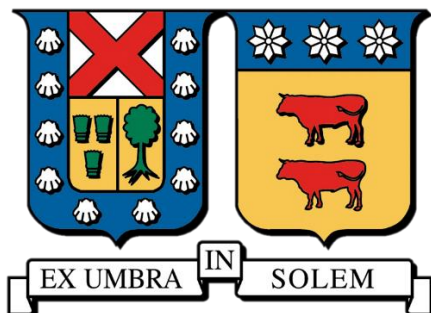


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INGENIERÍA MECÁNICA
VALPARAÍSO – CHILE



**“DESARROLLO E IMPLEMENTACIÓN DE METODOLOGÍA PARA
GENERAR MALLAS DE SUPERFICIES MOLECULARES DE MANERA
ÓPTIMA”**

ELMO JAVIER BEGAZO RIVERA

MEMORIA DE TITULACIÓN PARA OPTAR AL
TÍTULO DE: INGENIERO CIVIL MECÁNICO

PROFESOR GUÍA

CHRISTOPHER COOPER V.

PROFESOR CORREFERENTE

MICHAEL KARKULIK.

ENERO - 2022

Índice

Introducción	1
Objetivos	2
1. Marco Teórico	3
Figura 1.1: Diferencia gráfica entre ambos métodos	3
1.1 Modelo de Solvente Implícito	4
1.1.1 Ecuación de Poisson-Boltzmann	5
1.2 Condiciones de frontera	7
Figura 1.2: Frontera de Ω	7
1.3 Tipos de Energía a considerar	8
1.3.1 Energía de Coulomb	9
1.3.2 Energía de Solvatación	9
Figura 1.3: Diagrama de solvatación para representar las energías asociadas	10
1.4 Método de elementos de borde (BEM)	11
Figura 1.4: Sketch de una interfaz discretizada	11
1.4.1 Ecuación de Laplace	12
1.4.2 Espacios de las soluciones	14
1.4.3 Tipos de operadores a utilizar	16
1.5 Modelo para una interfaz única	17
2. Metodología	19
Figura 2.1 Tipos de Superficies Moleculares: Van der Waals (VdW), Superficie Accesible al Solvente (SAS) y Superficie Excluida de Solvente (SES)	19
2.1 MSMS	20
2.2 NanoShaper	20
2.3 Estimación de Error	22
2.3.1 Método de la ecuación adjunta	22
2.3.2 Estimación del potencial y su relación con el adjunto	25
2.3.3 Estimación del error local y refinamiento	26
Figura 3.2: Evolución de la malla y su algoritmo de refinamiento	28

2.3.4 Refinamiento adaptativo y no adaptativo.....	28
Figura 2.2: Representación gráfica de refinamiento no adaptativo vs refinamiento adaptativo.....	29
3. Análisis de los resultados.....	30
3.1 Código	30
Figura 3.1: Secuencia de construcción de la malla.....	30
Tabla 3.1: Resultados prueba lado derecho. Python 2, Bempp 3.3.4.	31
Tabla 3.2: Resultados prueba lado derecho. Python 3, Bempp 0.2.4.	31
3.2 Análisis del Criterio a Seleccionar	32
Tabla 3.3: Evaluación de criterios bajo Potencial constante	33
Tabla 3.4: Evaluación de criterios bajo Radio de curvatura constante.	33
3.3 Análisis del Refinamiento por Potencial	34
3.2.1 Metanol.....	34
Tabla 3.5: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 0.5 y 0.6 respectivamente.....	34
Tabla 3.6: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 1.0 y 1.5 respectivamente.....	35
Tabla 3.7: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 2.0, 2.5, 3.0 y 3.5 respectivamente.	36
3.2.2 Arginina	36
Tabla 3.8: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 0.5 generada en MSMS.....	37
Tabla 3.9: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 1.0 generada en MSMS.....	37
Tabla 3.10: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para Grid Scale 1.1 generada en Nanoshaper.	38
4. Conclusiones	39

Introducción

Gran parte de los procesos químicos y biológicos ocurren en medios acuosos y se han desarrollado un gran número de métodos para predecir energías de disolución de interés, tales como la Energía Libre de Solvatación [9][10][11], Energía de Unión, entre otras.

En el presente informe, se realizará un enfoque a la Energía Libre de Solvatación, la cual en términos simples corresponde al cambio de energía libre del soluto en el vacío comparado con la energía libre inmersa en el solvente debido a factores polares.

Actualmente existen diversos modelos que estudian la estimación de estas energías, los más importantes son los modelos explícitos y los modelos implícitos. La mayor diferencia entre ambos es la consideración del solvente, en el caso implícito, éste se compone de un dieléctrico continuo al cual se le introduce el soluto, y en el otro caso, se consideran explícitamente las moléculas del solvente. Cada modelo tiene sus ventajas y desventajas, el caso explícito representará de mejor manera y más cercana a la realidad los fenómenos asociados. Sin embargo, esta precisión se contrapone con los grandes tiempos de cálculo necesarios cuando la molécula estudiada es de gran tamaño.

A continuación, se estudiará principalmente el uso de modelos implícitos, como el aplicado por Yoon & Lenhoof [9], el cual implementa el método de elementos de frontera (BEM). Una de las ventajas de este método al compararlo con otros como puede ser el método de elementos finitos (FEM), es que, al trabajar en la frontera, las condiciones de borde están bien definidas y de esta manera no es necesario condiciones de contorno lejanas, además reduce una dimensión al problema y el tamaño de la malla es menor. Un problema para el caso de BEM es que se utilizan matrices densas, lo que significa que para resolver un problema con BEM, el tiempo y recursos a invertir generalmente serán mayores.

En este trabajo, se evaluará el mallado que se le debe dar a un problema de solvente implícito, es decir, el mallado a utilizar en diferentes superficies moleculares, para esto se presentarán diferentes herramientas disponibles, así como sus pros y sus contras.

Para esto, se utilizará la librería de Python BEMPP, así como también el análisis de softwares como MSMS, Nanoshaper, entre otros. BEMPP permite elegir una serie de espacios polinomiales para trabajar la malla superficial, llegando hasta orden 10[3]. La ventaja de esta herramienta es su sencilla implementación para este tipo de problemas, facilitando enormemente la discretización de los operadores asociados.

Objetivos

Los principales objetivos de esta investigación son:

1. Estudio de diferentes herramientas para el mallado de superficies moleculares como MSMS y Nanoshaper.
2. Actualizar código de refinamiento adaptativo desde Python 2 y una versión previa de BEMPP a Python 3 y BEMPP-CL 0.2.3.
3. Evaluar pros y contras de los softwares ya mencionados a la hora de generar mallas.
4. Evaluar distintos criterios y su relevancia en el refinamiento por error.

Este trabajo está pensado en el ahorro de recursos. Si bien las herramientas computacionales van en constante mejora, teniendo un mayor número de núcleos, más memoria disponible, mejores procesadores, entre otros; el uso de estos permite abarcar un mayor número de casos, especialmente en moléculas muy grandes, evitando la necesidad de utilizar un hardware más avanzado.

1. Marco Teórico

La electrostática representa una base importante para un gran número de aplicaciones, ya sean físicas, químicas, biomédicas, entre otras. A una pequeña escala, las moléculas se encuentran separadas por fuerzas atómicas inducidas por las interacciones entre sus propias cargas con el medio. Donde cada carga afecta a otra, logrando diferentes combinaciones y grados de libertad.

Dentro de los modelos más implementados, están los de mecánica molecular, estos consideran a los átomos e iones del solvente de manera explícita, significando un uso importante de recursos computacionales y tiempo.

Paralelamente se encuentran los métodos de solvente implícito, que sirven como una herramienta bastante precisa cuando no es posible utilizar métodos explícitos[12][13], como ocurre en los casos de moléculas de gran tamaño.

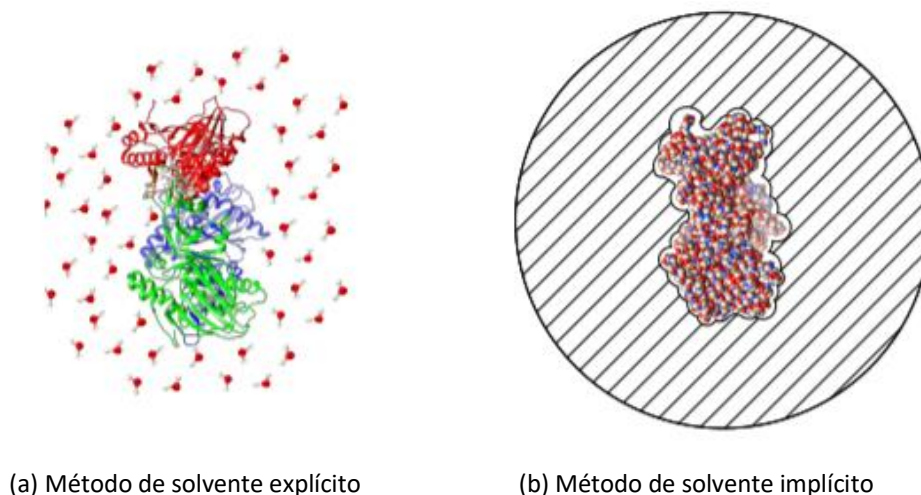


Figura 1.1: Diferencia gráfica entre ambos métodos[6].

Cada método cuenta con ventajas y desventajas, enfocadas principalmente en la representación del problema físico y el uso de recursos. Para ambos casos han surgido diferentes herramientas para mejorar tiempos de cálculo y mejorar la representación del problema. Desde las primeras publicaciones a mediados del siglo 20 por autores como John G. Kirkwood[14] y William McMillan junto a Joseph Mayer[15], y por autores más contemporáneos como Yoon A. M. Lenhoof, Byung Yung[9], M. Altman, Jaydeep Bardhan[10] y G. M. Ullman[13], Gilson y Hoing[12][16].

1.1 Modelo de Solvente Implícito

Molecularmente, tanto proteínas como moléculas están ubicadas en un medio, el cual está compuesto de moléculas de agua, iones y otros elementos, los cuales vienen a determinar las propiedades físicas del medio. Al considerarlo como un medio continuo dieléctrico con una alta polarización ($\epsilon = 80$), el potencial electrostático promedio representa de buena forma el comportamiento de las propiedades físicas de la molécula y su interacción con el solvente[11]. Este modelo además separa el problema en dos dominios, el soluto y el solvente.

Para comprender un poco más a fondo la teoría detrás de estos modelos, se define primero la región interior de una proteína donde se encuentran las cargas. Cada una de estas cargas (q) se encuentra afectada a un campo magnético E . Esta carga se enfrenta a una fuerza proporcional a este campo, mediante la relación:

$$F = qE \quad (1.1)$$

La Ley de Gauss en su forma diferencial para el campo eléctrico establece que la divergencia del campo eléctrico es proporcional a su densidad de carga (ρ).

$$\nabla \cdot E = \frac{\rho}{\epsilon_0} \quad (1.2)$$

Donde ϵ_0 corresponde a la permitividad o constante dieléctrica del medio, y representa la tendencia del medio a polarizarse en presencia de un campo eléctrico. Al integrar la ecuación (1.2) sobre un dominio se obtiene:

$$\int_{\Omega} \nabla \cdot E \, d\Omega' = \int_{\Omega} \frac{\rho}{\epsilon_0} \, d\Omega'$$

Ya que en la parte izquierda de esta ecuación es posible aplicar teorema de la divergencia, queda como:

$$\oint_{\partial\Omega} E \cdot \hat{n} \, d\Gamma = \int_{\Omega} \frac{\rho}{\epsilon_0} \, d\Omega \quad (1.3)$$

Con $\partial\Omega$ como la frontera de Ω . Además, dado que el campo es irrotacional, entonces existe un campo escalar ' u ', tal que $-\nabla u = E$. Por lo que de la ecuación (1.2) se desprende que:

$$\nabla^2 u = - \frac{\rho}{\varepsilon_0} \quad (1.4)$$

Donde u se conoce como potencial electrostático. Finalmente, si se considera una distribución de cargas puntuales descrita por $\rho(x) = q\delta(x - x_q)$ en la cual $\delta(x)$ corresponde a una función delta de Dirac, y x_q simboliza la posición de la carga. Cuando existe una única carga, la solución de la ecuación (1.4) se convierte en el potencial de Coulomb:

$$u^c(x) = \frac{q}{4\pi\varepsilon_0|x - x'|} \quad (1.5)$$

Si una molécula se encuentra compuesta por más cargas, el potencial de Coulomb estará dado por la suma de los potenciales de cada una de éstas.

