

Development of a SIMPLE plane-wave basis pseudopotential code

Summer, 2009



Byoungseon Jeon

Dept. Applied Science

University of California, Davis

* Updated June 29, 2009



Contents

- Day 1
 - Review of UNIX and C/C++
 - Electronic structure analysis and DFT
 - Reciprocal space and plane waves
 - FFT
- Day 2
 - Plane waves and Hamiltonian components
- Day 3
 - Mixing
 - Self-consistent loop
 - Eigen-solver

Pre-requisite

- Unix command and C/C++ programming
- Density Functional Theory
- Background of quantum chemistry or solid state physics
- Experience of plane-wave pseudopotential code package
 - Abinit, vasp, qbox, quantum espresso ...
- Use of numerical library
 - GSL, LAPACK, FFTW, ffe ...

Objective of the study

- Building the simplest plane-wave pseudopotential code
 - Analysis of a single hydrogen
- Figuring out how the BLACK BOX code is actually working
- Understanding the basic nature of electronic structure analysis



Day 1

... admitted that the Unix operating system and C programming language created by them is an elaborate prank kept alive for over 20 years... At one time, we joked about selling this to the Soviets to set their computer science progress back 20 or more years. Unfortunately, AT&T and other US corporations actually began using Unix and C...

From: Creators admit Unix, C, hoax



Contents

- Review of Unix and C/C++ programming
- Electronic structure Analysis
 - Density Functional Theory
 - Born-Oppenheimer Approximation
 - Plane wave basis
- Pseudopotential
- Configuration of plane waves/grids
- Reciprocal space
- FFT
 - Install FFTW library at a local directory



Unix and C/C++

- Unix/Linux command
 - Editor: vi, (x)emacs
 - How to implement Makefile
- C/C++
 - Compiler: gcc/g++
 - How to use library
 - Procedural programming



Electronic Structure Analysis

- Ground state of materials
- Dependent on the states of electrons
 - Mostly, valence electrons
- Hartree/Hartree-Fock
 - Classical approach/LCAO
- Density Functional Theory (DFT)
 - Favored by the computational efficiency

Density Functional Theory

- Breakthrough of quantum theory in many physical/chemical applications
- Walter Kohn: Nobel Laureate '98
- Many electrons \Rightarrow single electron density
- Mean-field theory
- Trade-off

Variations of DFT

- Basis
 - LCAO, PW, APW, LAPW, fully #
- Exchange-correlation
 - LDA, LSDA, GGA, PBE
- Electron-ion interaction
 - All-electron full, all-electron muffin-tin, pseudopotential, jellium

Variations of DFT

- Basis
 - LCAO, PW, APW, LAPW, fully #
- Exchange-correlation
 - LDA, LSDA, GGA, PBE
- Electron-ion interaction
 - All-electron full, all-electron muffin-tin, pseudopotential, jellium

Born-Oppenheimer approx.

- Mass of ion \gg mass of electron
 - Simplify the problem
- Adiabatic approximation
- Decouple the interactions
 - Electron-electron, electron-ion, and ion-ion
 - Build each Hamiltonian



Hamiltonian

- Variations of total energy by electron probability density
- Hartree
 - Electron-electron interaction
 - NOT Hartree/Hartree-fock of LCAO
- Pseudopotential
 - electron-ion interaction
- Exchange-correlation
 - Exchange and ? of electron-electron interaction
- Kinetic energy of the electrons

Hartree

- Electron-electron interaction
 - Void of exchange components compared to Hartree-Fock of LCAO

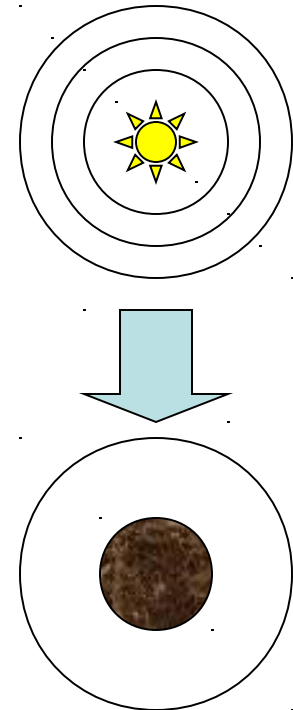
$$V_{HT} = \iint \frac{zz'}{|r - r'|} dr dr'$$

- Poisson equation in reciprocal space
 - Easy to solve

$$V_{HT}(G) = \frac{4\pi\rho(G)}{|G|^2}$$

Pseudopotential

- Pseudize core electrons + ion charge
- Not REAL potential
- Mostly works for valence electron interaction
- GTH functional form
 - Real/reciprocal space



$$V(G) = A \frac{e^{-(Gr_{loc})^2/2}}{G^2} + Be^{-(Gr_{loc})^2/2} (C_1 + C_2(3 - (Gr_{loc})^2))$$

Exchange-correlation

- Hartree component in DFT doesn't have exchange terms
- Fitting quantum Monte-Carlo results using Pade approximants
 - Parameterized by electron probability density
- Approach
 - Linear Density Approximation (LDA)
 - Generalized Gradient Approximation (GGA)
- GTH pp. provides LDA forms.

