Universidad Autónoma del Estado de Morelos

Instituto de Investigación en Ciencias Básicas y Aplicadas

SIMULACIÓN MONTE CARLO USANDO EL CAMPO DE FUERZA POLARIZABLE AMOEBA UN PROGRAMA EN FORTRAN90

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Licenciado en Ciencias
(Bioquímica y Biología Molecular)

PRESENTA:

Froylán Oswaldo Martínez Vázquez

DIRECTOR DE TESIS:
Dr. Minhhuy Hô

Cuernavaca, Mor., 9 de noviembre de 2020

A la Universidad Autónoma del Estado de Morelos y la Facultad de Ciencias
Por la formación que me han dado.
A mi madre Rosario. V. H. Mis profesores, mi amigo Mario Andres. L. B
Es gracias a ustedes que es posible el presente trabajo.
En verdad, gracias.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de está tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en está o cualquier otra universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Froylán Oswaldo Martínez Vázquez. Cuernavaca, Mor., 9 de noviembre de 2020

Resumen

El campo de fuerzas AMOEBA09, define una superficie de energía potencial como la suma de términos intramoleculares e intermoleculares.

$$\mathrm{U}_{total} = \mathrm{U}_{bond} + \mathrm{U}_{angle} + \mathrm{U}_{b\theta} + \mathrm{U}_{oop} + \mathrm{U}_{\phi} + \mathrm{U}_{vdW} + \mathrm{U}_{ele}^{perm} + \mathrm{U}_{ele}^{ind}$$

cada uno de estos terminos es particular del campo AMOEBA09. Los parámetros con los que dispone son muy robustos, lo cual permite estudiar un gran número de móleculas, proteínas, y cristales. En este trabajo implementamos los primeros cinco terminos de energía correspondiente a la contribución intramolecular. Desarrollamos algoritmos de conectividad para la construcción de cinco redes: *red enlace*, *red ángulo*, *red torsión*, *red de acoplamiento*, y finalmente la *red impropio*, es decir cada red se refiere a la pareja de átomos enlazados, triada de átomos formando un ángulo o a la cuarteta de átomos que forman un ángulo diedro con ello a partir de la geometría molécular incluyendo sus contribuciones energéticas de cada tipo de red. El programa **M**ecánica **M**olecular **F**ina incluye subrutinas y funciones divididas en 13 módulos escritos en lenguaje Fortran90. Probamos las moléculas: *amoniaco*, *indol*, *eter*, *benzamidina* y *nitrilo* obteniendo resultados consistentes a los resportados en la literatura mostrando que la implementación es correcta.

Un simple algoritmo de simulación Monte Carlo basado en este campo de fuerzas fue implementado. Este algoritmo consiste en 6 pasos: generar un esqueleto de carbono de manera aleatoria; agregar los átomos de hidrógeno; convertir la matriz Z a las coordenadas cartesianas con parámetros apropiados; evaluar la energía del campo de fuerzas MMF; probar la condición de Metropolis.

Para probar este algoritmo, las simulaciones de tres geometrías iniciales de la molécula de pentano fueron realizadas. Para cada geometría se evaluarón cinco tipos de energía sumando sus contribuciones obteniendo así la energía total inicial de cada una de estas, posteriormente se hizo un cambio aleatorio al ángulo diedro modificando su geometría, con esta nueva geometría tenemos una segunda energía total y podemos aplicar el método de Monte Carlo y el algoritmo de metropolis usando la condición de Boltzmann para encontrar la energía minima. Se presenta una breve discusión de la implementación y los resultados de las simulaciones.

Índice general

Índice de figuras

Índice de cuadros

Índice de Algoritmos

Índice de Códigos

Capítulo 1

Introducción

1.1. Presentación

La computadora es probablemente la invención más importante del siglo XX además es un instrumento indispensable para desarrollarse en el ámbito científico, permitiendo que surgan nuevas disciplinas como la química, física, y biología computacional, con ello la integración de métodos numéricos a partir de la teoría permite generar un gran número de *Software* científico. Sin embargo estos omiten la mayor parte del procedimiento por esta razón es importante conocer el método, para generar los datos correspondientes particularmente en el campo de la química computacional un método muy conocido es la mecánica molecular.

> «Debemos aprender no sólo a usar programas sino a hacerlos.»
> *Douglas Rushkoff.*

El método de mecánica molecular (MM) permite describir los cambios de energía que ocurren en las moléculas cuando modifican su geometría dichos cambios resultan de variar la longitud de enlaces, ángulos y propiedades electrostáticas generando una *Energía* comúnmente conocida como *superficie de energía potencial* (SEP) con ello podemos evaluar la estabilidad asociada a cada una de ellas, es decir a partir de una función de energía potencial simple se puede conocer y simular cualquier sistema molecular. La suma de estas energías potenciales es lo que se conoce como campo de fuerzas.

Una gran variedad de campos de fuerzas ha surgido a través del tiempo adaptando el mayor número de contribuciones intramoleculares e intermoleculares como la polarizabilidad una propiedad electrostática que redistribuye la carga de la molécula. Es un efecto que incluyen pocos campos de fuerzas además existen parámetros físicos, propiedades atómicas y estructurales característicos de cada campo de fuerzas, la asignación de estos es un proceso que se conoce como *Parametrización*. Es una elección de datos experimentales o teóricos que definen valores ideales para las longitudes de enlace, ángulos y constantes, esto implica que se necesitan suficientes datos para *Parametrizar* un campo de fuerzas.

El campo de fuerzas AMOEBA (Atomic Multipole Optimized Energetics for Biomolecular Applicationes) incorpora los parámetros mencionados anteriormente y es ampliamente utilizado para sistemas biológicos.

1.2. Planteamiento del problema

Como primer paso para implementar el campo de fuerzas en particular AMOEBA09, es necesario evaluar cada una de las contribuciones energéticas, entender el método de MM, aspectos geométricos, métodos numéricos y el lenguaje de programación Fortran además de la técnica de simulación Monte Carlo (MC). La implementación de este código dará origen a un programa mucho más didáctico con una estructura que permita abordar de manera más sencilla y directa los métodos para calcular distancias ángulos, y energías de cada interacción, descritas en ecuaciones del campo de fuerzas AMOEBA09 (2.2).

1.3. Hipótesis

La implementación de un campo de fuerzas se adquiere un conocimiento profundo de simulaciones, el cual nos permite continuar un posgrado con bases sólidas en el área, manejando las herramientas necesarias.

1.4. Justificación

Comprender las interacciones descritas por el campo de fuerza AMOEBA09, es esencial, para continuar en el área de simulación molecular ya que la mayoría de programas utilizados actualmente omiten estos conceptos básicos que son la base de su diseño por ello una manera directa y eficiente es implementar un programa del campo de fuerzas en particular AMOEBA09, el cual es uno de los más recientes y robustos que existe actualmente.

1.5. Objetivo

Implementar un código en Fortran90 para el campo de fuerzas AMOEBA09, entender las interacciones de la mecánica molecular, probar el programa con los valores de la literatura e implementar la técnica de simulación Monte Carlo.

Capítulo 2

Marco teórico

2.1. Mecánica Molecular

La mecánica es una parte de la física que describe el movimiento de una partícula o un cuerpo y sus leyes fueron establecidas por Issac Newton, estas sirven para describir muy bien los cuerpos de gran tamaño, sin embargo para partículas muy pequeñas (moléculas) estas leyes no son aplicables. Investigaciones posteriores y la curiosidad observada en estos fenómenos dieron origen a lo que actualmente se conoce como la teoría cuántica, fundamental para describir las interacciones en el nivel molecular, pero no todo es como parece a pesar que la teoría cuántica describe muy bien las interacciones. Por ejemplo los sistemas biológicos que son parte central de muchos estudios están formados por cientos de átomos y cada uno de ellos por sus partículas correspondientes lo cual representa un problema debido al gran número de partículas ya que en un modelo cuántico las interacciones se toman sin hacer referencia a "enlaces químicos" considerando partículas fundamentales como los núcleos y eléctrones. Esto representa un costo elevado en tiempo y recurso de computo una limitación, para superar esta barrera computacional recurrimos a la *Mecánica Molecular*.

La Mecánica Molecular es una alternativa a la Mecánica Cuántica, un modelo clásico donde los átomos se toman como un conjunto de cargas puntuales, que interactúan entre si bajo un *Campo de Fuerzas*. Una simplificación muy poderosa donde las partículas elementales y sus parámetros están implícitos.

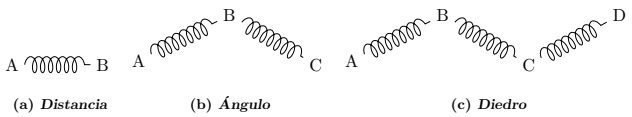
2.1.1. Importancia de la mecánica molecular

La MM nos permite estudiar moléculas de manera más práctica, al considerar "interacciones moleculares", (enlaces) con ello evaluar un número mayor de átomos sin hacer referencia a electrones y núcleos directamente, tomando en cuenta que cada molécula tiene una geometría y esta puede evaluarse con una ecuación de un potencial podemos entonces definir tal energía potencial para cada interacción en cualquier sistema molecular. Ademas existen muchos potenciales cada uno con sus propios parámetros y

constantes que se incluyen en diversos *Campos de Fuerzas*. Así la aplicación del método de MM combinado con técnicas de simulación como *Dinámica Molecular* y *Monte Carlo* son herramientas importantes para el estudio de procesos farmacéuticos, enzimáticos, de ingeniería química y la mayoría de procesos biológicos. Algunas propiedades importantes como la HFE *Hydration free energy* utilizada para evaluar la solubilidad molecular, Shi et al. (9) procesos enzimáticos implicados en la mayoría de sistemas biológicos participando activamente en las reacciones que pueden ser modeladas combinados métodos híbridos de (QM/MM). Mulholland (6) La comprensión de la acción enzima-sustrato en un nivel atómico necesariamente divide al sistema en dos regiones la región interior (región I) que contiene el sitio activo y la región que rodea la proteína (región II). Warshel (11). La región I implica el movimiento de electrones por esta razón se le da un tratamiento QM mientras que la región II se trata por MM. Ahora ¿cómo surge la idea de simular estos procesos? La idea de simular las propiedades de cualquier sistema molecular a partir de una función de energía potencial según describe Levit (5), John Kendrew habia escuchado de Shneior Lifson esta idea, por lo cual Kendrew envió a Levit al instituto Weizman en Israel para que trabajaran juntos en un programa que permitiera calcular la energía con respecto a las posiciones atómicas programa al que llamaron (CFF) *Consistent Force Field* el programa original escrito por Lifson y Warshel contribuyo a la próxima generación de campos de fuerzas como CHARM *Chemistry at HARvard Molecular Mechanics* en el grupo de Karplus en Harvard AMBER *Assisted Model Building with Energy Refinement* del grupo de Peter Kollman en la Universidad de California. OPLS *Optimized Potential for Liquid Simulations* desarrollado por Jorgensen en 1998 haciendo un énfasis en interacciones de no enlace para comparar estados termodinamicamente líquidos. GROMOS *GROningen MOlecular Simulation* desarrollado en 1978 Universidad de Harvard (USA) Universidad Groningen (Holanda) para el modelado de biomoléculas, el cual incluye un paquete para simulación dinámica que incorpora proteínas y ácidos nucléicos. Walter R. P Scott (10) Un gran número de campos de fuerza ha surgido a través de los años con aplicaciones en compuestos orgánicos, organometálicos, compuestos de coordinación, metales de transición, así como en la simulación de proteínas, estos trabajos son el punto de partida para el desarrollo de funciones de energía potencial. Baker (1)

2.1.2. ¿Qué es un Campo de Fuerzas?

Un *Campo de Fuerzas* consiste de una serie de ecuaciones para evaluar una superficie de energía potencial, (SEP) cada ecuación evalua una variable geometríca, figura (2.1) *enlaces*,(estiramiento) *ángulos*,(flexión, torsión, fuera de plano), *enlace–ángulo*(acoplamiento), contribuciones intramoleculares, además de interaciones intermoleculares *van der Waals*, *Polarizabilidad*. En otras palabras un *Campo de Fuerzas* esta en función de variables geometrícas que resultan en una energía potencial. Levit (5)



(a) *Distancia* (b) *Ángulo* (c) *Diedro*

Figura 2.1: Variables geométricas

2.2. Campo de fuerzas AMOEBA09

El campo de fuerzas AMOEBA09, define la energía como la suma de contribuciones intramoleculares e intermoleculares que incluyen enlaces, ángulos de enlace, acoplamiento enlace–ángulo, deformación del ángulo fuera de plano, rotación del ángulo diedro, términos electrostáticos de tipo van der Waals así como cargas permanentes e inducidas, estas características de AMOEBA09 constituyen un modelo polarizable. Ponder (7) Como se muestra a continuación.

$$
\begin{aligned}
U_{\text{total}} &= U_{\text{bond}} + U_{\text{angle}} + U_{b\theta} + U_{\text{oop}} + U_{\text{torsion}} + U_{\text{vdW}} + U_{\text{ele}}^{\text{perm}} + U_{\text{ele}}^{\text{ind}} \qquad (2.1)\\
U_{\text{bond}} &= K_b(b-b_0)^2[1 - 2.55(b-b_0) + \left(\tfrac{7}{12}\right)(2.55)(b-b_0)^2]\\
U_{\text{angle}} &= K_\theta(\theta-\theta_0)^2[1 - 0.014(\theta-\theta_0) + 5.6\cdot 10^{-5}(\theta-\theta_0)^2\\
&\quad + \ 7.0\cdot 10^{-7}(\theta-\theta_0)^3 + 2.2\cdot 10^{-8}(\theta-\theta_0)^4]\\
U_{b\theta} &= K_{b\theta}\left[(b-b_0) + (b'-b'_0)\right](\theta-\theta_0)\\
U_{\text{oop}} &= K_{\text{oop}}\chi^2\\
U_{\text{torsion}} &= \sum_{n=1}^{3} K_{n\phi}\left[1 + \cos(n\phi \pm \delta)\right]\\
U_{\text{vdw}} &= \epsilon_{ij}\left[\frac{1.07}{\rho_{ij}+0.07}\right]^7\left[\frac{1.12}{\rho_{ij}^7+0.12} - 2\right]
\end{aligned}
$$

Vamos a omitir las definiciones de las energías ya que no es parte de nuestra implementación.

2.2.1. Ecuaciones de AMOEBA09

El campo AMOEBA09 analiza las fuerzas que actúan para deformar los enlaces, ángulos, ángulos fuera de plano, torsión, interacción entre términos cruzados y términos electrostáticos esto produce una energía potencial [1] que resulta de la suma de ocho ecuaciones (2.1) las primeras cinco energías se describen a continuación.

[1] Cada campo de fuerzas tiene constantes y parámetros diferentes

2.2.1.1. Energía de enlace

La energía de estiramiento covalente para los átomos enlazados a través de la distancia $(b - b_0)$ donde b_0 es la distancia de enlace ideal, b la distancia de enlace actual y K_b es una constante de fuerza en el equilibrio, se evalúa con la siguiente ecuación esta energía es expresada como un potencial armónico con contribuciones de tercer y cuarto orden para la anarmonicidad.

$$U_{\text{bond}} = K_b(b - b_0)^2[1 - 2.55(b - b_0) + \left(\tfrac{7}{12}\right)(2.55)(b - b_0)^2] \tag{2.2}$$

2.2.1.2. Energía de ángulo

La energía para deformar un ángulo de valencia a un valor θ diferente desde un valor de referencia ideal θ_0 y una constante de fuerza K_θ se determina con la ecuación que aparece abajo esta expresión incluye término de tercer, cuarto, quinto y sexto orden para la anarmonicidad.

$$\begin{aligned} U_{\text{angle}} = K_\theta(\theta - \theta_0)^2[1 - 0.014(\theta - \theta_0) + 5.6 \cdot 10^{-5}(\theta - \theta_0)^2 \\ + \ 7.0 \cdot 10^{-7}(\theta - \theta_0)^3 + 2.2 \cdot 10^{-8}(\theta - \theta_0)^4] \end{aligned} \tag{2.3}$$

2.2.1.3. Energía acoplada

La interacción entre movimientos como la flexión del ángulo y el estiramiento de enlaces, es un fenómeno donde la deformación angular $(\theta - \theta_0)$ produce un cambio en la longitud de los enlaces $(b - b_0)$ y $(b' - b'_0)$ que forman el ángulo.

$$U_{b\theta} = K_{b\theta}\left[(b - b_0) + (b' - b'_0)\right](\theta - \theta_0) \tag{2.4}$$

2.2.1.4. Energía fuera de plano

Una distorsión que cambia la energía ocurre en moléculas planas con centros trigonales en ellas el átomo central se desplaza por encima o por debajo del plano, esto genera un ángulo χ^2 asociado a una fuerza K_{oop}, que puede evaluarse con la siguiente ecuación.

$$U_{\text{oop}} = K_{\text{oop}}\chi^2 \tag{2.5}$$

2.2.1.5. Energía de torsión

La energía de torsión se presenta cuando cuatro átomos unidos consecutivamente se disponen en dos planos formando el ángulo de torsión ϕ asociado a la constante $K_{n\phi}$ y una fase δ

$$U_{\text{torsion}} = \sum_{n=1}^{3} K_{n\phi}\left[1 + \cos(n\phi \pm \delta)\right] \tag{2.6}$$

2.2.1.6. Energía de van der Waals

La energía de van der Waals U_{vdW} de pares se evaluar por la forma de amortiguador de 14–7 donde ϵ_{ij} es la profundidad del pozo de la energía potencial

$$\epsilon_{ij} = \frac{4\epsilon_{ii}\epsilon_{jj}}{(\epsilon_{ii}^{1/2} + \epsilon_{jj}^{1/2})^2} \tag{2.7}$$
$$U_{\text{vdw}} = \epsilon_{ij}\left[\frac{1.07}{\rho_{ij} + 0.07}\right]^7 \left[\frac{1.12}{\rho_{ij}^7 + 0.12} - 2\right]$$

y $\rho_{ij} = R_{ij}/R_{ij}^0$ donde R_{ij} es la distancia entre el átomo i y j; R_{ij}^0 es la distancia de energía mínima.

2.3. Obtener parámetros geométricos

Para evaluar la energía es necesario conocer la posición de cada átomo presente en las moléculas, su conexión y ángulos dependen de la geometría. En un modelo de **MM** la distancia que existe entre dos puntos determina los enlaces y ángulos que forman entre los átomos a continuación se describe como calcular estos parámetros geométricos.

