

Wavefunction Report - How to extract atomic and molecular wavefunction data from a molden file, check its normalization and calculate overlaps with other wavefunctions

Antonis Hadjipittas

September 3, 2020

1 Introduction

In this report all the steps are explained in order to extract data from a molden file using one of two existing bash scripts, one for atoms and one for molecules. The two scripts, named 'moldenscriptAtom.sh' and 'moldenscriptMolecule.sh' take the data from molden and extract it in a format readable by mathematica. Another mathematica script called 'construct_wavefunction.n' is then used to calculate the wavefunction, calculate its normalization, check its overlap with other wavefuctions and finally plot it.

2 Theory - The mathematics of wavefunctions

In order to generate the wavefunctions for the orbitals of an atom or molecule, molpro needs a basis set to be specified. The basis set specifies how many primitive Gaussian type orbitals and contracted orbitals are used to generate a wavefunction. Gaussian type functions are used due to their computation efficiency and are of the form

$$G_{ijk}(x, y, z; \alpha) = N_{ijk} d x^i y^j z^k e^{-\alpha(x^2+y^2+z^2)} \quad (1)$$

where x , y and z represent the position of the nucleus in question, α is the exponent assigned by the basis set for each Gaussian type function and d is a primitive coefficient also assigned by the basis set for each Gaussian type orbital. The normalisation constant

$$N_{ijk} = \left(\frac{2\alpha}{\pi}\right)^{3/4} \left[\frac{(8\alpha)^{i+j+k} i! j! k!}{(2i)!(2j)!(2k)!}\right]^{1/2} \quad (2)$$

where the numbers i , j and k represent the three directions of the angular momentum l , where $l = i + j + k$. For each different orbital, s, p, d, f , the angular momentum number is different. For an s orbital, $l = 0$, p orbital $l = 1$ and so on. For a p orbital therefore, there are three possible combinations for the angular momentum, $i = 1, j = 0, k = 0$ or $i = 0, j = 1, k = 0$ or $i = 0, j = 0, k = 1$.

A number of primitive Gaussians with the same symmetry are then linearly combined in order to create a contracted Gaussian of the form

$$\phi_n(x, y, z) = \sum_{m=1}^K w_m^n G_{ijk}(x, y, z; \alpha_m^n) \quad (3)$$

where w_m^n is the contraction coefficient also given by the basis set. This equation holds for primitive Gaussians with identical angular momentum $l = i + j + k$ and the different primitive Gaussians vary only by their exponent coefficients α . In order to form the wavefunction of a certain atomic orbital, we then need to combine a number of these contracted coefficients together. For example, considering the Hydrogen 1s orbital, using the STO-2G basis, the wavefunction looks like

$$\psi_n(x, y, z) = w_1^n G_{000}(x, y, z; \alpha_1^n) + w_2^n G_{000}(x, y, z; \alpha_2^n) \quad (4)$$

3 Molden - How data is displayed

In order to generate a molden file, a molpro script needs to be executed, with the command 'put,molden,filename.molden;' at the end of the script, where 'filename' represents the name the user wants to give to the file. The molden file contains all the data needed for all the wavefunctions of the atom or molecule. The data given by Molden are the primitive coefficients d , the exponents α and the contraction coefficients w_m^n from equations 1 and 3. A molden file is shown below, for neutral carbon atom using basis cc-pvtz.