Why reciprocal space?

- Natural description of periodic system
 - Real space: actual wave function/probability density
 - Reciprocal space: magnitude and phase in terms of plane wave length
 - Coupled by FFT for numerical simulation
- Provides better solution for certain terms
 - Hartree
- Better choice for electron density calculations by FFT
 - $N_2 \rightarrow N \log N$ for computing cost



Calculations of electron probability density

- Brute-force

$$\rho = |\phi\rangle\langle\phi|$$

- N^2 calculation load

- FFT b/w reciprocal and real space

$$\begin{aligned}\rho(r) &= \langle r | \phi \rangle \langle \phi | r \rangle \\ &= |\langle r | \phi \rangle|^2\end{aligned}$$

- Order of N for real space calculation but the cost for FFT is $N\log N$



Why plane-waves?

- Natural to describe a periodic system
- Can be coupled with FFT/reciprocal space
- But many plane-waves are required to simulate actual wave-functions

Fourier Transform

- Transformation between real and reciprocal space
 - Formula?
 - Discrete Fourier Transform $\sim N^2$
 - Fast Fourier Transform $\sim N \log N$
 - What about 3D transform ? What if non-isotropic grids?
- We use external libraries
 - FFTW

FFT library

- FFTW

- Install at a local directory: /home/ngj/FFTW
 - Unpack source files at a temporary directory
 - # ./configure --prefix=/home/ngj/FFTW
 - # make
 - # make install

- FFTW data declaration

```
fftw_complex *ck, *cr, *backup;  
ck = new fftw_complex [Ntotal];  
cr = new fftw_complex [Ntotal];  
/* ... */
```

- FFTW execution

```
fftw_plan plan;  
plan = fftw_plan_dft_3d(Nx, Ny, Nz, ck, cr, FFTW_BACKWARD, FFTW_ESTIMATE);  
fftw_execute(plan);  
fftw_destroy_plan(plan);
```

FFTW library example

- Test example
 - At practice_1
 - Implement Forward transform routine (cr->ck)
 - Check the difference of cr[] and backup[]
 - Why there are differences? Backward+forward operations do not return to what they were?
- Normalization
 - Depends on the library

FFTW library example (cont.)

- Implement time checking routine
 - time() function from <ctime> or <time.h>
 - Returns the accumulated **seconds** since 1970
 - This method is not useful when $\Delta t < 1$ sec
- Extension of the simple test problem
 - Optimized grids?
 - Disable printing and check the computing resource for:

Grid size	127^3	128^3	129^3
Time(sec)			



Review of today

- Basic of electronic structure analysis
- Density Functional Theory
- Plane wave basis pseudopotential method
- Reciprocal space
- FFT



Day 2

Contents

- A single hydrogen analysis code
 - Configuration of plane waves and grids
 - How to handle wave-functions and density
 - Building Hamiltonian components
 - Fill out the codes

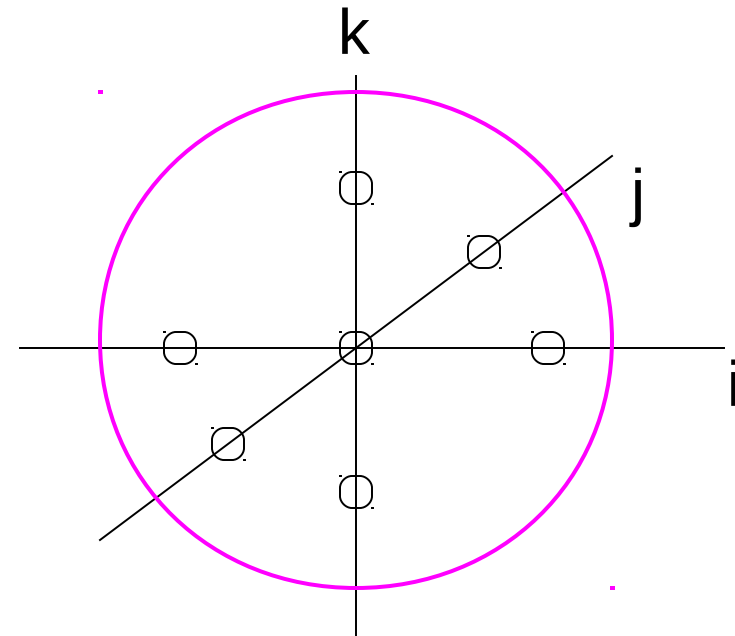
A single Hydrogen

- The most primitive example
 - Non-physical condition
 - Only numerical example
- A single hydrogen at 5x5x5 cubic periodic box
- Seven plane waves
- Gamma point integration only
- No non-local pseudo-potential
- Usage of FFT/eigen-solver
- A single electron with a single band

How many plane waves?

