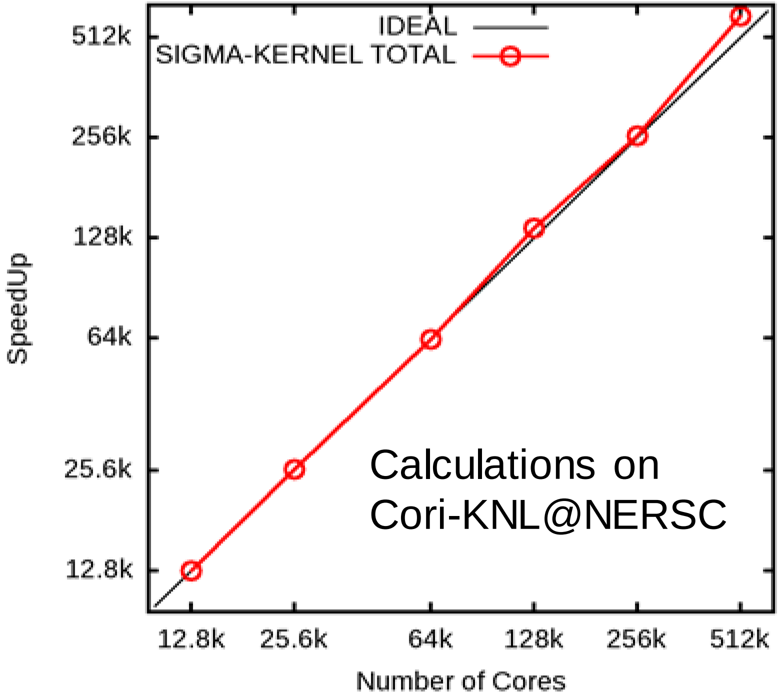
- Inter-Pool parallelization (*independent self-energy matrix elements*)
- **Intra-Pool parallelization** (*same self-energy matrix elements*)

