PROFESS 3.0 program structure

The subroutines and functions in PROFESS 3.0 have been organized into a set of source files according to their different functionalities. Some functions are obtained from public numerical software libraries. All modules can be separated into four categories: system initializer, energy functionals, optimization algorithms, and tools. Alternatively, the modules can also be arranged in a hierarchy: PROFESS → system initializer → optimizers → energy functionals→ tools. A detailed list of the source files is given in this section.

1. System initializer modules

There are three executables that can be compiled in PROFESS 3.0: *PROFESS* (or *pPROFESS* for the parallel version), *CINEB*, and *RhoConvert*. *PROFESS* is the major one to perform regular OFDFT calculations (see details of *CINEB* and *RhoConvert* in the following table). PROFESS.f90 is the main driver which calls the initializer and then the optimizer. Within the initializer, Initializer.f90 reads in input files and calls initialization subroutines throughout the program, while System.f90, CellInfo.f90, and PlaneWave.f90 are modules that contain global variables which can be used directly in other modules.

SUBJECT	NAME	DESCRIPTION
Independent code	*PROFESS.f90*	The main program starts from here.
	CINEB.f90	The saddle point search solver using the climbing-image nudged elastic band method, which calls the PROFESS executable to optimize the structure of images.
	RhoConvert.f90	Converts between different formats of charge densities.
Initializer	*Initializer.f90*	Initializes fundamental data structures, including allocation of global arrays, FFT size, plane waves, initial charge densities, etc.
	ReadInputFile.f90	Input reader for the .inpt file.
	ReadIonFile.f90	Input reader for the .ion file.
	ReadAtomDen.f90	Input reader for the atomic density file, if any.
	System.f90	Initializes global variables such as real space electron densities.
	CellInfo.f90	Keeps track of the cell size in both real space and reciprocal space.
	PlaneWave.f90	Initializes plane wave related arrays according to the energy cutoff and cell size.
	SetupFFT.f90	Used to set up FFT according to user inputs and requirements of FFTW 3.0 package.

	SetupKEDF.f90	Used to set up initial parameters for different KEDFs.
	LocalPseudoPot.f90	Includes subroutines that obtain the interpolated value of local pseudopotentials and their derivatives.

2. Optimization modules

Cell optimization and molecular dynamics are two mutually exclusive parts. There are three modules for optimization algorithms, which are CellOptimizers (for cell lattice vectors), IonOptimizers (for ion coordinates), and RhoOptimizers (for electron densities), respectively. The RhoOptimizers are also used in molecular dynamics simulations.

SUBJECT	NAME	DESCRIPTION
	Optimizer.f90	Wrapper for cell/ion/density optimizations.
Cell optimization	*CellOptimizer.f90*	Steepest-descent method for cell optimizations.
	CalStress.f90	Wrapper for stress calculation.
	RefreshCell.f90	Refresh cell-related variables when cell is changed.
Ion optimization	*IonOptimizer.f90*	Wrapper for ion optimizations.
	IonOptQui.f90	QUICKMIN method for ion optimizations.
	IonOptCON.f90	Conjugate Gradient (CG) method for ion optimizations by using Dcsrch subroutine.
	IonOptCG2.f90	CG method for ion optimizations based on home-made line search code, more robust than calling Dcsrch subroutine.
	IonOptBFGS.f90	BFGS method for ion optimizations.
	CalForce.f90	Wrapper for force calculations.
	RefreshIons.f90	Refresh necessary variables when ion positions are changed.
Density Optimization	*RhoOptimizer.f90*	Wrappers for density optimizations.
	CalPotential.f90	Wrappers for potential calculations.
	RhoOptN.f90	Rho optimization method with conserved electron numbers during line search for sqrt(rho). Three methods, TN, CG, and BFGS, are included in this module; the line search directions are

		calculated by calling one of the RhoDir subroutines depending on the method chosen.
	RhoLineSearch.f90	Line search method (for RhoOptN.f90).
	RhoDirNEW.f90	Obtain directions using truncated Newton method (for RhoOptTN.f90).
	RhoDirCG.f90	Obtain directions using CG method (for RhoOptTN.f90).
	RhoDirBFGS.f90	Obtain directions using BFGS method (for RhoOptTN.f90).
	RhoOptNEW.f90	Sqrt truncated newton optimization method.
	RhoOptSQR.f90	Sqrt conjugate gradient optimization method.
	RhoOptLOG.f90	Truncated-Newton method for log(rho).
Molecular Dynamics	*MolecularDynamics.f90*	Subroutines and functions that are used in the following modules.
	MolecularDynamicsNVT.f90	Parrinello-Rahman isothermal-isobaric ensemble (NPT).
	MolecularDynamicsNPT.f90	Nos é-Hoover canonical ensemble (NVT).
	MolecularDynamicsNVE.f90	NVE ensemble.