- Cut-off energy
- Seven plane-waves

$$\left. \begin{array}{l} - (0,0,0) \\ - (\pm 1,0,0) \\ - (0,\pm 1,0) \\ - (0,0,\pm 1) \end{array} \right\} \times \frac{2\pi}{l}$$



@reciprocal space

```
int n;  
int pw[Npw][3] = { [redacted], {1,0,0} };  
int Npw3 = Npw*Npw*Npw;
```

*Practice_2

- Fill out the pw[][] components at main.cc

How many grids?

- $2*N_{\max}+1 = 5$

- why?

- Density operation

$$\rho = |\phi\rangle\langle\phi| = \exp(-Gr)\exp(G'r)$$

- $[-G-G', G+G']$

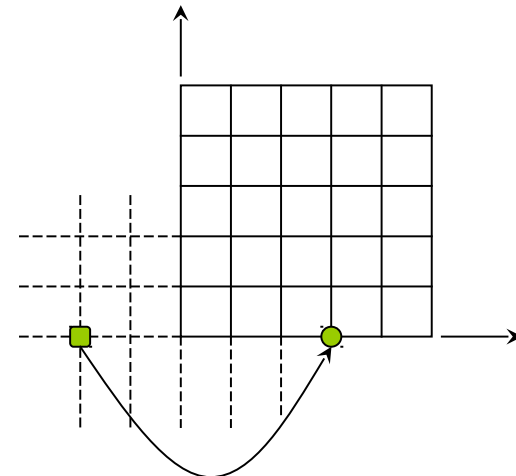
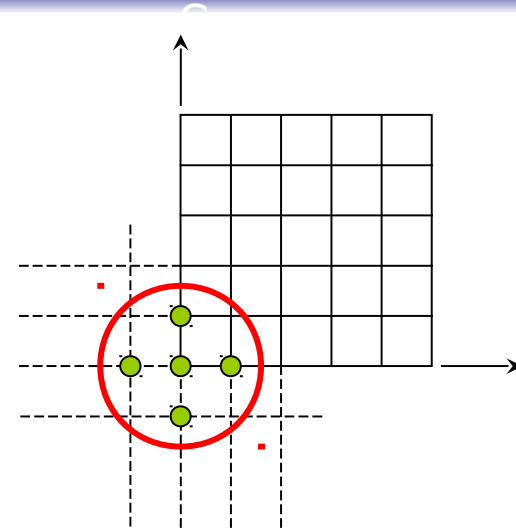
- $-1-1=-2$