1.1.1 Ecuación de Poisson-Boltzmann

Un continuo electrostático dentro de un medio dieléctrico y una distribución de cargas de un solo punto en un medio dieléctrico son generalmente usadas para representar la región del solvente y el soluto respectivamente. En sistemas biomoleculares normalmente se considera al solvente como una solución de agua con sal con cierta concentración, mientras que el soluto es una proteína o alguna otra biomolécula.

Cuando la molécula en estudio se encuentra estática, sus cargas se encuentran en posiciones y orientaciones específicas. Pero en el caso del solvente, particularmente cuando se habla de agua, los iones de Cloro (Cl^-) y Sodio (Na^+) y las moléculas de agua se ubican especialmente atraídos o repelidos por las cargas y fuerzas inducidas.

Bajo el supuesto de que los iones son esferas puntuales y que la fuerza que domina el comportamiento de los iones es electrostática y de largo alcance, se puede considerar una distribución radial para los iones. Luego, la densidad de carga (ρ) queda definida por: [11]

$$\rho = \sum_i^{Ni} q_i c_i(x) = \sum_k^{Ni} q_i c_{0i} e^{\frac{-\phi q_i}{k_B T}} \quad (1.6)$$

Donde c_{0i} es el número promedio de especies i . La sal, se disocia con el agua, por lo que es de esperarse que se encuentren proporciones similares de Cloro con carga negativa y

de Sodio con carga positiva, por ende, el número promedio de especies químicas es similar:

$$\begin{aligned} \sum_k^{Ni} q_i c_{0i} e^{\frac{-\varphi q_i}{k_B T}} &= -n_0 \left(e^+ e^{\frac{-\varphi e^+}{k_B T}} + e^- e^{\frac{-\varphi e^-}{k_B T}} \right) \\ &= -n_0 e^+ \left(e^{\frac{-\varphi e^+}{k_B T}} - e^- e^{\frac{-\varphi e^-}{k_B T}} \right) = 2n_0 e^+ \sinh \left(\frac{-\varphi e^+}{k_B T} \right) \end{aligned} \quad (1.7)$$

Cuando $\left| \frac{\varphi q}{k_B T} \right| \sim 0$ es factible linealizar el término hiperbólico, con lo cual, la distribución de cargas queda como:

$$\rho = 2n_0 e^+ \sum_i^{Ni} n_{0i} q_i \quad (1.8)$$

Luego, a partir de la ecuación (1.4), el potencial electrostático promedio en el solvente queda como:

$$\nabla^2 u = u \frac{1}{\varepsilon k_B T} \sum_i^{Ni} n_{0i} q_i \quad (1.9)$$

En esta ecuación, el término de la derecha es conocido como el cuadrado de la inversa del largo de Debye (λ), que simboliza el efecto de “apantallamiento” producido por las cargas de la solución iónica[9], siendo este parámetro determinado por la proporción y tipos de sales disueltas en el solvente. Por ejemplo, para una solución salina normal, es decir, una proporción equivalente a 1:1 de Sodio: Cloro, $\kappa = 0.125 [\text{\AA}^{-1}]$.

1.2 Condiciones de frontera

La región del solvente está dada por cuánto se puede acercar una molécula de agua a las cavidades del soluto. Esta superficie intermedia se conoce como la Superficie Excluida por el o de Solvente (SES) y está determinada por el radio de Van der Waals y el tamaño de una molécula de agua. Considerando un elemento superficial de la SES:

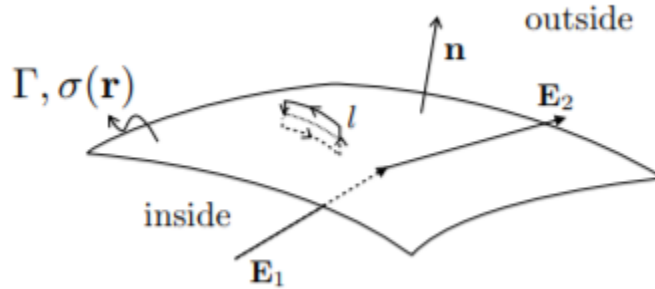


Figura 1.2: Frontera de Ω [11].

Al plantear la integral de línea del campo eléctrico (E) sobre una curva l , el resultado de esta queda dado por la diferencia de potencial eléctrico entre el punto inicial y el punto final, esto gracias a que el campo electrostático es irrotacional. Si se escoge un punto A y un punto B como punto interior y exterior respectivamente de la frontera, se llega a:

$$\int_A^B E \cdot dl = -(u_b - u_a)$$

Por lo que al desplazar los puntos A y B a Γ , se obtiene $u_{in} = u_{out}$, lo que significa que el potencial eléctrico es continuo en la frontera.

Tomando ahora un elemento de superficie Γ , cualquiera, con una carga superficial θ , lo que satisface:

$$\int_{\Gamma} e_{in} \cdot d\Gamma = \frac{\theta}{\epsilon_{in}} \quad ; \quad \int_{\Gamma} e_{out} \cdot d\Gamma = \frac{\theta}{\epsilon_{out}}$$

Dado que θ es común para ambas relaciones, para un elemento diferencial de Γ , para esta última relación se establece:

$$\epsilon_{in} E_{in} \cdot \hat{n} = \epsilon_{out} E_{out}$$

Así, reemplazando el campo por el potencial eléctrico se obtiene finalmente:

$$\varepsilon_{in} \nabla u_{in} \cdot \hat{n} = \varepsilon_{out} \nabla u_{out} \cdot \hat{n} \quad (1.10)$$

Aquí, agregando el término $\varepsilon \frac{\delta u}{\delta n}$, el cual corresponde a la componente normal del desplazamiento eléctrico y al igual que u , es continuo en Γ .

En resumen, se tienen las siguientes condiciones de frontera:

$$\begin{aligned} u_{in} &= u_{out} \\ \varepsilon_{in} \frac{\delta u_{in}}{\delta \hat{n}} &= \varepsilon_{out} \frac{\delta u_{out}}{\delta \hat{n}}; \quad en \Gamma \end{aligned} \quad (1.11)$$

1.3 Tipos de Energía a considerar

Existen diferentes tipos de energía en los problemas de solvente implícito, a continuación, se describirán las que se han considerado como las más importantes.

Previamente se define el concepto de trabajo asociado a una fuerza cualquiera entre dos puntos, y este, físicamente está dado por:

$$W = \int_{Start}^{End} F dl \quad (1.12)$$

Donde l corresponde al camino recorrido. Como la fuerza es proporcional al campo eléctrico, donde la constante de proporcionalidad corresponde a la carga de la partícula y el campo eléctrico (E) también está relacionado con el potencial electrostático, de esta manera se tiene un trabajo dado por:

$$W = \frac{1}{2} \int_{\Omega} \rho(x') u(x') d\Omega' \quad (1.13)$$

1.3.1 Energía de Coulomb

Las cargas y su disposición espacial son posibles gracias a uniones producidas por fuerzas de atracción y repulsión inducidas por diferencias de carga. La fuerza que mantiene atraídas las cargas se calcula como:

$$F^c(r) = \frac{q_1 q_2}{4\pi\epsilon_0 |r|^2} \quad (1.14)$$

Con 1 y 2 simbolizando las especies a considerar, por ejemplo, para el caso de NaCl, q_1 para el sodio (Na^+) y q_2 para el cloro (Cl^-), mientras que r representa la distancia entre ambas. De esta forma, el trabajo de desplazar la carga desde una zona hasta su posición final en esta molécula vendría dado como:

$$W = - \int_{\infty}^{r_{\text{end}}} F^c(x) dl = \frac{q_1 q_2}{4\pi\epsilon_0 r_{\text{end}}^2} \quad (1.15)$$

Y para una distribución de cargas puntuales, tenemos:

$$E^c = \frac{1}{2} \frac{1}{4\pi\epsilon_0} \sum_{i \neq j} \sum_j \frac{q_i q_j}{|x_i - x_j|} \quad (1.16)$$

En el caso de que esta energía sea mayor que 0, por ejemplo, cuando todas las cargas son del mismo signo, significa que existe un agente externo aportando algún tipo de interacción para lograr el equilibrio físico entre las fuerzas de repulsión, es decir, está aportando energía al sistema.

En el caso contrario, cuando la energía sea menor que 0, el agente externo se transforma en el receptor de la energía, es decir, es necesario aplicar trabajo para separar las cargas.

1.3.2 Energía de Solvatación

El proceso de Solvatación debe entenderse como la interacción entre el soluto y el solvente, lo que se traduce también en el proceso de estabilización en las partículas de este último. La energía de solvatación se define como el cambio en la energía libre de Gibbs cuando un ion o molécula es transferida del vacío, o de una fase gaseosa al solvente[17].

El proceso de solvatación se puede separar en cuatro etapas:

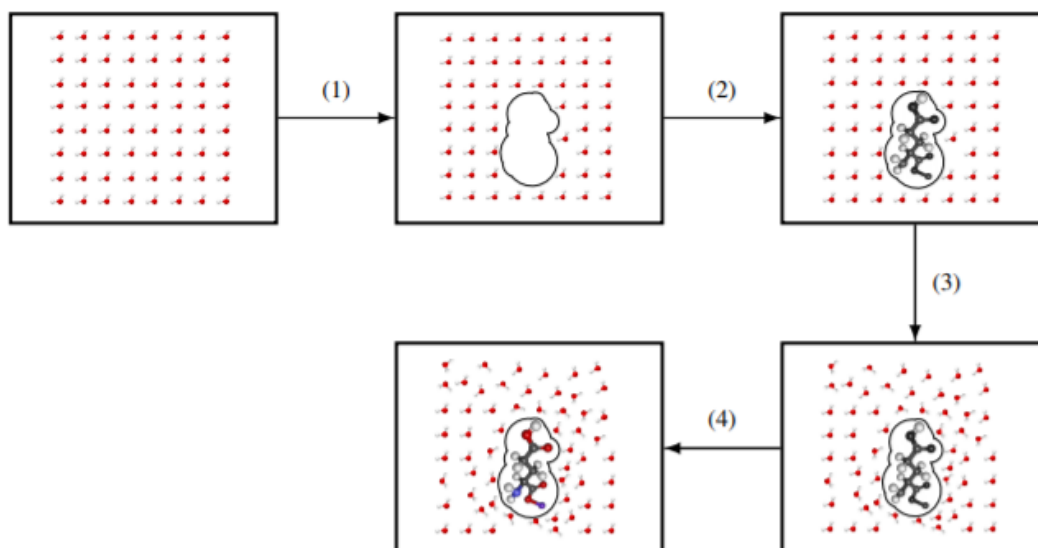


Figura 1.3: Diagrama de solvatación para representar las energías asociadas. El proceso (1) corresponde a generar la cavidad en el solvente representado por moléculas rojas y blancas (agua). El proceso (2) corresponde a agregar la distribución de cargas en la cavidad. Para (3) se agrega el efecto que tiene esta distribución de cargas sobre el solvente. Finalmente, (4) representa la polarización del soluto en respuesta a la polarización del solvente[6].