2.3.1. Distancia

La longitud de un vector $\mathbf{r}$ a menudo se denomina **norma** de $\mathbf{r}$ y se denota por $\|\mathbf{r}\|$. Si $\mathbf{r}_1(x_1, y_1, z_1)$ y $\mathbf{r}_2(x_2, y_2, z_2)$ son vectores en $\mathbf{R}^3$.

El vector $\mathbf{r_1r_2}$ $(x_2 - x_1, y_2 - y_1, z_2 - z_1)$ con distancia d

$$d = \sqrt{\mathbf{r_1r_2} \cdot \mathbf{r_1r_2}}$$
$$\|\mathbf{r}_1\mathbf{r}_2\| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{2.8}$$

La ecuación (2.8) define la distancia que hay entre dos puntos.

2.3.2. Ángulo

El ángulo entre tres puntos $\mathbf{r}_1(x_1, y_1, z_1)$, $\mathbf{r}_2(x_2, y_2, z_2)$ y $\mathbf{r}_3(x_3, y_3, z_3)$ se puede calcular si se define dos vectores $\mathbf{u}$ y $\mathbf{v}$ vectores en $\mathbf{R}^3$ como se muestra a continuación.

Para el vector $\mathbf{u}$

$$\mathbf{u} = \mathbf{r}_2 - \mathbf{r}_1$$
$$\mathbf{u} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$
$$\|\mathbf{u}\| = \sqrt{u_1^2 + u_2^2 + u_3^2}$$

y el vector $\mathbf{v}$

$$\begin{aligned}\mathbf{v} &= \mathbf{r}_3 - \mathbf{r}_2 \\ \mathbf{v} &= (x_3 - x_2, y_3 - y_2, z_3 - z_2) \\ \|\mathbf{v}\| &= \sqrt{v_1^2 + v_2^2 + v_3^2}\end{aligned}$$

Posteriormente hay que determinar su producto punto

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + u_3 v_3$$

y finalmente el ángulo

$$\theta = \cos^{-1} \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \ \|\mathbf{v}\|} \tag{2.9}$$

2.3.3. Torsión

Si cuatro puntos $\mathbf{p}_1(x_1, y_1, z_1)$ $\mathbf{p}_2(x_2, y_2, z_2)$ $\mathbf{p}_3(x_3, y_3, z_3)$ $\mathbf{p}_4(x_4, y_4, z_4)$ se encuentran en planos distintos tal que $\mathbf{p}_1$, $\mathbf{p}_2$ y $\mathbf{p}_3$ están en un plano, y $\mathbf{p}_2$, $\mathbf{p}_3$ y $\mathbf{p}_4$, en otro plano se puede conocer el ángulo ϕ de torsión a partir de estos cuatro puntos, definiendo los vectores $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$.

$$\begin{aligned}\mathbf{a} &= \mathrm{P}_1 - \mathrm{P}_2 \\ \mathbf{b} &= \mathrm{P}_3 - \mathrm{P}_2 \\ \mathbf{c} &= \mathrm{P}_2 - \mathrm{P}_3 \\ \mathbf{d} &= \mathrm{P}_4 - \mathrm{P}_3\end{aligned}$$

ahora generamos dos vectores $\mathbf{n_1}$ y $\mathbf{n_2}$

$$\mathbf{n_1} = \mathbf{a} \times \mathbf{b} \qquad \mathbf{n_2} = \mathbf{c} \times \mathbf{d}$$

$$\mathbf{n_1} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \qquad \mathbf{n_2} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ c_1 & c_2 & c_3 \\ d_1 & d_2 & d_3 \end{vmatrix}$$

$$\mathbf{n_3} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ n_1 & n_1 & n_1 \\ n_2 & n_2 & n_2 \end{vmatrix}$$

los cuales se encuentran en planos distintos unidos por **r**

$$\begin{aligned}\mathbf{r} &= \sqrt{b \cdot b} \\ \mathbf{x} &= \mathbf{n_3} \cdot \mathbf{b} \\ \mathbf{y} &= \mathbf{n_1} \cdot \mathbf{n_2} \\ &= \tan^{-1} \frac{\mathbf{x}}{\mathbf{y} \cdot \mathbf{r} \cdot \mathbf{r}} \\ \phi &= \tan^{-1} \frac{\mathbf{n_3} \cdot \mathbf{b}}{\mathbf{n_1} \cdot \mathbf{n_2} \cdot \|\mathbf{b}\|}\end{aligned} \tag{2.10}$$

2.3.4. Ángulo fuera de plano

Desde cuatro puntos se evalúa $\mathbf{a}(x_a, y_a, z_a)$ $\mathbf{b}(x_b, y_b, z_b)$ $\mathbf{c}(x_c, y_c, z_c)$ $\mathbf{d}(x_d, y_d, z_d)$

$$\begin{aligned}\mathbf{ab} &= (x_a - x_b, y_a - y_b, z_a - z_b) \\ \mathbf{cb} &= (x_c - x_b, y_c - y_b, z_c - z_b) \\ \mathbf{db} &= (x_d - x_b, y_d - y_b, z_d - z_b) \\ \mathbf{ad} &= (x_a - x_d, y_a - y_d, z_a - z_d) \\ \mathbf{cd} &= (x_c - x_d, y_c - y_d, z_c - z_d)\end{aligned}$$

$$\begin{aligned}\|\mathbf{ab}\|^2 &= x_{ab}^2 + y_{ab}^2 + z_{ab}^2 \\ \|\mathbf{cd}\|^2 &= x_{cd}^2 + y_{cd}^2 + z_{cd}^2 \\ \|\mathbf{db}\|^2 &= x_{db}^2 + y_{db}^2 + z_{db}^2 \\ \|\mathbf{ad}\|^2 &= x_{ad}^2 + y_{ad}^2 + z_{ad}^2 \\ \|\mathbf{cd}\|^2 &= x_{cd}^2 + y_{cd}^2 + z_{cd}^2 \\ \mathbf{ad} \cdot \mathbf{cd} &= (x_{ad} * x_{cd}) + (y_{ad} * y_{cd}) + (z_{ad} * z_{cd}) \\ \mathbf{r}_{ad,cd} &= \mathbf{ad} \cdot \mathbf{cd}\end{aligned}$$

$$cc = \|\mathbf{ad}\|^2 * \|\mathbf{cd}\|^2 - \mathbf{r}_{adcd} * \mathbf{r}_{adcd}$$

$$\mathbf{ee} = \mathbf{ab} \times \mathbf{cb}$$

$$\mathbf{ee} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_{ab} & y_{ab} & z_{ab} \\ x_{cb} & y_{cb} & z_{cb} \end{vmatrix}$$

$$e = \mathbf{ee} \cdot \mathbf{db}$$

$$\begin{aligned}bkk2 &= \|\mathbf{db}\|^2 - \frac{e * e}{cc} \\ \cos &= \sqrt{\frac{bkk2}{\|\mathbf{db}\|^2}}\end{aligned} \tag{2.11}$$

2.4. El lenguaje de programación Fortran90

FORTRAN es un acrónimo ingles **For**mula **Tran**slating (Traductor de Formulas). Desarrollado por John Backus, antes de graduarse cuando hizo una visita a IBM allí vio la **S**elective **S**equence **E**lectronic **C**alculator (SSEC), IBM lo contrató para dar mantenimiento a la máquina además de escribir programas tarea que resulto ser aún más interesante, permitiéndole desarrollar un primer lenguaje al que llamó SpeedCoding, más adelante IBM desarrollo la máquina 704 dirigida principalmente a científicos y matemáticos pero era difícil de programar y requería mucho tiempo al darse cuenta de esto se propuso diseñar un programa para la 704, que traduciría las formulas del usuario así fue como nació el proyecto FORTRAN uno de los primeros lenguajes de "alto nivel" sencillo, fácil de aprender y ampliamente utilizado en aplicaciones científicas y de ingenieria.

> «Fortran no es una flor sino una mala hierba resistente, florece de vez en cuando y crece en cada computadora»
> *Alan J. Perlis*

2.4.1. Estructura básica

El desarrollo de un programa en Fortran90 requiere escribir un archivo con extensión nombre.f90, usualmente se usa un editor de texto plano donde se incluyen los comandos y sentencias que el compilador traduce para generar un archivo ejecutable, la estructura básica de un programa se muestra a continuación.

```
1 program nombre !Inicio del programa
2 !
3 implicit none  !Declaracion de variables
4 !
5    [Especificacion]
6    [Ejecucion]
7 !
8 end program nombre !Fin del programa
```

Códigos 2.1: Estructura básica.f90

el programa inicia en la línea 1 con la palabra `program` seguida del nombre del programa, posteriormente la declaración `implicit none` línea 3, permite al programador declarar variables explicitamente, el cuerpo del programa inica en la línea 4 y contienen la parte de especificación y ejecución por último el final del programa `end program` línea 8 por razones historicas las variables en FORTRAN de definen:

$$\underbrace{ABCDEFGH}_{REAL}\overbrace{IJKLMN}^{INTEGER}\underbrace{OPQRSTUVWXYZ}_{REAL}$$

2.4.2. Tipo de datos, constantes y variables

Los datos de un programa pueden ser constantes o variables, un valor que no cambia durante su ejecución se considera constante y el que cambia una variable, estos valores son asignados a un tipo de dato y cada uno de ellos con operaciones específicas, el programador tiene la tarea de asignar los valores a un tipo de dato esto se conoce como declaración de variables, en Fortran existen cinco tipos de datos tres de los cuales son numéricos y los dos restantes de tipo lógico y carácter como se describe a continuación.

1. **Numéricos.**
   Este tipo de datos permite asignar variables numéricas

   a) **Enteros:**
   Las variables de tipo entero permiten guardar valores positivos o negativos sin punto decimal.

   ```
   integer          :: num1 = 260, num2 = -260, num3 = 26000
   ```

   b) **Real:**
   Las variables de tipo real permiten guardar valores positivos o negativos con punto decimal.

   ```
   real             :: num1 = 1.1, num2 = -2.60, num3 = 3.1416
   ```

   c) **Complejos.**

   ```
   complex          :: z
   z = cmplx(2,4)
   ```

2. **Lógicos:**
   Estas variables pueden tomar dos valores verdadero o falso.

   ```
   logical          :: verdadero = .true.
   logical          :: falso     = .false.
   ```

3. **Caracteres:**
   Estas variables permiten asignar caracteres.

   ```
   character            :: c
   character(len = 7)   :: cadena = 'cadena'
   ```

```
1  program tipo_de_datos
2  !
3    implicit none
4  !
5    integer  :: base = 1, altura = 2
6    real     :: area
7    complex  :: z
8    complex  :: z1
9    complex  :: z2
10
11   z1 = cmplx(3,2)
```

```
12   z2 = cmplx(4,1)
13   z  = z1 * z2
14 !
15 end program tipo_de_datos
```

Códigos 2.2: Tipos de datos.f90

El archivo 2.3 contiene el código fuente de FORTRAN las líneas 1 a 7 describen su objetivo así como los datos del programador y fecha el carácter "!" indica al compilador que es un comentario, posteriormente la línea 9 y 40 muestran el inicio y fin. En FORTRAN las variables $\{i, j, k, l, m, n\}$ no declaradas tienen un tipo `integer` implícito es común. Entre los programadores agregar la línea 11 `implicit none` para declarar variables explicitamente. La sección de declaración se encuentra entre las líneas 12 y 24 con el tipo de datos permitido `integer`, `real`, `logical`, `character`, `complex` líneas 15 a 18. El cuerpo del programa líneas 25 a 39 contiene la sección de ejecución, cada comando es procesado en orden de aparición así la línea 26 es la primera en ejecutarse pidiendo al usuario introducir dos números que son asignados a las variables a y b. Posteriormente los valores se muestran en pantalla línea 27. Los vectores $\mathbf{v}$ y $\mathbf{w}$ se suman y el resultado es asignado al vector $\mathbf{z}$ línea 30. El proceso continua con las operaciones básicas *suma*, *resta*, *producto* y *potencia* líneas 35, 36, 37 y 38, la ejecución termina cuando encuentra la última línea 40.

```
1  !          ###################################
2  !            Estructura de un programa en F90
3  !          ###################################
4  !
5  !       Fecha          Programador        Descripcion
6  !      ########      ##############      ##############
7  !      25/09/16      F. Oswaldo Mtz      Estructura
8           0          0          0          0          0          0          0          0
9    program Estructura
10 !
11       implicit none
12 !     [Declaracion de Variables]
13 !     ...
14 !     Tipo de variables
15       integer      :: i, j, exponente
16       real         :: a, b, c
17       logical      :: verdadero, falso
18       character    :: caracter
19 !     Arreglos
20
21       real, dimension(4) :: v = (/1.,2.,3.,4./)
22       real, dimension(4) :: w = (/5.,6.,7.,8./)
23       real, dimension(4) :: z
24 !     ...
25 !     ...
26       read(*,*) a, b
27       write(*,*)" Imprimir en panatalla ", a, b
28 !     Ciclos
29       do i = 1, 4
30         z(i) = v(i) + w(i)
31       end do
```

```
32 !
33 !     Operaciones basicas
34 !     ...
35       suma     = a + b
36       resta    = a - b
37       producto = a * b
38       potencia = a ** exponete
39 !     ...
40    end program Estructura
```

Códigos 2.3: Estructura.f90

2.4.3. Subprogramas, procedimientos

La solución de un problema particular generalmente es compleja y carece de solución única ante esta situación conviene dividir el problema. En programación ocurre algo similar. Para llegar a la solución un programa es dividido en pequeños programas encargados de una tarea particular que junto con otros subprogramas permite llegar a la solución general, es decir cada subprograma esta diseñado para una tarea específica. Esto es conocido como programación modular. En Fortran existen dos tipos de subprogramas o procedimientos externos: funciones y subrutinas, algunas ventajas son:

- Programa principal, con tareas resueltas por funciones y subrutinas.
- Independencia de tareas específicas.
- Reusabilidad

2.4.3.1. Funciones

En matemáticas una función es una expresión procedimientos evalúa uno o más valores de entrada y devuelve un único valor. En programación las funciones son procedimientos que pueden recibir como argumento distintos tipos de datos: numérico, lógico, carácter o arreglo; devolviendo un solo valor. A continuación se describe un ejemplo sencillo de una función que calcula la distancia entre dos puntos 5. La función inicia con la sentencia `real`, que indica el tipo de función, seguido por la sentencia `function` y a continuación el nombre de la función en este caso `Distancia` y los argumentos de entrada línea (7) los argumentos de la función deben especificarse de acuerdo al tipo de función 13 la función termina con la sentencia `end function`

```
1 ! Froylán Oswaldo Martínez Vázquez
2 ! Función calcula distancia entre dos puntos
3  p_i(x_i, y_i, z_i) y p_f(x_f, y_f, z_f)
4
5  Distancia = sqrt((x_f - x_i)^2 + (y_f - y_i)^2 + (z_f - z_i)^2)
6
7 real (dp) function Distancia (xi, yi, zi, xf, yf, zf)
8 !
```

```
use numerico
!
implicit none
!
real(dp)            :: xi, yi, zi, xf, yf, zf
!
Distancia = sqrt ((xf-xi)**2 + (yf-yi)**2 + (zf-zi)**2 )
!
end function Distancia
```

Códigos 2.4: FuncionDistancia.f90

2.4.3.2. Subrutinas

Es otro tipo de procedimiento que recibe valores de entrada y devuelve el resultado a través de la lista de argumentos, para iniciar una subrutina se escribe la sentencia `subroutine` seguido por la lista de argumentos en este caso p_i y p_f son los argumentos de entrada y un argumento de salida de **r** línea (22), los argumentos de entrada p_i y p_f se reconocen por el atributo `intent (in)` (25) y el argumento de salida r con el atributo `intent (out)` (26) a continuación se ejecuta la línea (31) para generar el argumento de salida, para llamar nuestra subrutina en el programa `program` sub lo hacemos a través de la palabra reservada `call` línea (18)

```
! Froylán Oswaldo Martínez Vázquez
! Subrutina calcula distancia entre dos puntos
 p_i(x_i, y_i, z_i) y p_f(x_f, y_f, z_f)

 Distancia = sqrt((x_f - x_i)^2 + (y_f - y_i)^2 + (z_f - z_i)^2)
!  Estrcutura de una subrutina

program sub

  implicit none

  real, dimension(3)  :: xi, xf
  real                :: d

  data  xi /1.0, 2.0, 3.0/
  data  xf /4.0, 5.0, 6.0/

  call distancia(xi, xf, d)
  write(*,*) 'Distancia',d
end program sub

subroutine distancia(pi, pf, r)

  implicit none
  real, dimension(3), intent(in)  :: pi, pf
  real, intent(out)               :: r
  real, dimension(3)              :: r2

  r2 = pf - pi
```

```
30   write(*,*) norm2(r2)
31   r = sqrt(dot_product(r2,r2))
32
33 end subroutine distancia
```

Códigos 2.5: SubrutinaDistancia.f90

2.4.3.3. Módulos

Un módulo es una unidad de programa compilado separadamente, el cual contiene datos que pueden compartirse entre unidades de programa. Un módulo comienza con la declaración `module` que asigna el nombre al módulo, para usar los valores declarados dentro de un módulo se debe incluir la sentencia `use` en el programa principal, de esta manera se pueden compartir datos entre módulos y programas, para indicar el fin del módulo la sentencia `end module`, a continuación se da un ejemplo sencillo. Obteniendo el siguiente resultado. Chapman (e.g. 3, p. 322) para compilar el programa seguimos la siguiente instrucción

<compilador> [flags] [−o executable] <codigo fuente>

```
usuario@pc:~$ gfortran Modulo.f90 -o modulo; ./modulo
 3.14159203       6.28318405       9.42477608       12.5663681       15.7079601
```

```
1
2    module shared_data
3    !
4    !  Purpose:
5    !    To declare data to share between two routines.
6
7    implicit none
8    save
9
10   integer, parameter :: NUM_VALS = 5        ! Max number of values in array
11   real, dimension(NUM_VALS) :: values       ! Data values
12
13   end module shared_data
14
15   program test_module
16   !
17   !  Purpose:
18   !    To illustrate sharing data via a module.
19   !
20   use shared_data               ! Make data in module "test" visible
21   implicit none
22
23   real, parameter :: PI = 3.141592  ! Pi
24
25   values = PI * (/ 1., 2., 3., 4., 5. /)
26
27   call sub1                     ! Call subroutine
```

```
28
29    end program test_module
30
31    !*****************************************************************
32    !*****************************************************************
33
34    subroutine sub1
35    !
36    !  Purpose:
37    !    To illustrate sharing data via a module.
38    !
39    use shared_data               ! Make data in module "test" visible
40    implicit none
41
42    write (*,*) values
43
44    end subroutine sub1
```

