

# User Guide for **ffr-LFDF**

Fadjar Fathurrahman

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation</b>	<b>1</b>
<b>3</b>	<b>Usage</b>	<b>2</b>
<b>4</b>	<b>Kohn-Sham equations</b>	<b>2</b>
<b>5</b>	<b>Lagrange basis function</b>	<b>2</b>
5.1	Periodic Lagrange function . . . . .	2
5.2	Cluster Lagrange function . . . . .	3
<b>6</b>	<b>Implementation</b>	<b>4</b>
6.1	Description of LF basis set . . . . .	4
6.2	Description of molecular or crystalline structure . . . . .	5
6.3	Pseudopotential . . . . .	6
6.4	Nonlocal pseudopotential . . . . .	6

## 1 Introduction

Welcome to **ffr-LFDF** documentation.

**ffr-LFDF** is a poor man's program (or collection of subroutines, as of now) to carry out electronic structure calculations based on density functional theory and Lagrange basis set.

How to compile

How to use

input parameters ...

subroutines ... (implementation)

Add tutorial on how to use **m-LF3d** module to solve Schrodinger equation in 1d.

In **LF3d** periodic, only gamma-point sampling is used.

## 2 Installation

Compiling and linking

A **Makefile** is provided. At the topmost part of the **Makefile** you need to specify which **make.inc** file you want to use. You need to decide which compiler to use if there are more than one compiler in you system. In the directory **platform** there are several **make.inc** files. Currently, **ffr-LFDF** is tested using the following compilers on Linux system:

- GNU Fortran compiler
- G95 Fortran compiler
- Intel Fortran compiler
- PGI Fortran compiler

- Sun (now part of Oracle) Fortran compiler

For typical Linux system, `make.inc.gfortran` is sufficient. You can manually edit the compiler options in the corresponding `make.inc` files.

Command `make` will build the library `libmain.a` and command `make main` will build the main executable `ffr LFDFT.x`.

### 3 Usage

PWSCF input file

### 4 Kohn-Sham equations

In this section a brief introduction to Kohn-Sham equation is given.

### 5 Lagrange basis function

#### 5.1 Periodic Lagrange function

For a given interval  $[0, L]$ , with  $L > 0$ , the grid points  $x_i$  appropriate for periodic Lagrange function are given by:

$$x_i = \frac{L}{2} \frac{2i - 1}{N} \quad (1)$$

with  $i = 1, \dots, N$ . Number of points  $N$  should be an odd number.

The periodic cardinal functions  $L_i^{\text{per}}(x)$ , defined at grid point  $i$  are given by:

$$L_i^{\text{per}}(x) = \frac{1}{\sqrt{NL}} \sum_{n=1}^N \cos\left(\frac{\pi}{L}(2n - N - 1)(x - x_i)\right). \quad (2)$$

The expansion of periodic function in terms of Lagrange functions:

$$f(x) = \sum_{i=1}^N c_i L_i^{\text{per}}(x) \quad (3)$$

with expansion coefficients  $c_i = \sqrt{L/N} f(x_i)$ . When doing variational calculation, the coefficients  $c_i$  are the variational parameters. The actual function values  $f(x_i)$  at grid points  $x_i$  is obtained by  $f(x_i) = \sqrt{N/L} c_i$ . The prefactor is sometimes abbreviated by  $h = L/N$  and is also referred to as scaling factor.

Consider periodic potential in one dimension:

$$V(x + L) = V(x). \quad (4)$$

Floquet-Bloch theorem states that the wave function solution for periodic potentials can be written in the form:

$$\psi_k(x) = e^{ikx} \phi_k(x) \quad (5)$$

where function  $\phi_k(x)$  and its first derivative  $\phi'_k(x)$  have the same periodicity as  $V(x)$  and  $k$  is a constant called the crystal momentum. Substituting this expression to Schrodinger equation we obtain:

$$\left[ -\frac{\hbar^2}{2m} \left( \frac{d^2}{dx^2} + 2ik \frac{d}{dx} - k^2 \right) + V(x) \right] \phi_k(x) = E \phi_k(k). \quad (6)$$

An alternative way of enforcing periodicity of the wave function is to require that:

$$\psi_k(x + L) = e^{ikL} \psi_k(x). \quad (7)$$

This condition follows from:

$$\begin{aligned} \psi_k(x + L) &= e^{ik(x+L)} \phi_k(x + L) \\ &= e^{ik(x+L)} \phi_k(x) \\ &= e^{ikL} e^{ikx} \phi_k(x) \\ &= e^{ikL} \psi_k(x) \end{aligned}$$