- $1+1=2$

- Periodic condition

- $-2 \Rightarrow N_{\text{grid}}-2$

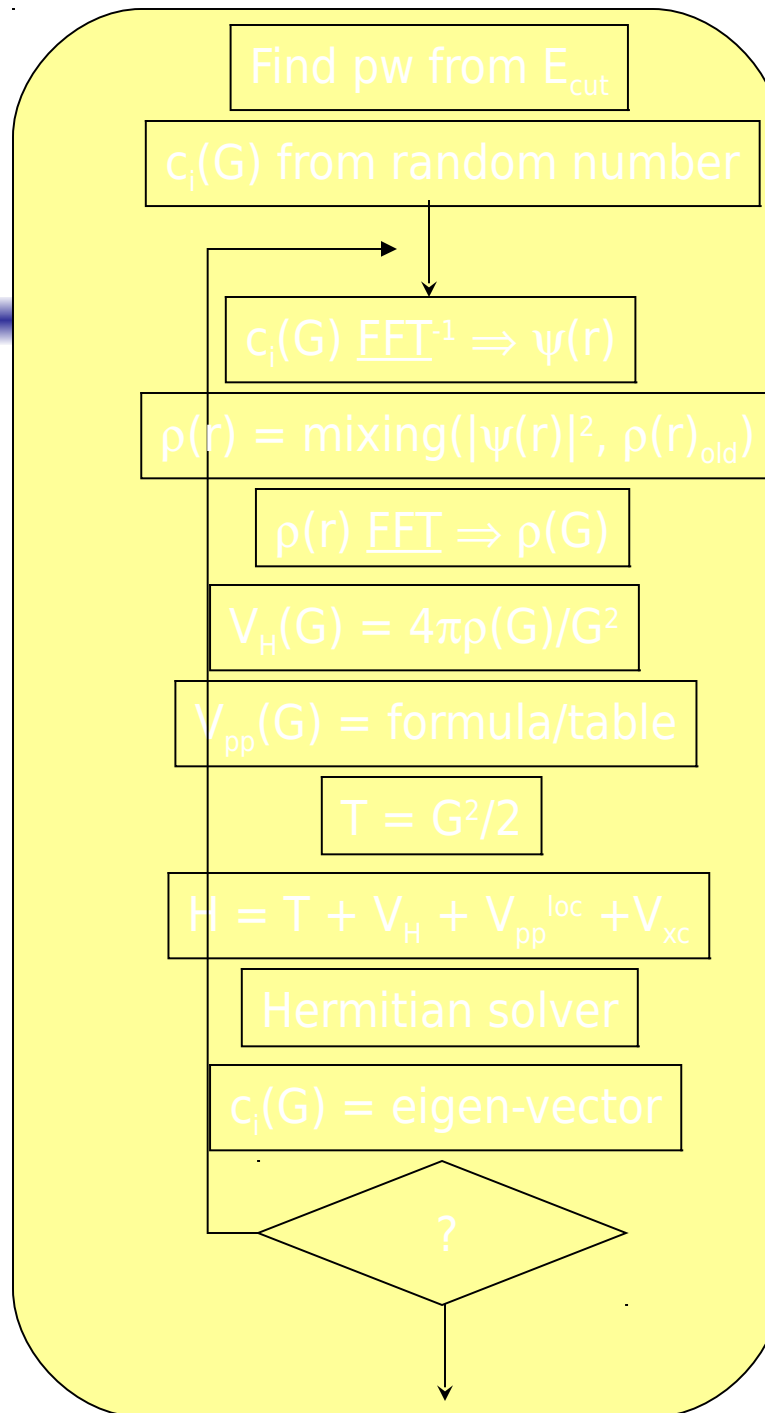
```
#define Ngr 5
```



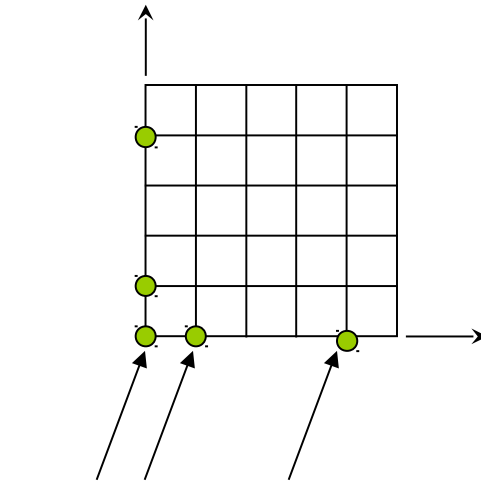
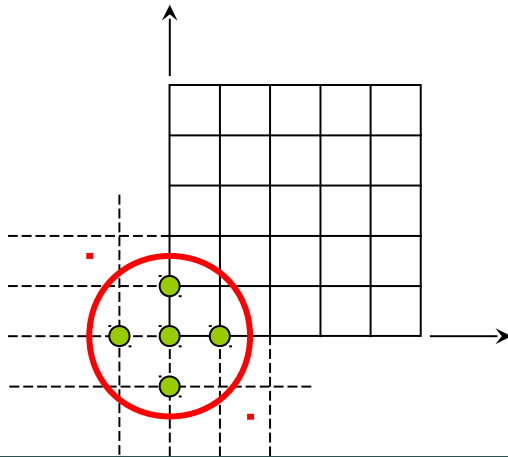
Self-consistent-loop

```
ck_initialize(pw, ck, nr, nk)

n = 0;
do {
    ck_fft_cr(ck, cr, l_box);
    nr_fft_nk(cr, nr, nk, alpha
    V_HT      (pw, nk, Hht, Eht
    V_loc      (pw, nk, Hloc, Eld
    V_xc      (    nr, Vxc, Exc
    H_kin      (pw, ck, Hkin, Eki
    Hermitian_solv(pw, Hht, Hld
    n ++;
} while (n<100);
```



How to initialize plane waves?



random numbers while the other grids are zero

```
//
// Initialization and normalization
x = 0.0;
for (i=0;i<Npw;i++) {
    kx = (pw[i][0] + Ngr)%Ngr;
    ky = (pw[i][1] + Ngr)%Ngr;
    kz = (pw[i][2] + Ngr)%Ngr;
    id = kz + Ngr*(ky + Ngr*kx);
    ck[id][0] = rand_double();
    ck[id][1] = rand_double();
    x += ck[id][0]*ck[id][0] + ck[id][1]*ck[id][1];
}
x = sqrt(x);
for (i=0;i<Ngr3;i++) {
    ck[i][0] = 
    ck[i][1] = 
}
```