Códigos 2.6: Modulo.f90

2.4.3.4. Precisión

El tipo de datos REAL (o punto flotante) se usa para representar números que contienen puntos decimales. En la mayoría de las computadoras una variable real predeterminada tiene una longitud de 4 bytes (o 32 bits). Es dividido en dos partes una mantisa y un exponente. Las computadoras modernas usan el IEEE 754 Estándar para variables de punto flotante para implementar números reales. 24 bits del número están dedicados a la mantisa y 8 bits son dedicado al exponente. Los 24 bits dedicados a la mantisa son suficientes para representar de 6 a 7 dígitos decimales significativos por lo que un número real puede tener hasta 7 dígitos significativos de manera similar los 8 bits del exponente son suficientes para representar números tan grandes como 10^{38} y tan pequeño como 10^{-38}

Expresar un número a más de siete dígitos significativos de precisión o para trabajar con números mayores de 10^{38} o menores que 10^{-38}. En cualquier caso, no podemos usar 32 bits. Fortran incluye al menos una versión más larga del tipo de datos reales para su uso en estas circunstancias. La versión más larga del tipo de datos REAL suele tener una longitud de 8 bytes (o 64 bits). En una implementación típica 53 bits del número están dedicados a la mantisa y 11 bits están dedicados al exponente. Los 53 bits dedicados a la mantisa son suficientes para representar de 15 a 16 dígitos decimales significativos del mismo modo los 11 bits del exponente son suficiente para representar números tan grandes como 10^{308} y tan pequeños como 10^{-308}

Una manera de declarar el tipo de precisión que se requiere en compiladores fortran es la variable `kind`, el tipo de valor real se especifica en paréntesis Chapman (3, p. 486)

```
real(kind = 1)
real(kind = 4)
real(kind = 8)
```

Para determinar el tipo de número asociado a la computadora simple precisión y doble precisión Chapman (3, p. 488)

```
1  PROGRAM kinds
2
3  !
4  ! Purpose:
5  !  To determine the kinds of single and double precision real
6  !  values on a particular computer.
7  !
8  IMPLICIT NONE
9  ! Escribe los valores para simple y doble presición
10 WRITE (*,'("The KIND para simple presicion is",I2)') KIND(0.0)
11 WRITE (*,'("The KIND para doble presicion is",I2)') KIND(0.0D0)
12 END PROGRAM kinds
13 selected_real_kind
```

Códigos 2.7: kind.f90

```
usuario@pc:~$ gfortran kind.f90 -o kind; ./kind
KIND para  simple precision is 4
KIND para  doble  precision is 8
```

El parametro `kind` es un entero que depende del procesador

- `real*4` - usa 4 bytes, aproximadamente $\pm 10^{-38}$ a $\pm 10^{38}$.
- `real*8` - usa 8 bytes, aproximadamente $\pm 10^{-308}$ a $\pm 10^{308}$.

La siguiente función especifica el rango y precisión en un procesador particular donde `precision` es el número de dígitos decimales requerido y `range` el exponente requerido en potencias de 10, estos dos argumentos son opcionales,

```
kind_number = SELECTED_REAL_KIND(p=precision,r=range)
```

con la ayuda de tres funciones podemos determinar el valor real de nuestra computadora estas funciones son

- `kind`(X) devuelve el tipo de número de X donde X es una variable o cualquier tipo de dato.
- `precision`(X) devuelve la precisión de X donde X es un valor real o complejo
- `range`(X) devuelve el rango del exponente decimal para X, donde X es un entero, real o valor complejo

Chapman (3, p. 491)

```
1  PROGRAM select_kinds
2  !
3  ! Purpose:
4  !  To illustrate the use of SELECTED_REAL_KIND to select
5  !  desired kinds of real variables in a processor-independent
```

```
6  !  manner.
7  !
8  ! Record of revisions:
9  !  Date Programmer Description of change
10 !  ==== ========== ====================
11 !  11/28/15 S. J. Chapman Original code
12 !
13 IMPLICIT NONE
14 ! Declare parameters:
15 INTEGER, PARAMETER :: SGL = SELECTED_REAL_KIND(p=6,r=37)
16 INTEGER, PARAMETER :: DBL = SELECTED_REAL_KIND(p=13,r=200)
17 ! Declare variables of each type:
18 REAL(kind=SGL) :: var1 = 0.
19 REAL(kind=DBL) :: var2 = 0._DBL
20 ! Write characteristics of selected variables.
21 WRITE (*,100) 'var1', KIND(var1), PRECISION(var1), RANGE(var1)
22 WRITE (*,100) 'var2', KIND(var2), PRECISION(var2), RANGE(var2)
23 100 FORMAT(A,': kind = ',I2,', Precision = ',I2,', Range = ',I3)
24 END PROGRAM select_kinds
```

Códigos 2.8: selecKind.f90

```
usuario@pc:~$ gfortran selectedkind.f90 -o selectedkind; ./selectedkind
var1: kind = 4, Precision = 6, Range = 37
var2: kind = 8, Precision = 15, Range = 307
```

2.4.4. Arreglos

Un arreglo es un grupo de variables y constantes de un mismo tipo, estos valores ocupan ubicaciones consecutivas en la computadora, cada valor es un elemento. Para declarar un arreglo en FORTRAN se hace de la siguiente manera

```
  real, dimension(16) :: miArreglo
  real                :: miArreglo(16)
```

ambas formas son validas pero el atributo `dimension` permite inicializar el tamaño del arreglo (miArreglo), a los 16 elementos de miArreglo se accede miArreglo(1), miArrreglo(2), ... miArreglo(16), un arreglo puede ser de tipo `integer`, `real`, `character` el siguiente ejemplo muestra las caracteristicas de los arreglos:

```
real, dimension(0:20)         :: a
real, dimension(3,0:5,-10:10) :: b
```

Rank: Número de dimensiones

a tiene rango 1 y **b** tiene rango 3

Bounds: Limite superior e inferior de cada dimensión del arreglo.

a tiene límites 0:20 y **b** límites 1:3, 0:5 y -10:10

Extent: Número de elementos en cada dimensión

a tiene 21 y **b** tiene 3,6 y 21

Size: Total del número de elementos

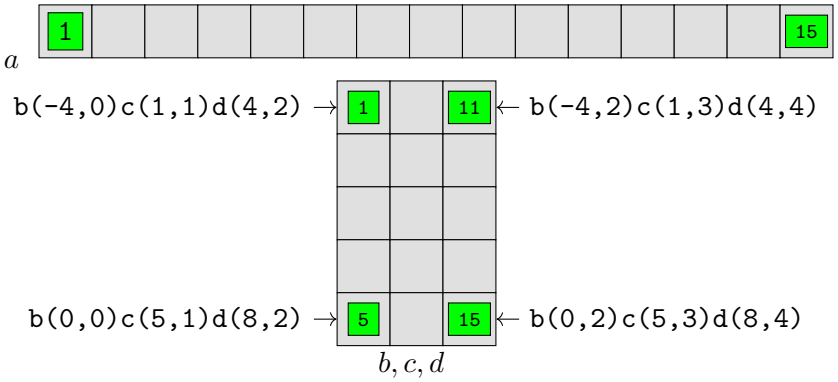
a tiene tamaño 21 y **b** tiene tamaño 30

Shape: La forma del arreglo es el rank y extent

a tiene 21 **b** tiene (3,6,21)

El siguiente ejemplo muestra cuatro arreglos a, b, c, d con el mismo size(15), b, c, d tiene el mismo rank(2) y a tiene rank(1) los bounds(límites) son diferentes para cada arreglo pero el extent (5) y (3) son iguales para b, c, d. Esta diferencia se puede observar mejor representando ambos arreglos, los cuadros verdes muestran el orden de como se guardan los elementos en la memoria, para acceder a cada elemento se hace a través de los indices correspondientes como se muestra al lado de cada cuadro verde para todos los arreglos b, c, d.

```
real,dimension(15) :: a
real,dimension(-4:0,0:2) :: b
real,dimension(5,3) :: c
real,dimension(4:8,2:4) :: d
```



2.4.4.1. Allocatable, allocate, deallocate

El atributo `allocatable` permite declarar un arreglo dinámico, que sera especificado por la sentencia `allocate` al momento de la ejecución, el tipo de arreglo es especificado en la declaración pero no el número de elementos en cada dimensión, la cual se especifica con dos puntos como un marcador de posición. Chapman (e.g. 3, p. 395)

```
REAL, ALLOCATABLE, DIMENSION(:) :: arreglo1
INTEGER, ALLOCATABLE, DIMENSION(:,:,:) :: indices
```

para asignar dinámicamente arreglo1 se utiliza la sentencia `allocate`

```
ALLOCATE (arreglo1(10000))
```

para liberar la memoria asignada se utiliza la siguiente sentencia

```
DEALLOCATE (arreglo1)
```

estas tres sentencias permiten trabajar con arreglos dinámicos en FORTRAN su importancia se manifiesta cuando en nuestro código los enlaces, ángulos diedros cambien de una molécula a otra.

Capítulo 3

Metodología

3.1. Cálculo de energías de AMOEBA

En esta sección se muestran ejemplos numéricos utilizando las ecuaciones 2.2.1 del campo de fuerzas AMOEBA09, este punto es clave porque se calcula las energías manualmente, cumpliendo así un objetivo entender el método de la mecánica molecular ya que cada ecuación consta de parámetros y constantes que deben ser sustituidos para generar el valor correspondiente, es importante mencionar que para conocer los parámetros y constantes nos ayudamos de TINKER versión 6.3.3 que consta de subrutinas para mecánica y dinámica molecular con 135,000 líneas de código fuente escrito en Fortran77, cuenta con un diseño que permite usar un conjunto de parámetros tales como AMMBER, CHARMM, MM2, MM3, OPLS-AA, OPLS-UA, y parámetros propios AMOEBA09 implementados en nuestro programa, un archivo nombreMolecula.xyz incluye las coordenadas cartesianas, el número de átomos, símbolo de los átomos, tipo de átomo, nombre de la molécula y conectividades.

8 Ammonia Dimer						
1	N	1.592728	0.000017	0.016491	61	2 3 4
2	H	2.080554	-0.812588	0.372825	62	1
3	H	2.084361	0.809532	0.374622	62	1
4	H	1.728078	0.002073	-0.987417	62	1
5	N	-1.674368	0.000159	-0.066149	61	6 7 8
6	H	-2.042826	0.810715	0.413439	62	5
7	H	-0.673083	0.000128	0.106233	62	5
8	H	-2.042629	-0.812314	0.410335	62	5
Átomos	**Nombre del Átomo**	**x**	**y**	**z**	**tipo**	conexión

Cuadro 3.1: Formato TINKER.xyz

El formato inicia la primer línea con el número de átomos seguido del nombre de la molécula. Las líneas siguientes están divididas en 7 columnas la primera es la numeración de los átomos seguida por el símbolo atómico las tres siguientes son las coordenadas x, y, z geometría la sexta columna es el tipo de átomo y finalmente la conectividad. En este caso se tomo como ejemplo el dímero amoniaco observe la primera y última columna los números 2, 3 y 4 corresponden a H y 1 al N, lo mismo ocurre para 7, 8 y 9 corresponden a H y 5 al N de esta manera el programa *analize* cálcula la energía potencial total del sistema mostrando en detalle la contribución de cada función potencial, lista los parámetros requeridos para el cálculo de cada interacción individual. Este programa requiere como entrada el archivo TINKER .xyz y un archivo TINKER.key el cual especifica el tipo de parámetros.

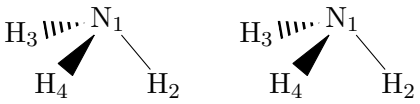


Figura 3.1: Dímero de amoniaco.

3.1.1. Estiramiento del enlace

Para calcular la energía de enlace ecuación 2.2 ejecutamos el programa ANALIZE con las siguientes opciones.

```
usuario@pc:~$ ./analize
Electrostatic Moments and Principle Axes   [M]
Internal Virial, dE/dV Values & Pressure   [V]
Connectivity Lists for Each of the Atoms   [C]
General System and Force Field Information [G]
Force Field Parameters for Interactions    [P]
Total Potential Energy and its Components  [E]
Energy Breakdown over Each of the Atoms    [A]
List of the Large Individual Interactions  [L]
Details for All Individual Interactions    [D]
Electrostatic Moments and Principle Axes   [M]
Internal Virial, dE/dV Values & Pressure   [V]
Connectivity Lists for Each of the Atoms   [C]
```

como resultado se obtiene un archivo TINKER.OUT como el que se muestra a continuación.

```
usuario@pc:~$
```

```
Atom Type Definition Parameters :

   Atom  Symbol  Type  Class  Atomic   Mass  Valence  Description

     1     N       61     45     7    14.007    3     Ammonia N
     2     H       62     46     1     1.008    1     Ammonia H3N
     3     H       62     46     1     1.008    1     Ammonia H3N
     4     H       62     46     1     1.008    1     Ammonia H3N
     5     N       61     45     7    14.007    3     Ammonia N
     6     H       62     46     1     1.008    1     Ammonia H3N
     7     H       62     46     1     1.008    1     Ammonia H3N
     8     H       62     46     1     1.008    1     Ammonia H3N

 Bond Stretching Parameters :

          Atom Numbers                        KS       Length

     1        1     2                      516.500    1.0120
     2        1     3                      516.500    1.0120
     3        1     4                      516.500    1.0120
     4        5     6                      516.500    1.0120
     5        5     7                      516.500    1.0120
     6        5     8                      516.500    1.0120


 Angle Bending Parameters :

             Atom Numbers                      KB       Angle   Fold    Type

     1        2     1     3                43.520   106.800
     2        2     1     4                43.520   106.800
```

```
   3        3     1     4              43.520   106.800
   4        6     5     7              43.520   106.800
   5        6     5     8              43.520   106.800
   6        7     5     8              43.520   106.800


Individual Bond Stretching Interactions :

Type              Atom Names                 Ideal    Actual      Energy

Bond            1-N        2-H              1.0120    1.0126      0.0002
Bond            1-N        3-H              1.0120    1.0126      0.0002
Bond            1-N        4-H              1.0120    1.0130      0.0005
Bond            5-N        6-H              1.0120    1.0113      0.0002
Bond            5-N        7-H              1.0120    1.0160      0.0082
Bond            5-N        8-H              1.0120    1.0113      0.0002


Individual Angle Bending Interactions :

Type              Atom Names                   Ideal    Actual      Energy

Angle       2-H        1-N        3-H      106.8000  106.4520      0.0016
Angle       2-H        1-N        4-H      106.8000  106.6195      0.0004
Angle       3-H        1-N        4-H      106.8000  106.5003      0.0012
Angle       6-H        5-N        7-H      106.8000  106.1777      0.0052
Angle       6-H        5-N        8-H      106.8000  106.7262      0.0001
Angle       7-H        5-N        8-H      106.8000  106.1944      0.0049
```

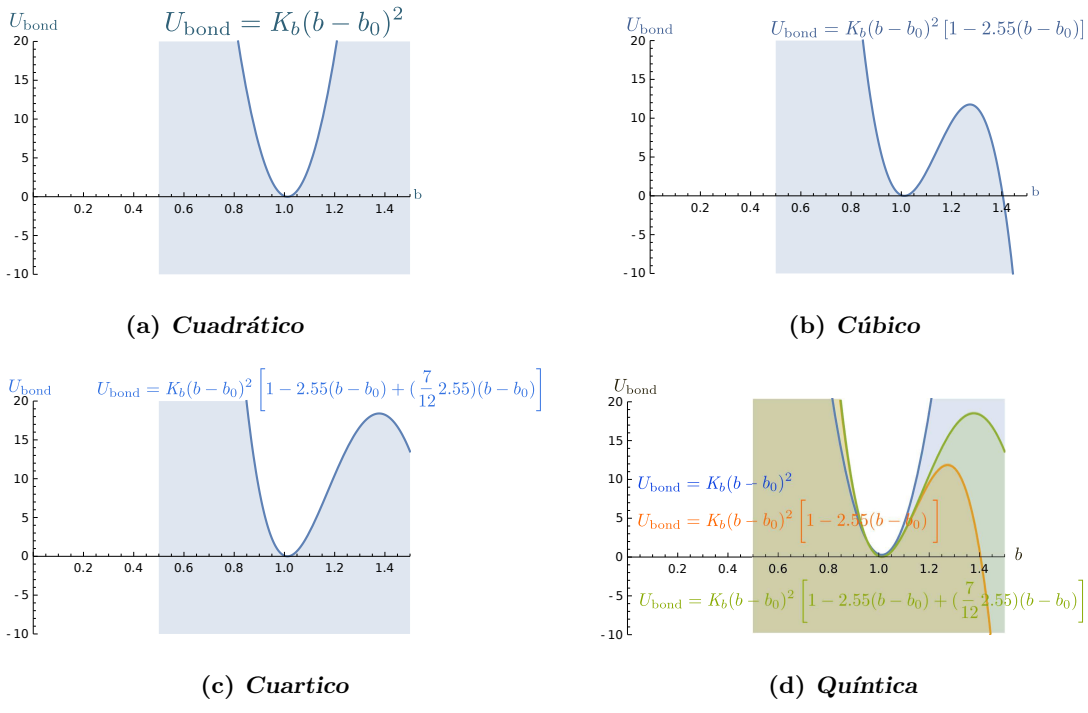
Este archivo contiene los parámetros y constantes además de la conectividad molecular, en nuestro ejemplo la conectividad para el amoniaco así como la contribución individual de energía, para cada una de las seis conexiones nosotros tomaremos la primera conectividad correspondiente a los átomos de Hidrógeno 2-H y Nitrógeno 1-N los parámetros y constantes se encuentran en la sección:

Interacciones tipo enlace-estiramiento individual

$$\text{Ideal} = b_0 = 1.0120$$
$$\text{Actual} = b = 1.0126$$
$$\text{KS} = K_b = 516.500$$

sustituyendo estos valores en la ecuación 2.2 se obtiene la energía de enlace para los átomos 1-N y 2-H también se incluye la gráfica de energía fig (3.2) donde se puede observar más claramente la importancia de agregar términos cúbicos y cuartícos para un mejor ajuste con la función de Morse

$$U_{\text{bond}} = K_b(b - b_0)^2 \left[1 - 2.55(b - b_0) + (\frac{7}{12}2.55)(b - b_0)\right]$$
$$U_{\text{bond}} = 0.001856555118$$



(a) ***Cuadrático*** **(b)** ***Cúbico***

(c) ***Cuartico*** **(d)** ***Quíntica***

Figura 3.2: Análisis energía de enlace

3.1.2. Doblamiento del ángulo

En esta sección se describe los parámetros para las seis conexiones que forman un ángulo para la molécula de amoniaco los átomos 2-H, 1-N, 3-H forman un ángulo con

los siguientes valores.