3.1 Atomic Wavefunction

```
1. [Molden Format]
2. [Atoms] Angs
3. C 1 6 0.0000000000 0.0000000000 0.0000000000
4. [GTO]
5. 1 0
6. s 10 1.00
7. 0.8236000000D+04 0.5309998617D-03
8. 0.1235000000D+04 0.4107998930D-02
9. 0.2808000000D+03 0.2108699451D-01
10. 0.7927000000D+02 0.8185297868D-01
11. 0.2559000000D+02 0.2348169388D+00
12. 0.8997000000D+01 0.4344008869D+00
13. 0.3319000000D+01 0.3461289099D+00
14. 0.9059000000D+00 0.3937798974D-01
15. 0.3643000000D+00 -0.8982997660D-02
16. 0.1285000000D+00 0.2384999379D-02
17. s 10 1.00
18. 0.8236000000D+04 -0.1129999414D-03
19. 0.1235000000D+04 -0.8779995449D-03
20. 0.2808000000D+03 -0.4539997647D-02
21. 0.7927000000D+02 -0.1813299060D-01
22. 0.2559000000D+02 -0.5575997110D-01
23. 0.8997000000D+01 -0.1268949342D+00
24. 0.3319000000D+01 -0.1703519117D+00
25. 0.9059000000D+00 0.1403819272D+00
26. 0.3643000000D+00 0.5986836897D+00
27. 0.1285000000D+00 0.3953887950D+00
28. s 10 1.00
29. 0.8236000000D+04 0.0000000000D+00
30. 0.1235000000D+04 0.0000000000D+00
31. 0.2808000000D+03 0.0000000000D+00
32. 0.7927000000D+02 0.0000000000D+00
33. 0.2559000000D+02 0.0000000000D+00
34. 0.8997000000D+01 0.0000000000D+00
35. 0.3319000000D+01 0.0000000000D+00
36. 0.9059000000D+00 0.1000000000D+01
37. 0.3643000000D+00 0.0000000000D+00
38. 0.1285000000D+00 0.0000000000D+00
39. s 10 1.00
40. 0.8236000000D+04 0.0000000000D+00
41. 0.1235000000D+04 0.0000000000D+00
42. 0.2808000000D+03 0.0000000000D+00
43. 0.7927000000D+02 0.0000000000D+00
44. 0.2559000000D+02 0.0000000000D+00
45. 0.8997000000D+01 0.0000000000D+00
46. 0.3319000000D+01 0.0000000000D+00
47. 0.9059000000D+00 0.0000000000D+00
48. 0.3643000000D+00 0.0000000000D+00
49. 0.1285000000D+00 0.1000000000D+01
50. p 5 1.00
```

51. 0.1871000000D+02 0.1403099838D-01
 52. 0.4133000000D+01 0.8686598995D-01
 53. 0.1200000000D+01 0.2902159664D+00
 54. 0.3827000000D+00 0.5010079421D+00
 55. 0.1209000000D+00 0.3434059603D+00
 56. p 5 1.00
 57. 0.1871000000D+02 0.0000000000D+00
 58. 0.4133000000D+01 0.0000000000D+00
 59. 0.1200000000D+01 0.0000000000D+00
 60. 0.3827000000D+00 0.1000000000D+01
 61. 0.1209000000D+00 0.0000000000D+00
 62. p 5 1.00
 63. 0.1871000000D+02 0.0000000000D+00
 64. 0.4133000000D+01 0.0000000000D+00
 65. 0.1200000000D+01 0.0000000000D+00
 66. 0.3827000000D+00 0.0000000000D+00
 67. 0.1209000000D+00 0.1000000000D+01
 68. d 1 1.00
 69. 0.1097000000D+01 0.1000000000D+01
 70. d 1 1.00
 71. 0.3180000000D+00 0.1000000000D+01
 72. f 1 1.00
 73. 0.7610000000D+00 0.1000000000D+01
 74.
 75.
 76. [MO]
 77. Sym= 1.1
 78. Ene= -11.3601
 79. Spin= Alpha
 80. Occup= 2.000000
 81. 1 1.00016454879686
 82. 2 0.000036118949355
 83. 3 -0.000301912804874
 84. 4 -0.000010163251979
 85. 5 0.0
 86. 6 0.0
 87. 7 0.0
 88. 0.0
 89. 9 0.0
 90. 10 0.0
 91. 11 0.0
 92. 12 0.0
 93. 13 0.0
 94. 14 0.000502821503131
 95. 15 -0.000251410751443
 96. 16 -0.000251410751688
 97. 17 0.0
 98. 18 0.0
 99. 19 0.0
 100. 20 0.000424271157570
 101. 21 -0.000212135578357

```

102. 22 -0.000212135579213
103. 23 0.0
104. 24 0.0
105. 25 0.0
106. 26 0.0
107. 27 0.0
108. 28 0.0
109. 29 0.0
110. 30 0.0
111. 31 0.0
112. 32 0.0
113. 33 0.0
114. 34 0.0
115. 35 0.0
116. Sym= 2.1
117. Ene= -0.7225
118. Spin= Alpha
119. Occup= 2.000000
120. 1 0.000715219629512
121. 2 1.01036015543517
122. 3 0.003102820351950
123. 4 -0.013803569065178
124. ...