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

For each E_{qp}

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

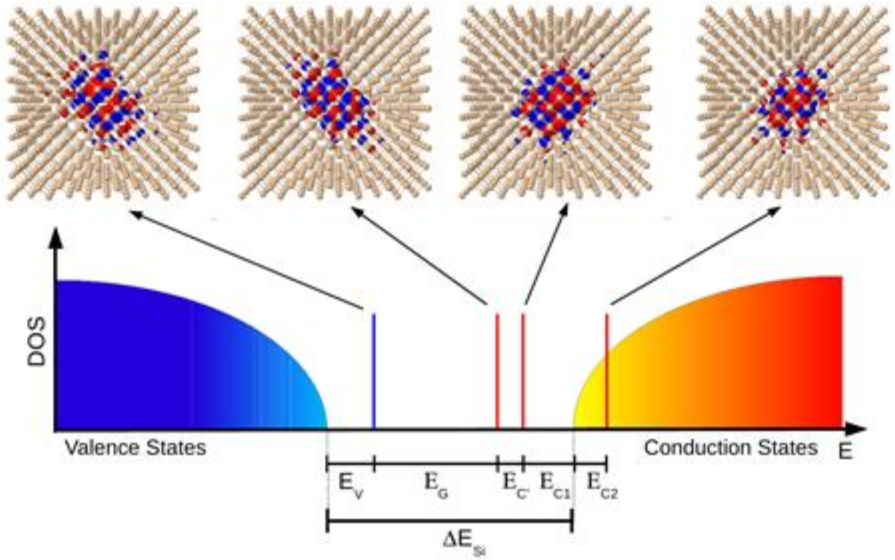
Sigma: Performance

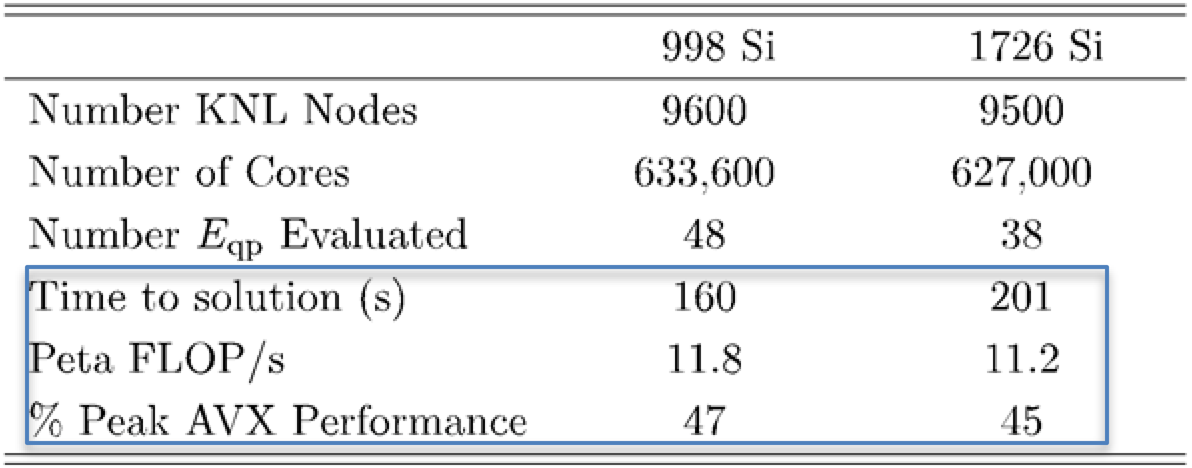


Calculations on Cori-KNL@NERSC

Divacancy in Silicon

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

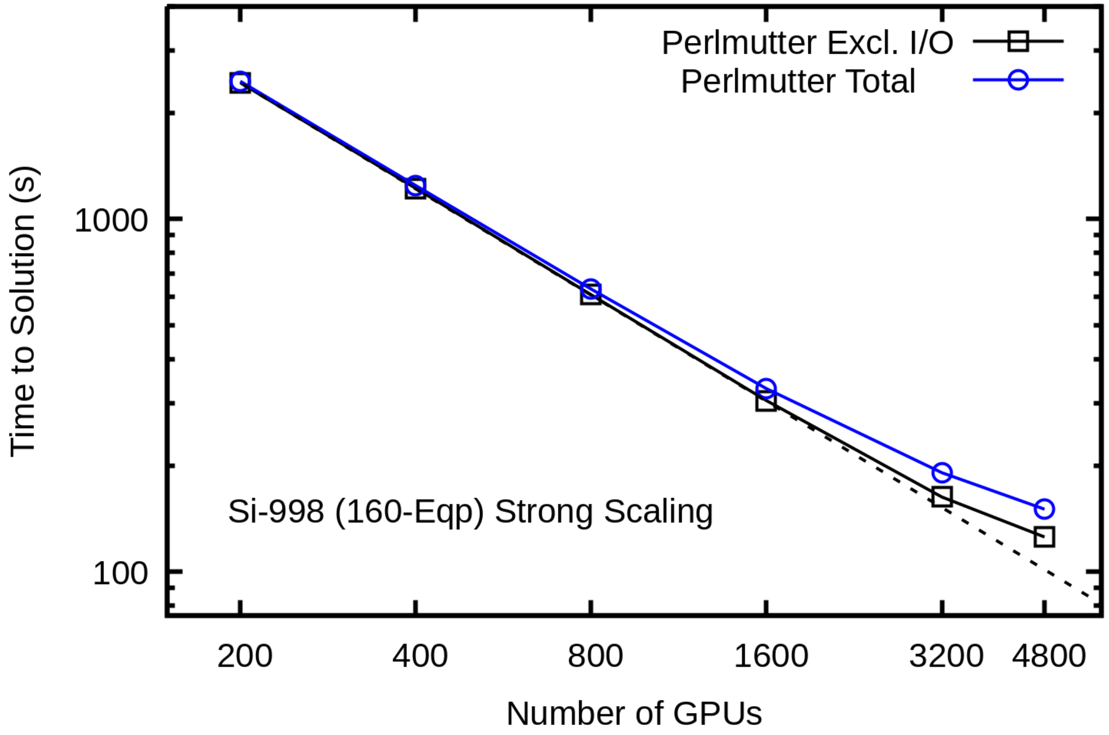


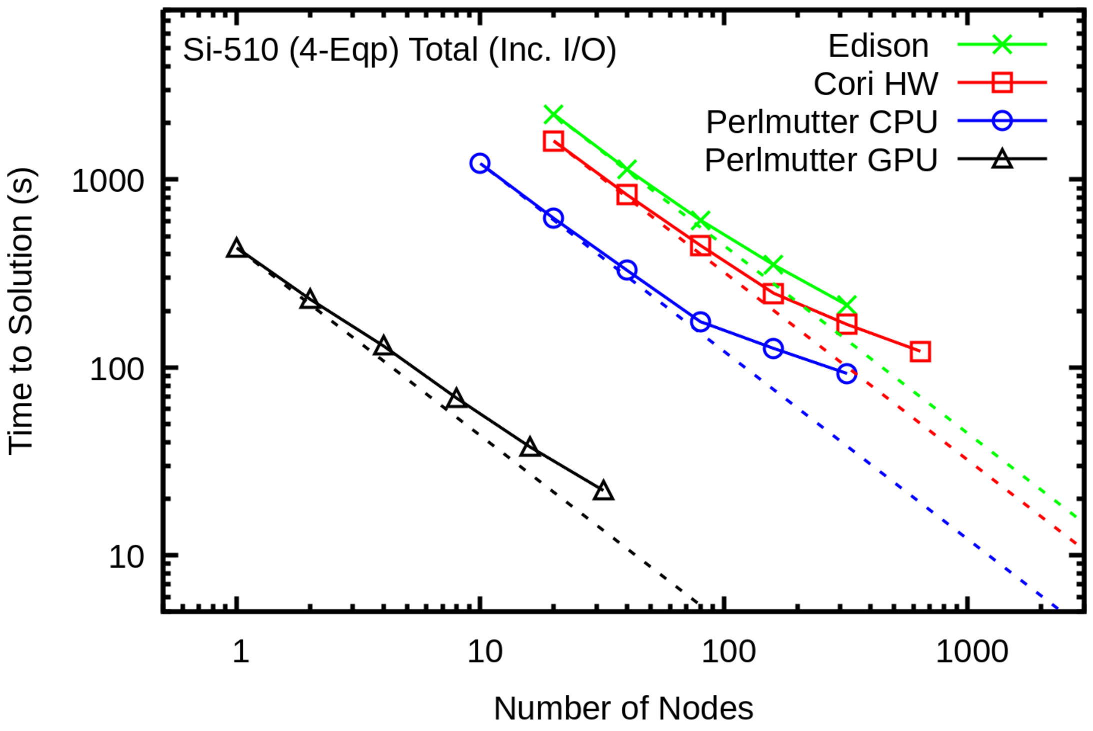


	998 Si	1726 Si
Number KNL Nodes	9600	9500
Number of Cores	633,600	627,000
Number E_{qp} Evaluated	48	38
Time to solution (s)	160	201
Peta FLOP/s	11.8	11.2
% Peak AVX Performance	47	45

M. Del Ben, F.H. da Jornada, A. Canning, N. Wichmann, K. Raman, R. Sasanka, C. Yang, S.G. Louie and J. Deslippe, *Comput. Phys. Commun.* **235**, 187-195 (2019)

Sigma: Performance on Perlmutter (GPU)

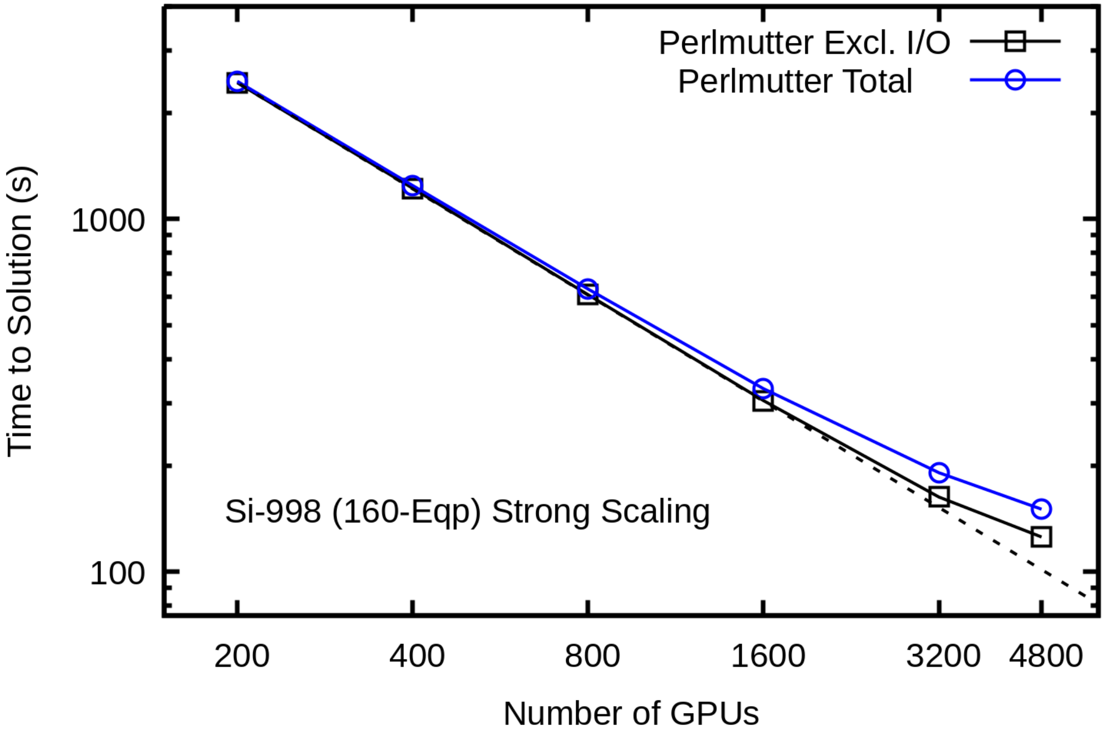


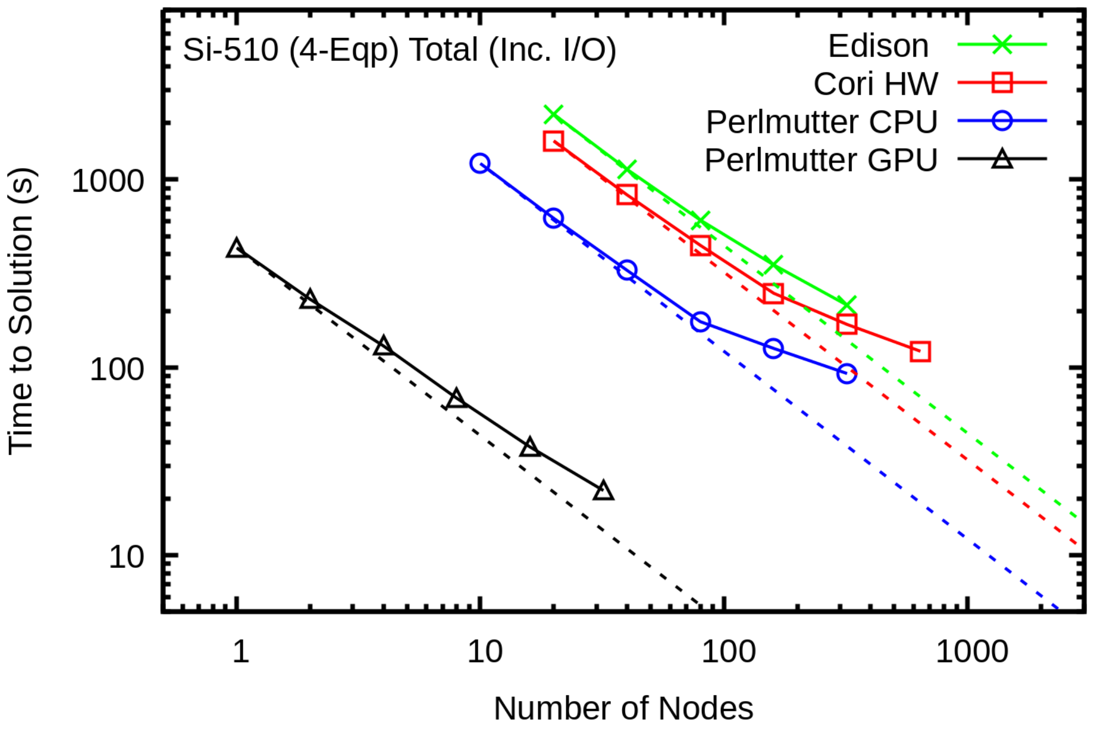


Strong Scaling for Sigma measured on Perlmutter@NERSC (Cray Shasta, Node: 2 AMD Milan + 4-A100 GPUs)

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

Sigma: Performance on Perlmutter (GPU)