*constraints of plane waves

1. normalized
2. complex number

*Practice_3

- Implement normalization routine for ck[]

Plane waves

- We have initialized wave functions, or the coefficients of plane-waves at reciprocal space
- We transform the coefficients into real space, using FFT
- The transformed value is the wave-function at real space

```
int i;  
int Ngr3 = Ngr*Ngr*Ngr;  
fftw_plan plan;  
:  
:  
:  
double omega = pow(l_box, 3.);  
  
for (i=0;i<Ngr3;i++) {  
    cr[i][0] = cr[i][0]/sqrt(omega);  
    cr[i][1] = cr[i][1]/sqrt(omega);  
}
```

*Practice_3

- At ck_fft_cr routine of density.cc, implement ck->cr transform

Electron (Probability) Density

- Density at rec. space $\rho = |\phi\rangle\langle\phi|$
- Density at real space $\rho(r) = \langle r|\phi\rangle\langle\phi|r\rangle$
 - Square of the plane waves at real space
- But we need mixing
 - Schrödinger equation is implicit
 - Linear mixing (mostly we don't use but this is an example)

```
for(i=0;i<Ngr3;i++) {
    rho[i] = 
}

for (i=0;i<Ngr3;i++) {
    nr[i][0] = (1. - alpha)*nr[i][0] + alpha*rho[i];
}
// ...
```

*Practice_3

- Fill out the routine of nr_fft_nk() at density.cc



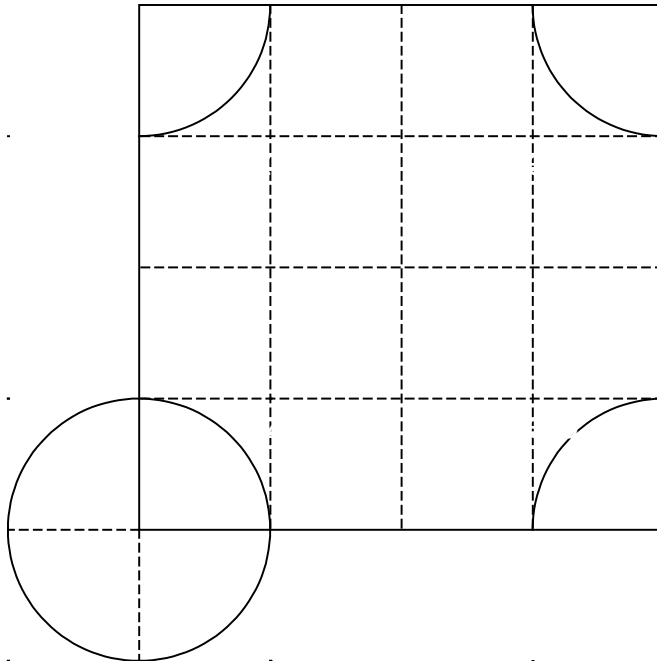
Electron Probability Density at Reciprocal Space

- We need electron probability density at reciprocal space
 - To calculate Hartree potential/energy and pseudopotential energy
- FFT again

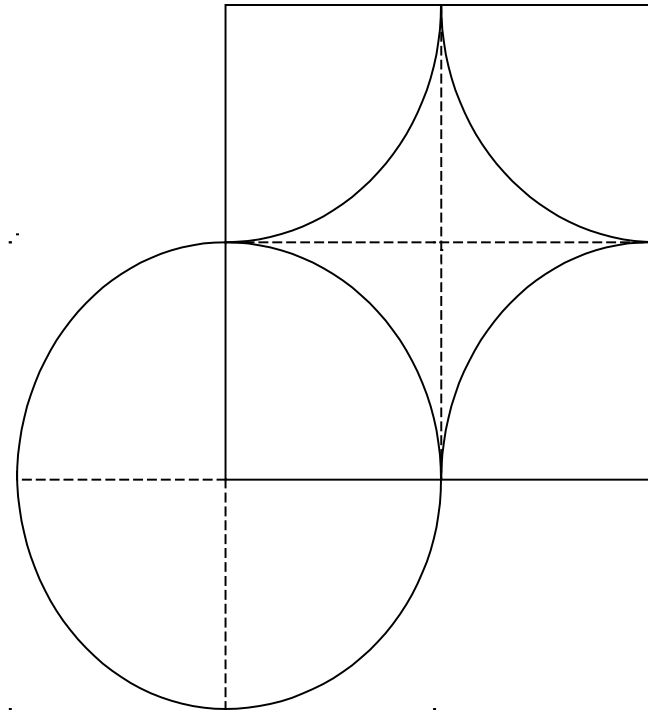
```
plan = fftw_plan_dft_3d(Ngr, Ngr, Ngr, nr, nk, FFTW_FORWARD, FFTW_ESTIMATE);  
fftw_execute(plan); fftw_destroy_plan(plan);  
for (i=0;i<Ngr3;i++) {  
    nk[i][0] = nk[i][0]/(double) Ngr3;  
    nk[i][1] = nk[i][1]/(double) Ngr3;  
}
```