```

Starting from the top, all the entries will be explained. At line 2 the units for distance are specified as angstroms. At line three, the atom used is specified as C, number shows shows the number of atom (for molecules), the number 6 shows the mass number of the nucleus, and the rest of the three numbers show the x , y and z position of the nucleus. Line 6 holds information for the first contracted Gaussian. It specifies its type as s and then it specifies the number of primitive Gaussians there are in that contracted as 10. The same is done for all other contracted Gaussians in lines 17, 28, 39, 50, 56, 62, 68, 70 and 72. For example line 50 shows a p type contracted Gaussian with 5 primitives. The numbers just below each line (lines 7-16, 18-27, etc) show two numbers, the first is the exponent α and the second is the primitive coefficient d , both shown in equation 1. This pattern continues until 73, where all the information about the primitive Gaussians is given.

From line 77 until line 115, all the information is given about the first atomic orbital. Line 77 specifies the symmetry of the orbital, where 1.1 stands for the 1st orbital (first 1) of symmetry 1 (second 1), meaning that it corresponds to the $1s$ orbital. Line 78 has the energy of the orbital in atomic units (check if true) and line 80 has the occupancy of the orbital, in this case being 2, i.e. full. This means that the atomic orbital is $1s^2$. From lines 81 to 115, the contracted coefficients making up the atomic orbital are given. In line 16 the symmetry of the next orbital is given being 2.1, meaning it is the 2nd orbital of the 1st symmetry, i.e. $2s$. Occupancy in line 119 is once again 2, therefore our orbital is $2s^2$. From lines 120 onwards are the contraction coefficients of this orbital.

3.2 Molecular Wavefunction

In the case of molecules, the data is displayed in a similar way, with the exception that every atom has its own atomic wavefunctions. An example is shown for neutral CH_4 , of how the primitive Gaussian exponents and coefficients are displayed.