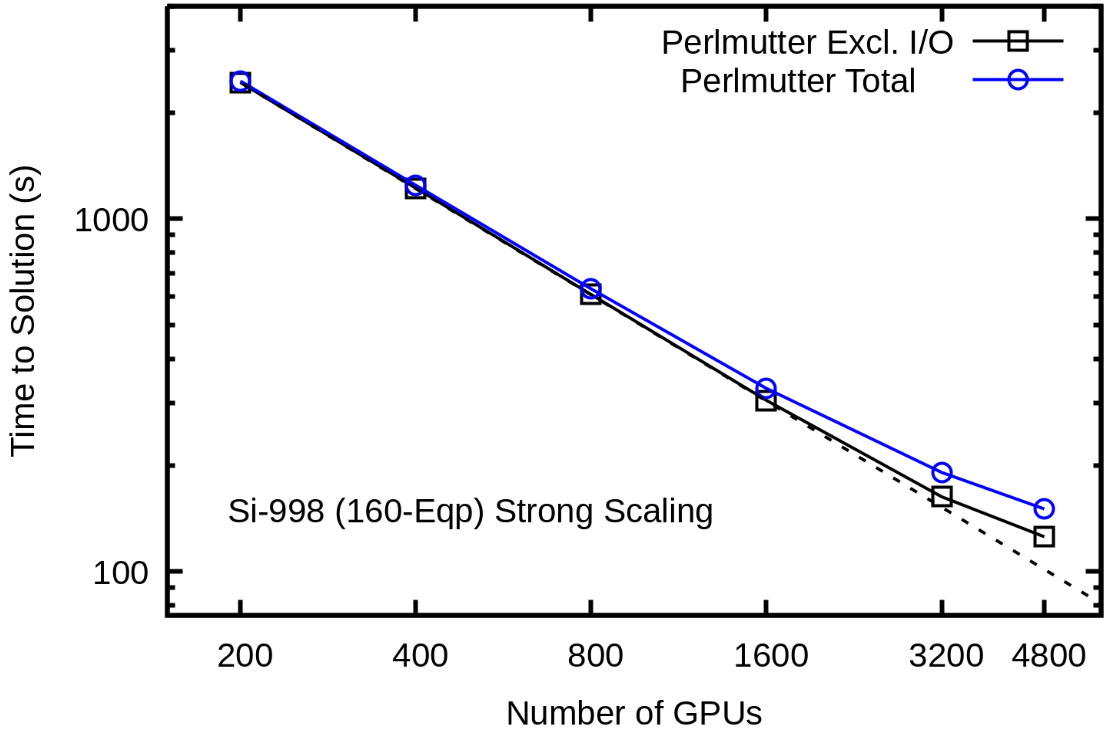


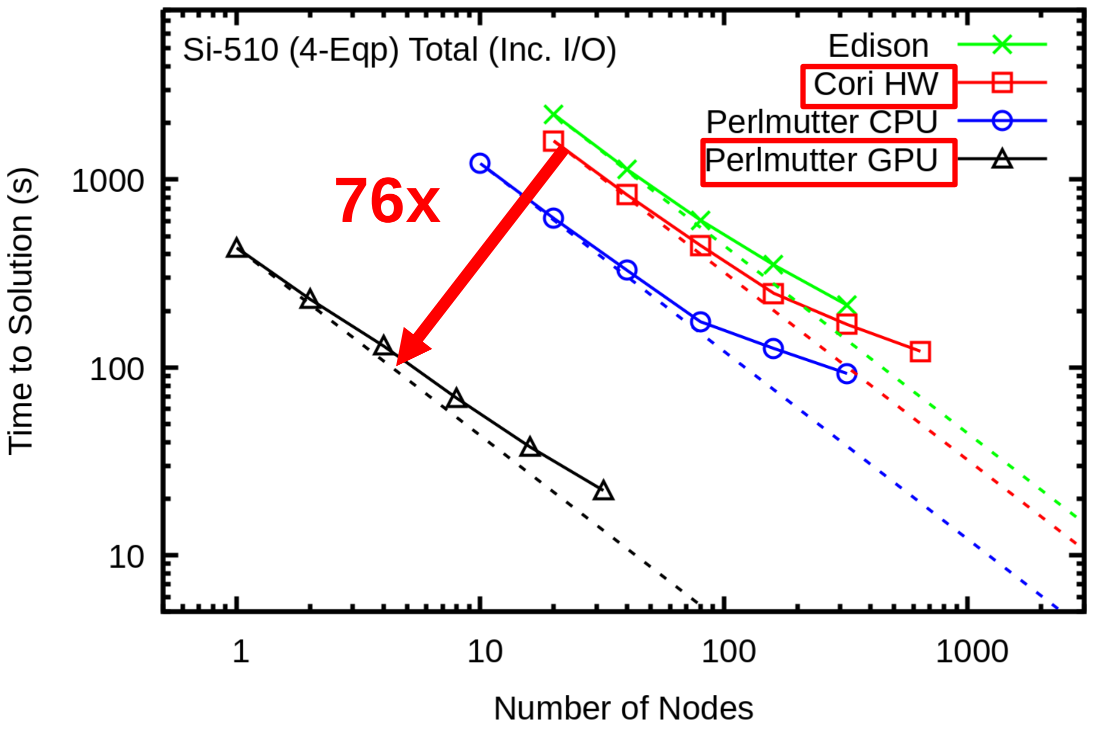


Strong Scaling for Sigma measured on Perlmutter@NERSC (Cray Shasta, Node: 2 AMD Milan + 4-A100 GPUs)

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

Sigma: Performance on Perlmutter (GPU)

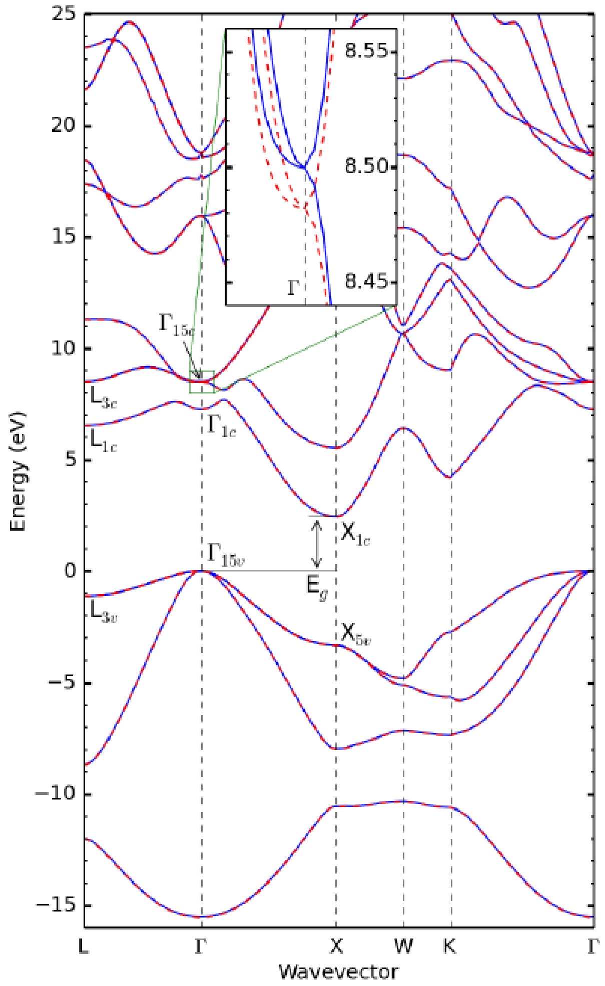


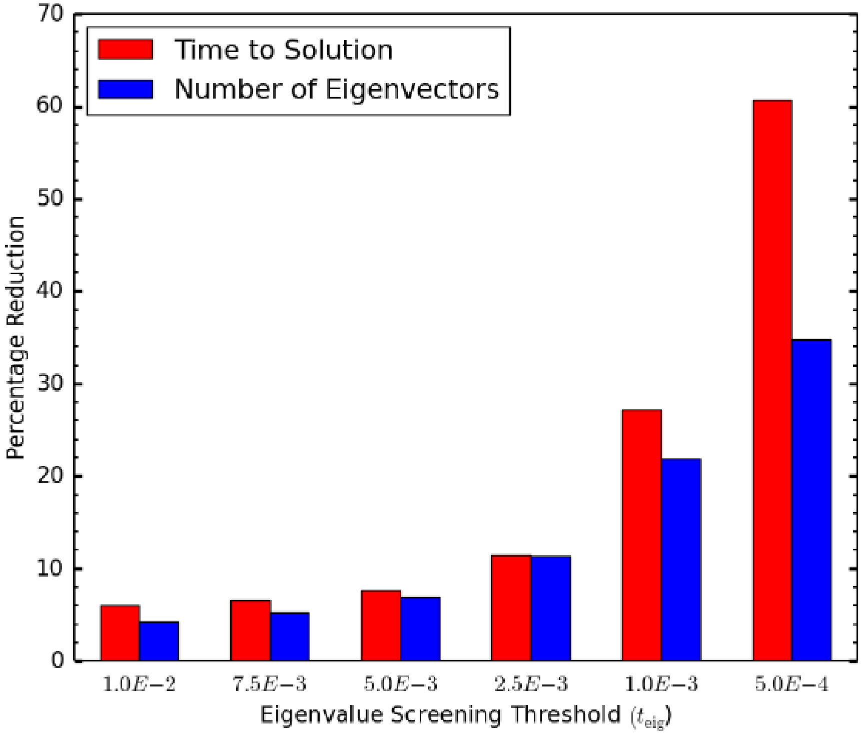


Strong Scaling for Sigma measured on Perlmutter@NERSC (Cray Shasta, Node: 2 AMD Milan + 4-A100 GPUs)