La energía para “solvatar” una molécula corresponde entonces al cambio de energía libre de solvatación dada por estos cuatro procesos:

$$\Delta G_{Solvat}^{Tot} = \Delta G_{cav} + \Delta G_{disp} + \Delta G_{el} \quad (1.17)$$

Estas fuentes de energía provienen de distintos principios físicos, ΔG_{cav} corresponde a la energía necesaria para generar la cavidad, ΔG_{disp} es la energía debido a los efectos de las fuerzas de corto alcance de Van der Waals entre el soluto y el solvente, mientras que ΔG_{el} es la componente de energía producida por los efectos electrostáticos inducidos por la polarización entre ambas sustancias[18].

La formación de la cavidad y las interacciones de Van der Waals son representadas en la figura 1.3 como el proceso (1) sumado con el (2) y (3) respectivamente, y corresponden

a la componente no polar. Mientras que para la componente polar se tienen únicamente los efectos electrostáticos inducidos por el soluto, es decir, el proceso (4).

La energía producida por el potencial electrostático está entonces determinada por el potencial de reacción producido por incorporar las cargas puntuales ρ del soluto como:

$$\Delta G_{Solvat} = \Delta G_{el} = \frac{1}{2} \int_{\Omega} \rho(x) u^R(x) d\Omega \quad (1.18)$$

Donde u^R es el potencial de reacción. En un sentido físico, es la diferencia de potencial de llevar cada carga desde el vacío, donde existen interacciones de Coulomb. Ya que Ω contiene únicamente cargas puntuales, esta ecuación queda como:

$$\Delta G_{Solvat} = \frac{1}{2} \sum_{k=1}^{N_q} q_k u^R(x_k) \quad (1.19)$$

De esta definición se obtiene una relación entre el potencial total y el potencial de reacción, la cual está dada por: $u = u^R + u^C$.

1.4 Método de elementos de borde (BEM)

Tanto los métodos de elementos finitos, métodos de diferencias finitas y los métodos de elementos de borde permiten resolver problemas a gran escala para casos de modelos implícitos, cada uno contando con ciertas ventajas y desventajas.

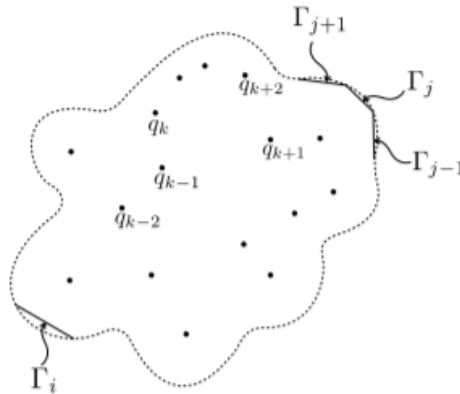


Figura 1.4: Sketch de una interfaz discretizada, Γ : Γ_i representa al panel que contiene el punto de colocación, Γ_j los paneles sobre los cuales se integra.

El método de elementos de borde o BEM discretiza la frontera en paneles como se muestra en la Figura 1.4 y asume una distribución de cantidades desconocidas dentro de esos paneles. Se usan paneles triangulares-planos y la técnica de colocación central para llegar a un sistema lineal.

BEM a lo largo de los años ha ido en constante progreso, gracias a los avances matemáticos y computacionales. Para entender un poco mejor las bases del método se debe considerar la ecuación de Laplace en su forma diferencial.

1.4.1 Ecuación de Laplace

Esta ecuación modela el comportamiento del potencial electrostático dentro de la región del soluto. Se puede definir como:

$$\nabla^2 u = b \quad \text{en } \Omega \quad (1.20)$$

En el lado derecho se puede apreciar una función b la cual será definida más adelante, al tomar toda la ecuación 1.20 y realizarle una convolución con una función ω sobre todo el dominio del espacio Ω , se tiene:

$$\int_{\Omega} \omega(x, x') \nabla^2 u(x') d\Omega' = \int_{\Omega} \omega(x, x') b(x') d\Omega' \quad (1.21)$$

Y utilizando las siguientes identidades:

$$\int \int \int_{\Omega} \nabla \cdot \mathbf{F} d\Omega' = \int \int_{\delta\Omega} \mathbf{F} \cdot \hat{\mathbf{n}} d\Gamma$$

$$\nabla \cdot (\gamma \mathbf{A}) = \nabla \gamma \cdot \mathbf{A} + \gamma \nabla \cdot \mathbf{A}$$

$$\int_{\Omega} f(x) \delta(x - x') dx' = f(x') \quad \text{para } x \in \Omega$$

Se llega a lo siguiente:

$$\begin{aligned} \int_{\Omega} \nabla \cdot (\nabla u(x') \omega(x, x')) d\Omega' - \int_{\Omega} \nabla \cdot (u(x') \nabla \omega(x, x')) d\Omega' \\ + \int_{\Omega} u(x') \nabla^2 \omega(x, x') d\Omega' = \int_{\Omega} \omega(x, x') b(x') d\Omega' \end{aligned} \quad (1.22)$$

Teniendo especial cuidado con el término de la derecha, ya que no siempre se puede obtener un valor directo y además se encuentra evaluada sobre todo Ω . Sin embargo, para el caso de la ecuación de Poisson-Boltzmann este lado derecho es igual a la distribución de cargas y b evaluada en la región del solvente es 0, lo que permite obtener un valor directo de esa integral para la frontera, que es lo importante.

Se escoge ω como una función del espacio libre de Green, la cual cumple con:

$$OG = -\delta(x - x') \quad (1.23)$$

Donde O es cualquier operador diferencial y G la función de Green asociada, mientras que $\delta(x - x')$ corresponde a la función Delta de Dirac. Las funciones de Green para el operador de la ecuación de Laplace y de Poisson-Boltzmann son conocidas y se encuentran en la literatura, estas son:

Ecuación de Laplace	Ecuación de Poisson-Boltzmann
$G_L(x, x') = \frac{1}{4\pi x - x' }$	$G_y(x, x') = \frac{e^{-k x - x' }}{4\pi x - x' }$

Con esto se puede nuevamente simplificar algunos términos de la ecuación anterior (1.22), quedando:

$$-u(x_{\Omega}) + \int_{\Gamma} G_L(x_{\Omega}, x'_{\Gamma}) \frac{\partial u}{\partial \hat{n}}(x'_{\Gamma}) d\Gamma' - \int_{\Gamma} u(x'_{\Gamma}) \frac{\partial G_L}{\partial \hat{n}}(x_{\Omega}, x'_{\Gamma}) d\Gamma' = B(x_{\Omega}) \quad (1.24)$$

En donde se han agregado los subíndices Γ y Ω para recalcar el dominio de dicha variable, también es posible verlo de la siguiente forma:

$$\begin{aligned} -u(x_{\Omega}) + \int_{\Gamma} G_L(x_{\Omega}, x'_{\Gamma}) \frac{\partial u}{\partial \hat{n}}(x'_{\Gamma}) d\Gamma' - \int_{\Gamma} u(x'_{\Gamma}) \frac{\partial G_L}{\partial \hat{n}}(x_{\Omega}, x'_{\Gamma}) d\Gamma' = \\ \int_{\Omega} \nabla^2 u(x'_{\Omega}) G_L(x_{\Omega}, x'_{\Omega}) d\Omega' \end{aligned} \quad (1.25)$$

Posteriormente, reemplazando el lado derecho por la densidad de cargas, permite obtener un valor para el potencial en cualquier punto interior del soluto o molécula (Ω_{in}).

$$\begin{aligned} -u(x_\Omega) + \int_{\Gamma} G_L(x_\Omega, x_{\Gamma'}) \frac{\partial u}{\partial \hat{n}}(x_{\Gamma'}) d\Gamma' - \int_{\Gamma} u(x_{\Gamma'}) \frac{\partial G_L}{\partial \hat{n}}(x_\Omega, x_{\Gamma'}) d\Gamma' \\ = - \sum_k^{N_q} \frac{q_k}{4\pi\epsilon_{in}|x_\Omega - x_k|} \end{aligned} \quad (1.26)$$

1.4.2 Espacios de las soluciones

Los espacios cumplen un rol fundamental, ya que permiten que las operaciones matemáticas se encuentren bien definidas. Dentro de los espacios relevantes, se encuentra la familia de los espacios de Lipschitz (L_p). Este se encuentra definido como el espacio de las funciones cuya p-ésima potencia en valor absoluto es integrable, es decir:

$$\|\varphi_p\| = \left(\int_{\Omega} |\varphi^p| d\Omega' \right)^{\frac{1}{p}} < \infty \quad (1.27)$$

Con esta definición, se crean los espacios de Sóbolev, tal que su α -ésima derivada también pertenezca al espacio L_p , entonces:

$$W_p^r(\Omega) := \{\Psi \in L_p(\Omega) \text{ tal que } \partial^\alpha \Psi \in L_p(\Omega), \text{ para } |\alpha| < r\}$$

El orden r del espacio de Sóbolev (W) también puede ser negativo o incluso una fracción. Dentro de los espacios de Sóbolev, se encuentran los espacios de Hilbert (H^s), los cuales se obtienen al restringir los espacios de Sóbolev W a los espacios de Lipschitz L_p con $p = 2$, esto viene siendo:

$$H^s = W_{p=2}^s = W_2^s$$

En particular, este último espacio cumple la propiedad:

$$\text{dual}(H^s(\Omega)) = H^{-s}(\Omega), \quad \text{con } 0 < s$$

Donde “dual()” denota el espacio dual del espacio H^s .

Definición de Trazas

En primera instancia, la traza interior de Dirichlet (γ_0^{int}) se define naturalmente sobre un conjunto $\Omega \subset \mathbb{R}^d$, la cual permite trasladar un elemento del dominio Ω a la frontera, formalmente:

$$\gamma_0^{int} f(x) = \lim_{\Omega \ni \bar{x} \rightarrow x \in \Gamma} f(\bar{x}) \quad \text{Para } x \in \Gamma = \partial\Omega \quad (1.28)$$

Además, para $\frac{1}{2} < s < k$, se tiene la identidad $\gamma_0^{int} H^s = H^{s-\frac{1}{2}}$. También se define la derivada normal o traza de Neumann sobre u como:

$$\gamma_1^{int} u(x) = \lim_{\Omega \ni \bar{x} \rightarrow x \in \Gamma} \nabla u(\bar{x}) \cdot \hat{n}(\bar{x}) \quad (1.29)$$

Estas dos definiciones son claves para definir los operadores asociados a BEM.

Construcción de los espacios

Junto con discretizar la superficie que tienen en común ambos dominios en paneles triangulares, como ya se ha mencionado, se deben discretizar los espacios $H^{1/2}$ y $H^{-1/2}$ donde habitan las soluciones de Dirichlet (ϕ) y Neumann ($\frac{\partial \phi}{\partial n}$) respectivamente. Para esto se considera una triangulación T_h del dominio Γ , con N elementos triangulares de superficie t_i , con sus nodos x_i asociados y M vértices p_i , de manera que $T_h = \cup t_i$, entonces, se definen dos espacios:

- Funciones constantes por elemento:

$$S_h^0 = \text{gen}\{\varphi_k^0\}, \quad \text{tal que } \varphi_k^0 = 1 \quad \forall x \in t_k, \text{ y } 0 \text{ en otro caso.}$$

Este espacio es posible construirlo en la herramienta BEMPP a través de los polinomios discontinuos (DP)

- Espacio S_h^1 para las funciones globalmente continuas y lineales ρ_i por elemento, tal que:

$$\rho(\rho_l) = \begin{cases} 1, & i = l \\ 0, & i \neq l \end{cases}$$

Este espacio puede construirse en BEMPP usando Polinomios continuos entre elementos (P), y debe tener al menos orden 1.