Plane-wave coefficients vs. electron probability density

ck - space



nk - space



Q: Then what about nr -space ?

Hartree Potential

- Hamiltonian matrix = $N_{pw} \times N_{pw}$
- (i,j) component of the matrix = the coupling of the i -plane wave to j -plane wave

$$V_{HT}(G - G') = \frac{4\pi\rho(G - G')}{|G - G'|^2}$$

*Practice_4

- Implement Hamiltonian component and the corresponding energy

```
for (i=0;i<Npw;i++) {
    ix = pw[i][0]; iy = pw[i][1]; iz = pw[i][2];
    for (j=0;j<Npw;j++) {
        jx = pw[j][0]; jy = pw[j][1]; jz = pw[j][2];
        kx = ix - jx; ky = iy - jy; kz = iz - jz;
        if (i != j) {
            G2 = FOURPI SQ * (double) (kx*kx + ky*ky + kz*kz) / (l_box*l_box);
            kx = (kx + Ngr)%Ngr; ky = (ky + Ngr)%Ngr; kz = (kz + Ngr)%Ngr;
            id = kz + Ngr*(ky + Ngr*kx);
            Hht[i][j][0] = XXXXXXXXXXXX
            Hht[i][j][1] = XXXXXXXXXXXX
            Eht += XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        }
        else {
            Hht[i][j][0] = Hht[i][j][1] = 0.0;
        }
    }
}
```

Hatree Hamiltonian matrix

$$\begin{pmatrix}
 0 & A1 & B1 & C1 & D1 & E1 & F1 \\
 A1^* & 0 & A2 & B2 & C2 & D2 & E2 \\
 B1^* & A2^* & 0 & A3 & B3 & C3 & D3 \\
 C1^* & B2^* & A3^* & 0 & A4 & B4 & C4 \\
 D1^* & C2^* & B3^* & A4^* & 0 & A5 & B5 \\
 E1^* & D2^* & C3^* & B4^* & A5^* & 0 & A6 \\
 F1^* & E2^* & D3^* & C4^* & B5^* & A6^* & 0
 \end{pmatrix}$$

Pseudopotential

- Not pseudized for hydrogen (no core electron) but practical for the implementation
- GTH/HGH functional form

$$V_{loc} = -4\pi \frac{Z_{ion}}{\Omega} \frac{e^{-(Gr_{loc})^2/2}}{G^2} + (2\pi)^{3/2} \frac{r_{loc}^3}{\Omega} e^{-(Gr_{loc})^2/2} (-4.00663326 + 0.6778322(3 - (Gr_{loc})^2))$$

*Practice_4

- Implement
Hamiltonian
component

```
for (i=0;i<Npw;i++) {
  ix = pw[i][0]; iy = pw[i][1]; iz = pw[i][2];
  for (j=0;j<Npw;j++) {
    jx = pw[j][0]; jy = pw[j][1]; jz = pw[j][2];
    kx = ix - jx; ky = iy - jy; kz = iz - jz;
    if (i != j) {
      G2 = FOURPIQ * (double) (kx*kx + ky*ky + kz*kz) / (l_box*l_box);
      kx = (kx + Ngr)%Ngr; ky = (ky + Ngr)%Ngr; kz = (kz + Ngr)%Ngr;
      id = kz + Ngr*(ky + Ngr*kx);
      A = exp(-G2*rs*rs*0.5);
      Hloc[i][j][0] = (c1 + c2*(3. - (G2*rs*rs*0.5))) * A;
      Hloc[i][j][1] = 0.0;
      Eloc += 0.5*omega*(Hloc[i][j][0]*nk[id][0] + Hloc[i][j][1]*nk[id][1]);
    }
    else {
      Hloc[i][j][0] = Hloc[i][j][1] = 0.0;
    }
  }
}
```