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

Speed-Up Full-Frequency GW Calculations





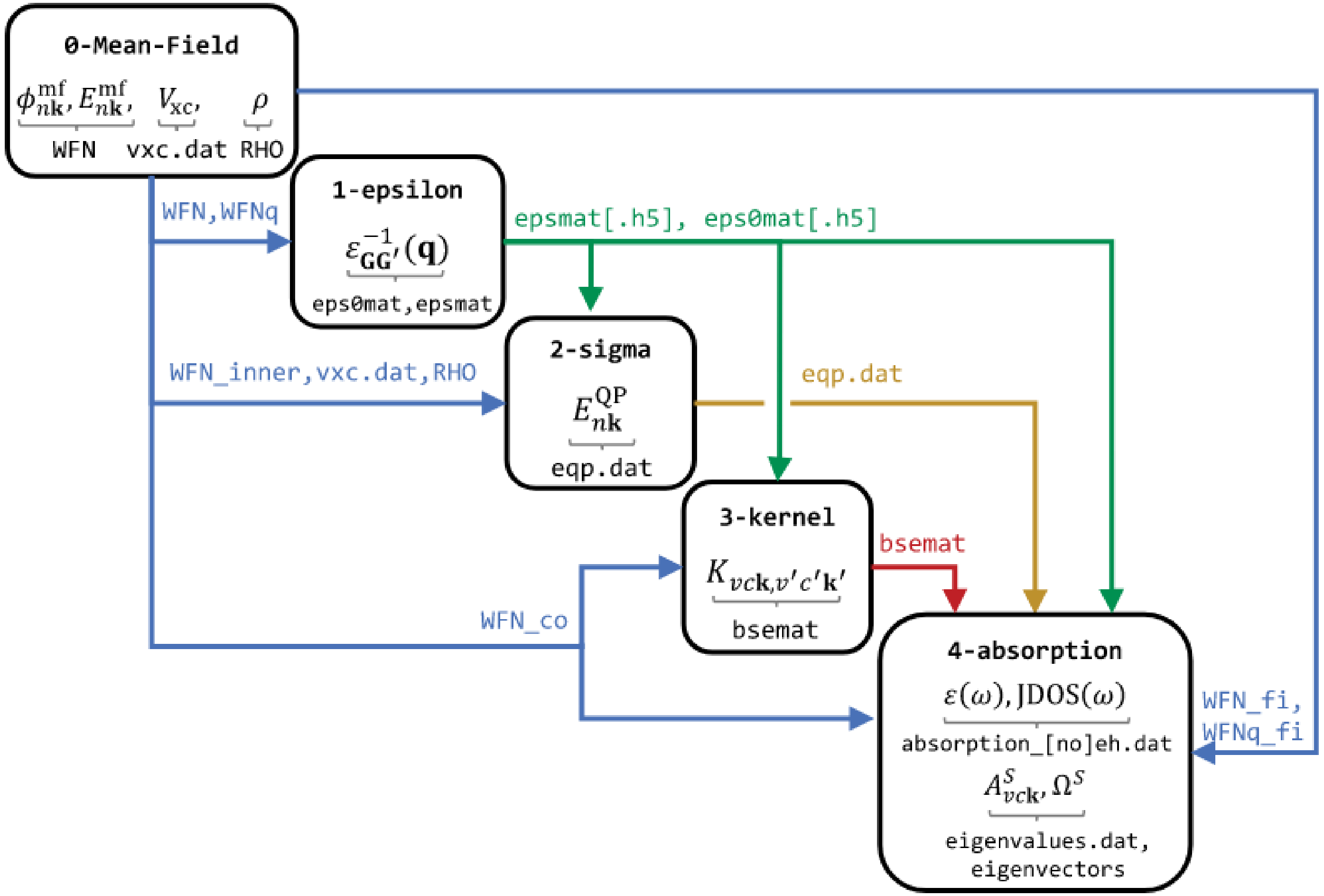
Silicon Carbide (β-SiC) band structure as obtained with and without approximation (0.01 threashold, 20x speed-up)

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

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

M. Govoni, and G.Galli, J. Chem. Theory Comput. 11, 2680 (2015) ; T.A. Pham, H.V. Nguyen, D. Rocca, and G. Galli, Phys. Rev. B 87, 155148 (2013) ; H.-V. Nguyen, T. A. Pham, D. Rocca, and G. Galli, Phys. Rev. B 85, 081101 (2012) ; H. F. Wilson, D. Lu, F. Gygi, and G. Galli, Phys. Rev. B 79, 245106 (2009) ; H F. Wilson, F. Gygi, and G. Galli, Phys. Rev. B 78, 113303 (2008) ; D. Lu, F. Gygi, and G. Galli, Phys. Rev. Lett. 100, 147601 (2008) ; M. Del Ben, F. H. da Jornada, J. Deslippe, S.G.Louie and A. Canning, Phys. Rev. B 99 (12), 125128 (2019)