1.4.3 Tipos de operadores a utilizar

Ya que ambas integrales se encuentran en los problemas tanto de Laplace como de Poisson-Boltzmann. Se utilizan operadores Single-Layer y Double-Layer, que mediante una discretización de Galerkin permiten construir un sistema de ecuaciones de la forma $Au=b$, en donde u corresponde al potencial electrostático evaluado en la frontera Γ , A es la matriz asociada ensamblada con los operadores y b el lado derecho del sistema (rhs).

Operador Single-Layer

Dado G , una Función de Green asociada al operador diferencial A en la frontera Γ del espacio Ω .

$$\hat{V}_A : H^{1/2} \rightarrow H^1$$

$$\varphi(r_\Gamma) \rightarrow [\hat{V}_A^{r_\Gamma}](\varphi) = \int_{\Gamma} \varphi(r'_\Gamma) G_A(r_\Omega, r_\Gamma) d\Gamma'$$

Ahora, este operador es posible llevarlo a la frontera Γ , esto es, aplicando la traza γ_0^{int} .

$$V_A = \gamma_0 \hat{V}_A : H^{-1/2} \rightarrow H^{1/2}$$

Operador Double-Layer

Considerando lo mismo que en el operador anterior, se tiene:

$$\hat{K}_A : H^{1/2} \rightarrow H^1$$

$$\varphi(r_\Gamma) \rightarrow [\hat{K}_A^{r_\Gamma}](\varphi) = \int_{\Gamma} \varphi(r'_\Gamma) \gamma_1^{int} G_A(r_\Omega, r_\Gamma) d\Gamma'$$

De igual manera, al llevarlo a la frontera:

$$K_A = \gamma_0 \hat{K}_A : H^{-1/2} \rightarrow H^{1/2}$$

En ambos casos, la función φ es una función cualquiera, el subíndice A denota el operador asociado (L para Laplace e Y para Steffan-Boltzmann) y como se puede observar V representa al operador Single-Layer y K al operador Double-Layer. Para facilitar el uso de estos operadores se utiliza la herramienta BEMPP, la cual permite obtener la discretización de Galerkin de estos operadores y trabajar de una manera relativamente sencilla. Utilizando la ecuación (1.26) es posible obtener la discretización de Galerkin para ambos operadores.

$$[V]_{ij} = \langle V_{\varphi j}; \varphi i \rangle_{\Gamma}$$

$$[K]_{ij} = \langle K_{\varphi j}; \varphi i \rangle_{\Gamma}$$

Pero, al hacer esto, la matriz K se vuelve no cuadrada, debido a que los espacios S_h^0 y S_h^1 tendrían un número diferente de elementos. Una solución a esto es “discretizar ambos espacios con el mismo espacio continuo $S_h^{1''}$ ”[3].

1.5 Modelo para una interfaz única

En el marco de los métodos de elementos de borde, el trabajo desarrollado por Yoon & Lenhoof[9] por los años 90 muestra la eficiencia de estos métodos en problemas donde sí se consideran efectos de los electrolitos sobre una geometría esférica. La transición rápida de la permitividad en la interfaz puede ser estudiada incorporando un Stern-Layer, pero para el estudio realizado, esto no se considerará.

Como ya se ha mencionado, en el medio se cumplen las siguientes relaciones:

$$\Delta u_{in} = \frac{1}{\varepsilon_{in}} \sum_{k=0}^{N_q} \delta q_k, \text{ en } \Omega_{in} \text{ y } \Delta u_{out} = k^2 u_{out}, \text{ en } \Omega_{out}$$

$$u_{in} = u_{out} \text{ y } \varepsilon_{in} \frac{\partial u_{in}}{\partial \hat{n}} = \varepsilon_{out} \frac{\partial u_{out}}{\partial \hat{n}} \text{ en } \Gamma.$$

Donde u corresponde al potencial eléctrico total, Ω_{in} es el dominio del soluto y Ω_{out} el dominio del solvente, mientras que Γ representa la interfaz entre ambos. Al hacer una convolución con la respectiva Función de Green en cada dominio, usar la ecuación (1.25) y además considerando los operadores antes mencionados, se tiene:

$$u_{in}(x_{\Omega_{in}}) + [K_L^{x_{\Omega_{in}}}] (u_{in}) - [V_L^{x_{\Omega_{in}}}] \left(\frac{\partial u_{in}}{\partial n} \right) = \frac{1}{\varepsilon_{in}} \sum_{k=1}^{N_q} q_k G_L(x_{\Omega_{in}}, r_k) \quad (1.30)$$

$$u_{out}(x_{\Omega_{out}}) + [K_Y^{x_{\Omega_{out}}}] (u_{out}) - [V_L^{x_{\Omega_{out}}}] \left(\frac{\partial u_{out}}{\partial n} \right) = 0 \quad (1.31)$$

Lo siguiente que se debe hacer es llevar el problema a la **frontera** y así utilizar las condiciones de contorno presentadas en (1.11).

$$\begin{bmatrix} \frac{1}{2} + K_L & -V_L \\ \frac{1}{2} - K_Y & \frac{\varepsilon_{in}}{\varepsilon_{out}} V_Y \end{bmatrix} \begin{pmatrix} u_{in} \\ \frac{\partial u_{in}}{\partial \hat{n}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\varepsilon_{in}} \sum_{k=0}^{N_q} q_k G_L \\ 0 \end{pmatrix} \quad (1.32)$$

Por conveniencia y para abarcar de mejor manera el problema, se define la normal n apuntando hacia el exterior de Ω_{in} , por ende, los signos de K_L y V_Y son cambiados.

Con el potencial total en la frontera, es posible determinar la energía de solvatación de este potencial, es decir:

$$\Delta G_{Solvat} = \sum_{k=1}^{N_q} q_k \left[V_{L-\Gamma}^{x_q} \left(\frac{\partial u_{in}}{\partial n} \right) - K_{L-\Gamma}^{x_q} u_{in} \right] \quad (1.33)$$

2. Metodología

A continuación, se presentarán diferentes herramientas para el mallado de superficies moleculares. También el funcionamiento del refinamiento adaptativo y su comparación con un refinamiento no adaptativo.

Para esto, se debe tener en cuenta que la superficie de Van der Waals[20] corresponde a la región del soluto formada por la unión de todas las porciones de las superficies atómicas sin ser sobrepuestas por otras. La Superficie Accesible al Solvente o SAS[21] es trazada por el centro de una esfera de prueba cuyo radio es igual al radio del solvente, cuyo trazo pasa sobre la superficie de Van der Waals. Finalmente, La Superficie Excluida de Solvente o SES[22] es definida como lo más cerca que una molécula de solvente puede estar del soluto, se traza al igual que en SES, pero desde el borde de la esfera de prueba.

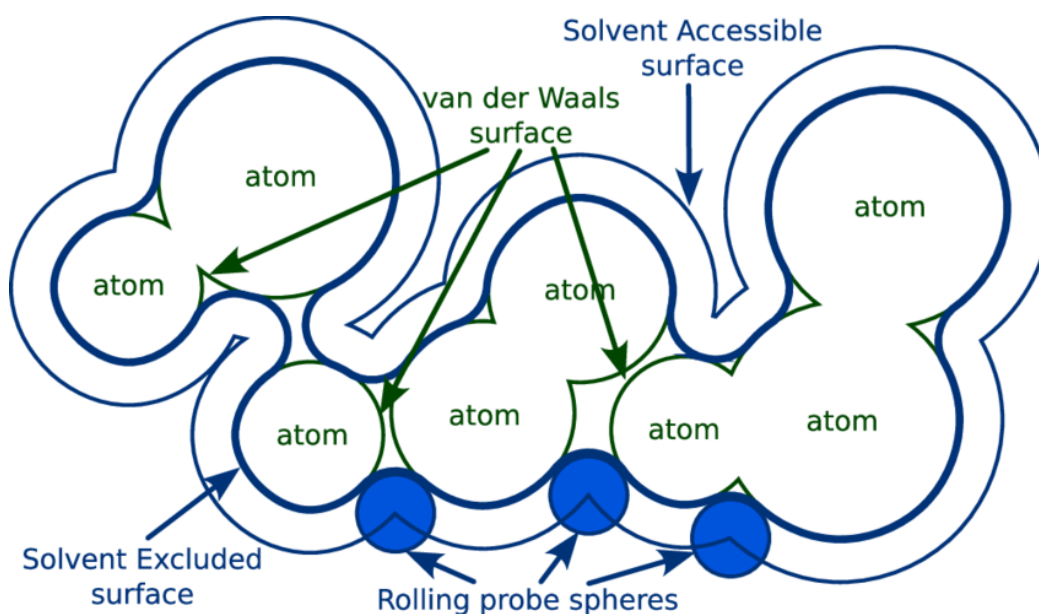


Figura 2.1 Tipos de Superficies Moleculares: Van der Waals (VdW), Superficie Accesible al Solvente (SAS) y Superficie Excluida de Solvente (SES)[27].

Como se mencionó anteriormente, se utilizarán superficies tipo SES para generar mallados de las superficies moleculares, las cuales se basan en triangulación y así realizar los diferentes análisis moleculares.

2.1 MSMS

Fue introducido en el año 1995 por Sanner[23] y a partir de entonces ha ganado popularidad debido a que fue una de las primeras herramientas en trabajar con triangulaciones para superficies del tipo SES. Después de crear una representación analítica de la superficie, el algoritmo genera una triangulación de una densidad especificada ajustando triángulos predefinidos según la superficie. La resolución de la malla es controlada según el usuario desee a través del parámetro de densidad d , la cual determina el número de triángulos usados para la triangulación de una superficie dada en unidades de vértices/angstrom².

2.2 NanoShaper

Fue introducido en el año 2012 por Decherchi y Rocchia[7], puede trabajar con superficies tipo SES y SAS, incluyendo superficies Gaussianas. Para construir una triangulación en superficies SES, Nanoshaper primero crea una descripción de la superficie, ya sea de manera analítica o con una aproximación. El código emplea un algoritmo en el cual se lanza un “rayo paralelo” a los ejes de coordenadas y así las intersecciones con la superficie son calculadas. Las posiciones de las intersecciones de los vértices son usadas por el algoritmo para así obtener la triangulación. La resolución de la malla es controlada por el usuario dado un parámetro escalar llamado s que establece la longitud del largo inverso de una celda cubica cuadriculada, todo esto en unidades de angstroms.

Para construir la malla superficial se debe seguir una serie de pasos, principalmente tener cuidado con el tipo de archivo, ya que algunas herramientas requieren de uno en específico.

Primero, se debe escoger la proteína a trabajar, esta puede ser seleccionada en la Research Collaboratory for Structural Bioinformatics - Protein Data Bank (RCSB PDB), en donde se puede obtener el archivo (.pdb), el cual utilizando PDB2PQR con algún tipo de forcefield permite obtener el archivo PQR (.pqr)[19], el cual contiene información con la ubicación, cargas y radio de Van der Waals de cada átomo. A continuación, a partir del archivo (.pqr), se debe generar un archivo (.xyzr), el cual es el input necesario para el mallador.