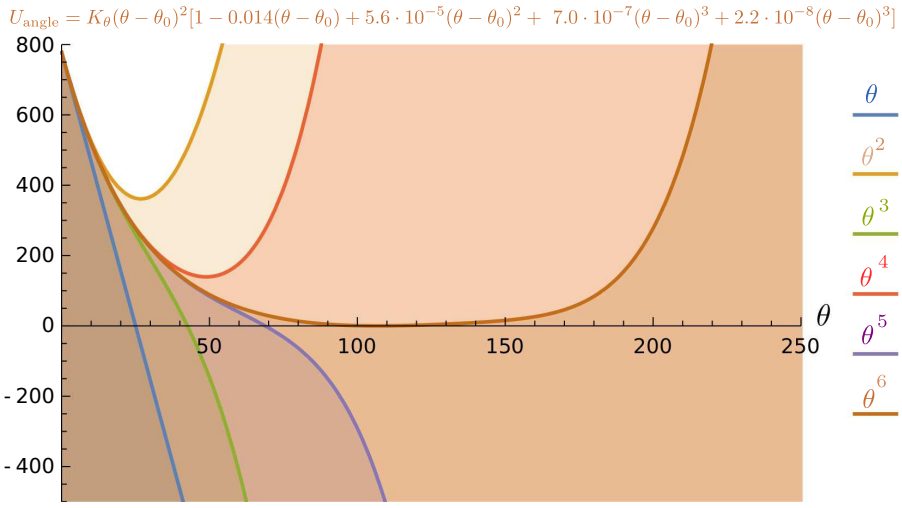
$$\begin{aligned} \text{Ideal} &= \theta_0 = 106.8000 \\ \text{Actual} &= \theta = 106.5420 \\ \text{KB} &= \text{K}_\theta = 43.520 \end{aligned}$$

es importante mencionar que para obtener unidades de energía se debe multiplicar la ecuación (2.3) por la unidad de conversión angunit

$$\text{angunit} = 1/(radian)^2$$

sustituyendo estos valores en la ecuación (2.3) se obtiene la contribución de la energía correspondiente a la formación del ángulo por los átomos 2-H, 1-N y 3-H

$$\begin{aligned} U_{\text{angle}} &= K_\theta(\theta-\theta_0)^2[1-0.014(\theta-\theta_0)+5.6\cdot10^{-5}(\theta-\theta_0)^2 \\ &\quad + \ 7.0\cdot10^{-7}(\theta-\theta_0)^3+2.2\cdot10^{-8}(\theta-\theta_0)^4] \\ U_{\text{angle}} &= 0.00161330247039 \end{aligned}$$



(a) ***Energía ángulo***

Figura 3.3: Análisis energía de ángulo

La Figura(3.3) muestra los seis términos que se incluyen para la anarmonicidad

3.1.3. Estiramiento y doblamiento

Esta sección se muestra los parámetros de los términos acoplados o cruzados, tomamos como ejemplo la molécula de **ether.xyz** debido a que el amoniaco no presenta

términos acoplados o no existen parámetros definidos en el campo de fuerzas AMOEBA09.

```
usuario@pc:~$ ether.out
```

```
Atom Type Definition Parameters :

   Atom  Symbol  Type  Class  Atomic   Mass  Valence  Description

     1     O       58     36     8    15.999    2     Methyl Ether O
     2     C       59     40     6    12.011    4     Methyl Ether CH3
     3     C       59     40     6    12.011    4     Methyl Ether CH3
     4     H       60     44     1     1.008    1     Methyl Ether H3C
     5     H       60     44     1     1.008    1     Methyl Ether H3C
     6     H       60     44     1     1.008    1     Methyl Ether H3C
     7     H       60     44     1     1.008    1     Methyl Ether H3C
     8     H       60     44     1     1.008    1     Methyl Ether H3C
     9     H       60     44     1     1.008    1     Methyl Ether H3C

 Stretch-Bend Parameters :

              Atom Numbers            KSB      Angle   Length1   Length2

     1        2     1     3       38.0000  106.0000    1.4130    1.4130
     2        1     2     7       16.7500  108.7000    1.4130    1.1120
     3        1     2     8       16.7500  108.7000    1.4130    1.1120
     4        1     2     9       16.7500  108.7000    1.4130    1.1120
     5        1     3     4       16.7500  108.7000    1.4130    1.1120
     6        1     3     5       16.7500  108.7000    1.4130    1.1120
     7        1     3     6       16.7500  108.7000    1.4130    1.1120


Individual Bond Stretching Interactions :

 Type              Atom Names              Ideal    Actual      Energy
```

Bond	1-O	2-C	1.4130	1.4212	0.0303
Bond	1-O	3-C	1.4130	1.4212	0.0306
Bond	2-C	7-H	1.1120	1.1128	0.0002
Bond	2-C	8-H	1.1120	1.1120	0.0000
Bond	2-C	9-H	1.1120	1.1128	0.0002
Bond	3-C	4-H	1.1120	1.1120	0.0000
Bond	3-C	5-H	1.1120	1.1128	0.0002
Bond	3-C	6-H	1.1120	1.1128	0.0002

Individual Angle Bending Interactions :

Type	Atom Names			Ideal	Actual	Energy
Angle	2-C	1-O	3-C	106.0000	113.4803	1.3549

$$
\begin{aligned}
\mathrm{K}_{b\theta} &= 38.000 \\
\theta_0 &= 106.0000 \\
\theta &= 113.4803 \\
b &= 1.4212 \\
b_0 &= 1.4130 \\
b' &= 1.4212 \\
b'_0 &= 1.4130
\end{aligned}
$$

Recuerde que (b_0) es la distancia ideal y (b) es la distancia actual del enlace 1-O y 2-C en la sección Individual Bond Stretching Interactions, para el enlace 1-O y 3-C los valores se encuentran en la misma sección y se denota por (b'_0) ideal y (b') el actual, sustituyendo estos valores en la ecuación (2.4) se obtiene la energía de Stretching and Bending.

$$
\begin{aligned}
U_{b\theta} &= K_{b\theta}\left[(b-b_0)+(b'-b'_0)\right](\theta-\theta_0) \\
U_{b\theta} &= 0.0813624144678
\end{aligned}
$$

el término angular se debe multiplicar por la unidad de conversión Stretching-Bending

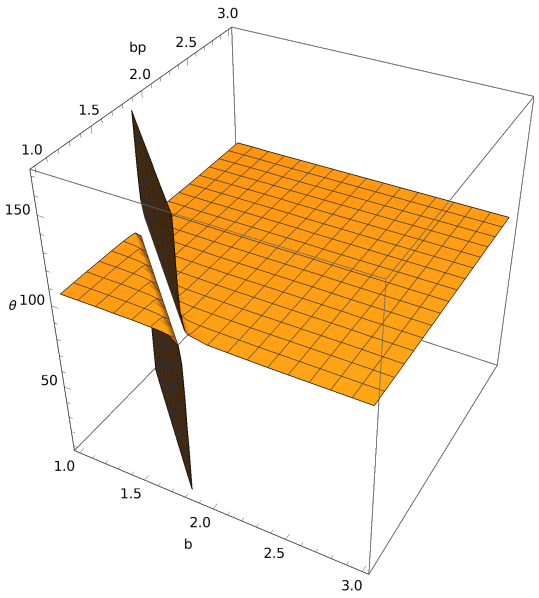
$$\text{stbnunit} = \frac{\pi}{180}$$

es importante mencionar que la constante KSB($K_{b\theta}$) depende de los dos enlaces descritos anteriormente. Estos valores pueden ser positivos para ambos, negativo para el primer enlace y positivo para el segundo, y finalmente positivo para el primer enlace y negativo para el segundo, para obtener el valor de la constante se debe promediar la suma de estos valores que se pueden encontrar en el archivo AMOEBA09.prm. Recuerde que cada átomo en la molécula está definido por un tipo, al que le corresponde una clase, a continuación se muestra un fragmento del archivo y la manera de calcular la constante para las dos primeras interacciones 2C —— 1O —— 3C y 1O —— 2C —— 7H

$$\mathrm{KSB}_{213} = \frac{38.00 + 38.00}{2} = 38.00$$

$$\mathrm{KSB}_{127} = \frac{38.00 - 4.50}{2} = 16.75$$

```
strbnd  40  36  40  38.00  38.00
strbnd  39  38  36  -4.50  38.00
strbnd  36  40  40  38.00  38.00
strbnd  36  40  41  11.50  11.50
strbnd  36  40  44  38.00  -4.50
```



(a) ***Energía StrBnd***

Figura 3.4: Análisis energía de acoplamiento

3.1.4. Fuera de plano

Para evaluar este tipo de energía nos apoyamos de otra molécula la benzamidina que es una de las 35 moléculas que se incluyen en los parámetros amoeba09 es importante mencionar que este número es mayor comparado con otros campos de fuerzas los cuales solo llegan a 13 o 15 moléculas. Los parámetros se encuentran en la sección Out-of-Plane Bending Parameters del archivo de salida del programa *analize.*

```
usuario@pc:~$ benzamidine.out
```

```
Atom Type Definition Parameters :

   Atom  Symbol  Type  Class  Atomic   Mass  Valence  Description

     1     N      333     97     7    14.007    3     Benzamidine N
     2     N      333     97     7    14.007    3     Benzamidine N
     3     C      335     99     6    12.011    3     Benzamidine N-C-N
     4     C      336     89     6    12.011    3     Benzamidine C1-CN2
     5     C      337     89     6    12.011    3     Benzamidine C2
     6     C      338     89     6    12.011    3     Benzamidine C3
     7     C      339     89     6    12.011    3     Benzamidine C4
     8     C      338     89     6    12.011    3     Benzamidine C3
     9     C      337     89     6    12.011    3     Benzamidine C2
    10     H      334     98     1     1.008    1     Benzamidine HN
    11     H      334     98     1     1.008    1     Benzamidine HN
    12     H      334     98     1     1.008    1     Benzamidine HN
    13     H      334     98     1     1.008    1     Benzamidine HN
    14     H      340     90     1     1.008    1     Benzamidine H2
    15     H      341     90     1     1.008    1     Benzamidine H3
    16     H      342     90     1     1.008    1     Benzamidine H4
    17     H      341     90     1     1.008    1     Benzamidine H3
    18     H      340     90     1     1.008    1     Benzamidine H2

Out-of-Plane Bending Parameters :
```

	Atom Numbers				KOPB
1	11	1	3	10	0.1793
2	10	1	3	11	0.1793
3	3	1	10	11	0.0500
4	13	2	3	12	0.1793
5	12	2	3	13	0.1793
6	3	2	12	13	0.0500
7	4	3	1	2	0.0195
8	2	3	1	4	0.0195
9	1	3	2	4	0.0195
10	9	4	3	5	0.2002
11	5	4	3	9	0.2002
12	3	4	5	9	0.1001
13	14	5	4	6	0.2099
14	6	5	4	14	0.2002
15	4	5	6	14	0.2002
16	15	6	5	7	0.2099
17	7	6	5	15	0.2002
18	5	6	7	15	0.2002
19	16	7	6	8	0.2099
20	8	7	6	16	0.2002
21	6	7	8	16	0.2002
22	17	8	7	9	0.2099
23	9	8	7	17	0.2002
24	7	8	9	17	0.2002
25	18	9	4	8	0.2099
26	8	9	4	18	0.2002
27	4	9	8	18	0.2002

Individual Out-of-Plane Bend Interactions :

Type	Atom Names	Angle	Energy

```
O-P-Bend       11-H       1-N        3-C       10-H       1.2683      0.00008632
O-P-Bend       10-H       1-N        3-C       11-H       1.2677      0.00008624
O-P-Bend        3-C       1-N       10-H       11-H       0.9745      0.00001428
O-P-Bend       13-H       2-N        3-C       12-H       1.2667      0.00008609
O-P-Bend       12-H       2-N        3-C       13-H       1.2661      0.00008601
O-P-Bend        3-C       2-N       12-H       13-H       0.9732      0.00001424
O-P-Bend        4-C       3-C        1-N        2-N       0.0116      0.00000000
O-P-Bend        2-N       3-C        1-N        4-C       0.0134      0.00000000
O-P-Bend        1-N       3-C        2-N        4-C       0.0134      0.00000000
O-P-Bend        9-C       4-C        3-C        5-C       0.3516      0.00000750
O-P-Bend        5-C       4-C        3-C        9-C       0.3516      0.00000750
O-P-Bend        3-C       4-C        5-C        9-C       0.3189      0.00000309
O-P-Bend       14-H       5-C        4-C        6-C       0.1661      0.00000176
O-P-Bend        6-C       5-C        4-C       14-H       0.1308      0.00000104
O-P-Bend        4-C       5-C        6-C       14-H       0.1302      0.00000103
O-P-Bend       15-H       6-C        5-C        7-C       0.3037      0.00000587
O-P-Bend        7-C       6-C        5-C       15-H       0.2396      0.00000349
O-P-Bend        5-C       6-C        7-C       15-H       0.2393      0.00000348
O-P-Bend       16-H       7-C        6-C        8-C       0.0618      0.00000024
O-P-Bend        8-C       7-C        6-C       16-H       0.0488      0.00000014
O-P-Bend        6-C       7-C        8-C       16-H       0.0488      0.00000014
O-P-Bend       17-H       8-C        7-C        9-C       0.2351      0.00000352
O-P-Bend        9-C       8-C        7-C       17-H       0.1852      0.00000209
O-P-Bend        7-C       8-C        9-C       17-H       0.1855      0.00000209
O-P-Bend       18-H       9-C        4-C        8-C       0.1705      0.00000185
O-P-Bend        8-C       9-C        4-C       18-H       0.1343      0.00000110
O-P-Bend        4-C       9-C        8-C       18-H       0.1336      0.00000109

Torsional Angle Parameters :

                Atom Numbers          Amplitude, Phase and Periodicity
```

```
 1      10     1     3     2      4.000 180/2
 2      10     1     3     4      4.000 180/2
 3      11     1     3     2      4.000 180/2
 4      11     1     3     4      4.000 180/2
 5      12     2     3     1      4.000 180/2
 6      12     2     3     4      4.000 180/2
 7      13     2     3     1      4.000 180/2
 8      13     2     3     4      4.000 180/2
 9       1     3     4     5      2.304 180/2
10       1     3     4     9      2.304 180/2
11       2     3     4     5      2.304 180/2
12       2     3     4     9      2.304 180/2
13       3     4     5     6     -0.610   0/1   4.212 180/2
14       3     4     5    14      6.104 180/2
15       9     4     5     6     -0.670   0/1   4.004 180/2
16       9     4     5    14      0.550   0/1   4.534 180/2  -0.550   0/3
17       3     4     9     8     -0.610   0/1   4.212 180/2
18       3     4     9    18      6.104 180/2
19       5     4     9     8     -0.670   0/1   4.004 180/2
20       5     4     9    18      0.550   0/1   4.534 180/2  -0.550   0/3
21       4     5     6     7     -0.670   0/1   4.004 180/2
22       4     5     6    15      0.550   0/1   4.534 180/2  -0.550   0/3
23      14     5     6     7      0.550   0/1   4.534 180/2  -0.550   0/3
24      14     5     6    15      4.072 180/2
25       5     6     7     8     -0.670   0/1   4.004 180/2
26       5     6     7    16      0.550   0/1   4.534 180/2  -0.550   0/3
27      15     6     7     8      0.550   0/1   4.534 180/2  -0.550   0/3
28      15     6     7    16      4.072 180/2
29       6     7     8     9     -0.670   0/1   4.004 180/2
30       6     7     8    17      0.550   0/1   4.534 180/2  -0.550   0/3
31      16     7     8     9      0.550   0/1   4.534 180/2  -0.550   0/3
32      16     7     8    17      4.072 180/2
33       7     8     9     4     -0.670   0/1   4.004 180/2
```

34	7	8	9	18	0.550	0/1	4.534	180/2	-0.550	0/3
35	17	8	9	4	0.550	0/1	4.534	180/2	-0.550	0/3
36	17	8	9	18	4.072	180/2				

Individual Torsional Angle Interactions :

Type	Atom Names				Angle	Energy
Torsion	10-H	1-N	3-C	2-N	173.4993	0.0513
Torsion	10-H	1-N	3-C	4-C	-6.4558	0.0506
Torsion	11-H	1-N	3-C	2-N	-2.3839	0.0069
Torsion	11-H	1-N	3-C	4-C	177.6610	0.0067
Torsion	12-H	2-N	3-C	1-N	173.5707	0.0502
Torsion	12-H	2-N	3-C	4-C	-6.4742	0.0509
Torsion	13-H	2-N	3-C	1-N	-2.3181	0.0065
Torsion	13-H	2-N	3-C	4-C	177.6370	0.0068
Torsion	1-N	3-C	4-C	5-C	134.3576	1.1778
Torsion	1-N	3-C	4-C	9-C	-44.4437	1.1296
Torsion	2-N	3-C	4-C	5-C	-45.5989	1.1761
Torsion	2-N	3-C	4-C	9-C	135.5997	1.1279
Torsion	3-C	4-C	5-C	6-C	-179.6941	0.0001
Torsion	3-C	4-C	5-C	14-H	-0.1945	0.0001
Torsion	9-C	4-C	5-C	6-C	-0.8643	-0.6691
Torsion	9-C	4-C	5-C	14-H	178.6353	0.0019
Torsion	3-C	4-C	9-C	8-C	178.9944	0.0013
Torsion	3-C	4-C	9-C	18-H	-1.5191	0.0043
Torsion	5-C	4-C	9-C	8-C	0.1656	-0.6700
Torsion	5-C	4-C	9-C	18-H	179.6521	0.0001
Torsion	4-C	5-C	6-C	7-C	0.9082	-0.6690
Torsion	4-C	5-C	6-C	15-H	-179.9975	0.0000
Torsion	14-H	5-C	6-C	7-C	-178.6021	0.0020
Torsion	14-H	5-C	6-C	15-H	0.4922	0.0003
Torsion	5-C	6-C	7-C	8-C	-0.2357	-0.6699

Torsion	5-C	6-C	7-C	16-H	179.5804	0.0002
Torsion	15-H	6-C	7-C	8-C	-179.3333	0.0005
Torsion	15-H	6-C	7-C	16-H	0.4828	0.0003
Torsion	6-C	7-C	8-C	9-C	-0.4576	-0.6697
Torsion	6-C	7-C	8-C	17-H	-179.7590	0.0001
Torsion	16-H	7-C	8-C	9-C	179.7263	0.0001
Torsion	16-H	7-C	8-C	17-H	0.4249	0.0002
Torsion	7-C	8-C	9-C	4-C	0.4923	-0.6697
Torsion	7-C	8-C	9-C	18-H	-179.0052	0.0010
Torsion	17-H	8-C	9-C	4-C	179.7911	0.0000
Torsion	17-H	8-C	9-C	18-H	0.2937	0.0001

A continuación se muestran los valores de la constante y ángulo fuera de plano así como la contribución de energía para la cuarteta de átomos 11H —— 1N —— 3C —— 10H

$$\text{KOPB} = \text{K}_{oop} = 0.1793$$
$$\text{Angle} = \chi = 1.2683$$

Sustituyendo estos valores en la ecuación (2.5) la energía Out-of-Plane.