The BSE Workflow: Kernel + Absorption

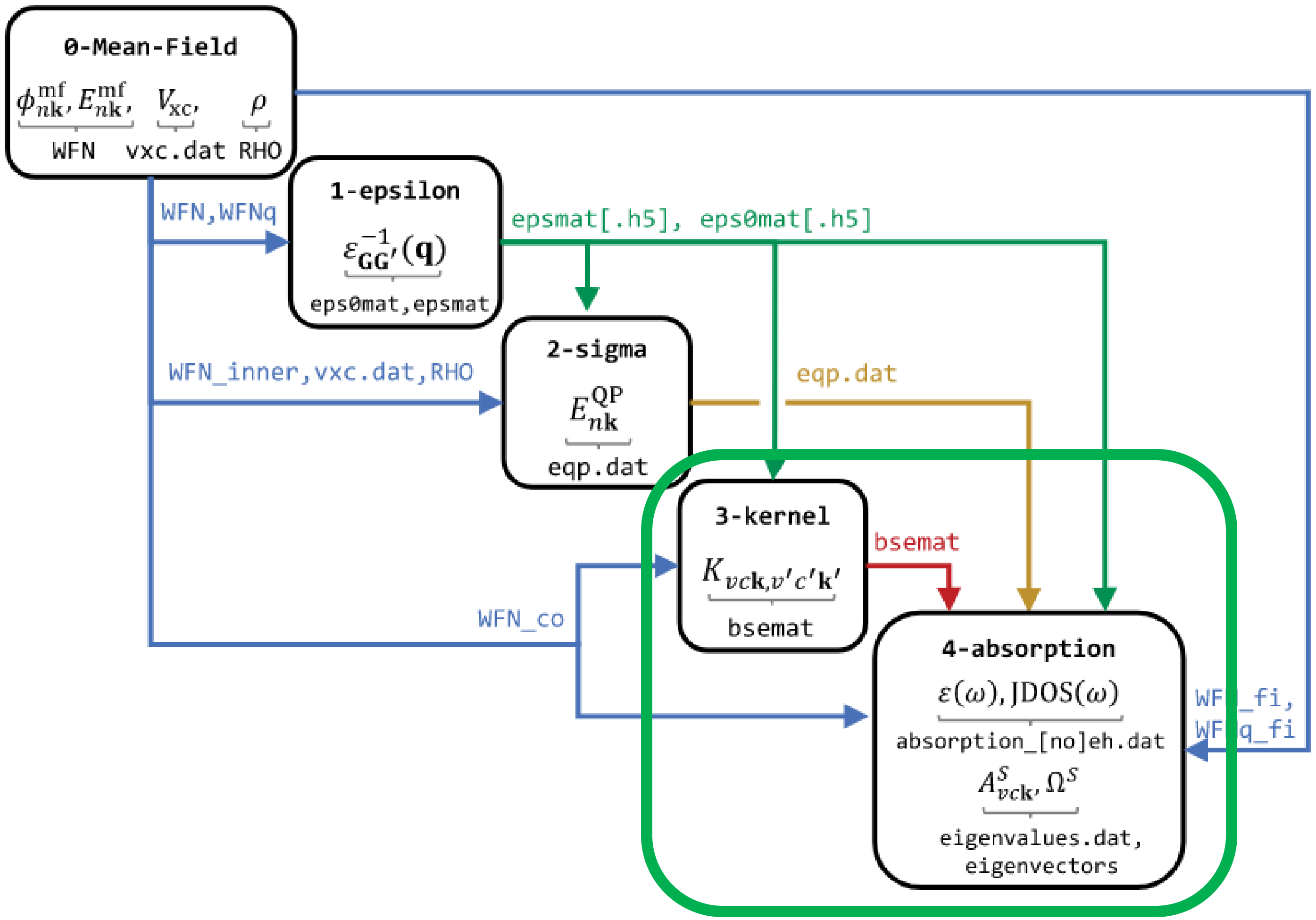

0-Mean-Field
$\underbrace{\phi^{\mathrm{mf}}_{n\mathbf{k}}, E^{\mathrm{mf}}_{n\mathbf{k}},}_{\mathtt{WFN}} \underbrace{V_{\mathrm{xc}},}_{\mathtt{vxc.dat}} \underbrace{\rho}_{\mathtt{RHO}}$
WFN,WFNq
1-epsilon
$\underbrace{\varepsilon^{-1}_{\mathbf{GG}'}(\mathbf{q})}_{\mathtt{eps0mat,epsmat}}$
epsmat[.h5], eps0mat[.h5]
WFN_inner,vxc.dat,RHO
2-sigma
$\underbrace{E^{\mathrm{QP}}_{n\mathbf{k}}}_{\mathtt{eqp.dat}}$
eqp.dat
3-kernel
$\underbrace{K_{vc\mathbf{k},v'c'\mathbf{k}'}}_{\mathtt{bsemat}}$
bsemat
WFN_co
4-absorption
$\underbrace{\varepsilon(\omega), \mathrm{JDOS}(\omega)}_{\mathtt{absorption\_[no]eh.dat}}$
$\underbrace{A^S_{vc\mathbf{k}}, \Omega^S}_{\mathtt{eigenvalues.dat,\ eigenvectors}}$
WFN_fi,
WFNq_fi

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

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

The BSE Workflow: Kernel + Absorption


0-Mean-Field
$\underbrace{\phi^{\mathrm{mf}}_{n\mathbf{k}}, E^{\mathrm{mf}}_{n\mathbf{k}}}_{\texttt{WFN}}, \underbrace{V_{\mathrm{xc}}}_{\texttt{vxc.dat}}, \underbrace{\rho}_{\texttt{RHO}}$
WFN,WFNq
1-epsilon
$\underbrace{\varepsilon^{-1}_{\mathbf{GG}'}(\mathbf{q})}_{\texttt{eps0mat,epsmat}}$
epsmat[.h5], eps0mat[.h5]
WFN_inner,vxc.dat,RHO
2-sigma
$\underbrace{E^{\mathrm{QP}}_{n\mathbf{k}}}_{\texttt{eqp.dat}}$
eqp.dat
3-kernel
$\underbrace{K_{vc\mathbf{k},v'c'\mathbf{k}'}}_{\texttt{bsemat}}$
bsemat
WFN_co
4-absorption
$\underbrace{\varepsilon(\omega), \mathrm{JDOS}(\omega)}_{\texttt{absorption\_[no]eh.dat}}$
$\underbrace{A^S_{vc\mathbf{k}}, \Omega^S}_{\texttt{eigenvalues.dat, eigenvectors}}$
WFN_fi,
WFNq_fi

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*:

$$\left(E_{c\mathbf{k}}^{\mathrm{QP}} - E_{v\mathbf{k}}^{\mathrm{QP}}\right) A_{vc\mathbf{k}}^{S} + \sum_{v'c'\mathbf{k}'} \langle vc\mathbf{k}| K^{\mathrm{eh}} |v'c'\mathbf{k}'\rangle A_{v'c'\mathbf{k}'}^{S} = \Omega^{S} A_{vc\mathbf{k}}^{S}$$

The BSE Workflow: Kernel + Absorption

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

$$\left(E^{\text{QP}}_{c\mathbf{k}} - E^{\text{QP}}_{v\mathbf{k}}\right) A^{S}_{vc\mathbf{k}} + \sum_{v'c'\mathbf{k}'} \langle vc\mathbf{k}|\, K^{\text{eh}}\, |v'c'\mathbf{k}'\rangle\, A^{S}_{v'c'\mathbf{k}'} = \Omega^{S} A^{S}_{vc\mathbf{k}}$$

→ Eigenvalue Problem

The BSE Workflow: Kernel + Absorption

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

$$\left(E^{\mathrm{QP}}_{c\mathbf{k}} - E^{\mathrm{QP}}_{v\mathbf{k}}\right) A^{S}_{vc\mathbf{k}} + \sum_{v'c'\mathbf{k}'} \langle vc\mathbf{k}|\, K^{\mathrm{eh}}\, |v'c'\mathbf{k}'\rangle\, A^{S}_{v'c'\mathbf{k}'} = \Omega^{S} A^{S}_{vc\mathbf{k}}$$

Eigenvalue Problem

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*:

$$\left(E^{\text{QP}}_{c\mathbf{k}} - E^{\text{QP}}_{v\mathbf{k}}\right) A^S_{vc\mathbf{k}} + \sum_{v'c'\mathbf{k}'} \langle vc\mathbf{k}| K^{\text{eh}} |v'c'\mathbf{k}'\rangle A^S_{v'c'\mathbf{k}'} = \Omega^S A^S_{vc\mathbf{k}}$$

⇨ Eigenvalue Problem

GW quasiparticle energies (diagonal matrix)

Electron-Hole interaction kernel (dense matrix)

The solution gives eigenvalues and eigenvectors:

- Excitation energy Ω^S
- Exciton wavefunction $A^S_{vc\mathbf{k}}$

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*:

$$\left(E^{\mathrm{QP}}_{c\mathbf{k}} - E^{\mathrm{QP}}_{v\mathbf{k}}\right) A^S_{vc\mathbf{k}} + \sum_{v'c'\mathbf{k}'} \langle vc\mathbf{k}|\, K^{\mathrm{eh}}\, |v'c'\mathbf{k}'\rangle\, A^S_{v'c'\mathbf{k}'} = \Omega^S A^S_{vc\mathbf{k}}$$

Eigenvalue Problem

GW quasiparticle energies (diagonal matrix)

Electron-Hole interaction kernel (dense matrix)

The solution gives eigenvalues and eigenvectors:

- Excitation energy Ω^S
- Exciton wavefunction $A^S_{vc\mathbf{k}}$

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*:

$$\left(E^{\mathrm{QP}}_{c\mathbf{k}} - E^{\mathrm{QP}}_{v\mathbf{k}}\right) A^S_{vc\mathbf{k}} + \sum_{v'c'\mathbf{k}'} \langle vc\mathbf{k}|\, K^{\mathrm{eh}}\, |v'c'\mathbf{k}'\rangle\, A^S_{v'c'\mathbf{k}'} = \Omega^S A^S_{vc\mathbf{k}} \Longrightarrow$$

Eigenvalue Problem

GW quasiparticle energies (diagonal matrix)

Electron-Hole interaction kernel (dense matrix)

The solution gives eigenvalues and eigenvectors:

- Excitation energy Ω^S
- Exciton wavefunction $A^S_{vc\mathbf{k}}$

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)**

The BSE Workflow: Kernel + Absorption

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

$$\left(E^{\text{QP}}_{c\mathbf{k}} - E^{\text{QP}}_{v\mathbf{k}}\right) A^S_{vc\mathbf{k}} + \sum_{v'c'\mathbf{k}'} \langle vc\mathbf{k}|\, K^{\text{eh}} \,|v'c'\mathbf{k}'\rangle\, A^S_{v'c'\mathbf{k}'} = \Omega^S A^S_{vc\mathbf{k}} \Longrightarrow$$

Eigenvalue Problem

GW quasiparticle energies (diagonal matrix)

Electron-Hole interaction kernel (dense matrix)

The solution gives eigenvalues and eigenvectors:

- Excitation energy Ω^S
- Exciton wavefunction $A^S_{vc\mathbf{k}}$

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 $K^{\mathrm{eh}} = K^{\mathrm{d}} + K^{\mathrm{x}}$.

