Plane-wave DFT: bottlenecks and algorithms

Antoine Levitt

January 30, 2024

Summary

1. PWDFT: equations and discretization

2. Implementation of PWDFT

Kohn-Sham density functional theory for a molecule

For the ground state of $2N$ spin-paired electrons in a potential V_{nucl}:

$$\begin{cases} -\frac{1}{2}\Delta\phi_n + V_{\mathrm{nucl}}\phi_n + V_{\mathrm{HXC}}[\rho]\phi_n = \lambda_n\phi_n, \quad \langle\phi_n,\phi_m\rangle_{L^2(\mathbb{R}^3)} = \delta_{mn} \\ \rho(x) = 2\sum_{n=1}^{N}|\phi_n(x)|^2 \\ V_{\mathrm{HXC}}[\rho] = \frac{1}{|x|} * \rho + V_{\mathrm{XC}}[\rho] \end{cases}$$

- V_{XC} is an approximation of the true exchange-correlation potential; e.g. LDA: $(V_{\mathrm{XC}}[\rho])(x) = v_{\mathrm{xc}}(\rho(x))$
- N nonlinear coupled PDEs in dimension 3
- Aufbau principle: $\{\lambda_n\}_{n=1,\dots,N}$ are the lowest eigenvalues of

$$H_\rho = -\frac{1}{2}\Delta + V_{\mathrm{nucl}} + V_{\mathrm{HXC}}[\rho]$$

- Alternatively: minimize the energy $E(\phi)$ under $\langle\phi_n,\phi_m\rangle_{L^2(\mathbb{R}^3)} = \delta_{mn}$

From the KS equations to ABINIT

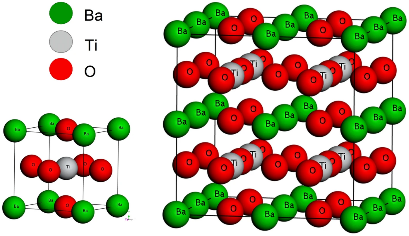
Roadmap:

- Truncate using a supercell (kpoints)
- Regularize the electron-nuclei Coulomb interaction with pseudopotentials
- Discretize in a plane-wave basis
- Set up the discrete equations as a fixed-point (SCF)
- Diagonalize the Hamiltonian iteratively
- Parallelize
- Do physics

The supercell method

Plot the EOS (energy-volume curve) of Silicon

- In theory, need to model whole crystal, with vacuum outside
- Surface effects not important: use a truncation method that preserves translational symmetry: the supercell



(a) (b)

- Solve the problem in the supercell, with periodic boundary conditions
- The (self-consistent) supercell Hamiltonian is (lattice) translation-invariant: Bloch theorem

$$\psi_{nk}(r) = e^{ikr} u_{nk}(r), \quad u_{nk} \text{ cell-periodic}$$

Supercell pbc ⇒ discretization of the Brillouin zone

The supercell Kohn-Sham equations

Make a $N_x \times N_y \times N_z$ supercell ⇒ sample the Brillouin zone on a $N_x \times N_y \times N_z$ grid

$$\begin{cases} -(-i\nabla + k)^2 u_{nk} + V_{\mathrm{nucl}} u_{nk} + V_{\mathrm{HXC}}[\rho] u_{nk} = \varepsilon_{nk} u_{nk}, \\ \langle u_{nk}, u_{mk} \rangle_{\text{unit cell}} = \delta_{mn} \\ \rho(x) = 2 \sum_{nk} |u_{nk}(x)|^2 \\ V_{\mathrm{HXC}}[\rho] = \text{"}\frac{1}{|x|} * \rho\text{"} + V_{\mathrm{XC}}[\rho] \end{cases}$$

for the u_{nk} on the unit cell with pbc.

Solve the supercell problem without ever computing on it!

Then, take the limit $N \to \infty$

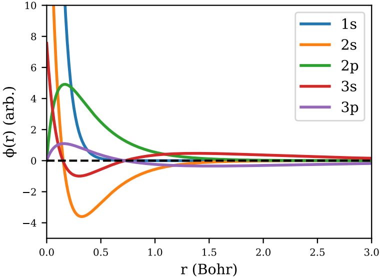
Discretization

$$-(-i\nabla + k)^2 u_{nk} + V_{\mathrm{nucl}} u_{nk} + V_{\mathrm{HXC}}[\rho] u_{nk} = \varepsilon_{nk} u_{nk}$$

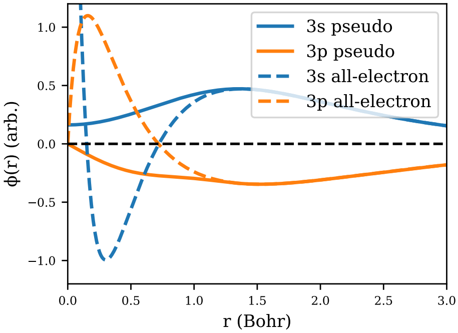
u_{nk} is periodic $\Rightarrow$ Fourier discretization natural

$$u_{nk}(r) = \sum_G c_G e^{iGr}$$

- The smoother u is, the faster c_G decays
- Several problems:
  1. Coulomb $\Rightarrow$ cusp at nuclei positions
  2. Core states strongly peaked
  3. Valence states strongly oscillatory