Existe documentación que realiza una comparativa entre Nanoshaper y MSMS, la cual principalmente considera los cálculos del área superficial S_a , la cual se compone de la

suma de las áreas de los triángulos que forman la superficie, y de energía de solvatación E_{solvat} que está definida previamente. Estos cálculos se han realizado para diferentes biomoléculas, entre las que se incluyen proteínas, péptidos y fragmentos de ácido nucleico. Además, se considera bastante la resolución de la malla, la cual en este caso viene dependiendo de la geometría de los triángulos, es decir, su tamaño, forma y relación de aspecto, de esta manera se pueden comparar cualitativamente las mallas de superficie generadas.

Algunos resultados muestran que MSMS tiene problemas para ciertos valores de d , especialmente para moléculas grandes, provocando incluso que no se genere la triangulación, para estos casos se recomienda utilizar la máxima resolución posible con la cual MSMS no falla.

Ambas herramientas producen un número de triángulos pequeños y/o delgados que reducen la eficiencia computacional y es común removerlos de las simulaciones. Uno debe seleccionar y ajustar el tamaño mínimo del área de los triángulos, por ejemplo, para un estudio realizado por Wilson & Krasny el año 2020[4], se consideró filtrar triángulos cuya área es menor a $1e-5$ angstrom cuadrado o si la distancia entre los centroides de dos triángulos vecinos es menor a $1e-5$ angstrom. Adicionalmente, aunque se filtren de esta manera los triángulos, posiblemente se generen triángulos de área 0, es decir, con un área menor a la precisión de la maquina usada, los cuales igual generan un porcentaje de error.

En el estudio ya mencionado de Wilson & Krasny, con una muestra de 38 biomoléculas diferentes, para el caso de MSMS se filtran triángulos igual al 0.064% del total y existen triángulos con área 0 igual al 0.052% del total, mientras que para Nanoshaper es 100 veces menor el número de triángulos filtrados y no existen triángulos de área 0.

Incluso después del filtrado mencionado, la relación de aspecto de los triángulos afecta el rendimiento computacional. La relación de aspecto, como ya es sabido, es definida como la distancia desde el lado más largo al más corto. Cuando se usa una pequeña cantidad de triángulos, la distancia promedio de las relaciones de aspecto en Nanoshaper son de hasta 30 veces menores que en MSMS, mientras que cuando se utilizan una gran cantidad de triángulos esta distancia es prácticamente la misma, pero inclinándose a ser mayor en el caso de MSMS, entonces se puede decir que con este parámetro Nanoshaper tiene un mejor rendimiento computacional que MSMS.

A través de diferentes simulaciones se puede ver que las mallas generadas por Nanoshaper son más suaves, incluso MSMS genera una irregularidad de malla mucho

mayor y converge en una mayor cantidad de iteraciones. Las irregularidades disminuyen la eficiencia de los cálculos realizados.

Para muchos casos la convergencia con N^{-1} es suave, en donde N corresponde al número de triángulos, esto sucede tanto para MSMS como Nanoshaper, pero MSMS alcanza su límite primero y además es incapaz de generar mallas confiables para un gran número de N y si se fuerza no se genera la malla o presenta muchas más irregularidades, mientras que, en general, Nanoshaper no tiene limitaciones, incluso si es necesario trabajar con mallas densas o biomoléculas de gran tamaño.

Para moléculas muy complejas (múltiples dominios), MSMS no genera la malla y en algunos casos, principalmente con moléculas muy pequeñas, Nanoshaper tampoco lo hace. Para ambos casos, el tiempo de generación de la malla es muy pequeño, otra diferencia se centra en la cantidad de iteraciones realizadas, en donde Nanoshaper tiene ventaja, además este último pocas veces presenta un límite de iteraciones, mientras que MSMS casi siempre presenta un límite. Hablando del tiempo requerido para las iteraciones, nuevamente Nanoshaper tiene ventaja, en donde para converger necesita aproximadamente $2/3$ del tiempo y $1/4$ de las iteraciones realizadas en MSMS, esto se debe principalmente a que MSMS presenta una gran cantidad de triángulos con una relación de aspecto muy grande.

2.3 Estimación de Error

A lo largo del tiempo las herramientas para representar algún fenómeno físico han ido evolucionando. Utilización de espacios de alto orden, enriquecimiento de malla, preconditionamiento, métodos de resolución de problemas matriciales y aproximación de integrales y derivadas que en conjunto buscan representar de mejor manera un problema con el objetivo de ahorrar recursos y obtener una solución más confiable.

A continuación, se aborda el método de la ecuación adjunta, el cual permite estimar el error observado en alguna propiedad de interés, en este caso el error en la energía de solvatación y como cada elemento de la malla influye en ello.

2.3.1 Método de la ecuación adjunta

Este método se basa en propiedades expresadas en forma integral, en función de un campo escalar determinado previamente. Esta ecuación permite generar un estimador

local([28],[29]). Para eso, se define la propiedad de interés (*QoI* por sus siglas en inglés) como:

$$QoI = \int_{\Omega} \Psi(x') f(x') dx' \quad (2.1)$$

Con $\Psi(x)$ una función que “pondera” los valores del campo escalar u en la región Ω que para el caso de la energía de solvatación está definida en la ecuación (1.18), en donde Ψ corresponde a la densidad de carga y f la componente regular del potencial.

El siguiente paso es definir Ψ de otra manera, para esto, se define el adjunto φ como tal.

$$D\varphi = \Psi \quad (2.2)$$

La búsqueda de condiciones de frontera para φ y para definir el operador diferencial D se definen como tales que cumplen con la siguiente relación:

$$QoI = \int_{\Omega} D\varphi(x') u^R(x') dx' = \int_{\Omega} \varphi(x') D^* u^R(x') dx' \quad (2.3)$$

Donde $D^* u^R(x')$ es el operador diferencial para la EDO del potencial de reacción. Esta ecuación diferencial regularizada, y conocida como RPBE[30] está dada por:

$$-\nabla(\varepsilon(x) \nabla u^R(x)) + k^2 u^R = \nabla((\varepsilon(x) - \varepsilon_{in}) \nabla u^C(x)) \quad (2.4)$$

Con las condiciones de frontera

$$\begin{aligned} u^R(\infty) &= 0 \\ [u^R]_{\Gamma} &= (\varepsilon_{in} - \varepsilon_{out}) u^C(\varepsilon_{\Gamma}) \\ \left[\varepsilon(x) \frac{\delta u^R}{\delta \hat{n}} \right]_{\Gamma} &= (\varepsilon_{in} - \varepsilon_{out}) \frac{\delta u^C(x_r)}{\delta \hat{n}} \quad ; \quad x_r \in \Gamma \end{aligned}$$

Si bien, se presenta la ecuación diferencial para esta componente, utilizarla discretamente para estimarla no es posible en el marco de los elementos de borde, luego de que no existe una función de Green apropiada para trabajar con la región exterior sin realizar una aproximación. Pero esta definición permite elegir al operador del adjunto como:

$$D\varphi = -\nabla(\varepsilon(x)\nabla\varphi) + k^2\varphi \quad (2.5)$$

Las condiciones de frontera son una consecuencia de la relación (2.3), para lo cual se deben buscar condiciones sobre φ para que esta se cumpla. En base a esto, se utiliza a φ y su derivada normal en Γ como continuo en Ω . Imponer estas 2 condiciones produce también que el adjunto sea idéntico en definición al potencial electrostático, esto significa la ventaja de que es posible resolverlo en la frontera con métodos de elementos de borde. Y así obtener un indicador de error local para la energía de solvatación.

La derivación del estimador de error en la frontera parte de la definición de un residual (E) definido de manera global:

$$E(\Delta G_{solvat}) = \int_{\Omega} \Psi(x) (F(x) - f(x)) dx \quad (2.6)$$

Con F una solución numérica y f la solución exacta.

Al separar esta definición en una parte que corresponda a la energía de solvatación exacta (Ex) y otra dada por una aproximación (Aprox) permite generar un indicador de error (E) o residual, expresado como la diferencia entre el valor observado entre estas 2 soluciones.

$$E(\Delta G_{solvat}) = \Delta G_{solvat}^{Ex} - \Delta G_{solvat}^{Aprox} \quad (2.7)$$

ΔG_{solvat}^{Ex} descompone al potencial electrostático en 3 componentes, únicamente con fines demostrativos. Para eso, se definen las siguientes componentes[29]:

Componente singular

Corresponde al potencial de Coulomb, y está definido de manera de simplificar el lado derecho de la LPBE.

$$-\nabla(\varepsilon\nabla u^C) = \sum_{i=1}^{N_q} q_i \delta(x - x_i) \quad (2.8)$$

Componente armónica

$$-\nabla^2 u^h = 0 \quad (2.9)$$

Con las condiciones de frontera $u^h = -u^c$. La elección de esta condición es una consecuencia de que el potencial electrostático total u sea continuo en la frontera.

Componente regular

Se define de tal manera que cumpla la ecuación de Poisson-Boltzmann y sus condiciones de borde.

$$-\nabla(\varepsilon \nabla u^{regular}) + k^2 u^{regular} = 0 \quad (2.10)$$

Con las condiciones de borde:

$$\begin{aligned} [u^{regular}(x)]_{\Gamma} &= 0 \\ [u^{regular}(x \rightarrow \infty)] &= 0 \\ \left[\varepsilon(x) \frac{\delta u^{regular}}{\delta n} \right]_{\Gamma} &= \varepsilon \frac{\delta}{\delta \hat{n}} (u^h + u^c) \end{aligned}$$

Como el potencial de reacción u^R es la suma de la componente regular y armónica, se define el valor de la energía de solvatación “exacta” por la ecuación (2.6).

$$\Delta G_{solvat}^{Ex} = (\Psi, u^{regular} + u^h) \quad (2.11)$$

$$\Delta G_{solvat}^{Ex} = \int_{\Gamma} \varepsilon \frac{\delta \varphi}{\delta n} u^c d\Gamma' - \int_{\Gamma} \varphi \varepsilon \frac{\delta u^c}{\delta n} d\Gamma' \quad (2.12)$$

Por otra parte, la solución aproximada se genera utilizando la separación convencional de 2 términos (Potencial de reacción y de Coulomb) y así se deriva un valor aproximado, usando las identidades de Green y las respectivas ecuaciones diferenciales.

$$\Delta G_{solvat}^{Aprox} = (\Psi, U^R) \quad (2.13)$$

$$\Delta G_{solvat}^{Aprox} = \int_{\Gamma} \varphi \left(\varepsilon \frac{\delta U^R}{\delta n} \right) d\Gamma' - \int_{\Gamma} U^R \varepsilon \frac{\delta \varphi}{\delta n} d\Gamma' \quad (2.14)$$

2.3.2 Estimación del potencial y su relación con el adjunto

Para estimar el potencial se debe resolver el problema en la frontera, para esto se requiere que el dominio, rango y espacio dual del operador sea el mismo. Los espacios

utilizados para resolverlo corresponden al espacio de polinomios constantes para cada elemento (DP-0) y el espacio de polinomios continuos entre elementos (P-1). A continuación, se detalla el dominio para cada variable.

- Para ΔG_{solvat}^{Trad} se utiliza el espacio constante entre elementos para U y $\frac{\delta U}{\delta n}$.
- Para $\Delta G_{solvat}^{Aprox}$ se utiliza el espacio constante entre elementos para U y $\frac{\delta U}{\delta n}$, y el espacio de los polinomios de primer orden continuos entre elementos para el adjunto φ y su derivada normal en la superficie.
- Para ΔG_{solvat}^{Ex} se utiliza el espacio de los polinomios de primer orden continuos entre elementos para el adjunto φ y su derivada normal en la superficie. Además, se utiliza la solución analítica para el potencial de Coulomb.