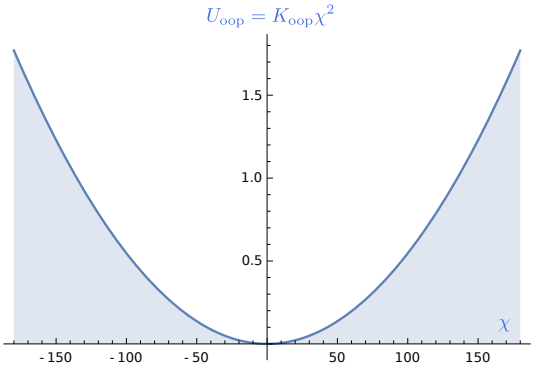
$$U_{\text{oop}} = K_{\text{oop}}\chi^2$$
$$U_{\text{oop}} = 8.78575340903e^{-05}$$

Individual Out-of-Plane Bend Interactions :

Atom Numbers					KOPB
1	11	1	3	10	0.1793

Individual Out-of-Plane Bend Interactions :

Type	Atom Names				Angle	Energy
O-P-Bend	11-H	1-N	3-C	10-H	1.2683	0.00008632



(a) ***Energía oop***

Figura 3.5: Análisis energía de ángulo fuera de plano

3.1.5. Torsión

La energía torsional también se observa en una cuarteta de átomos, en nuestro ejemplo tomamos como referencia la benzamidina que define su parámetros en la sección Torsional Angle Parameters del archivo benzamidine.out

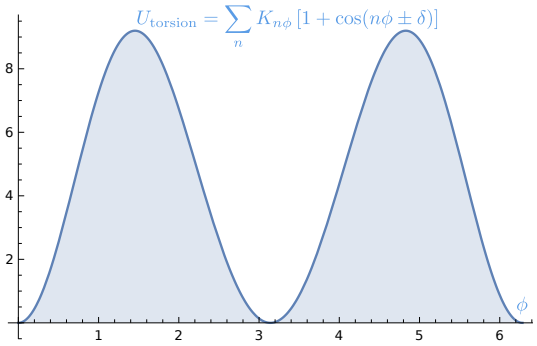
```
Torsional Angle Parameters

                  Atom Numbers          Amplitude, Phase and Periodicity

16        9     4     5    14       0.550   0/1   4.534 180/2  -0.550   0/3

Individual Torsional Angle Interactions :

 Type                        Atom Names                        Angle      Energy

 Torsion          9-C         4-C         5-C        14-H     178.6353      0.0019
```

para los átomos 9C——4C——5C——14H, observe que hay tres valores para la amplitud, phase y periodicity, mire con detalle las 36 interacciones correspondientes a la torsión. Note que en algunos casos los valores de amplitude, phase y periodicity son dos o incluso uno.

$$\begin{aligned}
\text{Amplitude} &= K_{\phi 1} = 0.550 \\
\text{Phase} &= \delta 1 = 0 \\
\text{Amplitude} &= K_{\phi 2} = 4.534 \\
\text{Phase} &= \delta 2 = 180 \\
\text{Amplitude} &= K_{\phi 3} = -0.550 \\
\text{Phase} &= \delta 3 = 0 \\
\text{Periodicity} &= n = 1, 2, 3 \\
\text{Angle} &= \phi = 178.6353
\end{aligned}$$

Sustituyendo estos valores en la ecuación (2.6) se obtiene la energía Torsional.

$$\begin{aligned}
\text{torsunit} &= 0.5 \\
U_{\text{torsion}} &= \sum_{n=1}^{3} K_{n\phi} \left[1 + \cos(n\phi \pm \delta)\right] \\
U_{\text{torsion}} &= 0.00194799006425
\end{aligned}$$



(a) ***Energía Tor***

Figura 3.6: Análisis energía de torsión

3.2. Conectividad Molecular

Las moléculas descritas anteriormente son poliatomicas, es decir están compuesta por más de un átomo, cuando esto ocurre es importante indicar el orden en que se

encuentran para determinar sus enlaces correspondientes cada átomo tiene propiedades diferentes, como son la electro negatividad el número de protones, electrones, masa atómica, radio de van der Waals, todas ellas participan en la formación del enlace, esto genera un gran número de combinaciones posibles restringidas por las propiedades antes mencionadas. Un ejemplo de lo anterior se puede observar en los grupos funcionales como son alcoholes, ester, cetonas, aldehídos, alcanos, alquenos y alquinos, por mencionar algunos, piense en los carbonos de la cetona y el aldehído el grupo carbonilo, se encuentra presente en ambos pero la posición le confiere propiedades muy distintas a cada compuesto, por esta razón es importante distinguir entre ambas *clases* de carbono, aún cuando son del mismo *tipo*, estos dos parámetros son muy importantes en nuestro modelo de mecánica molecular ya que cada átomo se toma como una pelota con propiedades intrínsecas que nos permite desarrollar de manera más sencilla las combinaciones entre átomos.

3.2.1. Red de enlace

La geometría de una molécula está determinada por la disposición de sus átomos, en nuestro modelo representamos cada átomo de una molécula como una partícula en el espacio x, y, z según la mecánica molecular, con propiedades que se describen a través de un campo de fuerzas, 2.1.2 por ejemplo una molécula pequeña como el agua está formada por tres átomos dos (Hidrógenos H) y uno de (Oxígeno O) es sabido que el átomo de Oxígeno es el centro al que se unen los Hidrógenos como se muestra a continuación H_1 — O_2 — H_3, o tal vez H_1 — H_3 — O_2, O_2 — H_1 — H_3 además también existe un ángulo formado por los enlaces que influye en su longitud (estiramiento), como se muestra en la Figura 3.7

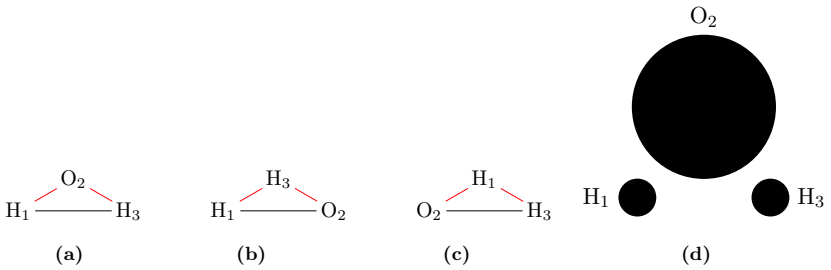


Figura 3.7: Conexiones

otros aspectos geométricos importantes (2.3) son necesarios para evaluar cualquier campo de fuerzas, entonces ¿Qué nos permite determinar la conectividad de los átomos en una molécula?. Tomando como ejemplo la molécula del agua en la Figura (3.7) se observa que la conectividad puede ser distinta incluso se podría poner una conexión donde no existe, como se observa en la Figura 3.7(a) donde los H_1 — H_3 están conectados o en 3.7(b) H_1 — O_2 y 3.7(c) O_2 — H_3 también notamos que su longitud cambia, cuando se

considera el ángulo que forman las conexiones, la Figura 3.7(d) muestra tres puntos de distinto tamaño en el espacio x, y, z que representan los tres átomos del agua donde el punto más grande representa el oxígeno y los dos pequeños el hidrógeno por la ecuación (2.8) conocemos la distancia que tienen cada uno de ellos con respecto a los demás, un parámetro importante es el radio de los puntos. Claramente el radio mayor corresponde al punto más grande y el radio menor a los dos más pequeños la Figura 3.8(a) muestra una representación del átomo de H y su radio de van der Waals $r_{hidrogeno}$ y en 3.8(b) el átomo de O y su radio de van der Waals $r_{oxigeno}$ a continuación se suman los radios de cada átomo $r_{h_1} + r_{h_2}$, $r_{h_1} + r_o$, $r_{h_2} + r_o$ estas sumas se comparan con la distancia que los separa cuando la suma de sus radios es mayor que la distancia consideramos que hay conexión Figura 3.8(c).

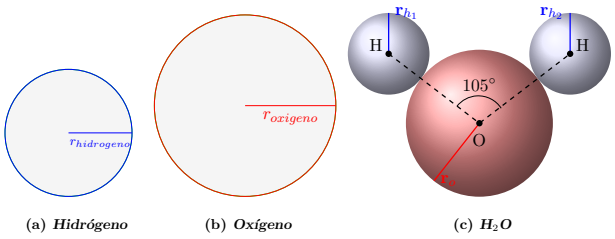


Figura 3.8: Radio van der Waals

Ahora supongamos que tenemos cinco puntos a, b, c, d, e, f con coordenadas x, y, z Figura 3.9(a) y conocemos sus radios, entonces calculamos las distancia y suma de radios en cada uno de estos 3.9(b) y 3.9(c) para obtener la conexión final 3.9(d). Lo anterior se resume en el algoritmo 1 para tener una matriz de $M_{(enlaces,2)}$

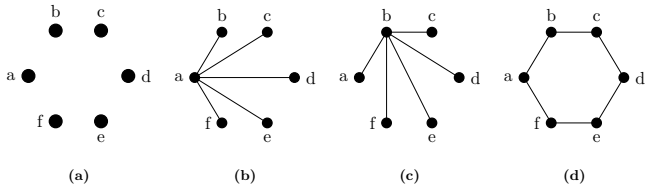


Figura 3.9: Conectividad

$$M_{(k,2)} = \begin{pmatrix} a_{1,1} & b_{1,2} \\ a_{2,1} & f_{2,2} \\ b_{3,1} & c_{3,2} \\ c_{4,1} & d_{4,2} \\ d_{5,1} & e_{5,2} \\ f_{6,1} & e_{6,2} \end{pmatrix}$$

Algoritmo 1 Conectividad

Input:

1: x, y, z (Geometría),

2: na (Número de átomos),

3: $rvdw$ (Radios de van der Waals)

Output:

4: $M_{(k,2)} = \begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \\ \vdots & \vdots \\ m_{k,1} & m_{k,2} \end{pmatrix}$ ▷ $k \rightarrow N.enlaces$

5: $k \rightarrow N.Enlaces$

6: **for each** $x, y, z, rvdw \rightarrow i, j$ **do**

7: $srvdw = \sum_{i=1}^{na-1} \sum_{j=i>1}^{na} (rvdw_i + rvdw_j)$ ▷ Suma Radios van der Waals

8: Distancia $= \sqrt{(\mathrm{x}_j - \mathrm{x}_i)^2 + (\mathrm{y}_j - \mathrm{y}_i)^2 + (\mathrm{z}_j - \mathrm{z}_i)^2}$

9: **if** Distancia $\leq$ srvdw **then** ▷ Determinar Conectividad

10: $m_{k,1} = i$ ▷ La matriz M cada elemento $m_{k,n}$ es un átomo

11: $m_{k,2} = j$ ▷ Cada fila representa átomos enlazados

12: $k = +1$

3.2.2. Red de ángulo

Como se menciono anteriormente en la sección 3.2 una parte importante de la conectividad cuando la molécula está formada por más de dos átomos es el átomo central que está determinado por dos enlaces unidos a éste, su variación aunado a sus propiedades y parámetros de cada campo de fuerzas permite evaluar una parte importante de la energía covalente, entonces ¿Cómo saber quien es el centro ó átomo central? Para responder esta pregunta retomaremos el ejemplo del agua Figura 3.8(c), y sabiendo que hay dos conexiones a–b y b–c Figura 3.10 a partir de la matriz de conectividad M (3.1) que contiene cuatro elementos donde el elemento común b puede encontrarse en diferentes posiciones. La importancia de este elemento resulta evidente cuando se comparan todos los casos posibles de tal manera que nos permite decir que tal elemento es el centro de dos conexiones. Además están conectados formando un ángulo con el átomo central a–b–c este resultado se guarda en otra matriz 3.2. El algoritmo 2 describe los pasos para generar la matriz angular a partir de la matriz de conectividad.

$$M_{(2,2)} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}, \begin{pmatrix} b & a \\ c & b \end{pmatrix}, \begin{pmatrix} b & a \\ b & c \end{pmatrix}, \begin{pmatrix} a & b \\ c & b \end{pmatrix} \tag{3.1}$$

$$Ang_{(k,3)} = \begin{pmatrix} a & b & c \end{pmatrix} \tag{3.2}$$

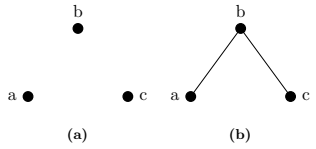


Figura 3.10: Átomo Central

$$Ang_{(2,3)} = \begin{pmatrix} a & b & c \\ b & a & d \end{pmatrix}, \begin{pmatrix} c & a & b \\ d & b & a \end{pmatrix}, \begin{pmatrix} c & a & b \\ a & b & d \end{pmatrix}, \begin{pmatrix} a & b & c \\ d & a & b \end{pmatrix} \tag{3.3}$$

$$Tor_{(k,4)} = \begin{pmatrix} a & b & c & d \end{pmatrix} \tag{3.4}$$

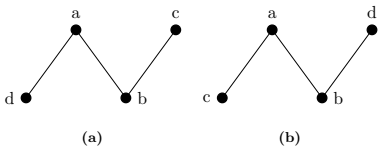


Figura 3.11: Diedros

$$Ang_{(2,3)} = \begin{pmatrix} a & b & c \\ a & b & d \end{pmatrix}, \begin{pmatrix} c & a & b \\ d & a & b \end{pmatrix}, \begin{pmatrix} c & a & b \\ b & a & d \end{pmatrix}, \begin{pmatrix} a & b & c \\ d & b & a \end{pmatrix} \tag{3.5}$$

$$Oop_{(k,4)} = \begin{pmatrix} a & b & c & d \end{pmatrix} \tag{3.6}$$

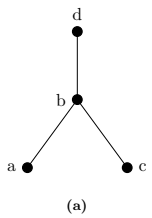


Figura 3.12: Diedros Impropios

Algoritmo 2 Átomo Central

Input:

1: (Matriz de Conectividad) $M_{(n,2)} = \begin{array}{c} \begin{matrix} a_1 & a_2 \end{matrix} \\ \begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \\ \vdots & \vdots \\ m_{n,1} & m_{n,2} \end{pmatrix} \end{array}$ ▷ $n \rightarrow No.enlaces$

Output: (Matriz angular) $Ang_{(k,3)} = \begin{pmatrix} ang_{1,1} & ang_{1,2} & \cdots & ang_{1,3} \\ ang_{2,1} & ang_{2,2} & \cdots & ang_{2,3} \\ \vdots & \vdots & \ddots & \vdots \\ ang_{k,1} & ang_{k,2} & \cdots & ang_{k,3} \end{pmatrix}$

2: ▷ $k \rightarrow No.angulos$

3: **for each** $a_{i,1}, a_{j,2} \rightarrow i = 1,\ enlaces - 1, j = i+1,\ enlaces$ **do**

4: **if** $a_{i,2} = a_{j,1}$ **then** ▷ Determinar Conectividad

5: $ang_{k,1} = a_{i,1}$

6: $ang_{k,2} = a_{i,2}$

7: $ang_{k,3} = a_{j,2}$

8: **else if** $a_{i,1} = a_{j,2}$ **then**

9: $ang_{k,1} = a_{i,2}$

10: $ang_{k,2} = a_{i,2}$

11: $ang_{k,3} = a_{j,1}$

12: **else if** $a_{i,2} = a_{j,2}$ **then**

13: $ang_{k,1} = a_{i,1}$

14: $ang_{k,2} = a_{i,2}$

15: $ang_{k,3} = a_{j,1}$

16: **else if** $a_{i,1} = a_{j,1}$ **then**

17: $ang_{k,1} = a_{i,2}$

18: $ang_{k,2} = a_{i,1}$

19: $ang_{k,3} = a_{j,2}$

20: $k = +1$

3.2.3. Red de acoplamiento

Anteriormente se determino el átomo central el cual es importante para la construcción de la siguiente red que describe el cambio en la longitud de los enlaces cuando varia el ángulo al que se encuentran unidos. Para construir esta red nos apoyamos de los parámetros definidos para el campo de fuerzas AMOEBA09 es decir a partir de la matriz de angulo que contiene la clase de cada átomo lo comparamos con el archivo de parámetros. Si la triada de clase se encuentra ésta es asignada a la matriz de acoplamiento como se describe en el siguiente algoritmo 3

Algoritmo 3 Acoplamiento

Input:

1: (Parametros str-bnd) ▷ Archivo de Parametros

2: $Ang_{(k,3)} = \begin{pmatrix} ang_{1,1} & ang_{1,2} & \cdots & ang_{1,3} \\ ang_{2,1} & ang_{2,2} & \cdots & ang_{2,3} \\ \vdots & \vdots & \ddots & \vdots \\ ang_{k,1} & ang_{k,2} & \cdots & ang_{k,3} \end{pmatrix}$

