1 Subroutine `setup_ffr_LFDFT()`

This subroutine prepares various tasks before actually solving the Kohn-Sham equation.

```
SUBROUTINE setup_ffr_LFDFT()

  USE m_options, ONLY : FREE_NABLA2, I_POISSON_SOLVE
  USE m_input_vars, ONLY : assume_isolated
  !
  INTEGER :: Narg   ! number of argument
  INTEGER :: iargc  ! needed for several compilers
  CHARACTER(64) :: filein
```

We check first number of argument(s) give to the program using built-in function `iargc()` and save the result to variable `Narg` Currently, we only support one argument, i.e. path to input file. The program will stop and display error message if `Narg` is not equal to one.

```
  Narg = iargc()
  IF( Narg /= 1 ) THEN
    WRITE(*,*)
    WRITE(*,*) 'ERROR: exactly one argument must be given: input file path'
    STOP
  ENDIF
```

We get the actual argument using built-in subroutine `getarg()`.

```
  CALL getarg( 1, filein )
```

We read the input file using subroutine `read_input()`

```
  CALL read_input( filein )
```

The following subroutine will initialize global variables related to basis function and grid points, molecular or crystalline structures, and pseudopotentials.

```
  CALL setup_from_input()
```

Various options, such as convergence criteria, choice of algorithms, etc which are given in the input file, will be converted to internal variables (mostly defined in module `m_options`).

```
  CALL setup_options()
```

The following calls will output information about molecular or crystalline structures, pseudopotentials, and basis function and grid points.

```
  CALL info_atoms()
  CALL info_PsPot()
  CALL info_LF3d()
```

This subroutine initialize nonlocal pseudopotential projectors. This subroutine must be called after variables from `m_LF3d` are initialized as they are defined on grid points.

TODO/FIXME: To be consistent, this call should be made in pseudopotential setup. (probably via subroutine `setup_from_input()`)

```
  CALL init_betaNL()
```

This call will initialize electronic states and occupation numbers.

```
  CALL init_states()
```

This call will initialize and calculate structure factor $S_f(\mathbf{G})$. This is required for periodic LF.
TODO/FIXME: This call should be made in `setup_from_input` or anther subroutine.

```
  CALL init_strfact_shifted()
```

Ewald energy

```
  IF( assume_isolated == 'sinc' ) THEN
    CALL calc_E_NN()
  ELSE
    CALL calc_Ewald_qe()
  ENDIF
```

The following call will initialize various global variables (arrays) needed to define Hamiltonian. It is mainly for storing potential terms.

```
  CALL alloc_hamiltonian()
```

Allocate local pseudopotential. For periodic sinc LF the potential is constructed directly on real space grid. For periodic LF, the potential is first constructed on reciprocal space and then transformed to real space grid via inverse FFT.

```
  IF( assume_isolated == 'sinc' ) THEN
    CALL init_V_ps_loc()
  ELSE
    CALL init_V_ps_loc_G()
  ENDIF

  ! Laplacian matrix
  CALL init_nabla2_sparse()
  ! ILU0 preconditioner based on kinetic matrix
  CALL init_ilu0_prec()

  IF( FREE_NABLA2 ) THEN
    CALL dealloc_nabla2_sparse()
  ENDIF

  IF( I_POISSON_SOLVE == 1 ) THEN
    CALL init_Poisson_solve_ISF()
  ELSEIF( I_POISSON_SOLVE == 2 ) THEN
    CALL init_Poisson_solve_DAGE()
  ENDIF
END SUBROUTINE
```