Para cada caso se utiliza un set de valores distintos. El cálculo de ΔG_{solvat}^{Trad} obedece principalmente a efectos comparativos, luego de que esta es la formulación que se encuentra más implementada en método de elementos frontera, y además es obtenible vía operadores de BEMPP.

El primer paso luego de cargar la malla a la rutina es crear los espacios de Dirichlet, Neumann y duales de la malla, que como se mencionó antes, para tener el adjunto como una matriz cuadrada, estos espacios deben ser idénticos. Teniendo estos espacios, se construye una versión discretizada de la matriz del sistema (1.32).

Esto se resuelve utilizando el algoritmo GMRES. Para tener una aproximación más cercana del error, se utilizan tolerancias pequeñas 10^{-8} o 10^{-5} y así evitar agregar errores de redondeo y cuantificar de buena manera el aporte de cada elemento a la energía para cada espacio de solución.

2.3.3 Estimación del error local y refinamiento

Con los valores del potencial y el adjunto se obtiene un indicador de error para cada elemento desglosando el residuo (2.6), por lo tanto:

$$E^i = \frac{\varepsilon}{2} \left| \left(\int_{\Gamma_i} \varphi \left(\frac{\delta U^R}{\delta n} \right) d\Gamma' - \int_{\Gamma_i} U^R \frac{\delta \varphi}{\delta n} d\Gamma' \right) - \left(\int_{\Gamma_i} \varphi \left(\frac{\delta \varphi}{\delta n} \right) u^c d\Gamma' - \int_{\Gamma_i} \varphi \frac{\delta u^c}{\delta n} d\Gamma' \right) \right| \quad (2.15)$$

Ya definido el error local, se procede a refinar un porcentaje determinado de elementos que más aportan al error. El número N_{ref} de elementos a refinar está dado por la siguiente relación:

$$\sum_{i=1}^{N_{ref}} E^i \leq P \sum_{i=1}^{N_{el}} E^i \quad (2.16)$$

Con P como el porcentaje de elementos a refinar. En general, se utiliza $P = 10\%$ para cada ciclo de refinamiento. Este método es iterativo, y se marcan los elementos a refinar apenas se supere este porcentaje.

Para refinar la malla, se toma el error local y se refina un porcentaje (P) determinado de elementos que aportan más al error. Además, para evitar problemas en algún ciclo de refinamiento el código utiliza el siguiente algoritmo, el cual asigna la variable “status” a cada elemento. Y su significado especifica cómo se refina el elemento.

- Status = 0 : No refinar el elemento.
- Status = 1 : Dividir el elemento en 2 nuevos triángulos.
- Status = 4 : Refinar el elemento en 4 nuevos triángulos.

Con lo cual se da inicio al algoritmo:

1. Asignar la variable Status = 4 a los elementos a refinar que aportan más al error.
2. Buscar todos los elementos adyacentes y sumar 1 a la variable Status por cada triángulo adyacente con Status = 4.

Cuando Status sea distinto de 1 o 4, cambiar su valor por 4.

Repetir esto hasta tener solamente Status con valores 0, 1 y 4.

3. Finalmente, se desarrolla la etapa de refinamiento, dependiendo de su Status y teniendo la precaución en el caso de que el Status sea igual a 1, se respete el nuevo vértice a agregar también sea agregado en el triángulo adyacente respectivo.

En el siguiente diagrama, se puede apreciar la evolución de la malla, para un input determinado de error y ya filtrado.

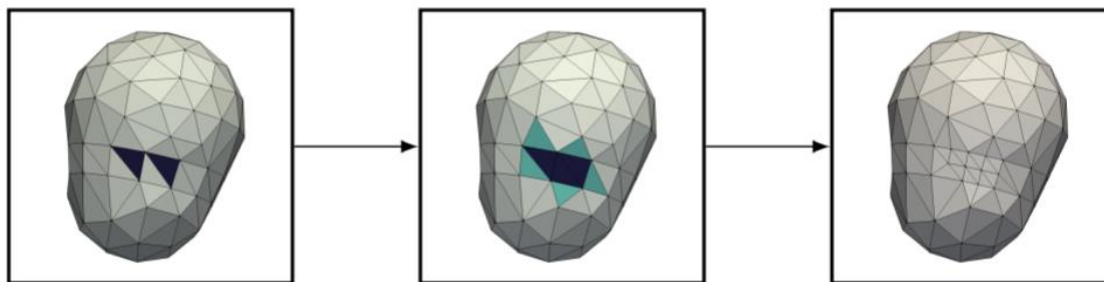


Figura 3.2: Evolución de la malla y su algoritmo de refinamiento. En blanco, *status* = 0, en celeste *status* = 1 y en azul oscuro *status* = 4. Primero, se recibe el input de los elementos a refinar, y se trabaja en el punto 2 hasta que no existan triángulos con un status entre 1 y 4. A la derecha, se observa la malla correctamente refinada[6].

La idea de esto es evitar la creación de elementos muy alargados, lo que ocurre para altos grados de refinamiento en los cuales se divide a la mitad un elemento de la malla más de 2 veces consecutivas.

2.3.4 Refinamiento adaptativo y no adaptativo

El concepto de adaptativo hace referencia a la utilización de la frontera real de la molécula o ion, y no adaptativo como el uso de la malla como la geometría sin variarla, es decir, sin llevar los elementos a la frontera real.

La implementación de estos conceptos significa una diferencia importante en cuanto a la convergencia de la Energía de Solvatación para mallas muy gruesas, ya que en mallas suficientemente finas (sobre 8 elementos por $\text{\AA}^2$), la adición de vértices en la frontera real, o sin extrapolarlo, no implica un gran cambio de la geometría ni en el suavizado de las esquinas.

1. Refinamiento no adaptativo: Al agregar nuevos vértices, estos se agregan en el mismo elemento, es decir, sin llevarlo a la frontera. Para un número alto de iteraciones generan triángulos alargados que impactan sobre el número de iteraciones y el uso de recursos, por lo que en general, el uso de este no debería ser comparable directamente en cuanto a rendimiento con el refinamiento adaptativo.

2. Refinamiento adaptativo: Los nuevos vértices se agregan de manera de representar la frontera real, por lo que se trasladan a la frontera utilizando una malla más fina.

La ventaja de agregar los vértices de esta manera es que se hace posible utilizar algunas herramientas para mejorar aspectos geométricos de esta. Para esto se utilizan herramientas como GAMer[25] (Geometry-preserving Adaptive MeshER).

Dentro de los recursos que entrega este software, se encuentra *ImproveSurfMesh*, el cual recibe un archivo de malla (.off) y reposiciona los vértices para flexibilizar la malla, mejorando así los tiempos de cálculo. Esto se logra limitando los ángulos de esta a ser mayores que 15° y menores que 150° .

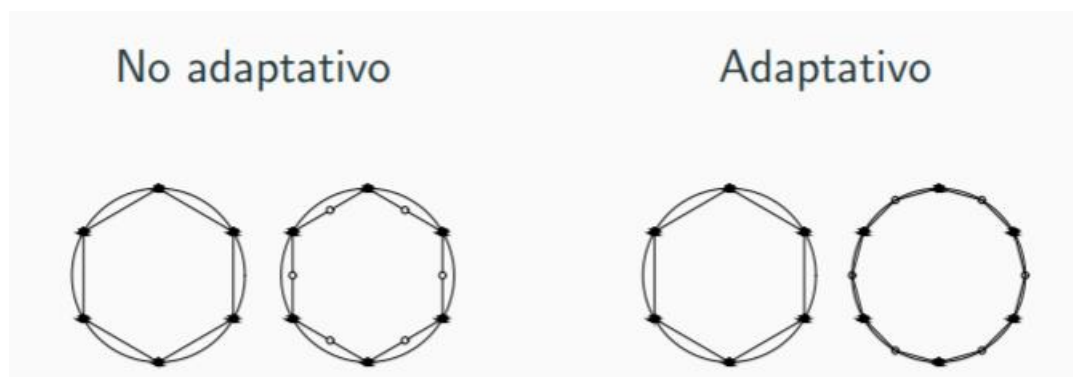


Figura 2.2: Representación gráfica de refinamiento no adaptativo vs refinamiento adaptativo[6].

En general, utilizar un refinamiento, ya sea adaptativo o no adaptativo disminuye el error local, sin embargo, en el caso de refinamiento no adaptativo la energía de solvatación puede converger a un valor diferente, pero esto sólo se da en mallas muy gruesas.

3. Análisis de los resultados

A continuación, se presentarán la mayoría de los resultados obtenidos. Primero se dará una explicación acerca del código adaptado y utilizado para obtener los resultados. Luego, se presentarán los análisis de dichos resultados y su validez.

3.1 Código

En breves palabras el código utilizado corresponde a un estimador de error, el cual permite realizar refinamientos adaptativos y no adaptativos a una malla dada. Además, permite calcular una estimación de Energía de Solvatación[6] aplicando los métodos previamente mencionados.

El código es capaz de reconocer y elaborar una malla mediante herramientas como MSMS y Nanoshaper, además puede trabajar con una malla ya importada o figuras preestablecidas de Bempp.

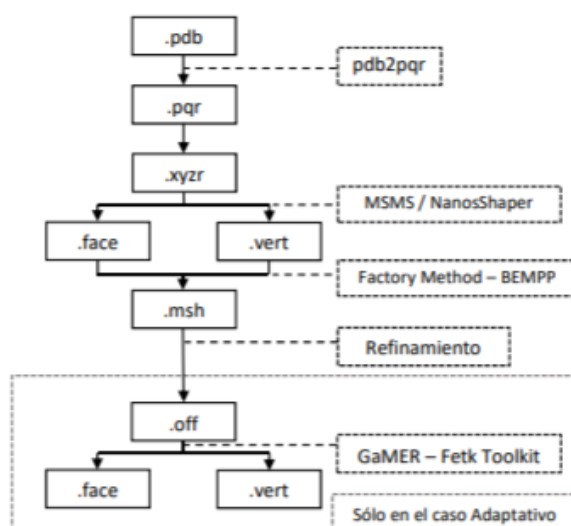


Figura 3.1: Secuencia de construcción de la malla[6].

El algoritmo para elaborar la malla consiste en utilizar un archivo .pdb el cual se obtiene desde RCSB, luego con la rutina *pdb_to_pqr* se obtiene el archivo con la ubicación, cargas y radio de Van der Waals para cada átomo, es decir, el archivo .pqr. El siguiente paso es generar el archivo que guarde la posición y el radio de Van der Waals de cada átomo (.xyzr) el cual es el input para cualquiera de las herramientas generadoras de mallas, ya

sea Nanoshaper o MSMS. La malla arrojada contiene 2 archivos, uno en el cual se guardan todos los vértices de la malla (.vert) y otro archivo donde se encuentran los elementos compuestos por los vértices, es decir, las caras (.face). Es en el siguiente punto el primer cambio notable que se tiene entre ambas versiones de Bempp, ya que en la versión 3.3.4 se utiliza el método Factory Method para construir el archivo que utiliza Bempp para leer la malla, pero en la versión más reciente, es decir, la 0.2.4 se utiliza *bempp.api.Grid*.

