Eigenproblems in MBPT and Practical Considerations

Jack Deslippe
NERSC

C2SEPEM
Who am I?

Jack Deslippe

- NERSC Application Performance Group Lead
- One of Lead Developers in BerkeleyGW project (C2SEPEM)
- US Exascale Computing Project Applications (ECP) Area Lead for Materials Sci. and Chemistry
What is GW

The “GW” method is an accurate approach to compute the “excited state” properties of materials through the Dyson equation ($\Sigma = iGW$). Examples:

- What happens when you add or remove an electron from a system
- How do electrons behave when you apply a voltage
- How does the system respond to light or x-rays

GW is complementary to the widely used density functional theory methods (DFT) which predict ground state properties of materials.
GW Reputation

\[ [E_{nk} - H_0(r) - V_H(r)] \psi_{nk}(r) - \int \Sigma(r, r', E_{n,k}) \psi_{nk}(r') dr' = 0 \]

The Good:
Quantitatively accurate for quasiparticle properties in a wide variety of systems.

Accurately describes dielectric screening important in excited state properties.

The Bad:
Prohibitively slow for large systems. Usually thought to cost orders of magnitude more time than DFT.

Memory intensive and scales badly. Exhausted by storage of the dielectric matrix and wavefunctions. Limited ~50 atoms.
GW Reputation

\[ [E_{nk} - H_0(r) - V_H(r)] \psi_{nk}(r) - \int \Sigma(r, r', E_{n,k}) \psi_{nk}(r') dr' = 0 \]

The Good:
Quantitatively accurate for quasiparticle properties in a wide variety of systems.
Accurately describes dielectric screening important in excited state properties.

The Bad:
Prohibitively slow for large systems. Usually thought to cost orders of magnitude more time than DFT.
Memory intensive and scales badly. Exhausted by storage of the dielectric matrix and wavefunctions. Limited ~50 atoms.
BerkeleyGW

A massively parallel highly optimized "GW" material science code that sits on top of DFT codes like Quantum ESPRESSO / PARATEC / PARSEC / SIESTA / Octopus / Abinit

Algorithms dominated by:
Dense Linear Algebra and Large Matrix/Tensor Reduction Loops, FFTs

SiC Diacancy Calculation. Run on All of Cori
GW Example : Computational Bottlenecks

A. Compute transition probabilities for electrons between orbitals (N^3) (FFT)

B. Sum (A) to form overall material response function (polarizability) (N^4) (ZGEMM)

C. Calculate interacting electron self energy in (N^4) (Hand Coded (GPP), ZGEMM (FF))

\[ \chi^0_{GG'}(\omega) = \sum_{n} \sum_{n'} [M^G_{nn'}]^* M^G'_{nn'} \frac{1}{E_n - E_{n'} - \omega} \]

\( M \) are complex double-precision arrays representing transition probabilities between orbitals
\( E_n \) are DFT orbital energies
Empty State Requirements

Lots of (empty) Orbitals Required. For meV accuracy 10%-100% necessary.

Typical DFT codes (QE, Abinit etc.) are not good at this! We find that when more than about 10% of eigenvalues are needed it is cheaper to use ScaLAPACK/ELPA to exactly diagonalize.

P. Zhang et al.
Approximations We’ve Employed

1. Assume high-energy orbitals are just plane-waves. Not good. Properties of dielectric depend sensitively on orthogonality of Eigenfunctions.

2. Form a new basis: exact low energy orbitals + orthogonalized PWs Is OK. Orthogonalization alone doesn’t quite give meV accuracy. And diagonalization in basis not much better than just diagonalizing H in PWs.
Smarter Ways to Generate the Orbitals

Spectrum Slicing - filtering spectrum to individual slices with polynomial filter.

Adds new level of parallelism on HPC systems.

W. Gao, K. Liou, C. Yang, J. Chelikowsky, Y. Saad
Other Eigen Problems

**Polarizability**  Construct low rank approximation at $\omega = 0$ to accelerate $\omega != 0$

\[
\chi_{GG'}^0(\omega) = \sum_{n}^{\text{occ}} \sum_{n'}^{\text{emp}} \left[ M_{nn'}^G \right]^* M_{nn'}^{G'} \frac{1}{E_n - E_{n'} - \omega}
\]


**Bethe-Salpeter Equation (BSE)**

Meiyue Shao, Chao Yang, Lin Lin Developed parallel solver for complex, non-Hermitian “Hamiltonian” - BSEPACK library

\[
H_{\text{BSE}} = \begin{bmatrix}
R & C \\
-\overline{C}^* & -\overline{R}^*
\end{bmatrix}
\]

Meiyue Shao, Felipe Homrich da Jornada, Chao Yang, Jack Deslippe, Steven G. Louie. Elsevier Linear Algebra and its Applications
Practical Use of ScaLAPACK + ELPA

<table>
<thead>
<tr>
<th>(nx,ny)</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>nb</td>
<td>1...100</td>
</tr>
<tr>
<td>N</td>
<td>20000</td>
</tr>
<tr>
<td>Architecture</td>
<td>KNL</td>
</tr>
<tr>
<td>Nodes</td>
<td>4</td>
</tr>
</tbody>
</table>
Parameter landscape - nb

<table>
<thead>
<tr>
<th>(nx, ny)</th>
<th>(10, 32)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>40000</td>
</tr>
<tr>
<td>Architecture</td>
<td>SKX</td>
</tr>
<tr>
<td>Nodes</td>
<td>8</td>
</tr>
</tbody>
</table>
OpenMP scalability

- OMP_NUM_THREADS * MPI_Ranks kept constant
- No OpenMP penalty would be a horizontal line
- ELPA is OK up to 4 threads
- Useful for e.g. Quantum Espresso Exact Exchange simulations

<table>
<thead>
<tr>
<th>(nx,ny)</th>
<th>best</th>
</tr>
</thead>
<tbody>
<tr>
<td>nb</td>
<td>best</td>
</tr>
<tr>
<td>Architecture</td>
<td>SKX</td>
</tr>
<tr>
<td>nodes</td>
<td>4</td>
</tr>
</tbody>
</table>
Scaling of ELPA and PZHEEVD

<table>
<thead>
<tr>
<th>(nx, ny)</th>
<th>best</th>
</tr>
</thead>
<tbody>
<tr>
<td>nb</td>
<td>best</td>
</tr>
<tr>
<td>Architecture</td>
<td>SKX</td>
</tr>
</tbody>
</table>

![Graph showing the scaling of ELPA and PZHEEVD]
ELPA profile

- 4 BDW Nodes, 20K Matrix
- obtained with Cray perftools
- MPI tradeoff
- nx=12 is best overall time
Processor evolution

solution
time on 4
nodes
Opportunity

Collaborate with HPC Facilities on Supporting Vendor Upcoming Roadmaps

NERSC - NESAP
OLCF - CAAR
ALCF - ESP
Extras
Why We Need GW

Many-body effects extremely important in **Excited-State properties** of Complex Materials.

Accurately describes properties important for:

- Photovoltaics
- LEDs
- Junctions / Interfaces
- Defect Energy Levels
- ....