1. [Molden Format]
2. [Atoms] Angs
3. C 1 6 -0.0000000000 0.0000000000 0.0000970175
4. H 2 1 -0.0000000000 0.0000000000 1.0870970175
5. H 3 1 1.0246513008 0.0000000000 -0.3627510385
6. H 4 1 -0.5123256504 -0.8873740565 -0.3627510385
7. H 5 1 -0.5123256504 0.8873740565 -0.3627510385
8. [GTO]
9. 1 0
10. s 10 1.00
11. 0.8236000000D+04 0.5309998617D-03
12. 0.1235000000D+04 0.4107998930D-02
13. 0.2808000000D+03 0.2108699451D-01
14. 0.7927000000D+02 0.8185297868D-01
15. 0.2559000000D+02 0.2348169388D+00
16. 0.8997000000D+01 0.4344008869D+00
17. 0.3319000000D+01 0.3461289099D+00
18. 0.9059000000D+00 0.3937798974D-01
19. 0.3643000000D+00 -0.8982997660D-02
20. 0.1285000000D+00 0.2384999379D-02
21. s 10 1.00
22. 0.8236000000D+04 -0.1129999414D-03
23. 0.1235000000D+04 -0.8779995449D-03
24. 0.2808000000D+03 -0.4539997647D-02
25. 0.7927000000D+02 -0.1813299060D-01
26. 0.2559000000D+02 -0.5575997110D-01
27. 0.8997000000D+01 -0.1268949342D+00
28. 0.3319000000D+01 -0.1703519117D+00
29. 0.9059000000D+00 0.1403819272D+00
30. 0.3643000000D+00 0.5986836897D+00
31. 0.1285000000D+00 0.3953887950D+00
32. s 10 1.00
33. 0.8236000000D+04 0.0000000000D+00
34. 0.1235000000D+04 0.0000000000D+00
35. 0.2808000000D+03 0.0000000000D+00
36. 0.7927000000D+02 0.0000000000D+00
37. 0.2559000000D+02 0.0000000000D+00
38. 0.8997000000D+01 0.0000000000D+00
39. 0.3319000000D+01 0.0000000000D+00
40. 0.9059000000D+00 0.1000000000D+01
41. 0.3643000000D+00 0.0000000000D+00
42. 0.1285000000D+00 0.0000000000D+00
43. s 10 1.00
44. 0.8236000000D+04 0.0000000000D+00
45. 0.1235000000D+04 0.0000000000D+00
46. 0.2808000000D+03 0.0000000000D+00
47. 0.7927000000D+02 0.0000000000D+00
48. 0.2559000000D+02 0.0000000000D+00
49. 0.8997000000D+01 0.0000000000D+00
50. 0.3319000000D+01 0.0000000000D+00

51. 0.9059000000D+00 0.0000000000D+00
 52. 0.3643000000D+00 0.0000000000D+00
 53. 0.1285000000D+00 0.1000000000D+01
 54. p 5 1.00
 55. 0.1871000000D+02 0.1403099838D-01
 56. 0.4133000000D+01 0.8686598995D-01
 57. 0.1200000000D+01 0.2902159664D+00
 58. 0.3827000000D+00 0.5010079421D+00
 59. 0.1209000000D+00 0.3434059603D+00
 60. p 5 1.00
 61. 0.1871000000D+02 0.0000000000D+00
 62. 0.4133000000D+01 0.0000000000D+00
 63. 0.1200000000D+01 0.0000000000D+00
 64. 0.3827000000D+00 0.1000000000D+01
 65. 0.1209000000D+00 0.0000000000D+00
 66. p 5 1.00
 67. 0.1871000000D+02 0.0000000000D+00
 68. 0.4133000000D+01 0.0000000000D+00
 69. 0.1200000000D+01 0.0000000000D+00
 70. 0.3827000000D+00 0.0000000000D+00
 71. 0.1209000000D+00 0.1000000000D+01
 72. d 1 1.00
 73. 0.1097000000D+01 0.1000000000D+01
 74. d 1 1.00
 75. 0.3180000000D+00 0.1000000000D+01
 76. f 1 1.00
 77. 0.7610000000D+00 0.1000000000D+01
 78.
 79. 2 0
 80. s 5 1.00
 81. 0.3387000000D+02 0.6067999217D-02
 82. 0.5095000000D+01 0.4530799416D-01
 83. 0.1159000000D+01 0.2028219738D+00
 84. 0.3258000000D+00 0.5039029350D+00
 85. 0.1027000000D+00 0.3834209505D+00
 86. s 5 1.00
 87. 0.3387000000D+02 0.0000000000D+00
 88. 0.5095000000D+01 0.0000000000D+00
 89. 0.1159000000D+01 0.0000000000D+00
 90. 0.3258000000D+00 0.1000000000D+01
 91. 0.1027000000D+00 0.0000000000D+00
 92. s 5 1.00
 93. 0.3387000000D+02 0.0000000000D+00
 94. 0.5095000000D+01 0.0000000000D+00
 95. 0.1159000000D+01 0.0000000000D+00
 96. 0.3258000000D+00 0.0000000000D+00
 97. 0.1027000000D+00 0.1000000000D+01
 98. p 1 1.00
 99. 0.1407000000D+01 0.1000000000D+01
 100. p 1 1.00
 101. 0.3880000000D+00 0.1000000000D+01