Kernel: Kernel Matrix Elements

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

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

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

Kernel: Kernel Matrix Elements

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

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)

$$\langle vc\mathbf{k}|K^{\mathrm{d}}|v'c'\mathbf{k}'\rangle = \sum_{\mathbf{G}\mathbf{G}'} M_{cc'}(\mathbf{k},\mathbf{q},\mathbf{G}) W_{\mathbf{G}\mathbf{G}'}(\mathbf{q};0) M^{*}_{vv'}(\mathbf{k},\mathbf{q},\mathbf{G}')$$

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

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

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

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

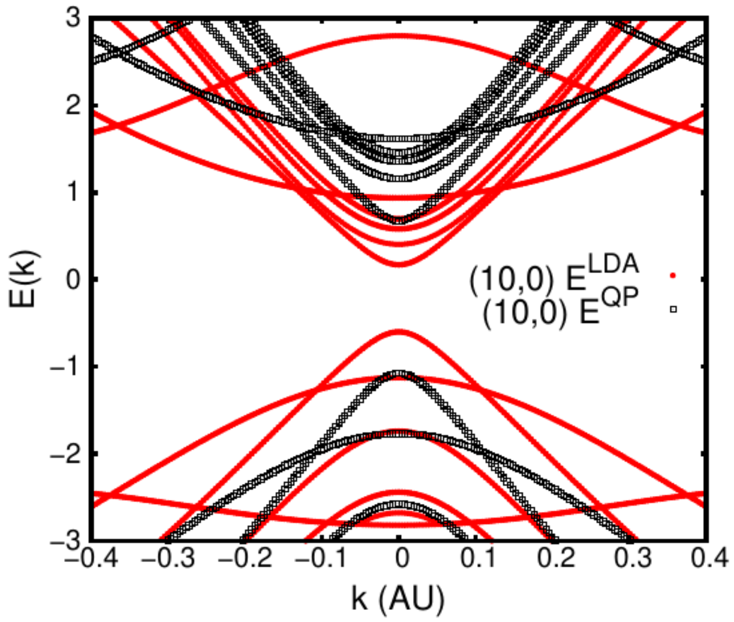
$$C^{\mathbf{k}_{\text{co}}}_{n,n'} = \int d\mathbf{r}\, u_{n\mathbf{k}_{\text{fi}}}(\mathbf{r}) u^{*}_{n'\mathbf{k}_{\text{co}}}(\mathbf{r})$$

2) Use overlaps to interpolate Kernel to Fine Grid

$$\begin{aligned}&\langle vc\mathbf{k}_{\text{fi}}|K|v'c'\mathbf{k}'_{\text{fi}}\rangle = \\ &\quad \sum_{n_1,n_2,n_3,n_4} C^{\mathbf{k}_{\text{co}}}_{c,n_1} C^{*\mathbf{k}_{\text{co}}}_{v,n_2} C^{*\mathbf{k}'_{\text{co}}}_{c',n_3} C^{\mathbf{k}'_{\text{co}}}_{v',n_4} \langle n_2 n_1 \mathbf{k}_{\text{co}}|K|n_4 n_3 \mathbf{k}'_{\text{co}}\rangle\end{aligned}$$

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

$$\begin{aligned}&E^{\text{QP}}_{n}(\mathbf{k}_{\text{fi}}) = \\ &\quad E^{\text{MF}}_{n}(\mathbf{k}_{\text{fi}}) + \langle \sum_{n'} |C^{\mathbf{k}_{\text{co}}}_{n,n'}|^2 (E^{\text{QP}}_{n'}(\mathbf{k}_{\text{co}}) - E^{\text{MF}}_{n'}(\mathbf{k}_{\text{co}}))\rangle_{\mathbf{k}_{\text{co}}}\end{aligned}$$



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

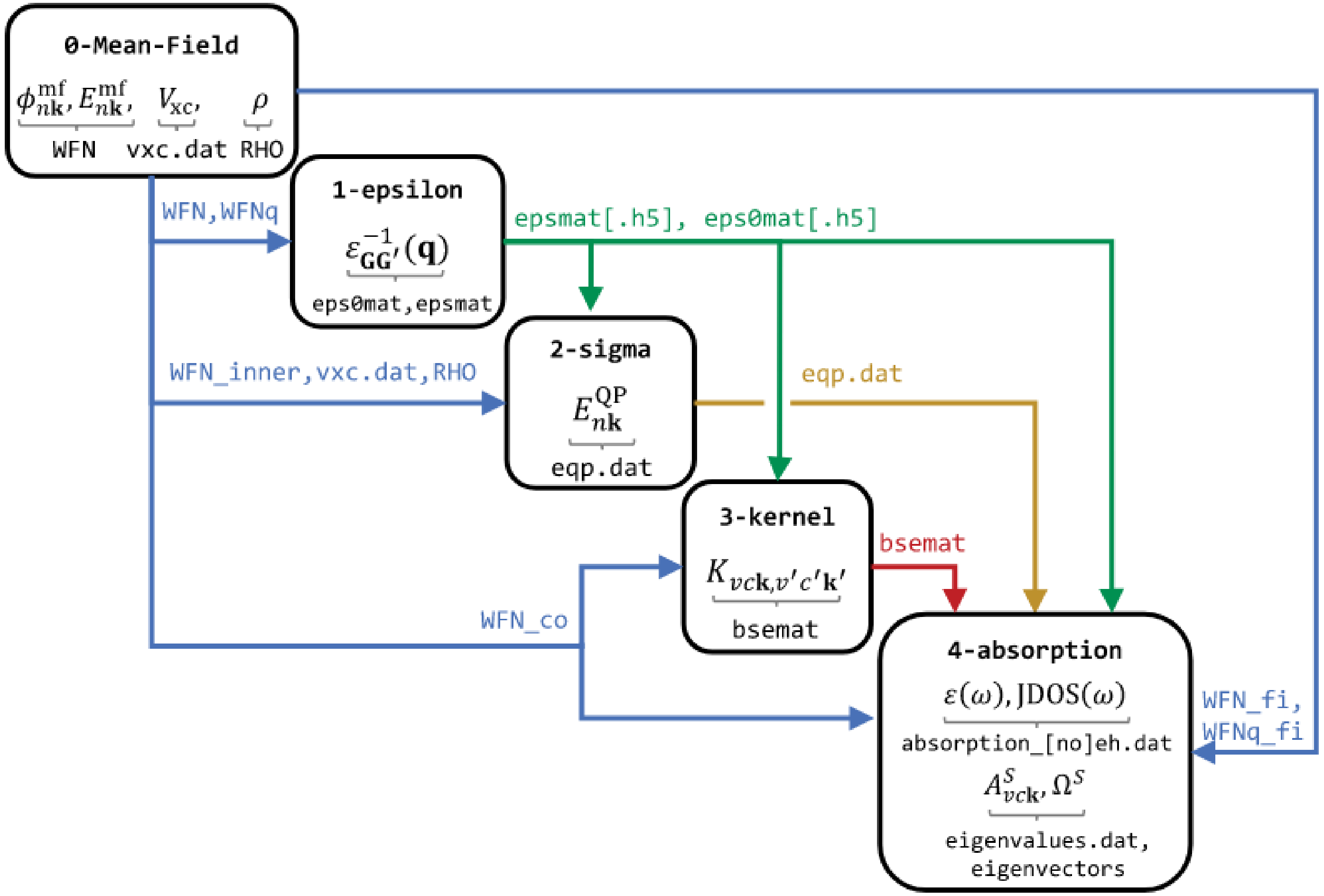
Fine **k**-grid: $$\left(E^{\text{QP}}_{c\mathbf{k}} - E^{\text{QP}}_{v\mathbf{k}}\right) A^S_{vc\mathbf{k}} + \sum_{v'c'\mathbf{k}'} \langle vc\mathbf{k}|\, K^{\text{eh}} \,|v'c'\mathbf{k}'\rangle\, A^S_{v'c'\mathbf{k}'} = \Omega^S A^S_{vc\mathbf{k}}$$

Interpolated E^{QP}

Interpolated kernel matrix elements

- 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


0-Mean-Field
$\underbrace{\phi^{\rm mf}_{n\mathbf{k}}, E^{\rm mf}_{n\mathbf{k}},}_{\text{WFN}} \underbrace{V_{\rm xc},}_{\text{vxc.dat}} \underbrace{\rho}_{\text{RHO}}$
WFN,WFNq
1-epsilon
$\underbrace{\varepsilon^{-1}_{\mathbf{GG}'}(\mathbf{q})}_{\text{eps0mat,epsmat}}$
epsmat[.h5], eps0mat[.h5]
WFN_inner,vxc.dat,RHO
2-sigma
$\underbrace{E^{\rm QP}_{n\mathbf{k}}}_{\text{eqp.dat}}$
eqp.dat
3-kernel
$\underbrace{K_{vc\mathbf{k},v'c'\mathbf{k}'}}_{\text{bsemat}}$
bsemat
WFN_co
4-absorption
$\underbrace{\varepsilon(\omega), \mathrm{JDOS}(\omega)}_{\text{absorption\_[no]eh.dat}}$
$\underbrace{A^S_{vc\mathbf{k}}, \Omega^S}_{\text{eigenvalues.dat, eigenvectors}}$
WFN_fi,
WFNq_fi