Using periodic cardinal the Schrodinger equation for periodic potential can be written as:

$$\sum_{j=1}^N \left[ -\frac{\hbar^2}{2m} \left( D_{jl}^{(2)} + 2ikD_{jl}^{(1)} - k^2 \delta_{jl} \right) + V(j) \delta_{jl} \right] \phi(j) = E \phi(l) \quad (8)$$

with  $l = 1, \dots, N$ .  $D_{jl}^{(1)}$  are matrix elements of the first derivatives:

$$D_{jl}^{(1)} = \begin{cases} 0 & j = l \\ -\frac{2\pi}{L} (-1)^{j-l} \left( 2 \sin \frac{\pi(j-l)}{N} \right)^{-1} & j \neq l \end{cases} \quad (9)$$

and  $D_{jl}^{(2)}$  are matrix elements of the second derivatives,  $N' = (N - 1)/2$ :

$$D_{jl}^{(2)} = \begin{cases} -\left( \frac{2\pi}{L} \right)^2 \frac{N'(N' + 1)}{3} & j = l \\ -\left( \frac{2\pi}{L} \right)^2 (-1)^{j-l} \frac{\cos(\pi(j-l)/N)}{2 \sin^2[\pi(j-l)/N]} & j \neq l \end{cases} \quad (10)$$

Note that,  $D_{jl}^{(1)}$  is not symmetric, but  $D_{jl}^{(1)} = -D_{lj}^{(1)}$ . Meanwhile, the second derivative matrix  $D_{jl}^{(2)}$  is symmetric, i.e.  $D_{jl}^{(2)} = D_{lj}^{(2)}$ . With the above expressions, first and second derivative of periodic cardinals can be expressed as

$$\frac{d}{dx} L_i^{\text{per}}(x) = \sum_{j=1}^N D_{ji}^{(1)} L_j^{\text{per}}(x) \quad (11)$$

$$\frac{d^2}{dx^2} L_i^{\text{per}}(x) = \sum_{j=1}^N D_{ji}^{(2)} L_j^{\text{per}}(x) \quad (12)$$

The previous approach also can be extended to periodic potential in 3D:

$$V(\mathbf{r}) = V(x, y, z) = V(x + L_x, y + L_y, z + L_z)$$

Using periodic LF, Schrodinger equation can be casted into the following form:

$$\left[ -\frac{\hbar^2}{2m} (\nabla^2 + 2i\mathbf{k} \cdot \nabla - \mathbf{k}^2) + V(\mathbf{r}) \right] \phi_{\mathbf{k}}(\mathbf{r}) = E \phi_{\mathbf{k}}(\mathbf{r}) \quad (13)$$

## 5.2 Cluster Lagrange function

For a given interval  $[A, B]$ , with  $B > A$ , the grid points  $x_i$  appropriate for cluster Lagrange function are given by:

$$x_i = A + \frac{B - A}{N + 1} i$$

where  $i = 1, \dots, N$ . Number of points  $N$  can be either odd or even number. The cluster Lagrange functions  $L_i^{\text{clu}}(x)$ , defined at grid point  $i$  are given by:

$$L_i^{\text{clu}}(x) = \frac{2}{\sqrt{(N+1)(B-A)}} \sum_{n=1}^N \sin(k_n(x_i - A)) \sin(k_n(x - A)). \quad (14)$$

where  $k_n = \pi n / (B - A)$ . The expansion of a function  $f(x)$  in terms of cluster Lagrange functions:

$$f(x) = \sum_{i=1}^N c_i L_i^{\text{clu}}(x) \quad (15)$$

with expansion coefficients  $c_i = \sqrt{(B-A)/(N+1)} f(x_i)$ . When doing variational calculation, the coefficients  $c_i$  are the variational parameters. The actual function values  $f(x_i)$  at grid points  $x_i$  is obtained by  $f(x_i) = \sqrt{(N+1)/(B-A)} c_i$ .

Matrix elements  $D_{jl}^{(2)}$  of the second derivatives for cluster Lagrange functions are

$$D_{jl}^{(2)} = \begin{cases} -\frac{1}{2} \left( \frac{\pi}{B-A} \right)^2 \frac{2(N+1)^2 + 1}{3} - \frac{1}{\sin^2[\pi j / (N+1)]} & j = l \\ -\frac{1}{2} \left( \frac{\pi}{B-A} \right)^2 (-1)^{j-l} \left[ \frac{1}{\sin^2\left[\frac{\pi(j-l)}{2(N+1)}\right]} - \frac{1}{\sin^2\left[\frac{\pi(j+l)}{2(N+1)}\right]} \right] & j \neq l \end{cases} \quad (16)$$

For free or cluster boundary condition, we don't need  $D_{jl}^{(1)}$ .