3: (Número de ángulos) $dang$

4: (Parámetro máximo acoplados) nstrbnd

Output:

5: (Matriz Acoplada) $SB(k,3) = \begin{pmatrix} sb_{1,1} & sb_{1,2} & \cdots & sb_{1,3} \\ sb_{2,1} & sb_{2,2} & \cdots & sb_{2,3} \\ \vdots & \vdots & \ddots & \vdots \\ sb_{k,1} & sb_{k,2} & \cdots & sb_{k,3} \end{pmatrix}$ ▷ $k \rightarrow No.Acoplados$

6: **for** $i \rightarrow nstrbnd$ **do**

7: **for** $j \rightarrow dang$ **do**

8: **if** $Ang = M_{str-bnd}$ **then** ▷ Evaluar elementos clase en cada matriz

9: $SB_{(k,:)} = Ang_{(j,:)}$ ▷ Asignar elementos a SB

10: $k = +1$

3.2.4. Red de torsión

La construcción de la red de torsión contiene los átomos que forman un ángulo diedro. Para poder definir un ángulo diedro se necesita de cuatro puntos es decir una tetrada de átomos dos de ellos forman una recta que se llama arista la cual limita dos

semi planos. A partir de la matriz de ángulo podemos generar la matriz de torsión si encontramos un arista es decir dos puntos que se encuentren al comparar la matriz de ángulo (3.3) la cual muestra de manera general cuatro tipos de casos que pueden presentarse. Para formar la matriz (3.4) la representación se muestra en la Figura (3.11), lo anterior se resume en el algoritmo Torsión(4)

Algoritmo 4 Torsión

Input:

1: $Ang_{(k,3)} = \begin{pmatrix} ang_{1,1} & ang_{1,2} & \cdots & ang_{1,3} \\ ang_{2,1} & ang_{2,2} & \cdots & ang_{2,3} \\ \vdots & \vdots & \ddots & \vdots \\ ang_{k,1} & ang_{k,2} & \cdots & ang_{k,3} \end{pmatrix}$

2: (Número de ángulos) $dang$

Output:

3: (Matriz Torsión) $Tor(k,3) = \begin{pmatrix} tor_{1,1} & tor_{1,2} & tor_{1,3} & \cdots & tor_{1,4} \\ tor_{2,1} & tor_{2,2} & tor_{1,3} & \cdots & tor_{2,4} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ tor_{k,1} & tor_{k,2} & tor_{k,3} & \cdots & tor_{k,4} \end{pmatrix}$

4: **for** $i \rightarrow dang$ **do**

5: **for** $j \rightarrow i+1, dang$ **do**

6: **if** $Ang_{i,:} = Ang_{j,:}$ **then** ▷ Evaluar elementos de cada matriz

7: **if** $a_{i,2} = a_{j,1}$ **then** ▷ Asignar elementos a tor

8: $tor_{(k,1)} = a_{(i,1)}$

9: $tor_{(k,2)} = a_{(i,2)}$

10: $tor_{(k,3)} = a_{(j,2)}$

11: $tor_{(k,4)} = a_{(j,2)}$

12: $k = +1$

3.2.5. Red de ángulo impropio o fuera de plano

Al igual que la red de torsión también está compuesta por una tetrada de átomos, pero con una ligera modificación de conexión en este caso no hay un arista que une los dos planos. Ahora debemos buscar un centro que une tres puntos, la matriz (3.5) permite generar la matriz (3.6) de ángulos impropios o fuera de plano Figura (3.12)

Algoritmo 5 Fuera de Plano

Input:

1: $Ang_{(k,3)} = \begin{pmatrix} ang_{1,1} & ang_{1,2} & \cdots & ang_{1,3} \\ ang_{2,1} & ang_{2,2} & \cdots & ang_{2,3} \\ \vdots & \vdots & \ddots & \vdots \\ ang_{k,1} & ang_{k,2} & \cdots & ang_{k,3} \end{pmatrix}$

2: (Número de ángulos) $dang$

3: (Número de valencia) val ▷ Valencia átomo central

Output:

4: (Matriz Oop) $oop(k,3) = \begin{pmatrix} Oop_{1,1} & Oop_{1,2} & Oop_{1,3} & \cdots & Oop_{1,4} \\ Oop_{2,1} & Oop_{2,2} & Oop_{1,3} & \cdots & Oop_{2,4} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Oop_{k,1} & Oop_{k,2} & Oop_{k,3} & \cdots & Oop_{k,4} \end{pmatrix}$

5: **for** $i \rightarrow dang$ **do**

6: **for** $j \rightarrow i+1, dang$ **do**

7: **if** $Ang_{(i,:)} = Ang_{(j,:)}$ *and* $val(:) = 3$ **then**

8: $Oop_{(k,1)} = a_{(i,1)}$

9: $Oop_{(k,2)} = a_{(i,2)}$

10: $Oop_{(k,3)} = a_{(j,2)}$

11: $Oop_{(k,4)} = a_{(j,2)}$

12: $k = +1$

3.3. Implementación

Ahora que ya sabemos calcular las energías correspondientes y determinar las conexiones *enlace, acoplamiento, ángulo, diedro y diedro impropio*, podemos pensar en una estructura general que nos permita trabajar para llegar a nuestro objetivo. Es decir con los **datos**, como *parámetros*, *geometría*, *radio, número, masa, símbolo, valencia* y nuestros algoritmos podemos iniciar el diseño de un programa al que nombramos **MMF** *mecánica molecular fina*. Como primer paso se muestra la Figura (3.13), en la cual se puede observar la estructura general de la implementación, tres condiciones necesarias, la geometría de la molécula un archivo de parámetros y finalmente el número de conexiones presentes en la molécula con ello generar las energías correspondientes a

cada tipo de interacción y finalmente una simulación de tipo Monte Carlo.

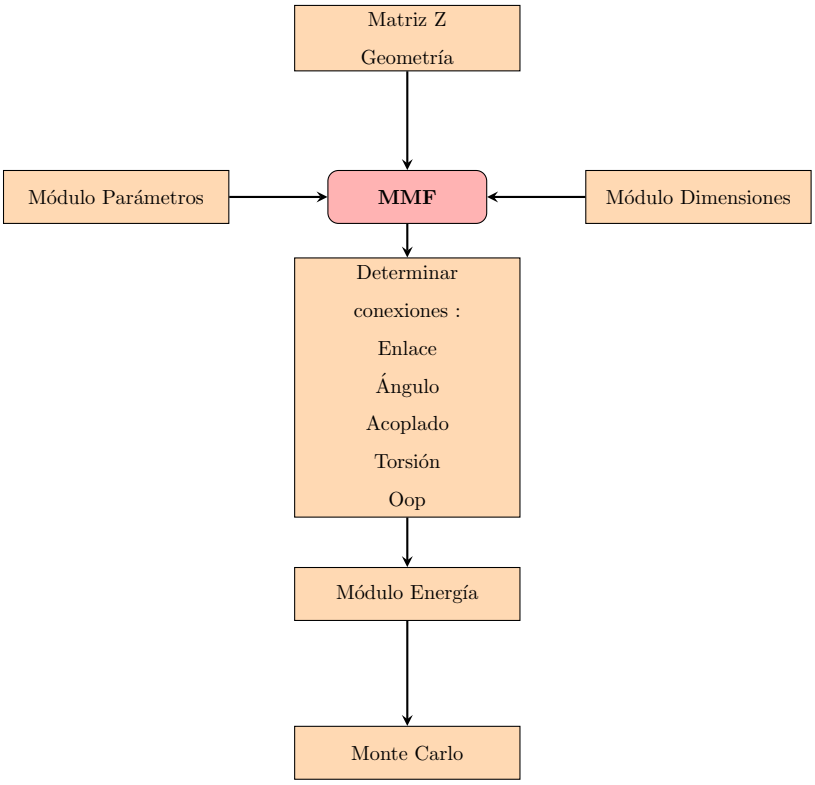


Figura 3.13: Estructura de Implementación

3.3.1. Módulos Implementación de Archivos

Siguiendo con la estructura de implementación asignando tareas especificas, que dividen en módulos procesos de lectura, asignación y evaluación de operaciones se llega a la solución general. En nuestro caso implementamos módulos de lectura de archivos.

3.3.1.1. ModParametros

Este módulo **ModParametros**, esta diseñado para leer el archivo de parámetros del campo de fuerzas AMOEBA09 y los radios de van der Waals, contiene de tres subrutinas.

3.3.1.2. Modxyz

Este módulo **Modxyz** sirve para leer la geometría y crear el archivo de conectividades contiene de dos subrutinas.

3.3.1.3. ModConstantes

En este módulo **ModConstantes** incluye la precisión las constantes y los parámetros definidos para el campo de fuerza AMOEBA09

3.3.1.4. ModGeom

En este módulo **ModGeom** incluye las funciones para hacer los cálculos geométricos incluye cuatro funciones

3.3.1.5. ModDimensiones

En este módulo **ModDimensiones** determina el número de conectividades para las matrices de conexiones contiene cinco subrutinas.

3.3.1.6. ModConexiones

En este módulo **ModConexiones** genera las matrices de conexión para enlace, ángulo y torsión, contiene tres subrutinas.

3.3.1.7. ModAtomprm

En este módulo **ModAtomprm** busca los parámetros y constantes para las matrices de conexión contiene tres subrutinas

3.3.1.8. ModIdealprm

En este módulo **ModAtomprm** busca los parámetros y constantes ideales para las matrices de conexión, contiene cinco subrutinas.

3.3.1.9. ModOop

En este módulo **ModOop** genera la matriz de conectividad fuera de plano asigna las constantes y parámetros ideales, contiene tres subrutinas.

3.3.1.10. ModAcoplado

En este módulo **ModAcoplado** genera la matriz de conectividad de acoplamiento contiene una subrutina.

3.3.1.11. ModAmoeba09

En este módulo **ModAmoeba09** contienen las funciones del campo de fuerzas AMOEBA para calcular las energías, contiene cinco funciones.

3.3.1.12. ModEnergia

En este módulo **ModEnergia** calula los tipos de energía enlace, ángulo, fuera de plano, torsión e impropio del campo de fuerzas AMOEBA09, contiene cuatro subrutinas

3.3.1.13. ModZtoc

Este módulo **ModZtoc** es para leer la matriz Z para el alcano de cinco carbonos, convertir esa matriz a coordenadas cartesianas, contiene tres subrutinas.

3.3.1.14. main

En este módulo **main** es el programa principal muestra las conexiones energías y contienen un módulo para generar números aleatorios, genera una simulación de tipo Monte Carlo para la molécula de pentano a partir de la matriz Z del alcano.

3.3.2. Visión de la simulación Monte Carlo

En la sección anterior describimos los módulos que conforman un programa para calcular energías de una molécula de interés a partir de condiciones iniciales como su geometría, pero ¿Qué pasa si movemos un poco cada átomo? Es decir generamos una perturbación ¿la energía será igual a la anterior? Para responder estas preguntas es necesario calcular la nueva energía y comparar ambas. Esto resulta relativamente sencillo para un par de comparaciones, pero nuestra muestra es muy pequeña con lo que estamos limitados para tener un buen resultado. Además también hemos provocado deliberadamente la perturbación. Si queremos tener mejores resultados debemos aumentar el tamaño de la muestra y que las perturbaciones no tengan un peso que cargue nuestros resultados. Para ello se dispone de métodos de muestreo representativos que nos permite asegurar distribuciones homogéneas del sistema de interés en nuestro caso una molécula. Se sabe que la naturaleza es parsimoniosa y las moléculas buscan un estado de mínima energía así que nos centramos en encontrar energías más pequeñas a la de inicio, energía mínima pero dando oportunidad a otras que se encuentran en un estado de mínima energía local. A este proceso se le conoce como simulación. Beddard (2), para una revisión amplia del método Monte Carlo véase, Jensen (4), Schlick (8)

3.3.2.1. Implementación método Monte Carlo

El algoritmo de método Monte Carlo, para nuestro interés nos permite generar nuevas configuraciones a partir de una porque podemos aumentar $\Delta\phi$, el cambio del ángulo diedro, cambio aleatorio a la geometría coordenadas x, y, z o cambiando el ángulo diedro en la matriz Z, esta nueva configuración aumentará o disminuirá la energía. Con esta tendremos dos energías E_1 y E_2 si la nueva energía es más baja aceptamos, si no la probabilidad de que E_2 contribuya es $e^{-(E2-E1)/(k_BT)}$ generamos un número aleatorio en el rango de 0 a 1, y aceptamos si E_2 es más pequeño que el número. Los pasos a seguir se conocen como el algoritmo de metrópolis descrito en la siguiente sección.

3.3.2.2. Algoritmo de Metrópolis

(1) Iniciar con una configuración y calcular E_1

(2) Iniciar ciclo de repetición.
Calcular E_2 usando otra configuración elegida al azar.

(I) Si $E_2 < E_1$ Conservar el estado E_2

(II) Si $E_2 > E_1$ calcular $e^{-(E2-E1)/(k_BT)}$

(a) Si $e^{-(E2-E1)/(k_BT)} > r$.
donde r es un número aleatorio entre 0 y 1, conservar E_2

(b) Si $e^{-(E2-E1)/(k_BT)} \leqslant r$.
conservar E_1

3.3.2.3. Generación de geometrías

El punto crucial de la implemantación es la generación de las geometrías. Empezamos con un esqueleto de la cadena de carbono donde los ángulos diedros están generados de manera aleatoria. En el segundo paso, agregamos los átomos de hidrógenos a esta cadena. Por el momento, empleamos los parámetros de sp_3 para los carbonos, es decir, los ángulos de enlace H-C-C es de 109.5° y el enlace H-C es $0.9A°$ los carbonos del extremo tiene tres átomos de hidrógeno y los demás tienen solo dos conectados. La distancia C-C es de 1.54° La Figura 3.14 muestra un ejemplo de la matriz Z de una cadena de cinco carbonos donde la última columna ángulos diedros se genera de manera aleatoria la Figura 3.15 muestra la gráfica del esqueleto especificado por esta matriz Z.

De la misma manera, la Figura 3.16 muestra la matriz Z completa con los $2n + 2$ átomos de hidrógeno agregados donde n es el número de carbono. La Figura 3.17 muestra la molécula completa. Ya que **MMF** sigue el formato TINKER, un cálculo empieza con el archivo de entrada en coordenadas cartesianas. Este formato es el de las bases de datos de proteína. Por este fin, usamos álgebra lineal y trigonometría para convertir la matriz Z a las coordenadas cartesianas seguimos el algoritmo de Frederic

A. van Catledge. Las coordenadas resultantes se muestran en la Figura 3.18. La Figura 3.19 muestra varios esqueletos de cadena de cinco carbonos.

Vale la pena mencionar que nuestro algoritmo no proporciona una manera de evitar generar geometrías no físicas, es decir es posible para una cadena más larga, los átomos de carbono queda con una distancia mucho más pequeña que la suma de sus radios de van der Waals. Sin embargo, consideramos que tal algoritmo es más costoso desarrollar y ejecutar comparando con el costo de evaluar la energía del campo AMOEBA09. Además, dicha geometría "no física" será rechazada por la condición de Boltzmann de cualquier manera.

Nuestra implementación tiene como *objetivo* probar nuestro programa **MMF** explorando una geometría de la cadena de pentano. A través de la cual aplicaremos los algoritmos desarrollados para la construcción de la molécula con la implementación de un módulo que genera una matriz Z de pentano. En donde primero se construyó la cadena de carbonos con $1.54A°$ como la distancia estándar de $C_{(sp_3)} - -C_{(sp_3)}$ y $109.5°$ el ángulo de enlace estándar para $C_{(sp_3)}$ ϕ_1 y ϕ_2 son dos números aleatorios que se generan para una configuración C_5

1	C						
2	C	1	1.5400				
3	C	2	1.5400	1	109.5000		
4	C	3	1.5400	2	109.5000	1	−73.3034
5	C	4	1.5400	3	109.5000	2	1.3517

Figura 3.14: Matriz Z de 5 Carbonos

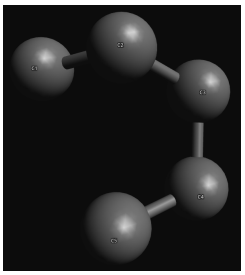


Figura 3.15: 5 Carbonos

1	C						
2	C	1	1.5400				
3	C	2	1.5400	1	109.5000		
4	C	3	1.5400	2	109.5000	1	−73.3034
5	C	4	1.5400	3	109.5000	2	1.3517
6	H	1	0.9000	2	109.5000	3	0.0000
7	H	1	0.9000	2	109.5000	3	−120.0000
8	H	1	0.9000	2	109.5000	3	120.0000
9	H	2	0.9000	1	109.5000	3	−120.0000
10	H	2	0.9000	1	109.5000	3	120.0000
11	H	3	0.9000	2	109.5000	4	−120.0000
12	H	3	0.9000	2	109.5000	4	120.0000
13	H	4	0.9000	3	109.5000	5	−120.0000
14	H	4	0.9000	3	109.5000	5	120.0000
15	H	5	0.9000	4	109.5000	3	0.0000
16	H	5	0.9000	4	109.5000	3	−120.0000
17	H	5	0.9000	4	109.5000	3	120.0000

Figura 3.16: Matriz Z con Hidrógenos

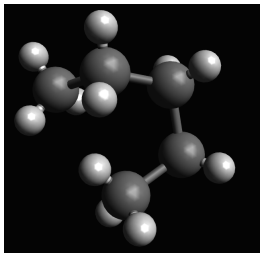


Figura 3.17: C_5H_{12}

Ahora que ya tenemos la cadena de carbonos y los hidrógenos agregados correctamente en una matriz Z. Debemos convertir a coordenadas cartesianas.