3. Energy functional modules

Kinetic energy density functionals (KEDFs) are a major part of PROFESS 3.0. We have a unique module for each type of KEDF, and also other energy terms in the total energy.

SUBJECT	NAME	DESCRIPTION
Kinetic Energy Density Functional (KEDF)	*KEDF_TF.f90*	Thomas-Fermi KEDF.
	KEDF_VW.f90	von Weizs äcker KEDF.
	KEDF_WT.f90	Wang-Teter KEDF.
	KEDF_WTkernel.f90	Fill WT KEDF kernel.
	KEDF_WGC.f90	Wang-Govind-Carter 99 KEDF.
	KEDF_WGCkernel.f90	Fill WGC99 KEDF kernel.
	KEDF_CAT.f90	CAT KEDF.
	KEDF_Q.f90	LQ and HQ KEDF.
	KEDF_HC10.f90	Huang-Carter KEDF.
	KEDF_WGCD.f90	WGC-decomposed (WGCD) KEDF.

	KEDF_DenDec.f90	Density decomposed KEDF.
	KEDF_GGA.f90	Generalized gradient approximation (GGA) type KEDF and also vWGTF KEDFs.
	KEDF_EvW.f90	EvW KEDF.
	KEDF_KernelODE.f90	Creates non-local KEDF kernels.
Ion-electron, electron-electron, ion-ion and exchange correlation terms.	*IonElectron.f90*	Used to calculate ion-electron energy, potential, force, and stress.
	IonElectronSpline.f90	Used to calculate ion-electron energy, potential, force, and stress using b-spline methods.
	CBSpline.f90	Used to set up needed arrays and support needed functions for the b-spline method.
	XC_LDA.f90	Perdew-Zunger or Perdew-Wang local density approximations (LDA) for exchange-correlation energy, potential, and stress.
	XC_PBE.f90	Perdew-Burke-Ernzerhof (PBE) form of generalized gradient approximation (GGA) for exchange-correlation energy, potential, and stress.
	Hartree.f90	Used to calculate Hartree electron-electron energy, potential, and stress.
	Ewald.f90	Used to calculate ion-ion energy, potential, force, and stress.

4. Tools modules

The files with the prefix "Math" implement a diverse set of mathematical functions used throughout the computation. The Fast Fourier Transform interface has its own module. Depending on whether the serial version or the parallel version of PROFESS is compiled, the module Fourier-serial or Fourier-parallel is used. Another set of modules provide useful tool functions. For example, fundamental constants (*e.g.*, Hartree, Bohr radius) are stored in Constants, and the Timer module is used extensively for benchmarking calculations.

SUBJECT	NAME	DESCRIPTION
Math functions	*MathFunctions.f90*	Various useful math functions.
	MathSpline.f90	Spline subroutines are included.
	MathNMS.f90	Hermite interpolation method.
	MathLNsrch.f90	Line search algorithm written by Chen Huang.
	MathLBFGS.f	Used in BFGS ion optimization method.

	MathRKsuite.f	Runge-Kutta code from www.netlib.org, written by R.W. Brankin, I. Gladwell, and L.F. Shampine.
	MathDCsrch.f	Line search code from the MINPACK project at Argonne National Laboratory and University of Minnesota.
	MathDCstep.f	Line search code from the MINPACK project at Argonne National Laboratory and University of Minnesota.
Tool functions	*MPI_Functions.f90*	MPI helper functions are wrapped in this module.
	Neighbor.f90	Helps to prevent distances between atoms from getting too small.
	OutputFiles.f90	Control output log file name, etc.
	Output.f90	Prints out various information in a pretty form.
	Report.f90	Reports various information in a pretty form during optimization.
	Timer.f90	Useful for recording time consumption of subroutines and functions.
	Constants.f90	Physical constants.
Fast Fourier Transform (FFT)	*Fourier.f90*	FFT interfaces translating PROFESS3 style to PROFESS2 interfaces.
	Fourier_v1.f90	New FFTW3 interface for serial version of PROFESS.
	Fourier_v1_para.f90	New FFTW3 interface for parallel version of PROFESS.