102. d 1 1.00
 103. 0.1057000000D+01 0.1000000000D+01
 104.
 105. 3 0
 106. s 5 1.00
 107. 0.3387000000D+02 0.6067999217D-02
 108. 0.5095000000D+01 0.4530799416D-01
 109. 0.1159000000D+01 0.2028219738D+00
 110. 0.3258000000D+00 0.5039029350D+00
 111. 0.1027000000D+00 0.3834209505D+00
 112. s 5 1.00
 113. 0.3387000000D+02 0.0000000000D+00
 114. 0.5095000000D+01 0.0000000000D+00
 115. 0.1159000000D+01 0.0000000000D+00
 116. 0.3258000000D+00 0.1000000000D+01
 117. 0.1027000000D+00 0.0000000000D+00
 118. s 5 1.00
 119. 0.3387000000D+02 0.0000000000D+00
 120. 0.5095000000D+01 0.0000000000D+00
 121. 0.1159000000D+01 0.0000000000D+00
 122. 0.3258000000D+00 0.0000000000D+00
 123. 0.1027000000D+00 0.1000000000D+01
 124. p 1 1.00
 125. 0.1407000000D+01 0.1000000000D+01
 126. p 1 1.00
 127. 0.3880000000D+00 0.1000000000D+01
 128. d 1 1.00
 129. 0.1057000000D+01 0.1000000000D+01
 130.
 131. 4 0
 132. s 5 1.00
 133. 0.3387000000D+02 0.6067999217D-02
 134. 0.5095000000D+01 0.4530799416D-01
 135. 0.1159000000D+01 0.2028219738D+00
 136. 0.3258000000D+00 0.5039029350D+00
 137. 0.1027000000D+00 0.3834209505D+00
 138. s 5 1.00
 139. 0.3387000000D+02 0.0000000000D+00
 140. 0.5095000000D+01 0.0000000000D+00
 141. 0.1159000000D+01 0.0000000000D+00
 142. 0.3258000000D+00 0.1000000000D+01
 143. 0.1027000000D+00 0.0000000000D+00
 144. s 5 1.00
 145. 0.3387000000D+02 0.0000000000D+00
 146. 0.5095000000D+01 0.0000000000D+00
 147. 0.1159000000D+01 0.0000000000D+00
 148. 0.3258000000D+00 0.0000000000D+00
 149. 0.1027000000D+00 0.1000000000D+01
 150. p 1 1.00
 151. 0.1407000000D+01 0.1000000000D+01
 152. p 1 1.00


```

153. 0.3880000000D+00 0.1000000000D+01
154. d 1 1.00
155. 0.1057000000D+01 0.1000000000D+01
156.
157. 5 0
158. s 5 1.00
159. 0.3387000000D+02 0.6067999217D-02
160. 0.5095000000D+01 0.4530799416D-01
161. 0.1159000000D+01 0.2028219738D+00
162. 0.3258000000D+00 0.5039029350D+00
162. 0.1027000000D+00 0.3834209505D+00
163. s 5 1.00
164. 0.3387000000D+02 0.0000000000D+00
165. 0.5095000000D+01 0.0000000000D+00
166. 0.1159000000D+01 0.0000000000D+00
167. 0.3258000000D+00 0.1000000000D+01
168. 0.1027000000D+00 0.0000000000D+00
169. s 5 1.00
170. 0.3387000000D+02 0.0000000000D+00
171. 0.5095000000D+01 0.0000000000D+00
172. 0.1159000000D+01 0.0000000000D+00
173. 0.3258000000D+00 0.0000000000D+00
174. 0.1027000000D+00 0.1000000000D+01
175. p 1 1.00
176. 0.1407000000D+01 0.1000000000D+01
177. p 1 1.00
178. 0.3880000000D+00 0.1000000000D+01
179. d 1 1.00
180. 0.1057000000D+01 0.1000000000D+01