Pseudopotentials



- For an atom, replace $-Z/|r|$ by

  $$V_{\mathrm{loc}}(r) + \sum_{ij} D_{ij} |p_i\rangle\langle p_j|$$

  Choose $V_{\mathrm{loc}}, D_{ij}, p_i$ so that the valence eigenvalues and orbitals after some cutoff r_{cut} match the exact ones.
- For a molecule or solid, just add them and hope for the best.
- More sophisticated technologies: US, PAW

Plane-wave discretization

$$-(-i\nabla + k)^2 u_{nk} + V_{\mathrm{pseudo}} u_{nk} + V_{\mathrm{HXC}}[\rho] u_{nk} = \varepsilon_{nk} u_{nk}$$
$$u_{nk}(r) = \sum_G c_G e^{iGr}$$

How to truncate G vectors? Kinetic energy cutoff

$$\frac{1}{2}|k+G|^2 \leq E_{\mathrm{cut}}$$

Galerkin method: require that

$$\forall G \text{ s.t. } \frac{1}{2}|k+G|^2 \leq E_{\mathrm{cut}},$$
$$\langle e^{iGr}, H_k u_{nk} \rangle = \varepsilon_{nk} \langle e^{iGr}, u_{nk} \rangle$$

Plane waves are orthogonal: yields an $N_{\mathrm{pw}} \times N_{\mathrm{pw}}$ algebraic system

$$\sum_{G'} H_{k,GG'} c_{G'} = \varepsilon_{nk} c_G$$

The Hamiltonian

$$H = \frac{1}{2}(-i\nabla + k)^2 + V_{\mathrm{loc,HXC}} + \sum_{mn} D_{mn} |p_m\rangle\langle p_n|$$

$$\begin{aligned} H_{GG'} &= \frac{1}{\Omega}\langle e^{iGr}, H_k e^{iG'r}\rangle \\ &= \frac{1}{2}|G+k|^2 \delta_{GG'} + \widehat{V}_{\mathrm{loc,HXC}}(G'-G) + \sum_{mn} D_{mn} \widehat{p}_m(G)\overline{\widehat{p}_n(G')} \end{aligned}$$

Cell integrals turn into whole-space Fourier transforms of localized quantities:

$$V_{\mathrm{loc}}(r) = \sum_{a \in \mathrm{ats}, R \in \mathrm{latt}} v_{\mathrm{loc,a}}(r - a - R)$$

$$\int_{\mathrm{cell}} V_{\mathrm{loc}}(r) e^{-i(G'-G)r} dr = \sum_a e^{-i(G'-G)a} \int_{\mathbb{R}^3} v_{\mathrm{loc,a}}(r) e^{-i(G'-G)r} dr$$

Forming the Hamiltonian

- V_{loc} and V_{nonloc} are built from radial pseudopotential information (.upf, .pspnc...)
- The density

  $$\rho(r) = 2 \sum_{nk} |u_{nk}(r)|^2$$

  is computed on a real-space grid
- The exchange-correlation potential

  $$V_{\mathrm{XC}}(r) = V_{\mathrm{LDA}}(\rho(r))$$

  also

  (GGA: same principle. Hybrids: much more expensive!)
- The Hartree potential

  $$-\Delta V_{\mathrm{H}} = 4\pi\rho$$

  is computed in Fourier space
- All of these steps are not very costly

Self-consistent equations

$$H[\rho]u_{nk} = \varepsilon_{nk} u_{nk}$$

$$\rho(r) = 2\sum_{nk} |u_{nk}(r)|^2$$

- Nonlinear problem: get iterative solution
- Simple SCF:

$$H[\rho^i]u_{nk}^{i+1} = \varepsilon_{nk}^{i+1} u_{nk}^{i+1}$$

$$\rho^{i+1}(r) = 2\sum_{nk} |u_{nk}^{i+1}(r)|^2$$

- Usually does not converge!
- Simple mixing:

$$\rho^{i+1} = \rho^i + \alpha \left(2\sum_{nk} |u_{nk}^{i+1}(r)|^2 - \rho^i \right)$$

  converges for $\alpha > 0$ small enough, but slowly
- State of the art: advanced mixing (dependent on $\rho^{i-1}, \rho^{i-2}, \dots$) and preconditioning (strategic damping of some wave vectors)

Diagonalization of the Hamiltonian

The SCF loop

1. Initialize system, precompute things
2. Diagonalize the Hamiltonian
3. Update the Hamiltonian
4. If not converged, go back to (2)
5. Postprocess

$$Hu_{nk} = \varepsilon_{nk} u_{nk}$$

Characteristics:

- Need to solve for $N_{\mathrm{el}} \ll N_{\mathrm{pw}}$ states ($N_{\mathrm{el}} \approx 1\%$ of N_{pw})
- Matrix dense but efficient matvecs available
- Need smallest eigenvalues
- Scale separation: characteristic gap $\approx 1eV$, largest eigenvalue (spectral width) tens of Hartree $\Rightarrow$ conditioning issues
- Usually have good initial guess

