Project Directions

Hamiltonian

Fortan code build_KEmats.f90

- lines 17 & 18 are now member data in *FEMDVR* class.
- lines 21 & 22 are now taken care of in the *Quadrature* class.
- lines 24 – 62 builds the laplacian in our DVR representation. Notice on line 41 the scale factor "scal_w" just places you in the right finite element, **but on line 47 the scale factor is for the radau complex tail and this is needed to compute the Hamiltonian on the complex tail.**
- Line $64-71$ builds the centrifugal term $\frac{l(l+1)}{r^2}$ which makes the Hamiltonian a $3D$ cube data structure with the third dimension having a size of *lmax*.
- Lines $72-82$ builds the inverse Hamiltonian using *zgesv* from LAPACK. Therefore, the makefile will need to incorporate LAPACK libraries and link properly.
- Also, two member datas this class **has to have** is of course the eigenvectors and eigenvalues of this Hamiltonian given in our DVR representation. This means this class should diagonalize the Hamiltonian matrix. This is done in **cdiag.f90** subroutine using the *zgeev* LAPACK routine.

Design Interface

- Pass in Potential class so the user knows what kind of Hamiltonian is being constructed. The potential is diagonal in the DVR representation so just need to add the potential to the diagonal (i.e. [i][i]).

2-electron Integrals code build_R12matsRad.f90

- If there is time checkout this subroutine that computes the r^{-12} term in the Hamiltonian.

Potential

- The Fortran function V_pot.f95 has some examples of different potentials. This function is just a simple function and isn't very powerful but making a *Potential* class with various potentials that inherits the FEMDVR grid can be useful and more flexible.
- The potential needed to calculate the cross-sections is the Coulomb potential $\frac{Z}{r}$, where Z is the atom's charge.
- This class should be able to plot the various potentials on the DVR grid. I am making a gnuplot class that will use *gnuplot* external plotting program. This class will take actual gnuplot commands and create a nice figure. You can look up the various tutorials on gnuplot on the interwebs but I should have a plotting member function for FEMDVR class soon so you can check that out as an example.

Design Interface

- This is still a question for me. I know we were talking about having a base potential that inherits the grid points and then on top of that have the different potentials with their respected forms. This is still up for creative input and bad ass discussion.

Cross-Section

- I added a new directory to the fortran directory called *cross_section.* This directory has the various other pieces we need in order to solve our single ionization problem.
- First we need to get our right hand side, $\mu\Psi_0$. This subroutine is found in the file **Right_Hand_Side.f95**. This subroutine computes the angular part using *GauntCoef* subroutine and has the radial analytic form of the wave function.

- Second we need to get our scattered wave equation Ψ^{sc}. This is created by solving our driven equation

  $$(ES - H)\Psi^{sc} = \mu\Psi_0.$$

  To solve this system of linear equations we need to use *zgesv* and all of this can be found in **Driven_Eq_Solve.f95**. The S in the equation above is the overlap of the eigenfunctions. Look in the fortran subroutines for it's calculation.
- Now that we have the right hand side and our scattered wave function, we can solve the optical theorem using the equation

  $$\sigma_{tot} = -\frac{8\pi}{k^2}\text{Im} < \Psi^{sc}|\mu|\Phi_0 > .$$

  The file **Opt_Theorem.f95** has the fortran code that one can follow for calculating this. It's simply the dot product of the two vectors.
- The file **Flux.f95** uses a different formula that calculates the cross-section. This is useful to have for checking the optical theorem calculation is correct.
- The **main.f95** file shows the full logic of the cross-section calculation. **The comments should help here.**

Dividing up the work

Let me know your progress and we can talk about the creating the cross-section pieces together and who well work on what piece. But know you guys have the fortran subroutines that are very self-explanatory for the calculation we need to do.