## 6 Implementation

### 6.1 Description of LF basis set

Description of LF basis set in 3d is given in module `m_LF3d`. All global variables in this module is given prefix `LF3d`.

```
MODULE m_LF3d
  IMPLICIT NONE
  INTEGER, PARAMETER :: LF3d_PERIODIC = 1
  INTEGER, PARAMETER :: LF3d_CLUSTER = 2
  INTEGER, PARAMETER :: LF3d_SINC = 3
  INTEGER :: LF3d_TYPE
  INTEGER, DIMENSION(3) :: LF3d_NN
  REAL(8), DIMENSION(3) :: LF3d_LL
  REAL(8), DIMENSION(3) :: LF3d_AA, LF3d_BB
  REAL(8), DIMENSION(3) :: LF3d_hh
  INTEGER :: LF3d_Npoints
  REAL(8) :: LF3d_dVol
  REAL(8), ALLOCATABLE :: LF3d_grid_x(:)
  REAL(8), ALLOCATABLE :: LF3d_grid_y(:)
  REAL(8), ALLOCATABLE :: LF3d_grid_z(:)
  REAL(8), ALLOCATABLE :: LF3d_D1jl_x(:, :)
  REAL(8), ALLOCATABLE :: LF3d_D1jl_y(:, :)
  REAL(8), ALLOCATABLE :: LF3d_D1jl_z(:, :)
  REAL(8), ALLOCATABLE :: LF3d_D2jl_x(:, :)
  REAL(8), ALLOCATABLE :: LF3d_D2jl_y(:, :)
  REAL(8), ALLOCATABLE :: LF3d_D2jl_z(:, :)
  REAL(8), ALLOCATABLE :: LF3d_lingrid(:, :)
  INTEGER, ALLOCATABLE :: LF3d_xyz2lin(:, :, :)
  INTEGER, ALLOCATABLE :: LF3d_lin2xyz(:, :)
  REAL(8), ALLOCATABLE :: LF3d_G2(:)
  REAL(8), ALLOCATABLE :: LF3d_Gv(:, :)
END MODULE
```

Variables in `m_LF3d` is initialized by calling the subroutine `init_LF3d_XX()`, where `XX` may be one of:

- `p`: periodic LFF
- `c`: cluster LF
- `sinc`: sinc L

```
SUBROUTINE init_LF3d_p( NN, AA, BB )
SUBROUTINE init_LF3d_c( NN, AA, BB )
SUBROUTINE init_LF3d_sinc( NN, hh )
```

In the above subroutines:

- `NN`: an array of 3 integers, specifying sampling points in  $x$ ,  $y$  and  $z$  direction.
- `AA`: an array of 3 floats, specifying left ends of unit cell.
- `BB`: an array of 3 floats, specifying right ends of unit cell.
- `hh`: an array of 3 floats, specifying spacing between adjacent sampling points.

Note that for periodic and cluster LF we have to specify `NN`, `AA`, and `BB` while for sinc LF we have to specify `NN` and `hh`. Note that for periodic LF `NN` must be odd numbers.

Example:

```
NN = (/ 35, 35, 35 /)
AA = (/ 0.d0, 0.d0, 0.d0 /)
BB = (/ 6.d0, 6.d0, 6.d0 /)
CALL init_LF3d_p( NN, AA, BB )
```

## 6.2 Description of molecular or crystalline structure

Description of molecular or crystalline structure is given in module `m_atoms`. Note that, unit cell for crystalline structure (currently only orthorombic structure is possible) is specified by `AA` and `BB` in call to `init_LF3d_p()`

```
MODULE m_atoms
  IMPLICIT NONE
  INTEGER :: Natoms
  INTEGER :: Nspecies
  REAL(8), ALLOCATABLE :: AtomicCoords(:, :)
  INTEGER, ALLOCATABLE :: atm2species(:)
  CHARACTER(5), ALLOCATABLE :: SpeciesSymbols(:)
  REAL(8), ALLOCATABLE :: AtomicValences(:)
  COMPLEX(8), ALLOCATABLE :: StructureFactor(:, :)
END MODULE
```

Currently, variables in module `m_atoms` are initialized by subroutine `init_atoms_xyz()`.

```
SUBROUTINE init_atoms_xyz( fil_xyz )
```

This subroutine takes one argument `fil_xyz` which is the path to XYZ file describing the molecular structure or crystalline structure.

### 6.3 Pseudopotential

Module m\_PsPot

```
MODULE m_PsPot
  USE m_Ps_HGH, ONLY : Ps_HGH_Params_T
  IMPLICIT NONE
  CHARACTER(128) :: PsPot_Dir = './HGH/'
  CHARACTER(128), ALLOCATABLE :: PsPot_FilePath(:)
  TYPE(Ps_HGH_Params_T), ALLOCATABLE :: Ps_HGH_Params(:)
  INTEGER :: NbetaNL ! max(NbetaNL)
  REAL(8), ALLOCATABLE :: betaNL(:, :) ! (Npoints, NbetaNL)
  INTEGER, ALLOCATABLE :: prj2beta(:, :, :, :) ! (iprj, Natoms, l, m)
  INTEGER :: NprojTotMax
END MODULE
```

We currently support HGH pseudopotential only. The HGH pseudopotential parameter is described by an array of type Ps\_HGH\_Params\_T which is defined in m\_Ps\_HGH:

```
TYPE Ps_HGH_Params_T
  CHARACTER(5) :: atom_name
  INTEGER :: zval
  REAL(8) :: rlocal
  REAL(8) :: rc(0:3)
  REAL(8) :: c(1:4)
  REAL(8) :: h(0:3, 1:3, 1:3)
  REAL(8) :: k(0:3, 1:3, 1:3)
  INTEGER :: lmax
  INTEGER :: Nproj_l(0:3) ! number of projectors for each AM
  REAL(8) :: rcut_NL(0:3)
END TYPE
```

### 6.4 Nonlocal pseudopotential

Nonlocal HGH pseudopotential action can be defined as follows:

$$\hat{V}_{\text{NL}}\psi = \tag{17}$$