GW Workflow

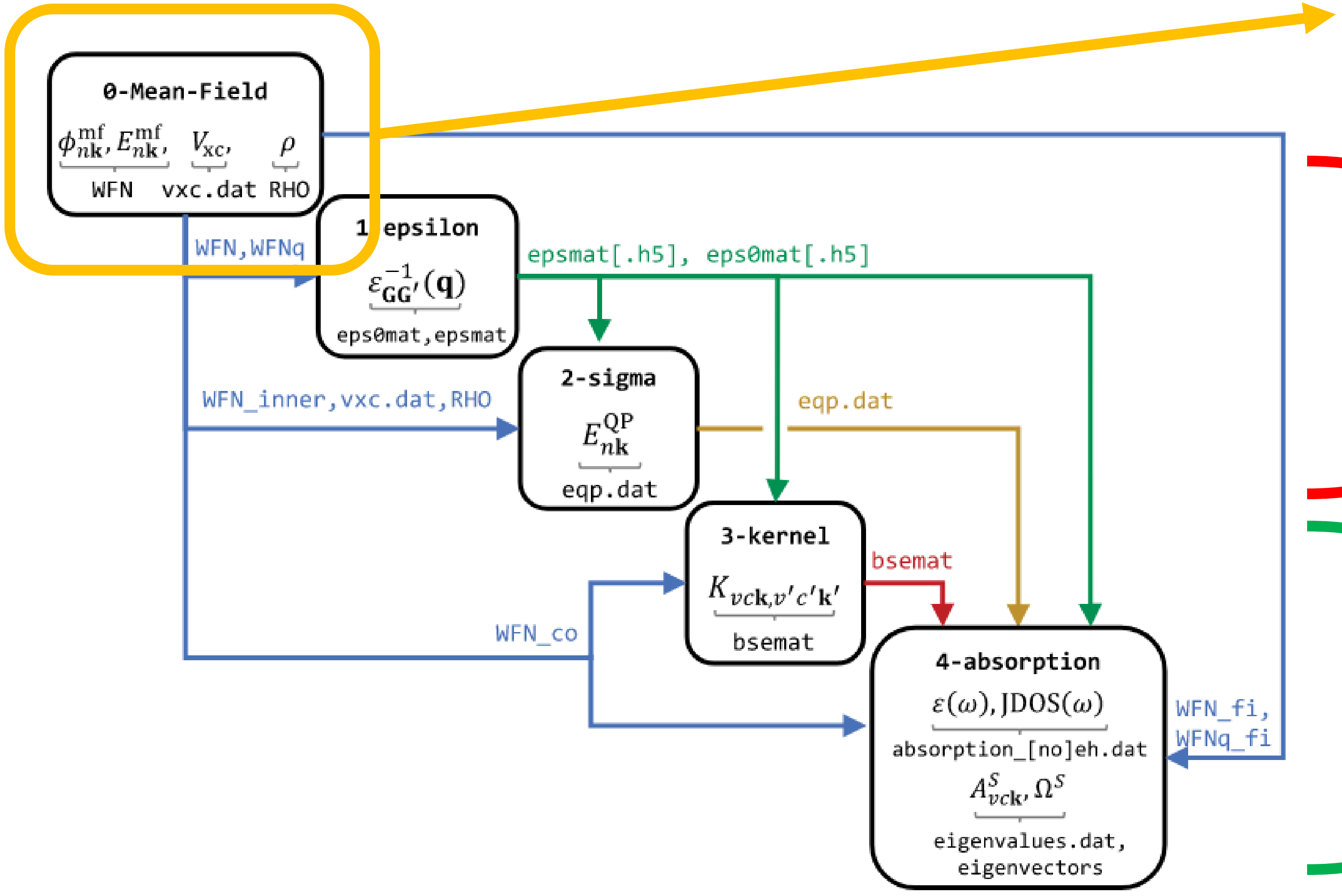
epsilon + sigma

BSE Workflow

kernel + absorption

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

The ParaBands Tool


0-Mean-Field
$\underbrace{\phi^{\text{mf}}_{n\mathbf{k}}, E^{\text{mf}}_{n\mathbf{k}}}_{\text{WFN}}, \underbrace{V_{\text{xc}}}_{\text{vxc.dat}}, \underbrace{\rho}_{\text{RHO}}$
WFN,WFNq
1-epsilon
$\underbrace{\varepsilon^{-1}_{\mathbf{GG'}}(\mathbf{q})}_{\text{eps0mat,epsmat}}$
epsmat[.h5], eps0mat[.h5]
WFN_inner,vxc.dat,RHO
2-sigma
$\underbrace{E^{\text{QP}}_{n\mathbf{k}}}_{\text{eqp.dat}}$
eqp.dat
3-kernel
$\underbrace{K_{vc\mathbf{k},v'c'\mathbf{k'}}}_{\text{bsemat}}$
bsemat
WFN_co
4-absorption
$\underbrace{\varepsilon(\omega), \text{JDOS}(\omega)}_{\text{absorption\_[no]eh.dat}}$
$\underbrace{A^S_{vc\mathbf{k}}, \Omega^S}_{\text{eigenvalues.dat, eigenvectors}}$
WFN_fi,
WFNq_fi

Generate a large number of empty states
parabands

GW Workflow
epsilon + sigma

BSE Workflow
kernel + absorption

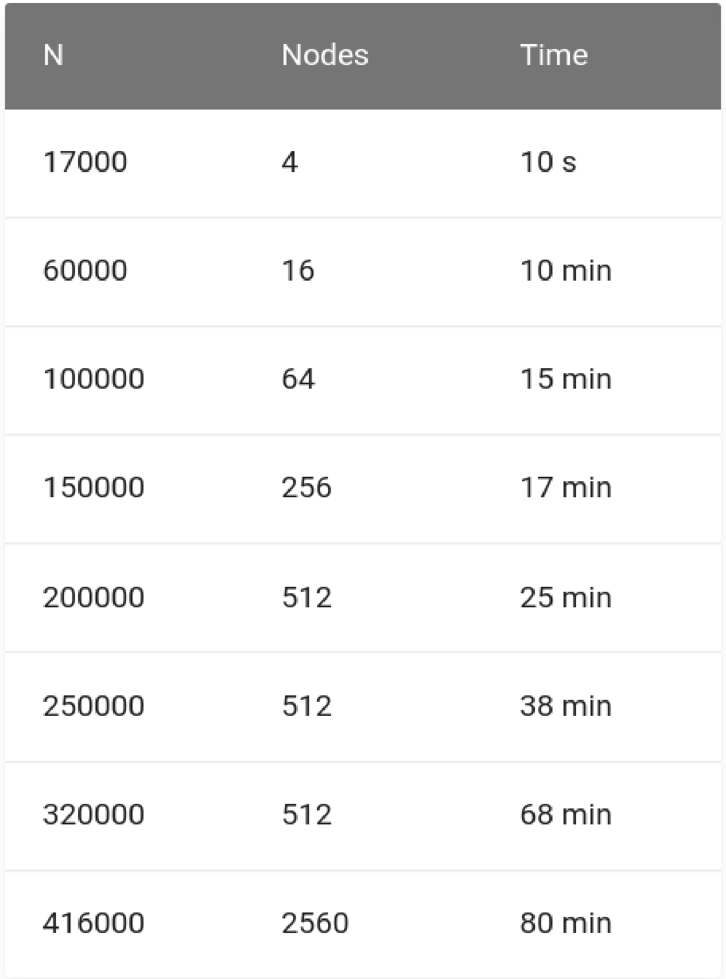
http://manual.berkeleygw.org/3.0/overview-workflow/

The ParaBands Tool

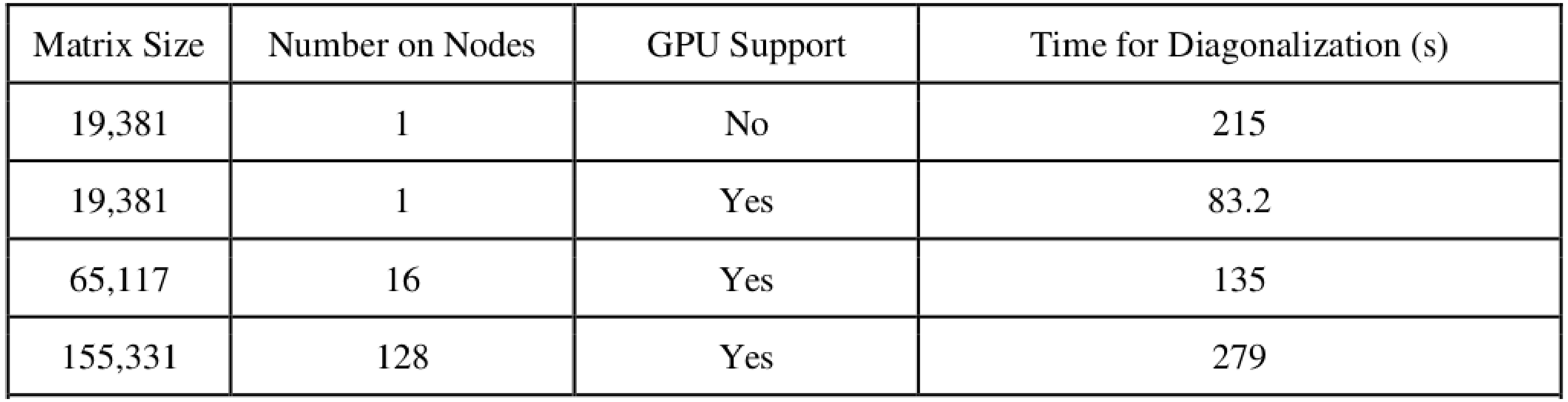
ParaBands builds and diagonalizes the dense DFT Hamiltonian in plane wave basis using a direct solver to obtain a large number of empty bands.