1	C	0.000000	0.000000	0.000000
2	C	1.540000	0.000000	0.000000
3	C	2.054063	1.451668	0.000000
4	C	1.832513	2.075465	−1.390465
5	C	1.139786	1.051548	−2.308791
6	H	−0.300426	0.848377	0.000000
7	H	−0.300426	−0.424189	0.734716
8	H	−0.300426	−0.424189	−0.734716
9	H	1.840426	−0.424189	−0.734716
10	H	1.840426	−0.424189	0.734716
11	H	2.932602	1.459267	0.195218
12	H	1.605853	1.929094	0.617391
13	H	2.626273	2.307506	−1.745574
14	H	1.313933	2.806893	−1.312435
15	H	0.998986	0.287496	−1.854479
16	H	1.663648	0.884259	−3.021239
17	H	0.351308	1.383646	−2.588100

Figura 3.18: Coordenadas cartesianas

Las figuras mostradas anteriormente se resumen en los siguientes pasos:

(1) Generación de cadenas o esqueletos de carbonos.
A partir de la matriz Z Figura 3.14 se pueden generar configuraciones como se muestra en la Figura 3.15 observe que los ángulos diedros ϕ fueron generados aleatoriamente. En la Figura 3.19 se muestran cadenas de carbono generadas con este procedimiento.

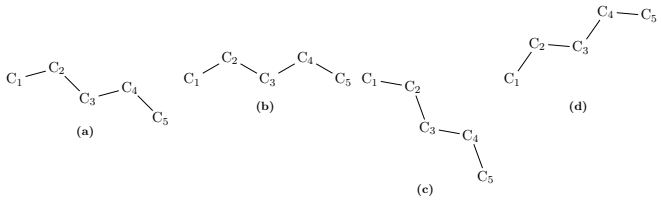


Figura 3.19: Esqueleto de Carbonos

Los parámetros de enlace, ángulo y ángulo diedro para el esqueleto de carbonos. Son los siguientes:

$$R_{c-c} = 1.53\mathring{A}$$
$$\theta_{c-c-c} = 109^{\circ}$$
$$\phi_{c-c-c-c} = (0 \quad a \quad 2\pi)$$

Donde ϕ es el ángulo diedro generado por un número aleatorio de $(0° - 360°)$ No verificamos si la geometría es viable es decir si la distancia entre dos carbonos es menor que la suma de sus radios de van der Waals, por la razón de que es más eficiente evaluar la energía de configuración y usar la condición de Boltzmann que verificar la geometría.

(2) Agregar Hidrógenos al esqueleto de carbonos.

El número de átomos conectados a un carbono depende de su valencia, y el número de hidrógenos en un alcano esta determinado por la formula $2(nc) + 2$ donde nc es el número de carbonos del alcano y el número de hidrógenos que corresponde a los extremos es seis si restamos esta cantidad al total de hidrógenos (n_h) podemos conocer el número que corresponde a la cadena que forma el alcano (n_{h2}) entonces definimos el número de hidrógenos como.

$$
\begin{aligned}
n_h &= 2 * nc + 2 \\
n_{h2} &= nh - 6 \\
R_{HC} &= 0.9\mathring{A} \\
\theta_{HCC} &= 109.5° \\
\theta_{HCH} &= 109.5°
\end{aligned}
$$

Con estos parámetros agregamos los hidrógenos a la matriz Z. Las Figuras 3.16 y Figura 3.17 muestran los hidrógenos agregados al esqueleto de carbono correctamente. Finalmente en la Figura 3.20 muestra ejemplos de las estructuras generadas.

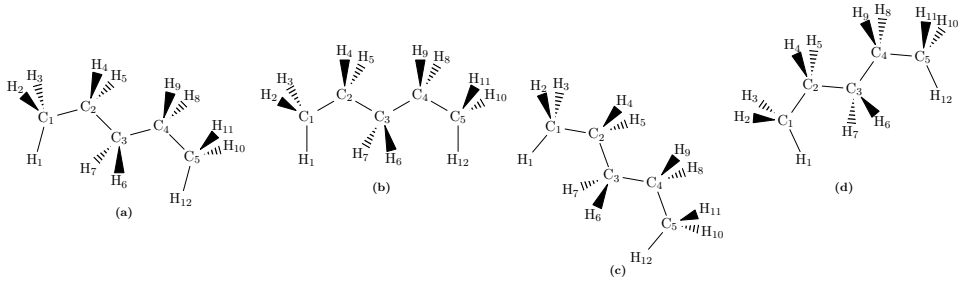


Figura 3.20: Esqueleto de Carbonos e Hidrógenos

(3) Conversión de coordenadas internas (matriz Z) a coordenadas cartesianas.
Para convertir las coordenadas internas a cartesianas seguimos el algoritmo Frederic A. van Catledge el cual se basa en cuatro átomos principalmente con los siguientes pasos.

a) Colocar el átomo 1 en origen $a = (0, 0, 0)$

b) Colocar el átomo 2 en eje x $b = (r_{12}, 0, 0)$

c) Colocar el átomo 3 en el plano xz $c = (x - \cos(\theta_{123}), 0, \sin(\theta_{123}))$

d) Transladar los 3 átomos para que el átomo 2 este en el origen

e) Rotar los 3 átomos para que 2 y 3 esten en el eje z

$$x_4 = r\sin(\theta_{234})\cos(\phi_{1234})$$
$$y_4 = r\sin(\theta_{234})\sin(\phi_{1234})$$
$$z_4 = z_3 - cos(\theta_{234})$$

para verificar que el algoritmo funciona mostramos las coordenadas (x, y, z) Figura 3.18 y evaluamos sus parámetros geométricos sección (2.3) distancia, ángulo y diedro. Esta evaluación se hizo con el módulo *ModGeom* del programa **MMF** que recibe como entrada un archivo con la geometría del pentano y nos devuelve la energía de interacción de cada uno de los átomos que participan en ella. El resultado es un archivo *n-alcano1.out*. Solo mostramos las secciones de enlace, ángulo y diedro del archivo. En la sección de torsión del programa **MMF** encontramos en la línea 28 la cuarteta de átomos que forman el ángulo diedro y el valor de $\mathbf{-73.3034}°$ correspondiente a ϕ_1, también la línea 31 la cuarteta de átomos y el valor de ϕ_2, generados aleatoriamente. Lo mismo ocurre con la conexión, en la sección *redenlace* se observa que el enlace $1 - -2$ la distancia tiene un valor de $\mathbf{1.5400}A°$. Con esto verificamos que la conversión y la implementación del algoritmo funcionan.

```
Iniciar MMF con la Molécula: n-alcano1.xyz
C           0.0000         0.0000         0.0000   29
C           1.5400         0.0000         0.0000   31
C           2.0541         1.4517         0.0000   31
C           1.8325         2.0755        -1.3905   31
C           1.1398         1.0515        -2.3088   29
H          -0.3004         0.8484         0.0000   30
H          -0.3004        -0.4242         0.7347   30
H          -0.3004        -0.4242        -0.7347   30
H           1.8404        -0.4242        -0.7347   32
H           1.8404        -0.4242         0.7347   32
H           2.9326         1.4593         0.1952   32
H           1.6059         1.9291         0.6174   32
H           2.6263         2.3075        -1.7456   32
H           1.3139         2.8069        -1.3124   32
```

```
H           0.9990         0.2875        -1.8545   30
H           1.6636         0.8843        -3.0212   30
H           0.3513         1.3836        -2.5881   30
 ================================================

        redenlace       kb       bo  distancia_actual  energia
 ================================================
   1   C  C  1     2  323.0000   1.5247  1.5400    0.0727
   2   C  H  1     6  341.0000   1.1020  0.9000   23.2350
   3   C  H  1     7  341.0000   1.1020  0.9000   23.2350
   4   C  H  1     8  341.0000   1.1020  0.9000   23.2350
   5   C  C  2     3  323.0000   1.5247  1.5400    0.0727
   6   C  H  2     9  341.0000   1.1120  0.9000   26.2239
   7   C  H  2    10  341.0000   1.1120  0.9000   26.2239
   8   C  C  3     4  323.0000   1.5247  1.5400    0.0727
   9   C  H  3    11  341.0000   1.1120  0.9000   26.2241
  10   C  H  3    12  341.0000   1.1120  0.9000   26.2239
  11   C  C  4     5  323.0000   1.5247  1.5400    0.0727
  12   C  H  4    13  341.0000   1.1120  0.9000   26.2238
  13   C  H  4    14  341.0000   1.1120  0.9000   26.2237
  14   C  H  5    15  341.0000   1.1020  0.9000   23.2350
  15   C  H  5    16  341.0000   1.1020  0.9000   23.2350
  16   C  H  5    17  341.0000   1.1020  0.9000   23.2349
Energia Total de enlace :      297.0439
 ================================================

       redangulo      KB    anguloideal angulo_actual energia
 ================================================
   1   2     1     6  42.4400   110.7000  109.5000     0.0189  C  C  H
   2   2     1     7  42.4400   110.7000  109.5000     0.0189  C  C  H
   3   2     1     8  42.4400   110.7000  109.5000     0.0189  C  C  H
   4   1     2     3  48.2000   111.0000  109.5000     0.0337  C  C  C
   5   1     2     9  42.4400   109.3100  109.5000     0.0005  C  C  H
   6   1     2    10  42.4400   109.3100  109.5000     0.0005  C  C  H
   7   6     1     7  39.5700   107.8000  109.4425     0.0318  H  C  H
```

8	6	1	8	39.5700	107.8000	109.4425	0.0318	H	C	H
9	7	1	8	39.5700	107.8000	109.4424	0.0318	H	C	H
10	3	2	9	42.4400	109.3100	109.4424	0.0002	C	C	H
11	3	2	10	42.4400	109.3100	109.4424	0.0002	C	C	H
12	2	3	4	48.2000	111.0000	109.5000	0.0337	C	C	C
13	2	3	11	42.4400	109.3100	109.5000	0.0005	C	C	H
14	2	3	12	42.4400	109.3100	109.5000	0.0005	C	C	H
15	9	2	10	39.5700	107.6000	109.4424	0.0399	H	C	H
16	4	3	11	42.4400	109.3100	109.4425	0.0002	C	C	H
17	4	3	12	42.4400	109.3100	109.4424	0.0002	C	C	H
18	3	4	5	48.2000	111.0000	109.5000	0.0337	C	C	C
19	3	4	13	42.4400	109.3100	109.5000	0.0005	C	C	H
20	3	4	14	42.4400	109.3100	109.5000	0.0005	C	C	H
21	11	3	12	39.5700	107.6000	109.4425	0.0399	H	C	H
22	5	4	13	42.4400	109.3100	109.4425	0.0002	C	C	H
23	5	4	14	42.4400	109.3100	109.4425	0.0002	C	C	H
24	4	5	15	42.4400	110.7000	109.5000	0.0189	C	C	H
25	4	5	16	42.4400	110.7000	109.5000	0.0189	C	C	H
26	4	5	17	42.4400	110.7000	109.5000	0.0189	C	C	H
27	13	4	14	39.5700	107.6000	109.4424	0.0399	H	C	H
28	15	5	16	39.5700	107.8000	109.4424	0.0318	H	C	H
29	15	5	17	39.5700	107.8000	109.4424	0.0318	H	C	H
30	16	5	17	39.5700	107.8000	109.4425	0.0318	H	C	H

Energía Total Angulo: 0.5292

	redtorsion				energia	Angulo_actual
1	6	1	2	3	0.3410	0.0000
2	6	1	2	9	0.2990	-120.0000
3	6	1	2	10	0.2990	120.0000
4	7	1	2	3	0.3410	-120.0000
5	7	1	2	9	0.2990	119.9999

6	7	1	2	10	0.2990	0.0000
7	8	1	2	3	0.3410	120.0000
8	8	1	2	9	0.2990	0.0000
9	8	1	2	10	0.2990	-119.9999
10	9	2	3	4	0.0125	46.7319
11	9	2	3	11	0.0346	-73.2682
12	9	2	3	12	0.0346	166.7318
13	10	2	3	4	0.0126	166.6614
14	10	2	3	11	0.0350	46.6613
15	10	2	3	12	0.0350	-73.3387
16	11	3	4	5	0.3406	121.3869
17	11	3	4	13	0.2986	1.3869
18	11	3	4	14	0.2986	-118.6131
19	12	3	4	5	0.3406	-118.6835
20	12	3	4	13	0.2986	121.3164
21	12	3	4	14	0.2986	1.3165
22	13	4	5	15	0.2990	120.0352
23	13	4	5	16	0.2990	0.0353
24	13	4	5	17	0.2990	-119.9648
25	14	4	5	15	0.2990	-120.0352
26	14	4	5	16	0.2990	119.9648
27	14	4	5	17	0.2990	-0.0353
28	4	3	2	1	0.2191	-73.3034
29	11	3	2	1	0.0397	166.6966
30	12	3	2	1	0.0397	46.6966
31	5	4	3	2	0.9615	1.3517
32	13	4	3	2	0.1079	-118.6484
33	14	4	3	2	0.1079	121.3517
34	15	5	4	3	0.3410	0.0000
35	16	5	4	3	0.3410	-120.0000
36	17	5	4	3	0.3410	120.0000

Energía Total torsion : 9.1499

(4) Evaluar energía E_1
Ahora que hemos verificado que la conversión funciona correctamente. Estamos listos para evaluar la energía de esta geometría E_1 que corresponde a la energía total de la primera conformación con un valor de $E_1 = 327.81968432864704$ $kcal/mol$

(5) Generar nueva geometría variando los ángulos diedros ϕ_1, ϕ_2
Al variar los ángulos diedros cambiamos la matriz Z entonces debemos convertir a coordenadas cartesianas para evaluar E_2. Energía total del paso 2: $E_2 = 325.06847330100265$ $kcal/mol$. Mostramos la geometría de ambas conformaciones. Observe que la "geometría" para ambas conformaciones son distintas. Porque los ángulos diedros ϕ_1 y ϕ_2 han cambiado desde la matriz Z **matz 1** y **matz 2**

```
matz 1:   -73.303399999999996        1.3516999999999999
   0.0000000000000000        0.0000000000000000        0.0000000000000000
   1.5400000000000000        0.0000000000000000        0.0000000000000000
   2.0540625632200076        1.4516678962819547        1.4516678962819547
   1.8325131290520011        2.0754652673517882        2.0754652673517882
   1.1397858550912674        1.0515480200998839        1.0515480200998839
  -0.3004261733103939        0.8483773419829605        0.8483773419829605
  -0.3004261733103939       -0.4241886709914801       -0.4241886709914801
  -0.3004261733103939       -0.4241886709914801       -0.4241886709914801
   1.8404261733103939       -0.4241886709914801       -0.4241886709914801
   1.8404261733103939       -0.4241886709914801       -0.4241886709914801
   2.9326023685234297        1.4592673497941371        1.4592673497941371
   1.6058526519959120        1.9290940655163704        1.9290940655163704
   2.6262728922899159        2.3075057756598558        2.3075057756598558
   1.3139331063635191        2.8068927420706649        2.8068927420706649
   0.9989856490027461        0.2874956079442641        0.2874956079442641
   1.6636479899976573        0.8842591127335854        0.8842591127335854
   0.3513082040712598        1.3836460791443943        1.3836460791443943

matz 2:  -57.060659418969024      -60.289223650264390

   0.0000000000000000        0.0000000000000000        0.0000000000000000
   1.5400000000000000        0.0000000000000000        0.0000000000000000
```

```
 2.0540625632200076       1.4516678962819547       0.0000000000000000
 1.4815902457374248       2.1997335778595031      -1.2183075070613492
 1.9384563670666568       1.5005093043299764      -2.5121230899357743
-0.30042617331039390      0.84837734198296055      0.0000000000000000
-0.30042617331039390     -0.42418867099148011      0.73471633015236226
-0.30042617331039390     -0.42418867099148011     -0.73471633015236226
 1.8404261733103939      -0.42418867099148011     -0.73471633015236226
 1.8404261733103939      -0.42418867099148011      0.73471633015236226
 2.9530104440757396       1.4520404712114541      -4.3503662982660563E-002
 1.7905293783806913       1.8636966067893497       0.75550155671981223
 1.7786232373278643       3.0493016996347202      -1.2159819167786465
 0.58252055791413859      2.1966053109414911      -1.1775164494208821
 2.4512707611098583       0.79051601927528536     -2.3049252800068309
 2.4137899972968242       2.0772451092732078      -3.0135547914352845
 1.2176873178830983       1.2245487205799785      -2.9750893240775200

 Energía total del paso 2:  325.06847330100265
```

(6) Finalmente comparamos E_1 y E_2 siguiento el algoritmo (3.3.2.2) y aplicamos la condición de Boltzmann.

Capítulo 4

Análisis de Resultados

4.1. Energía de AMOEBA09

Los resultados de nuestra implementación con los Módulos descritos anteriormente estan diseñados para calcular las energías de interacción de tipo intramolecular reunidos en un programa principal al que llamamos **MMF**, los resultados son consistentes con el programa *Tinker* para las moléculas de Amonio, Eter, Indole y Benzamidina. Como se muestra en la tabla 4.1 y 4.2, Para el dímero de amonio se muestran dos tipos de energía U_b y U_θ, seis interacciones de tipo enlace y seis interacciones de tipo ángulo, para el Eter cuatro tipos de energía U_b, U_θ, $U_{b\theta}$ y U_ϕ

Tinker ...*						
Molécula	U_b	U_θ	$U_{b\theta}$	U_{oop}	U_ϕ	**Interacciones**
Dimero amoniaco	0.0096	0.0134				6 6
Eter	0.0620	1.8634	0.1324		0.0017	8 13 7 6
Indole	1.3240	18.8126	−0.2228	0.0000	−12.1200	17 27 27 24 40
Benzamidina	3.5309	1.0182	−0.3504	0.0004	0.8365	18 27 18 27 36

*. Energía kcal/mol.

Cuadro 4.1: Tinker

MMF ...*						
Molécula	U_b	U_θ	$U_{b\theta}$	U_{oop}	U_ϕ	**Interacciones**
Dimero amoniaco	0.0096	0.0134				6 6
Eter	0.0620	1.8634	0.1324		0.0017	8 13 7 6
Indole	1.3240	18.8126	−0.2228	0.0000	−12.1200	17 27 27 24 40
Benzamidina	3.5309	1.0182	−0.3504	0.0004	0.8365	18 27 18 27 36

*. Energía kcal/mol.