Kinetic term

- Kinetic energy of electrons
- Only diagonal terms exist

$$T(G) = \frac{1}{2} G^2$$

*Practice_4

- Implement
Hamiltonian
component

```
Ekin = 0.0;
for (i=0;i<Npw;i++) {
    ix = pw[i][0]; iy = pw[i][1]; iz = pw[i][2];
    i2 = ix*ix + iy*iy + iz*iz;
    ix = (ix+Ngr)%Ngr; iy = (iy+Ngr)%Ngr; iz = (iz+Ngr)%Ngr;
    id = iz + Ngr*(iy + Ngr*ix);
    G2 = FOURPISQ*(double) i2 / (l_box*l_box);
    Hkin[i] = G2;
    Ekin += Hkin[i]*(ck[id][0]*ck[id][0] + ck[id][1]*ck[id][1]);
}
```

Exchange-Correlation

- Fitting to quantum Monte-Carlo data
- How to fit?
 - LDA: local density
 - GGA: density + density gradient
 - With or without spin polarization
- GTH paper provides the fitting function
 - Electron probability density
 - Wigner Seitz radius
 - Hamiltonian component
- Calculated → real space electron density
 - FFTed for reciprocal space Hamiltonian

Local Density Approximation

$$n = 1/V = 1/(4\pi r_s^3/3)$$

$$\varepsilon_{xc}(r) = -\frac{a_0 + a_1 r_s + a_2 r_s^2 + a_3 r_s^3}{b_1 r_s + b_2 r_s^2 + b_3 r_s^3 + b_4 r_s^4}$$

$$V_{xc}(r) = \varepsilon_{xc}(r) + n(r) \frac{\delta \varepsilon_{xc}(r)}{\delta n(r)}$$

*Practice_4

- Implement Wigner-Seitz radius calculation

```
for (i=0;i<Ngr3;i++) {
    rs = XXXXXXXXXX;
    asum = a[0] + a[1]*rs + a[2]*rs*rs + a[3]*rs*rs*rs;
    aasum = a[1] + 2.*a[2]*rs + 3.*a[3]*rs*rs;
    bsum = b[0]*rs + b[1]*rs*rs + b[2]*rs*rs*rs + b[3]*pow(rs,4,0);
    bbsum = b[0] + 2.*b[1]*rs + 3.*b[2]*rs*rs + 4.*b[3]*rs*rs*rs;
    exc = -asum/bsum;
    ndedn = rs*(aasum*bsum - asum*bbsum)/bsum/bsum/3.;
    Wxc[i][0] = exc + ndedn; Wxc[i][1] = 0.0;
    Exc += exc*nr[i][0]*dv;
}
fftw_plan plan;
plan = fftw_plan_dft_3d(Ngr,Ngr,Ngr, Wxc, Vxc, FFTW_FORWARD, FFTW_ESTIMATE);
fftw_execute(plan); fftw_destroy_plan(plan);

for (i=0;i<Ngr3;i++) {
    Vxc[i][0] = Vxc[i][0]/(double) Ngr3;
    Vxc[i][1] = Vxc[i][1]/(double) Ngr3;
}
```

Day 3

Q: What can one do when convergence is bad?

A: Pray

From: Hands on Session IV, VASP



Contents

- Hamiltonian matrix
 - Eigen-solver
- Mixing and self-consistent loop
- Analysis of the result
 - Comparison to abinit

Hamiltonian Matrix

- Add-up of all the Hamiltonian components
- (ij) component corresponds to the effect of plane-wave i to plane-wave j
- Hermitian matrix
 - Real eigen-value
 - Complex eigen-vector

*Practice_5

- Fill out the routine

```

for (i=0;i<Npw;i++) {
    ix = pw[i][0]; iy = pw[i][1]; iz = pw[i][2];
    for (j=0;j<Npw;j++) {
        jx = pw[j][0]; jy = pw[j][1]; jz = pw[j][2];
        kx = ix - jx; kx = (kx+Ngr)%Ngr;
        ky = iy - jy; ky = (ky+Ngr)%Ngr;
        kz = iz - jz; kz = (kz+Ngr)%Ngr;
        id = kz + Ngr*(ky + Ngr*kx);

        a[i+Npw*j].real = 
        a[i+Npw*j].imag = 
        if (i == j) a[i+Npw*j].real += Hkin[i];
    }
}
    
```

Eigen-solver

- LAPACK
 - ZHEEV, ZHEEVX, ...
 - OSX has built-in library of LAPACK
- GSL
 - gsl_eig_hermv()
- For 7×7 matrix
 - 7 eigen value (band energy)
 - 7×7 matrix for eigen-vector or 7 eigen-vectors which have 7 elements

Eigen solution

- E-values/vectors of the Hamiltonian matrix are required
- Bad News!!!
 - You have to solve Big ($N_{pw} \times N_{pw}$) matrix
- Good News!!!
 - You don't need to solve all of them
 - Lowest and its neighboring (bands) eigen values/vectors