- 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



N	Nodes	Time
17000	4	10 s
60000	16	10 min
100000	64	15 min
150000	256	17 min
200000	512	25 min
250000	512	38 min
320000	512	68 min
416000	2560	80 min



Matrix Size	Number on Nodes	GPU Support	Time for Diagonalization (s)
19,381	1	No	215
19,381	1	Yes	83.2
65,117	16	Yes	135
155,331	128	Yes	279

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

Summary

NSF

U.S. DEPARTMENT OF ENERGY

TACC
TEXAS ADVANCED COMPUTING CENTER

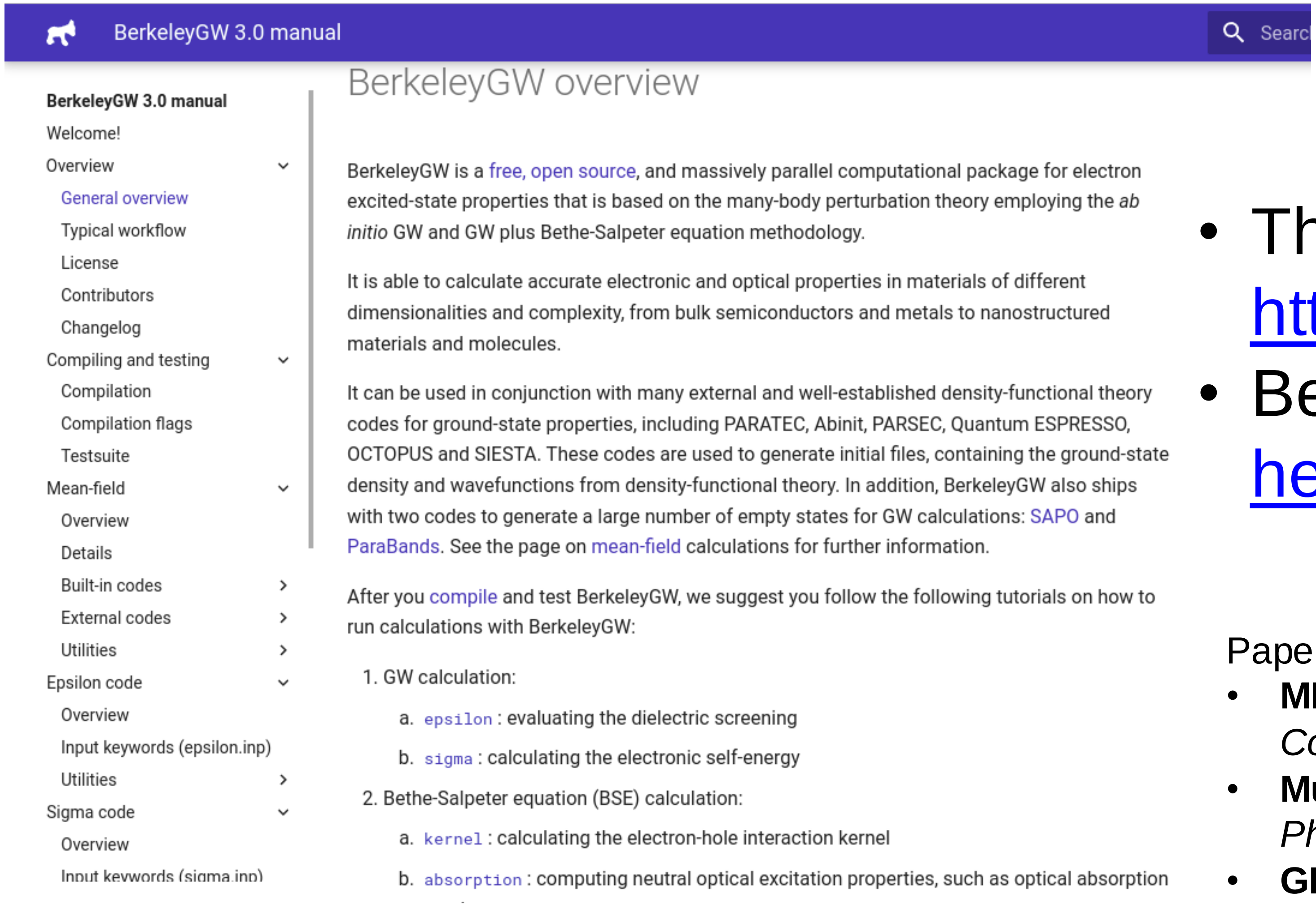
BerkeleyGW Summary

- 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 Useful Resources



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 GW 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 http://manual.berkeleygw.org/3.0/
- BerkeleyGW-Help mailing list: help@berkeleygw.org

Papers about implementation:

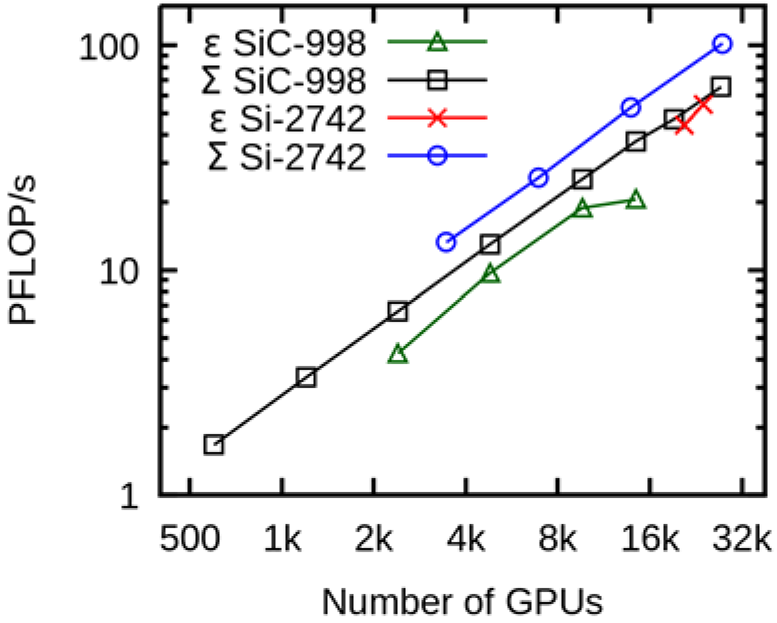
- **MPI and overview**: J Deslippe et al, *Computer Physics Communications* 183 (6), 1269-1289
- **Multi-core/OpenMP**: M Del Ben et al, *Computer Physics Communications* 235, 187-195
- **GPU**: M Del Ben et al, *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020, pp. 1-11

BerkeleyGW On the Path to Exascale

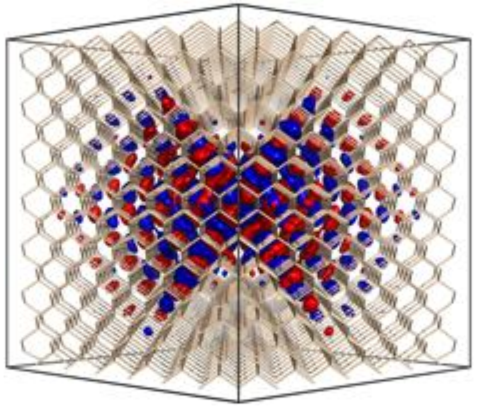
Foreseen exascale HPC systems will be GPU accellerated architectures

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



M. Del Ben, C. Yang, Z. Li, F. H. da Jornada, S. G. Louie and J. Deslippe, “Accelerating Large-Scale Excited-State GW Calculations on Leadership Class HPC Systems” in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, ser. SC ’20 No.4 pp.1 (2020), ACM Gordon-Bell Finalist



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.