This code is intended to provide auxiliary subroutines for the KEF project. Here is some general information about the code:

1. The ONLY input file the user will need to modify is "InputFile.inp". This file permits to set the parameters for constructing the grid. This code bases the calculations performed on the tensor product of a radial grid and Lebedev (angular) grids.
   These parameters are:
   Nr – Number of radial points
   Rm – Parameter for the radial grid. It is atom dependent and related to the atomic radius.
   In the same very file are some suggested values for some closed shell atoms.
   Lebedev_order – Permits to choose among several Lebedev grids
   Z_Atomic – Atomic number

   For further information about the utilized grids the user can check the following papers:

   - Chemical Physics Letters, **209**, 506-512 (1993)
   - Lebedev's papers: This are classic papers. They are cited by the Wikipedia article on Lebedev quadrature http://en.wikipedia.org/wiki/Lebedev_quadrature

2. Files that are needed but should not be modified by the user are:

   "Lebedev_XXXX_REC.txt", where XXXX = 38, 50, 86, 170, 350 and 1202

   These files contain the angular points of the Lebedev grids in Cartesian coordinates.

3. The file "rho_GAyers.txt" contains the coefficients and exponents used for expressing the atomic density as a sum of several (up to 20) Gaussians. The first column corresponds to the coefficients while the second one does to the exponents.
   It contains the parameters for atoms from Hydrogen to Argon and Cupper.

4. Since this code is intended to be run in SHARCNET, the MAKEFILE is included. Its name is

   "makefile_RogKEF-ifortlapack.txt"

   The SSH Secure Shell software has to be used in order to transfer the files from your local directory to sharcnet. Once installed and all the files have been transferred, the following instructions have to be typed in order to compile and run the code (it is suggested to run it in the work subdirectory of the user's account)

   To compile type (hit enter after each of them):
   i. make -f makefile-RogKEF-ifortlapack.txt realclean
   ii. make -f makefile-RogKEF-ifortlapack.txt clean
   iii. make -f makefile-RogKEF-ifortlapack.txt

After this instruction has been executed, a bunch of messages will be shown on the screen…that is OK.

iv. make -f makefile-RogKEF-ifortlapack.txt clean

To run:

i. Make sure the executable program has been created. This code has been named as “Auxiliary_KEF.x”.

ii. Submit your job to the queue by typing
sqsub -r 9000 -o screenoutput.out ./Auxiliary_KEF.x

iii. After completion the following OUTPUT files will have been created:
- “grid_points.txt” – This file contains the grid points used for the calculation.
- “Check_Normalization.out” – It contains the computed value of the density normalization.
- “Check_EigenVectors.out” – It contains the values of the eigenvalues as well as the error associated with them.

5. List of the FORTRAN files (.f90) and general information about the subroutines contained in each of them. For detailed the

I. FILE: NRTypeVar.f90
MODULE: The module ”nrtype” in which the type of used variables is declared (taken from Numerical Recipes)

II. FILE: Read_InputFile.f90
MODULE: The module “InputFile”.
SUBROUTINES:
- Input_File(Nr,Rm,Lebedev_order,Z_Atomic). Reads the parameters present in the InputFile.inp

III. FILE: Module_Grids.f90
MODULE: The module “GridSubroutines”
SUBROUTINES:

SUBROUTINE GetGrid_File(Nr,Lebedev_order,Rm,Z_Atomic,Ntot) which reads the file UNIT=15 'Lebedev_RECgrid.txt' This file contains the angular points in cartesian coordinates (x,y,z). Once the points are read they are changed to points in spherical coordinates (r,theta,phi). Finally the tensor product of the radial and angular grids is performed and the complete grid is printed out in the file UNIT=20 'grid.txt'

SUBROUTINE Grid_Points(NoPoints,Wtot,Xtot,Ytot,Ztot)
This subroutine reads the grid points of the complete grid from the file
"grid_points.txt" that has been identifed as UNIT 20

IV. FILE: Check_Norm.f90
MODULE: The module “CheckNormalization”
SUBROUTINES:
SUBROUTINE Check_Normalization(Nr,Lebedev_order,GPoints,Rm,Z_Atomic)
This subroutine computes the normalization of the density as a check.

V. FILE: Integrals_Module.f90
MODULE: The module “IntegralsOnGrid”
SUBROUTINES:
SUBROTUINE
IntegrateFunction3D(func3D,Nr,Rm,Lebedev_order,Value3DIntegral)
This subroutine computes the integral of a 3D function

SUBROUTINE
IntegrateFunction6D(func6D,Nr,Rm,Lebedev_order,Value6DIntegral)
This subroutine computes the integral of a 6D function

VI. FILE: Density_Function.f90
MODULE: The module “DensityFunction”
FUNCTIONS:
FUNCTION Density(Z_Atomic,r,theta,phi)
This function is the density of some closed shell atoms I have been using
It is obtained based on the fittings James, myself and some others have done.

FUNCTION FuncTest3D(r,theta,phi)
This function is just a random one chosen to check if the
3D integrating routine works

FUNCTION FuncTest6D(r1,theta1,phi1,r2,theta2,phi2)
This function is just a random one chosen to check if the
6D integrating routine works

VII. FILE: MatrixCalc.f90
MODULE: The module “CalcMatrix”
SUBROUTINES:
SUBROUTINE EigenMatrix(m,A,EigenValues,EigenVectors)
This subroutine computes the eigenvalues and eigenvectors of a given square matrix. The matrix has to be symmetric.
This subroutine uses the following LAPACK routine

SUBROUTINE DSYEV(JOBZ, UPLO, N, A, LDA, W, WORK, LWORK, INFO)
For more detailed information about this subroutine please follow the link http://netlib.org/lapack/double/dsyev.f

VIII. FILE: DRIVER.F90

This is the driver from which all the subroutines are called.

Rogelio Cuevas
February 11, 2010