Cuadro 4.2: MMF

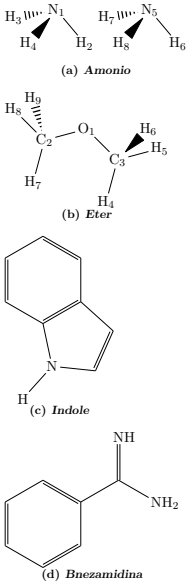


Figura 4.1: Estructuras químicas

4.2. Simulación de Monte Carlo

Nuestra simulación para la molécula de pentano

```
 ++++++++++++++++++
 real: x y,z
   0.0000000000000000        0.0000000000000000        0.0000000000000000
   1.5400000000000000        0.0000000000000000        0.0000000000000000
   2.0540630000000002        1.4516680000000000        0.0000000000000000
   3.5940629999999998        1.4516680000000000        0.0000000000000000
   4.1081250000000002        2.9033359999999999        0.0000000000000000
 -0.30042600000000003      -0.42418899999999998      -0.73471600000000004
 -0.30042600000000003       0.84837700000000005        0.0000000000000000
 -0.30042600000000003      -0.42418899999999998       0.73471600000000004
   1.8404259999999999      -0.42418899999999998       0.73471600000000004
   1.8404259999999999      -0.42418899999999998      -0.73471600000000004
   1.7536360000000000        1.8758570000000001       0.73471600000000004
   1.7536360000000000        1.8758570000000001      -0.73471600000000004
   3.8944890000000001        1.0274790000000000       0.73471600000000004
   3.8944890000000001        1.0274790000000000      -0.73471600000000004
   3.4086940000000001        3.4697239999999998       -0.0000000000000000
   4.6082669999999997        3.0449329999999999       0.73471600000000004
   4.6082669999999997        3.0449329999999999      -0.73471600000000004
denlac   16     297.0440
dangul   30       0.5292
dstr   21      21.0995
dtor   36       5.6340
doop    0       0.0000
 ****************************
 energia inicial emin   324.30664139692851
```

partiendo de esta geometría calculamos la contribución de las cinco energías correspondientes y su energía minima "global' es decir la energía más baja de toda su trayectoría de la simulación Monte Carlo. Posteriormente como se describio en la sección (3.3.2.3) generamos una perturbación cambiando los ángulos diedros en la matriz

Z, con esta nueva matriz se tiene una nueva geometría y calculamos la energía comparando ambas estos pasos repitieron 9000 veces para finalmente llegar a la siguiente configuración.

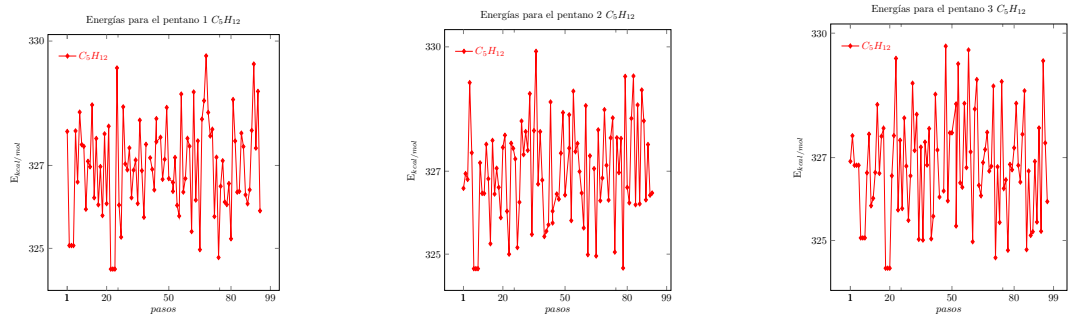
```
   0.0000000000000000        0.0000000000000000        0.0000000000000000
   1.5400000000000000        0.0000000000000000        0.0000000000000000
   2.0540625632200076        1.4516678962819547        0.0000000000000000
   3.5939342554914044        1.4517133324316522        1.9878865077780174E-002
   4.1077968022292577        2.9033693603905792        3.5372720330800626E-002
 -0.30042617331039390      -0.42418867099148011      -0.73471633015236226
 -0.30042617331039390       0.84837734198296055        0.0000000000000000
 -0.30042617331039390      -0.42418867099148011       0.73471633015236226
   1.8404261733103939      -0.42418867099148011       0.73471633015236226
   1.8404261733103939      -0.42418867099148011      -0.73471633015236226
   1.7441318197741471        1.8713558725771431       0.73335185062856800
   1.7630991588540361        1.8803237346213657      -0.73593102229577256
   3.8848389045827045        1.0308964100049542       0.76034331590333526
   3.9038068614626273        1.0241795041908122      -0.70895156360802591
   3.4083623827511511        3.4697265281507130        2.9800344172994132E-002
   4.5983982890426391        3.0404946365039858       0.77733420166748146
   4.6173656281225277        3.0494624985482082      -0.69194867125685933
denlac   16     297.0433
dangul   30       0.5420
dstr   21      21.0748
dtor   36       5.6346
doop    0       0.0000
 energia total:    324.29471744600556
```

En la Figura 4.2 gráficamos la energía vs. Los pasos para tres simulaciones donde empezamos con tres diferentes conformaciones de la cadena de carbono. Observamos que la energía varia entre 330 y 320 $kcal/mol$ implicando que las conformaciones generadas son muy razonables. Dos cosas indican que la energía es muy razonable. La energía es compatible con la de AMOEBA (sin las interacciones no enlazantes) y los carbonos se extienden para minimizar la repulsión (efecto estérico) como aprendimos en química orgánica. En tres simulaciones, la energía "mínima global" se encuentra muy cerca del inicio. Una explicación posible es que la geometría inicial Figura 4.3 ya esta muy cerca a esta energía mínima global. La idea es que probamos la implemen-

tación, no queremos saber si el método de MC funciona. Puesto que la teoría de MC nos lleva a la energía minima independientemente de la configuración. Suponemos que para moléculas más complicadas, este no es necesariamente el caso

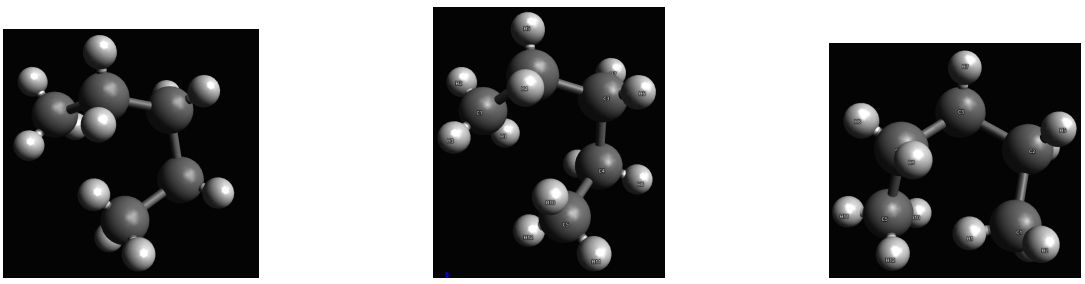
La Figura 4.4 muestra las configuraciones finales para las tres simulaciones. Podemos ver que las tres son muy similares a pesar de que empezamos con diferentes geometrías iniciales. En esta configuración final, los cinco carbonos se extienden de manera que la distancia de C_1 y C_5 es máxima de acuerdo con la explicación simple basado en la repulsión por sobre población de los átomos muy cercanos.

Para referencia, incluimos las coordenadas cartesianas de las geometrías iniciales y finales en las Figuras 4.5 - 4.10, hemos diseñado e implementado un algoritmo simple de la simulación Monte Carlo para un alcano. Los resultados confirman que para casos sencillos de pentano, el programa funciona correctamente.



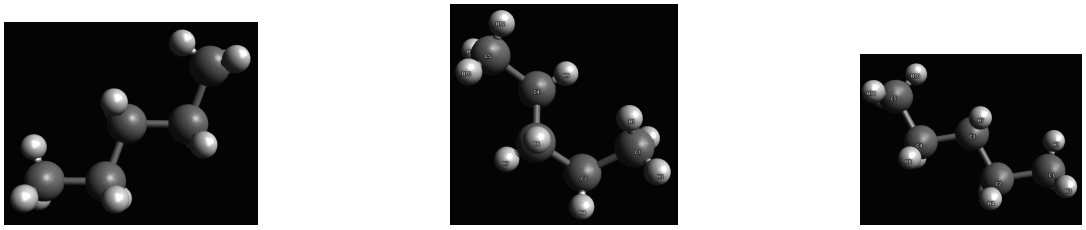
(a) **(b)** **(c)**

Figura 4.2: Energías Pentanos



(a) ***n1*** C_5H_{12} **(b)** ***n2*** C_5H_{12} **(c)** ***n3*** C_5H_{12}

Figura 4.3: Conformación Inicial



(a) ***n1*** C_5H_{12} **(b)** ***n2*** C_5H_{12} **(c)** ***n3*** C_5H_{12}

Figura 4.4: Conformación Final

1	C	0.000000	0.000000	0.000000
2	C	1.540000	0.000000	0.000000
3	C	2.054063	1.451668	0.000000
4	C	1.832513	2.075465	−1.390465
5	C	1.139786	1.051548	−2.308791
6	H	−0.300426	0.848377	0.000000
7	H	−0.300426	−0.424189	0.734716
8	H	−0.300426	−0.424189	−0.734716
9	H	1.840426	−0.424189	−0.734716
10	H	1.840426	−0.424189	0.734716
11	H	2.932602	1.459267	0.195218
12	H	1.605853	1.929094	0.617391
13	H	2.626273	2.307506	−1.745574
14	H	1.313933	2.806893	−1.312435
15	H	0.998986	0.287496	−1.854479
16	H	1.663648	0.884259	−3.021239
17	H	0.351308	1.383646	−2.588100

Figura 4.5: Geometría $n1$ $(C_5H_{12})_i$

1	C	0.000000	0.000000	0.000000
2	C	1.540000	0.000000	0.000000
3	C	2.054062	1.451667	0.000000
4	C	3.566441	1.461449	−0.290199
5	C	4.092653	2.907380	−0.227075
6	H	−0.3004261	0.848377	0.000000
7	H	−0.3004261	−0.424188	0.734716
8	H	−0.3004261	−0.424188	−0.734716
9	H	1.840426	−0.424188	−0.734716
10	H	1.840426	−0.424188	0.734716
11	H	1.901010	1.824573	0.804684
12	H	1.624109	1.922628	−0.635087
13	H	3.717127	1.123360	−1.110558
14	H	3.993340	0.960237	0.323432
15	H	3.414104	3.465813	−3.285008
16	H	4.447802	3.132837	−1.022712
17	H	4.724015	2.969714	0.411278

Figura 4.6: Geometría $n1$ $(C_5H_{12})_f$

1	C	0.000000	0.000000	0.000000
2	C	1.540000	0.000000	0.000000
3	C	2.054063	1.451668	0.000000
4	C	1.116249	2.329108	−0.849826
5	C	0.782680	1.600903	−2.165139
6	H	−0.300426	0.848377	0.000000
7	H	−0.300426	−0.424189	0.734716
8	H	−0.300426	−0.424189	−0.734716
9	H	1.840426	−0.424189	−0.734716
10	H	1.840426	−0.424189	0.734716
11	H	2.883968	1.476490	−0.347334
12	H	2.073083	1.763639	0.843986
13	H	1.521973	3.110885	−1.034792
14	H	0.356615	2.486424	−0.393528
15	H	1.200606	0.803994	−2.181675
16	H	1.058786	2.098498	−2.862394
17	H	−0.106573	1.474037	−2.221130

Figura 4.7: Geometría $n2$ $(C_5H_{12})_i$

1	C	0.000000	0.000000	0.000000
2	C	1.540000	0.000000	0.000000
3	C	2.054062	1.451667	0.000000
4	C	1.627534	2.148052	1.305650
5	C	2.082313	3.619070	1.275963
6	H	−0.300426	0.848377	0.000000
7	H	−0.300426	−0.424188	0.734716
8	H	−0.300426	−0.424188	−0.734716
9	H	1.840426	−0.424188	−0.734716
10	H	1.840426	−0.424188	0.734716
11	H	1.706213	1.893554	−0.702664
12	H	2.952034	1.452386	−6.037824
13	H	2.007840	1.731002	2.006675
14	H	0.731824	2.112971	1.386100
15	H	2.509021	3.786030	0.501337
16	H	2.640046	3.775059	1.964876
17	H	1.364030	4.157027	1.344300

Figura 4.8: Geometría $n2$ $(C_5H_{12})_f$

1	C	0.000000	0.000000	0.000000
2	C	1.540000	0.000000	0.000000
3	C	2.054063	1.451668	0.000000
4	C	0.979551	2.377515	0.599861
5	C	−0.184799	2.527069	−0.396895
6	H	−0.300426	0.848377	0.000000
7	H	−0.300426	−0.424189	0.734716
8	H	−0.300426	−0.424189	−0.734716
9	H	1.840426	−0.424189	−0.734716
10	H	1.840426	−0.424189	0.734716
11	H	2.232283	1.707264	−0.844339
12	H	2.804656	1.504575	0.493771
13	H	1.338132	3.186273	0.765181
14	H	0.672582	2.023203	1.368127
15	H	−0.011125	2.044340	−1.136361
16	H	−0.279576	3.393731	−0.620308
17	H	−0.945126	2.230662	−0.017362

Figura 4.9: Geometría $n3$ $(C_5H_{12})_i$

1	C	0.000000	0.000000	0.000000
2	C	1.540000	0.000000	0.000000
3	C	2.054062	1.451667	0.000000
4	C	3.594059	1.451669	3.215516
5	C	4.107887	2.899064	0.115583
6	H	−0.300426	0.848377	0.000000
7	H	−0.300426	−0.424188	0.734716
8	H	−0.300426	−0.424188	−0.734716
9	H	1.840426	−0.424188	−0.734716
10	H	1.840426	−0.424188	0.734716
11	H	1.752955	1.877002	0.733774
12	H	1.756024	1.875915	−0.735654
13	H	3.896082	1.085057	−0.761230
14	H	3.893023	0.972400	0.703874
15	H	3.408366	3.463785	0.157545
16	H	4.609534	3.096573	−0.605069
17	H	4.606475	2.983917	0.860035

Figura 4.10: Geometría $n3$ $(C_5H_{12})_f$

Capítulo 5

Conclusiones

Hemos implementado el campo de fuerzas AMOEBA09 en el lenguaje de programación Fortran 90 como resultado 1427 líneas de código organizado en 13 módulos 12 funciones y 37 subrutinas. Que permiten calcular la energía de una molécula nuestros resultados son consistentes con los valores reportado en la literatura para las moléculas de amonio, nitrilo, benzamidina, y con nuestro campo de fuerza, implementamos una simulación tipo Monte Carlo que consiste en los algoritmos de generar nuevas configuraciones moleculares y el algoritmo de Metropolis. Probamos con un sencillo ejemplo de la molécula de pentano.

Bibliografía

[1] Baker, C. M. (2015). Polarizable force fields for molecular dynamics simulations of biomolecules. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(2):241–254. 4

[2] Beddard, G. S. (2011). Using the metropolis algorithm to calculate thermodynamic quantities: An undergraduate computational experiment. *Journal Chemical Education*, 8:574–580. 48

[3] Chapman, S. (2018). *Fortran for Scientists and Engineers.* Mcgraw-Hill, fourth edition. 15, 16, 17, 19

[4] Jensen, F. (2007). *Introducction to Computational Chemistry.* Wiley, second edition. 48

[5] Levit, M. (2001). The birth of computational structural biology. *Nature structure molecular biology.*, 8:392. 4

[6] Mulholland, A. J. (2008). Computational enzymology: modelling the mechanisms of biological catalysts. *Biochemical Society Transactions*, 36(1):22–26. 4

[7] Ponder, J. W. (2010). Current status of the amoeba polarizable force field. *J. Phys. Chem. B*, 114:2549. 5

[8] Schlick, T. (2002). *Molecular Modeling and Simulation: An Interdisciplinary Guide.* Springer, 2 edition. 48

[9] Shi, Y., Wu, C., Ponder, J. W., and Ren, P. (2011). Multipole electrostatics in hydration free energy calculations. *Journal of Computational Chemistry*, 32(5):967–977. 4

[10] Walter R. P Scott, P. H. H. (1999). The gromos biomolecular simulation program package. *J. Phys. Chem. B*, 103:3596–3607. 4

[11] Warshel, A. (2003). Computer simulations of enzyme catalysis: Methods, progress, and insights. *Annual Review of Biophysics and Biomolecular Structure*, 32(1):425–443. PMID: 12574064. 4

Universidad Autónoma del Estado de Morelos
Av. Universidad 1001
Col. Chamilpa
62209 Cuernavaca, Morelos, México

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

10 Marzo 2020

Froylán Oswaldo Martínez Vázquez
Lic. Bioquímica y Biología Molecular
Centro de Investigación en Dinámica Celular

Comité Tutorial

APROBACIÓN DE TESIS

Toda vez que el trabajo de **Tesis** realizado por el C. **Froylán Oswaldo Martínez Vázquez** de la Licenciatura en **Ciencias, Bioquímica y biología molecular**, con numero de matrícula **2009500073**, y que lleva por título "**SIMULACIÓN MONTE CARLO USANDO EL CAMPO DE FUERZA POLARIZABLE AMOEBA UN PROGRAMA EN FORTRAN90**" ha sido concluido en fondo y forma, me permito emitir mi **VOTO APROBATORIO**

Dr. Humberto Saint-Martin Posada
PRESIDENTE

Dr. Minhhuy Hô
SECRETARIO

Dr. Ramón Hernández Lamoneda
VOCAL

Dr. Jorge Hernández Cobos
SUPLENTE

Saludos.

Froylán Oswaldo Martínez Vázquez.
Estudiante

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

RAMON HERNANDEZ LAMONEDA | Fecha:2020-11-07 13:01:03 | Firmante

VCpJsEpIqpfUkgFcqTmwlcjiWA+9nLG9aRoLoVg5pUhpChMhwi8KahCRIQOJ88iOO3kAS+ftw1TvSC91Pn/IfNsYY9mpD4yZ3V4SPSMEETMEqyfxPEeI3NPMEY1c7zKsFGUoSO2ZdLGf81QVmA4bPcWG87K16z2w4sKG60EFXZFLYChvxhlKPOo6Eh7XsFTwt+750Rn5d9R200riKqYYtQ0aQWEEeTj0BKHmlNQnfXJLvp6uXOFLzkj3k3u74szgYh8/H76v83GuP8hQ626Li98Os3rykdrUQQS+g1ahFyvreinZKCnqAafZy3sO77tEzV1/z+rEzOnbLLMOONDPxQ==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:

Wa7YHJ

https://efirma.uaem.mx/noRepudio/XLWy48HsbKy5cyGhpBnFtOYcTUQN2FDL