```

The difference between the atomic and molecular code can be seen in lines 2-7 where the information of all the atoms is displayed on their own. Then lines 9, 79, 105, 131 and 157, show the atom that the information below the line is dealing with. For example, line 9 has the number 1, meaning information about the first atom is displayed. According to lines 2-7, atom number 1 is carbon. Then line 79, says that the information displayed is for atom 2, and so on. Each When calculating the wavefunction of each atom, the position of each atom needs to be taken into account, which is why a different script was created for the calculation of the molecular orbitals, since in the case of a single atom, it is always found at the origin.

4 How to combine data to form wavefunctions

Let us try to use the data from section 3.1 to form the wavefunction for an atomic orbital $1s^2$, specified in lines 77-115. The first thing we notice is that there are 35 contracted coefficients (lines 77-115) and 58 primitives (lines 6-73). So how exactly are they combined to form the wavefunction for the orbital?

Every contraction coefficient, (lines 81-115) corresponds to a collection of primitives. The contraction Gaussians are also shown in lines 6, 17, 28, 39, 50, 56, 62, 68, 70 and 72, with each one having a symmetry s , p , d or f and each one having a number of corresponding primitives. For example there are 10 primitives for the contracted Gaussian specified in lines 6, 17, 28 and 39, 5 primitives for the contracted Gaussians of lines 50, 56 and 62 and 1 primitive for the contracted Gaussians of lines 68, 70 and 72. The first contracted coefficient (line 81) corresponds to the first contracted Gaussian (line 6) and multiplies all ten of its primitives. The second contracted coefficient (line 82) corresponds to the second contracted Gaussian (line 17) and multiplies all ten of its primitives. This pattern continues, until we reach the first p orbital in line 50. This orbital is the 5th one, so the 5th contracted coefficient (line 85) corresponds to it. However, since it is a p orbital and its angular momentum number $l = 1$, there are three possible combinations for the i , j and k numbers since $l = i + j + k$. These are $i = 1, j = 0$ and $k = 0$, or $i = 0, j = 1$ and $k = 0$ or $i = 0, j = 0$ and $k = 1$. There are therefore, three contracted coefficients, with the same primitive coefficients and exponents. The three contracted coefficients in lines 85-87, are assigned for the three different contracted Gaussians of line 50.

This is true for all other contracted Gaussians as well, shown in lines 56, 62, 68, 70 and 72. If the symmetry is p , there are three contracted Gaussians are formed, with three coefficients assigned to each, if it is d , there are six contracted Gaussians and if it is f there are 10. Adding all these up, results in total 35 contracted Gaussians, exactly the number of coefficients.

When it comes to the molecular molden file, the same procedure applies.

5 Atomic Script Description

4 In order to extract the data for an atom a bash script was written called 'moldenscriptAtom.sh'. The end goal is to extract all data in a csv format, where each line has the following data for each primitive in the order of i, j, k , primitive exponent *alpha*, primitive weight, x, y, z and contraction coefficient. The x, y and z specify the position of the nucleus.

The script starts by reading the x, y and z position of the atom and extracting it in three txt files on lines 17-19. It then separates the data of the primitive Gaussians and contracted coefficients in two txt files, basis.txt and orbitals.txt. Further simplifications are done until the primitive orbitals are separated further, having the exponents in one file called Exponent-Temp.txt, and the primitive weights in another called WeightTemp.txt. The symmetry of the contracted orbitals and the number of primitives each contracted orbital has is also stored in two different files, called TypeOfOrb.Final.txt and NumOfPrims.Final.txt. At this point, all useful information has been extracted.