Posteriormente se realizan los cálculos para la obtención del potencial electrostático, el cual es necesario para calcular la Energía de Solvatación. Es en esta instancia donde surgen los primeros problemas en la adaptación del código, ya que a la hora de generar el lado derecho de la matriz los resultados no coincidían.

```
[ 2.58808169e-03  1.62113861e-03 -1.11696735e-03 -2.15266546e-03
-2.46334254e-03 -3.09950626e-03 -2.61145968e-03 -2.93678831e-03
-1.77125587e-03 -1.43403586e-03  1.01500815e-03  3.36363501e-03
 1.14924191e-03  1.18562558e-03  1.05150484e-03  1.06429670e-03
 1.19507411e-03  1.06544648e-03  7.74953415e-04  6.79938924e-04
 8.65031437e-04  7.44037648e-04  8.36009817e-04  4.80808091e-04
-2.58169946e-04 -5.89747381e-04  5.18739817e-04  1.04774819e-03
 1.38958536e-03  1.33637908e-03  1.05145670e-03  3.38541739e-04
-5.89747381e-04 -1.31735267e-05  6.64722359e-04 -7.62338706e-04
-7.57556692e-05 -9.98110465e-04 -2.78101810e-03 -1.19882240e-03
 4.81071294e-03  7.92556116e-03 -9.52790243e-04 -2.67325708e-03
-3.57150524e-03 -4.00463556e-03 -3.93678667e-03 -3.23170577e-03]
```

Tabla 3.1: Resultados prueba lado derecho. Python 2, Bempp 3.3.4.

```
[ 2.68077887e-03  1.72350181e-03 -1.09620401e-03 -2.13171001e-03
-2.44974289e-03 -3.08238570e-03 -2.59941868e-03 -2.91958936e-03
-1.75970792e-03 -1.41166288e-03  9.94886500e-04  3.37391114e-03
 1.14632156e-03  1.18374147e-03  1.04708970e-03  1.06163142e-03
 1.18907821e-03  1.05999867e-03  7.71188745e-04  6.75841208e-04
 8.62376575e-04  7.37604551e-04  8.30037621e-04  4.78408893e-04
-2.64724580e-04 -5.83201524e-04  5.12565243e-04  1.03384144e-03
 1.37909481e-03  1.33982432e-03  1.05092081e-03  3.37087761e-04
-5.83201524e-04 -4.07561780e-05  6.54254625e-04 -7.79496974e-04
-8.87699528e-05 -9.98824941e-04 -2.63849804e-03 -1.11752329e-03
 4.80079233e-03  7.24754435e-03 -8.53885572e-04 -2.64647524e-03
-3.55016220e-03 -3.97178518e-03 -3.90518339e-03 -3.15682491e-03]
```

Tabla 3.2: Resultados prueba lado derecho. Python 3, Bempp 0.2.4.

Como se observa en las tablas 3.1 y 3.2 existe una pequeña diferencia en las pruebas realizadas, esto sucede con cualquier molécula o proteína a comparar y se debe a que el número de puntos de cuadratura gausseana es menor en Bempp 3.3.4. Esto mismo ocurre a la hora de generar el lado derecho de la matriz en el cálculo del adjunto[6].

Finalmente, se realiza el refinamiento deseado, ya sea adaptativo o no adaptativo, el cual se mencionó anteriormente y mejora aún más los aspectos geométricos de la malla.

Sin embargo, se opta por utilizar PyGaMER principalmente por la mayor facilidad de acceso y de uso que esta herramienta presenta en contraste a su otra versión, ya que se puede instalar fácilmente como una extensión de Python.

Uno de los principales problemas de todo el algoritmo es el tiempo y uso de recursos que se tiene, el algoritmo GMRES tiende a demorar mucho sobre todo con moléculas grandes y con una gran cantidad de elementos, y este se utiliza tanto en el cálculo de potencial como del adjunto, además, las matrices generadas por BEMPP son bastante densas. Lo mismo sucede a la hora del refinamiento, al tener moléculas grandes los recursos invertidos son mayores, si bien modificar el número de iteraciones puede ayudar un poco con el tiempo de cálculo, siempre será necesario un considerable uso de recursos.

Es por esto último que se buscan alternativas para optimizar el cálculo del error y de esta manera mejorar los aspectos antes mencionados.

3.2 Análisis del Criterio a Seleccionar

Una de las ideas principales del código, como ya se ha mencionado, es la estimación del error en cada elemento y de esta manera disminuir el uso de recursos y obtener mejores resultados de energía. El problema principal de esta metodología es el tiempo de cálculo, lo cual se desea disminuir lo más posible. Es por esto que se desea verificar de entre una serie de criterios cual es el más adecuado, es decir, el que entrega los resultados más similares al de refinamiento por error y de esta manera comprobar si es posible optimizar dicho proceso.

Para esto se utilizan esferas generadas por Bempp y se tendrá en cuenta el radio de curvatura ($1/r$), el potencial de cada elemento, el área de la esfera y la forma de cada elemento (alargamiento de los triángulos al refinar, por ejemplo), la cual estará representada por un ratio de entre el lado más largo del triángulo dividido por el lado más pequeño, es decir, h_{max}/h_{min} .

A continuación, se ordenan los índices de los resultados de los criterios ya mencionados en orden ascendente, para lo cual se utiliza la función de numpy *argsort* para posteriormente calcular una distancia euclidiana mediante la función *linalg.norm* de entre los índices de los criterios vs los índices del error para cada elemento. Mientras la sumatoria de esta distancia sea más cercana a 0, mejor es dicho criterio.

Finalmente, se realizan 2 tipos de pruebas, una manteniendo el potencial global constante y otra manteniendo el radio de curvatura constante. Para cada prueba se

selecciona un rango de valores de carga y radio de una molécula arbitraria, en este caso la 1CID.

Para facilitar la generación de una tabla comparativa, los criterios y sus diferentes relaciones serán descritos como:

- 0 : Potencial.
- 1 : Ratio.
- 2 : Área.
- 3 : Potencial * Ratio.
- 4 : Potencial * Área.
- 5 : Ratio * Área.
- 6 : Potencial / Ratio.
- 7 : Ratio / Potencial.
- 8 : Área / Ratio.
- 9 : Ratio / Área.
- 10 : Área / Potencial.

Potencial Constante						
Radio (r)	0,6	0,8616	1,1232	1,3848	1,6464	1,908
Carga (q)	0,6	0,8616	1,1232	1,3848	1,6464	1,908
Criterio	0, 3, 6, 10	0, 3, 4, 10	0, 4, 6, 3	3, 4, 0, 6	0, 4, 6, 10	0, 4, 3, 6

Tabla 3.3: Evaluación de criterios bajo Potencial constante.

Radio de Curvatura Constante						
Radio (r)	1,459	1,459	1,459	1,459	1,459	1,459
Carga (q)	-0,8627	-0,5579	0,039	0,1305	0,3592	0,7341
Criterio	0, 3, 4, 6	0, 3, 4, 6	0, 3, 4, 6	0, 3, 4, 6	0, 3, 4, 6	0, 3, 4, 6

Tabla 3.4: Evaluación de criterios bajo Radio de curvatura constante.

Como se puede observar en los resultados para ambas tablas el criterio más relevante corresponde al potencial de cada elemento, además tienen bastante relevancia sus diferentes relaciones. Es por esto que se decide optar por realizar refinamientos según el potencial y así evaluar realmente que tanta relevancia tiene este frente a un refinamiento por error, además de comparar ambos resultados. De ser un resultado favorable significaría un buen ahorro de recursos computacionales y tiempo.

3.3 Análisis del Refinamiento por Potencial

Realizando un refinamiento según el potencial electrostático que posee cada elemento de la malla se busca comparar y anticipar los resultados obtenidos al realizar un refinamiento en base al error. Este se realiza de la misma forma que el refinamiento por error[6], tan sólo que los elementos seleccionados son ordenados según el potencial de cada uno.

En las tablas presentadas para cada molécula de prueba se muestran los refinamientos relacionados al error en el lado izquierdo y los relacionados al potencial en el lado derecho. Adicionalmente, se tendrán en cuenta 3 parámetros; N_ref refiriéndose al número de refinamientos, la cantidad de elementos de la malla y S como la energía de solvatación en $\left[\frac{kcal}{mol}\right]$.

A continuación, se incorporan los resultados obtenidos bajo los conceptos ya mencionados.

3.2.1 Metanol

Para construir la malla superficial de esta molécula se utiliza MSMS[23]. A partir de las diferentes mallas generadas según su densidad, se realizan refinamientos adaptativos en todos los casos, tanto por error como por potencial. Se utiliza 1 refinamiento adaptativo en el caso del potencial debido a que si se hacen más se estarían refinando los mismos elementos. Para el caso del refinamiento por error se realizan los suficientes refinamientos con tal de encontrar una similitud con el caso del potencial.

	Malla = 0.5				Malla = 0.6	
N_ref	Elementos	S		N_ref	Elementos	S
0	48	-5,1798		0	48	-5,1798
1	54	-4,0859		1	54	-4,0859
	Malla = 0.6				Malla = 0.6	
N_ref	Elementos	S		N_ref	Elementos	S
0	52	-4,2079		0	52	-4,2079
1	58	-3,9682		1	58	-3,9682

Tabla 3.5: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 0.5 y 0.6 respectivamente.

Como se puede observar en la Tabla 3.5, para el caso del Metanol con una densidad de malla baja, es decir, con una malla gruesa, la energía de solvatación tanto para el caso del refinamiento por error como por potencial son iguales.

	Malla = 1.0				Malla = 1.0	
N_ref	Elementos	S		N_ref	Elementos	S
0	86	-3,6019		0	86	-3,6019
1	92	-3,4629		1	96	-3,3526
2	98	-3,0875				
	Malla = 1.5				Malla = 1.5	
N_ref	Elementos	S		N_ref	Elementos	S
0	142	-3,2345		0	142	-3,2345
1	148	-3,1186		1	156	-3,0846
2	154	-3,0875				

Tabla 3.6: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 1.0 y 1.5 respectivamente.

En la Tabla 3.6 se puede apreciar una diferencia entre ambos tipos de refinamiento, la cual se acentúa un poco más en la malla de densidad 1.5, esta última tiene una mayor cantidad de elementos. Al ver estos resultados se podría decir que, a mayor cantidad de elementos entonces 1 refinamiento por potencial significa una “anticipación” de hasta 2 refinamientos por error.

	Malla = 2.0				Malla = 2.0	
N_ref	Elementos	S		N_ref	Elementos	S
0	188	-3,0902		0	188	-3,0902
1	206	-2,9587		1	206	-3,3526
	Malla = 2.5				Malla = 2.5	
N_ref	Elementos	S		N_ref	Elementos	S
0	232	-3,0189		0	232	-3,0189
1	238	-2,9441		1	254	-2,9006
2	244	-2,9302				
3	254	-2,8955				
	Malla = 3.0				Malla = 3.0	
0	286	-2,9546		0	286	-2,9546
1	292	-2,8939		1	314	-2,8480
2	304	-2,8870				
3	314	-2,8580				
4	322	-2,8362				
	Malla = 3.5				Malla = 3.5	
0	328	-2,9405		0	328	-2,9405
1	334	-2,8866		1	360	-2,8449
2	348	-2,8622				

Tabla 3.7: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 2.0, 2.5, 3.0 y 3.5 respectivamente.

Finalmente, como se muestra en la Tabla 3.7, al tener una densidad de malla 2.0 se tiene un caso único dentro de los casos estudiados, ya que al refinar por potencial equivale a 1 sólo refinamiento por error. Para los casos posteriores, es decir, con un mallado más fino, equivalente a una mayor cantidad de elementos, se tiene que el refinamiento por potencial equivale hasta aproximadamente 4 refinamientos por error, el cual se da en la densidad de malla 3.0.