A prototype diagonalization algorithm: subspace iteration

Goal: partially diagonalize a given Hamiltonian H
Input: set of vectors $X \in \mathbb{C}^{N_{\mathrm{pw}} \times N_{\mathrm{el}}}$

1. Form the block matvec $Y = HX$
2. Find an orthonormal basis Z of the range of (X, Y)
3. Solve for the N_{pw} lowest eigenvalues of $(Z^* HZ)c = \varepsilon c$
4. Obtain the new $X = Zc$

The Hamiltonian

$$
\begin{aligned}
H &= \frac{1}{2}(-i\nabla + k)^2 + V_{\mathrm{loc,HXC}} + \sum_{mn} D_{mn} |p_m\rangle\langle p_n| \\
H_{GG'} &= \frac{1}{\Omega}\langle e^{iGr}, H_k e^{iG'r}\rangle \\
&= \frac{1}{2}|G+k|^2 \delta_{GG'} + \widehat{V}_{\mathrm{loc,HXC}}(G'-G) + \sum_{mn} D_{mn} \widehat{p}_m(G)\overline{\widehat{p}_n(G')}
\end{aligned}
$$

Dense matrix, but very structured

1. Kinetic energy: diagonal
2. Local potential: convolution
3. Nonlocal potential: separable (low-rank)

Convolutions and the FFT

To perform a matrix-vector product, need

$$\sum_{G'} \widehat{V}_{\mathrm{loc,HXC}}(G' - G) c_{G'}$$

- Discrete convolutions can almost be performed with an FFT:

Convolution theorem

If f and g are in $\mathbb{C}^N$ and

$$(f * g)_i = \sum_{j=1}^{N} f_{i-j} g_j$$

with the implicit periodization convention that $f_{i+N} = f_i$, then

$$f * g = F^{-1}(F(f)F(g)).$$

- FFTs imply implicit periodization ⇒ zero-padding
- Reason why $N_{\mathrm{pw}} \neq N_{\mathrm{FFT}}$

How to apply the Hamiltonian

$$H_{GG'} = \frac{1}{2}|G+k|^2\delta_{GG'} + \widehat{V}_{\mathrm{loc,HXC}}(G'-G) + \sum_{mn} D_{mn}\widehat{p}_m(G)\overline{\widehat{p}_n(G')}$$

- Matrix-vector products can be performed by simple scaling, FFT and low-rank multiplication
- Operation cost for one matvec:

$$O(N_{\mathrm{pw}} + N_{\mathrm{FFT}}\log(N_{\mathrm{FFT}}) + N_{\mathrm{proj}}N_{\mathrm{pw}})$$

- Orders of magnitudes (taken from Silicon Ecut 30Ha):
  1. $N_{\mathrm{el}} \approx 5N_{\mathrm{at}}$
  2. $N_{\mathrm{proj}} \approx 5N_{\mathrm{at}}$
  3. $N_{\mathrm{pw}} \approx 200N_{\mathrm{at}}$
  4. $N_{\mathrm{FFT}} \approx 5000N_{\mathrm{at}}$
- Asymptotically, the nonlocal operator dominates (but only for very big systems)

Computational primitives

Need

- FFT
- Linear algebra (matrix-matrix multiply, cholesky, eigenvalue decomposition...)

Kernels

- Standard (FFTW, MKL...)
- Can use threads
- Exist on GPU

Parallelism

Dominant paradigm: MPI+X

- MPI to distribute data between different nodes
  - Communication by collective (MPI_SUM, MPI_ALL_TO_ALL, etc) or point-to-point (MPI_SEND, etc)
- $X =$ threads, GPU, to exploit different cores inside a node. Data shared between cores.
  - a

MPI is developer's job. X is mostly the job of libraries: developer handles communication with device and maybe some computational kernels.

Parallelism in PWDFT

The main data structure: u_{Gnk}, distributed with MPI (npfft, npband, np_spkpt)

Operations

- Parallelism over k mostly trivial (small and infrequent reductions)
- The Hamiltonian requires no communication over n, but communication over G, parallelized somewhat (MPI FFTW, MPI matmul)
- Linear algebra operations need communication over both G and (especially) n, parallelized somewhat (Scalapack)

Parallelization limited: ideally, minimize communication between vectors

- In ABINIT, two main data distributions: split G and n for Hamiltonian application, and split only G for linear algebra
- Switch with MPI_ALL_TO_ALL

Summary of operations

Operation	Cost	G Parallelism	n parallelism
H: FFTs	$N_{\mathrm{el}} N_{\mathrm{FFT}} \log N_{\mathrm{FFT}}$	Moderate	Perfect
H: nonlocal	$N_{\mathrm{el}} N_{\mathrm{pw}} N_{\mathrm{proj}}$	Good	Perfect
Orthogonalization	$N_{\mathrm{el}}^2 N_{\mathrm{pw}}$	Good	Bad
Rayleigh-Ritz	N_{el}^3	No	Moderate
ALL_TO_ALL	$N_{\mathrm{el}} N_{\mathrm{pw}}$	Bad	Bad