Next, the information needs to be modified. In line 79 the while loop, loops through all the primitive exponents. The primitive exponents are separated by the symmetry of the orbital. When a line contains the symmetry of the orbital (if statements, line 95-99) The number of iterations is adjusted accordingly. Then in the next for loop (line 110), the exponents are multiplied as many times as it is needed depending on the symmetry of the orbital. In the same

loop, the primitive weights are also multiplied in the same way, depending on the symmetry of the orbital (line126). This way the final primitive weights and exponents are written in files `Weight_Final.txt` and `Exponent_Final.txt`.

In the next for loop in line 151, work is done on the contracted coefficients and to align them with the correct i , j and k values. The loop scans through the files each contracted Gaussian, and records the symmetry and number of primitives for each. It then proceeds in the next for loop, to copy the values of i , j and k , as many times as it is needed to span all primitive Gaussians. The original data for i , j and k are found in the pre-existing files `ijk0.txt`, `ijk1.txt`, `ijk2.txt` and `ijk3.txt` and they have the values in the same order as `molden`. The data are stored in a file `ijk_Final.txt`.

Next the contracted coefficients are altered depending on the number of primitives each contracted Gaussian has. The final data are stored in a file called `ContractedCoeffs_Final.txt`

At this point all data are stored in seven files, the `x.txt`, `y.txt` and `z.txt` where all the position are found, and the `ijk_final.txt`, `Exponent_Final.txt`, `Weight_Final.txt`, `ContractedCoeffs_Final.txt`. The last four should have equal number of lines (excluding possible empty lines at the end of the file). The data are combined and exported in a csv format. The final line `'sed -E 's/([+-]?[0-9.]+)[D]\+?(-?)([0-9]+)/\1E\2\3/g' $1.csv >temp'` changes all values from `1.213D+01` to `1.213E+01`, so that mathematica can recognise it.

Wherever there are manual changes that need to take place, they are found right after a comment starting with `#TODO:` followed by the change that needs to take place. At the time of writing this report only such occasion occurs, in line 38. Other comments should also be useful.

6 Molecular Script Description

In the case of the molecular script, the same procedures take place. The difference however is found in the position of the atoms. In the case of the atomic script, position was standard, where as for molecules it is different for each atom. In the first lines (13-34), the positions are taken from the `molden` file and extracted in 3 files `x_initial.txt`, `y_initial.txt` and `z_initial.txt`. Then in the first for loop in line 78, the primitive exponents and weights of all the atoms are collected in two files called `Exponents_Together.txt` and `Weights_Together.txt`.

In the next for loop in line 130 the data for the x , y and z positions are altered. In the loop, it is calculated depending on both the symmetry (s , p , d and f) as well as the number of primitives in each atom, how many times the position of each atomic nucleus will be multiplied. The final data are stored in three files, `x.txt`, `y.txt` and `z.txt`.

The rest of the code is identical to the atomic code, adjusting the primitive exponents, primitive weights and finally the contraction coefficients. Once again, anything that needs manual change is found right after a comment starting with `#TODO:.` There are five such instances at the time of writing this report

7 CH₄ normalisation tests and overlap integrals

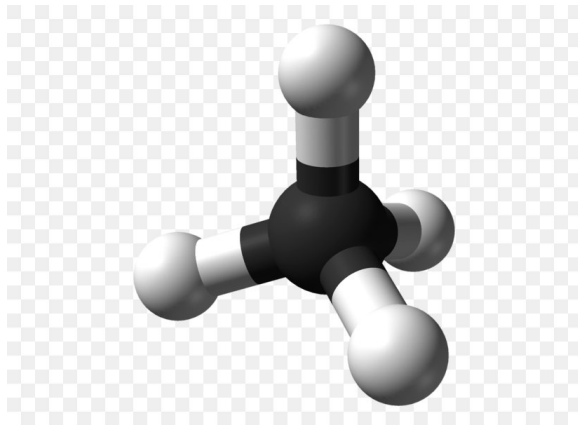
Using the above scripts, we have calculated the wavefunctions for the neutral atoms of C and H, as well as for neutral CH₄. The electronic configuration for these three atoms/molecule are the following:

- H - $1s^1$
- C - $1s^2 2s^2 2p^2$
- CH_4 - $(1\alpha_1)^2 (2\alpha_1)^2 (1t_{2x})^2 (1t_{2y})^2 (1t_{2z})^2$

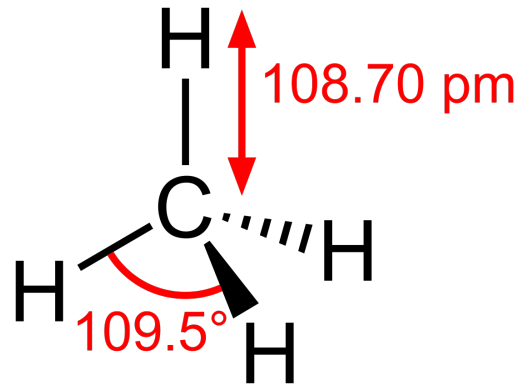
The energy for an electron to ionise from each orbital of CH_4 is the following

- $(1\alpha_1)^2$ - 11.2058 au
- $(2\alpha_1)^2$ - 0.9437 au
- $(1t_{2x})^2$ - 0.5454 au
- $(1t_{2y})^2$ - 0.5452 au
- $(1t_{2z})^2$ - 0.5452 au

CH_4 is a tetrahedral molecule belonging to the T_d point group. The shape of the molecule is shown in figures 1a and 1b.



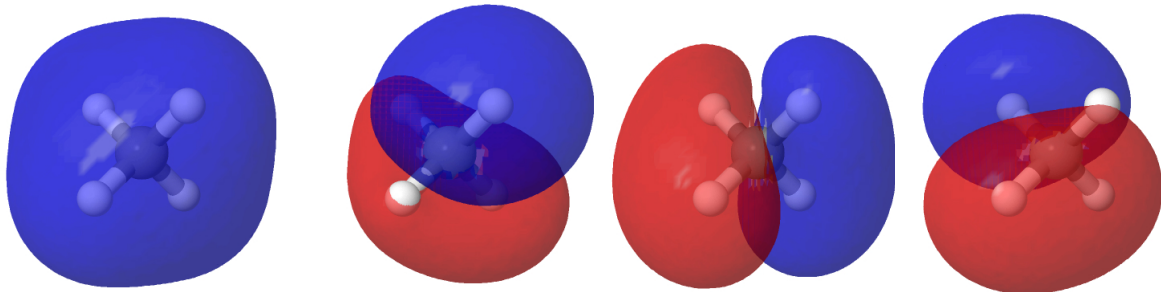
(a) 3d model of methane



(b) Atoms and dimensions of methane molecule

Figure 1: Shape of CH_4 molecule

Next we show the shape of the orbitals



(a) Shape of molecular orbitals a_1 . (b) Shape of molecular orbitals t_{2x} . (c) Shape of molecular orbitals t_{2y} . (d) Shape of molecular orbitals t_{2z} .

Figure 2: Shapes of the orbitals of CH_4 . The a_1 orbitals are shown in figure (a) and the three degenerate T_2 orbitals in the x, y and z directions are shown in figures (b), (c) and (d).

The innermost orbital was checked for H and the two innermost orbitals were checked for C and CH₄ shown above. The normalisation of all orbitals was found to be equal to 1.

The overlap was calculated between them as well. Below are the findings

- $C_{1s} - H_{1s} = 0.37503$
- $CH4_{1\alpha_1} - C_{1s} = 0.999996$
- $CH4_{1\alpha_1} - H_{1s} = 0.377356$
- $CH4_{2\alpha_1} - C_{2s} = 0.976755$