Numerical Algorithms Applied to Computational Quantum Chemistry

Homework 2: Analytical and Numerical Integration for Overlap Integrals

---

Link to Github Repo: https://github.com/Mattwaseem/chem279-Waseem-hw
Repo: publicly available.
Username: **MattWaseem**

The focus was on numerical concepts covered in the first five weeks of lecture on approaches to estimate energy of MO through a summation of atomic orbitals by calculating the overlap integrals between Gaussian functions using both numerical and analytical approaches. This was divided into two problems:

- **Problem 1:** Numerical 1D Overlap Integral
- **Problem 2:** Analytical 3D Overlap Integral

Each problem is addressed by implementing specific algorithms for integration, and the results are output into designated directories after processing the input files.

—

This is intentionally left blank please go to the next page.

```
.
├── CMakeLists.txt                # CMake build configuration file
├── Plots                         # Directory to store any generated plots (currently empty)
├── bin                           # Directory for compiled executable 'homework2_exec'
│   └── homework2_exec
├── build                         # Directory containing build-related files
├── calculated_output_p1          # Directory for output files of Problem 1 (Numerical)
│   ├── 1-output.txt
│   ├── 2-output.txt
│   └── 3-output.txt
├── calculated_output_p2          # Directory for output files of Problem 2 (Analytical)
│   ├── 1-output.txt
│   ├── 2-output.txt
│   └── 3-output.txt
├── include                       # Directory containing header files for the project
│   ├── CartesianGaussian.hpp
│   ├── Gaussian1D.hpp
│   ├── InputParser.hpp
│   ├── Integrator1D.hpp
│   └── OverlapIntegral.hpp
├── sample_input                  # Directory containing sample input files
│   ├── analytical                # Input files for Problem 2 (Analytical 3D)
│   ├── numerical                 # Input files for Problem 1 (Numerical 1D)
├── sample_output                 # Sample output files for comparison
├── src                           # Directory containing source code files
│   ├── CartesianGaussian.cpp
│   ├── Gaussian1D.cpp
│   ├── InputParser.cpp
│   ├── Integrator1D.cpp
│   ├── OverlapIntegral.cpp
│   └── main.cpp
│
```

How to Build and Run the Code

1. Go to the build directory
2. Run the following commands to utilize the Cmake build system set up to execute this code

```
cd build
cmake ..
Make
```

3. To run the executable

```
make run-all
```

4. This command will execute the program for all input files (numerical and analytical).
The output will be saved in the calculated_output_p1 and calculated_output_p2 directories, corresponding to Problem 1 and Problem 2, respectively.

1.1 Discussion of Problem 1:

This problem focused on calculating the overlap integral between two Gaussian functions using numerical integration methods. Specifically, the **Trapezoidal Rule** is implemented for evaluating the integral. There are other approaches, such as the rectangular approach and other shapes. The overall objective was to sum all the areas under the curve using an approximation approach. In this problem there were two integrals G_A and G_B one of them is shown below:

$$G_A(x) = (x - X_A)^{l_A} \exp\left[-\alpha\,(x - X_A)^2\right] \tag{1.1}$$

The overlap (`OverlapIntegral1D`) for the above integral was calculated based on the following equation:

$$\begin{aligned} S_x^{AB} &= \int_x G_A(x) G_B(x) && (1.2)\\ &= \int_x (x - X_A)^{l_A}\,(x - X_B)^{l_B} \exp\left[-\alpha\,(x - X_A)^2 - \beta\,(x - X_B)^2\right] && (1.3)\end{aligned}$$

The output for Problem 1 will be saved in the `calculated_output_p1` directory. For example, after running the program, the file `1-output.txt` will contain the overlap integral result. This was a numerical approach that used an indefinite integral calculation approach. A more precise approach that would be practical would be to evaluate the above approach in 3-dimension using analytical approach which is what was done for problem 2.

2.1 Discussion of Problem 2:

This problem focused on the overlap integral between two Gaussian functions in **3D** is calculated using an analytical approach. The idea behind this is to evaluate the overlap integral in 3D space between the spherical shell space of two normalized cartesian Gaussian functions. The normalized cartesian gaussian function (also known as primitive gaussian) is as follows:

$$\omega(\mathbf{r}) = N\,(x - X)^l \left(y - Y\right)^m (z - Z)^n \exp\left[-\alpha\,(\mathbf{r}\text{-}\mathbf{R})^2\right] \tag{2.1}$$

The overlap between the two is evaluated by taking evaluating the triple integral since the product of two primitive gaussian is a gaussian.

$$\begin{aligned} S^{AB} &= \int_x \int_y \int_z \omega_A(\mathbf{r})\omega_B(\mathbf{r}) && (2.2) \\ &= S_x^{AB} S_y^{AB} S_z^{AB} && (2.3) \end{aligned}$$

Understanding the product of two primitive gaussians is a gaussian the following equation can be derived below in order to more practically evaluate the overlap area between the different shells:

$$S_x^{AB} = \exp\left[-\frac{\alpha\beta(X_A - X_B)^2}{\alpha+\beta}\right]\sqrt{\frac{\pi}{\alpha+\beta}}\sum_{i=0}^{l_A}\sum_{j=0}^{l_B}\binom{l_A}{i}\binom{l_B}{j}\frac{(i+j-1)!!(X_P - X_A)^{l_A - i}(X_P - X_B)^{l_B - j}}{\left[2(\alpha+\beta)\right]^{(i+j)/2}} \tag{2.9}$$

Implementing the above equation into modular code as shown in the source (`src`) directory results in the sum of the overlaps in the output files. The output for Problem 2 is saved in the `calculated_output_p2` directory. The results are printed in a formatted manner that includes the overlap integrals and the angular momentum components. For example, after running the program, the file `1-output.txt` will be generated in the output directory.