3.2.2 Arginina

Para construir el mallado superficial de esta molécula se utiliza tanto Nanoshaper[26] como MSMS[23]. Esto debido a que el mallado con MSMS genera agujeros en la malla, lo cual, en algunos casos, principalmente en mallas finas genera errores a la hora de realizar los refinamientos. Por otro lado, Nanoshaper en algunas mallas genera problemas posteriores al refinamiento adaptativo, ya que genera ángulos de 0° , lo cual inhabilita el seguir refinando, así como el uso de Pygamer. Adicionalmente, se utiliza una densidad de malla de 80.0 en el caso de las mallas creadas por MSMS y de Grid Scale igual a 5.0 en el caso de Nanoshaper, esto principalmente para aminorar los problemas anteriormente mencionados.

Además, se consideran las mismas condiciones que en el caso del metanol, es decir, siempre se realizan refinamientos adaptativos en donde sólo se utiliza 1 en el caso del potencial y se refinará por error hasta obtener un resultado similar.

	Malla = 0.5				Malla = 0.5	
N_ref	Elementos	S		N_ref	Elementos	S
0	282	-78,9422		0	282	-78,9422
1	288	-73,2668		1	370	-70,1548
2	300	-73,2781				
3	312	-70,8314				
4	324	-69,5355				

Tabla 3.8: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 0.5 generada en MSMS.

En el caso de la molécula de arginina, la cual es bastante más grande que el metanol, se puede apreciar en la Tabla 3.8 que 1 refinamiento por potencial equivale hasta 4 refinamientos por error, lo cual podría fortalecer lo dicho para el caso de las mallas finas en el metanol, esto es, a mayor número de elementos mayor será la “anticipación” que tendrá el refinamiento por potencial.

	Malla = 1.0				Malla = 1.0	
N_ref	Elementos	S		N_ref	Elementos	S
0	366	-71,8099		0	366	-71,8099
1	372	-66,7753		1	388	-66,2999
2	388	-66,5528				
3	400	-65,8817				

Tabla 3.9: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para densidad de malla 1.0 generada en MSMS.

El caso presentado en la Tabla 3.9 es un caso especial, ya que se tuvo que refinar tan sólo el 2% de los elementos en el caso del potencial, esto se debe que al refinar una mayor cantidad de elementos se producía un error relacionado a vértices repetidos. Si bien este problema puede interpretarse como un mal calculo en los resultados, aun así tiene valor, principalmente porque sigue el mismo patrón que el resto de los resultados. De esta manera, se puede ver que el refinamiento por potencial equivale hasta 3 refinamientos por error.

	Grid Scale = 1.1				Grid Scale = 1.1	
N_ref	Elementos	S		N_ref	Elementos	S
0	692	-75,6761		0	692	-75,6761
1	698	-72,2603		1	854	-68,0579
2	704	-71,0613				
3	710	-69,3471				

Tabla 3.10: Refinamiento por Error a la izquierda vs Refinamiento por Potencial a la derecha para Grid Scale 1.1 generada en Nanoshaper.

Como último caso se realiza la prueba con una Grid Scale de 1.1 hecha en Nanoshaper, en donde se puede apreciar el mismo patrón mencionado en los casos anteriores.

4. Conclusiones

A partir de un trabajo anterior, en donde se buscaba reducir el error con una baja cantidad de recursos y así obtener resultados más acertados acerca de la energía de solvatación, surge la idea de verificar que aspecto es el que más influye en el cálculo del error. A priori se verificó que tanto influía la geometría de la molécula, el potencial de la misma y su radio de curvatura, en donde a base de las pruebas ya mencionadas en el punto anterior se verifica que el potencial electrostático de cada elemento es el que más influencia tiene en el error.

Al realizar las pruebas para las moléculas previamente mencionadas, tanto para el error como para el potencial se puede observar la relación que estos poseen. Principalmente en mallas gruesas con una pequeña cantidad de elementos se pudo observar que el refinamiento de elementos por potencial prácticamente equivale al refinamiento por error, lo que significa un gran ahorro de recursos y tiempo, ya que para el refinamiento por potencial no se necesita realizar los cálculos del adjunto, lo cual corresponde a una parte importante de la ganancia ya mencionada.

Sin embargo, en mallas más finas, es decir, con una mayor cantidad de elementos, el refinamiento por potencial parece anticiparse al refinamiento por error, lo cual significa una aún mayor ganancia en términos de recursos y tiempo. Pero, esto no puede ser verificado a totalidad, puesto que existen ciertos casos en donde el refinamiento por potencial se comporta de manera diferente y, además, por ahora no se puede medir que tanto se puede anticipar al refinamiento por error. Como se observa en los diferentes resultados para la molécula de arginina hay situaciones en donde el refinamiento por potencial se anticipa hasta 2, 3 o 4 refinamientos por error, esto en teoría podría ser aún más variable para moléculas más grandes, lo cual significa que es un tema de posterior estudio y análisis.

Otro aspecto posible por analizar en un futuro viene a ser el estudio acerca de qué otros aspectos podrían afectar el cálculo del error además de los ya mencionados, y qué tanta influencia estos podrían tener. De esta manera se podría saber en su totalidad que aspectos afectan mayormente al error y así anticiparse con mayor certeza y efectividad al mismo. Todo esto sin necesidad de realizar cálculos extensos y que requieren una gran cantidad de recursos computacionales.

Del mismo modo, también se debe estudiar en mayor profundidad otras herramientas que generen mallas como podría ser QuickSES, realizar pruebas del mismo a la hora de

probar mallas y así probablemente evitar los errores que generalmente arrojan MSMS y Nanoshaper, ya que estas dos herramientas generaron muchos problemas a la hora de realizar las pruebas del refinamiento por potencial, debido a agujeros en las mallas y ángulos cercanos a $0[^\circ]$. Por otra parte, también se pueden revisar algunas herramientas que permitan corregir problemas como los antes mencionados y de esta manera no recurrir a la prueba y error de un nuevo mallador.

Referencias

- [1] Martinez, Xavier, Michael Krone, and Marc Baaden. "QuickSES: A Library for Fast Computation of Solvent Excluded Surfaces." The Eurographics Association, 2019.
- [2] Cooper, Bardhan, Barba, A biomolecular electrostatics solver using Python, GPUs and boundary elements that can handle solvent-filled cavities and Stern layers, 2019.
- [3] Timo Betcke, Matthew W. Scroggs and Wojciech Smigaj. Product algebras for Galerkin discretisations of boundary operators and their applications, 2017.
- [4] Leighton Wilson, Robert Krasny, Comparison of the MSMS and NanoShaper molecular surface triangulation codes in the TABI Poisson–Boltzmann solver, 2020.
- [5] Kozlikova, B., Krone, M., Lindow, N., Falk, M., Baaden, M., Baum, D., Viola, I., Parulek, J., Hege, H.C. Visualization of molecular structure: State of the art revisited. Computer Graphics Forum (2016).
- [6] Vicente Ramm, Jehanzeb H. Chaudhry, Christopher D. Cooper. Efficient mesh refinement for the Poisson-Boltzmann equation with boundary elements, 2019.
- [7] Sergio Decherchi, Matteo Masetti, Ivan Vyalov, Walter Rocchia, Implicit solvent methods for free energy estimation, 2015.
- [8] Nathan A. Baker, Poisson–Boltzmann Methods for Biomolecular Electrostatics, 2004.
- [9] Byung Jun & Yoon A. M. Lenhoff (1990). A boundary element method for molecular electrostatics with electrolyte effects.
- [10] Michael D. Altman, Jaydeep P. Bardhan, Jacob K. White, Bruce Tidor. Accurate Solution of Multi-Region Continuum Biomolecule Electrostatic Problems Using the Linearized Poisson–Boltzmann Equation with Curved Boundary Elements. Wiley InterScience, Junio 2008.
- [11] Christopher Cooper Villagran. Biomolecular Electrostatics with Continuum models: A Boundary Integral Implementation and application to Biosensors.
- [12] Barry Honig and Anthony Nicholls, Classical Electrostatics in Biology and Chemistry. Science, New Series, Vol. 268, No. 5214. (Mayo 26 de 1995)

- [13] Ullmann G.M., Bombarda E. (2014) Continuum Electrostatic Analysis of Proteins. In: Naray-Szabó G. (eds) Protein Modelling. Springer, Cham.
- [14] Kirkwood J. G.; Buff F. P. The statistical mechanical theory of solutions. I. JCP 1951, 19, 774 -777.
- [15] William G. McMillan Jr. and Joseph E. Mayer. The Statistical Thermodynamics of Multicomponent Systems. J. Chem. Phys. 13, 276 (1945).
- [16] Gilson, M.; Honig, B. "Calculations of Electrostatic Potentials in an Enzyme Active Site" Nature, 330, 84 (1987).
- [17] IUPAC. Compendium of Chemical Terminology, 2nd ed. (the "Gold Book"). Compiled by A. D. McNaught and A. Wilkinson. Blackwell Scientific Publications, Oxford (1997). Online version (2019-) created by S. J. Chalk. ISBN 0-9678550-9-8.
- [18] Sergio Decherchi et al. Implicit solvent methods for free energy estimation. Eur J Med Chem. 2015 Feb 16; 0: 27–42.
- [19] Elizabeth Jurrus, Dave Engel, Keith Star, Kyle Monson, Juan Brandi, Lisa E. Felberg, David H. Brookes, Leighton Wilson, Jiahui Chen, Karina Liles, Minju Chun, Peter Li, David W. Gohara, Todd Dolinsky, Robert Konecny, David R. Koes, Jens Erik Nielsen, Teresa Head-Gordon, Weihua Geng, Robert Krasny, Guo-Wei Wei, Michael J. Holst, J. Andrew McCammon, and Nathan A. Baker. Improvements to the apbs biomolecular solvation software suite. Protein Science, 27(1):112–128, 2018.
- [20] Bash PA, Pattabiraman N, Huang C, Ferrin TE, Langridge R. (1983). Van Der Waals Surfaces in Molecular Modeling: Implementation with Real-Time Computer Graphics. Science.
- [21] Lee B, Richards FM. (1971). The Interpretation of Protein Structures: Estimation of Static Accessibility. Journal of Molecular Biology.
- [22] Connolly ML. (1983). Analytical Molecular Surface Calculation. Journal of Applied Crystallography.
- [23] Sanner, M.F., Spehner, J.-C., and Olson, A.J. (1996) Reduced surface: an efficient way to compute molecular surfaces. Biopolymers, Vol. 38, (3),305-320.

- [24] Lindow, N., Baum, D., Hege, H.C.: Ligand excluded surface: A new type of molecular surface. *IEEE Transactions on Visualization and Computer Graphics* 20(12), 2486–2495 (2014).
- [25] Z. Yu, M. Holst, Y. Cheng, and J.A. McCammon, Feature-Preserving Adaptive Mesh Generation for Molecular Shape Modeling and Simulation. *Journal of Molecular Graphics and Modeling*, 26(8) (2008), pp. 1370-1380.
- [26] S. Decherchi, W. Rocchia (2013), "A general and Robust Ray-Casting-Based Algorithm for Triangulating Surfaces at the Nanoscale".
- [27] Daberdaku Sebastian - 2018/11/20 - Identification of protein pockets and cavities by Euclidean Distance TransformER.
- [28] Steven M. Kast. An Introduction to Adjoints and Output Error Estimation in Computational Fluid Dynamics.
- [29] Jehanzeb H. Chaudhry, Posteriori Analysis and Efficient Refinement Strategies for the Poisson-Boltzmann Equation. *SIAM Journal on Scientific Computing* 2018, Vol 40, No. 4, pp. A2519-A2542A.
- [30] Stephen D. B, Jehanzeb Hameed Chaudhry, Eric C. Cyr, Lune N. Olson. A First-Order System Least-Squares Finite Element Method for the Poisson-Boltzmann Equation. 2009.