Mixing

- Why?
 - Nature of Schrödinger equation

$$H\psi = E\psi$$

- To find the electron density, we need to know electron density ... ?
 - Chicken or the egg
 - Gradual upgrade of electron density from a scratch

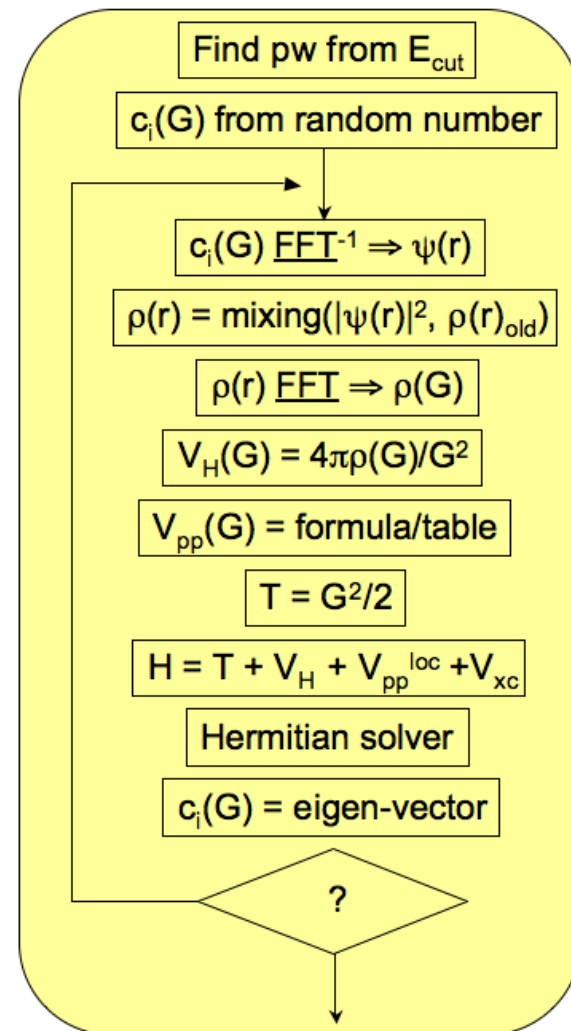
- Kinds of m

```
for (i=0;i<Ngr3;i++) {  
    nr[i][0] = (1. - alpha)*nr[i][0] + alpha*rho[i];  
}
```

- Linear, Broyden, Kerker, ...

Self-consistent loop

- How long do we repeat?
 - Certain criterion required
 - We just do 100 loops
 - Check how the total/band energy change
- For the convergence
 - Better mixing required
 - Approximation like smearing might be employed



Before going forward ...

- Now we solved 7×7 matrix
 - We began with one set (one band) of plane waves
 - 7 eigen-vectors which have 7 elements
 - We have to replace the plane waves with eigen-vectors, for the SCF loops
 - Which eigen-vector is the right one?
- Concept of energy band
 - We are doing Gamma-point calculation

Merging source codes

- Copy all the (modified) source codes into practice_5
- Run Makefile after editing the library
- Run the calculations
 - Check how total energy is changing
 - Check how eigen-value is changing

Analysis of the results

- Eigen system of the Hamiltonian matrix
 - Eigen value = band energy \neq total energy
 - Eigen vector = coefficients of plane wave
- Total energy = sum of each component
 $= HT + PP + XC + T + (\text{ion-ion})$
- Repeat the SCF up to the criterion
 - Total energy/Total+band energy

Results

- Comparison to ABINIT
 - GTH pseudopotential
 - Gamma point calculation

```

Kinetic energy = 4.11339775459721E-02 ----> 0.0411408
Hartree energy = 6.34879675028364E-03 ----> 0.00634736
XC energy      = -1.91265689957026E-01 ----> -0.191266
Ewald energy   = -2.83729747948065E-01 ----> N/A
PspCore energy = -3.84286430543779E-05 ----> N/A
Loc. psp. energy= -7.68362301718382E-02 ----> -0.0764223
NL psp energy= 0.0000000000000000E+00 ----> N/A
>>>>>>>> Etot= -5.04387322423728E-01
  
```

* We didn't use the smearing yet

Next Step

- Non-local pseudopotential
 - Effect of angular momentum
- Advanced mixing
 - Broyden/BFGS/Kerker
- Finding the occupation number
 - Per energy band
 - Bisection method is required
- Smearing
 - Fermi-Dirac distribution
 - For numerical convergence
- Multiple integration points
 - Monkhorst-pack or special point method
 - Symmetry operation by group theory