



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MECÁNICO

PROFESOR GUÍA: PHD. CHRISTOPHER COOPER
PROFESOR CORREFERENTE: PHD. ALEJANDRO PACHECHO

Implementación de campo de fuerza polarizable AMOEBA en modelación biomolecular utilizando el método de elementos de frontera

FELIPE VICENCIO HIDALGO
MAYO - 2021

Agradecimientos

No puedo dejar pasar cualquier oportunidad para agradecer de todo corazón a mis padres, a su apoyo constante y por creer siempre en mí.

Me gustaría agradecer también a mi hermano Francisco, por ser una constante fuente de risas e inspiración.

A toda mi familia, en especial a mi tía Claudia por estar siempre presente.

A todas las personas cercanas que conocí en este largo camino, sin duda hicieron de estos años un tiempo para recordar.

Quiero agradecer también a mi profesor guía, Christopher Cooper, por su tremenda disposición, por todo el tiempo dedicado a guiar este trabajo y por la gran motivación.

¡Muchas Gracias!

Índice

1. Introducción	7
2. Marco Teórico	8
2.1. Función de Green	8
2.2. Operadores de Borde	8
2.3. Segunda Identidad de Green	9
2.4. Método de Elementos de Frontera	9
2.5. Método de Galerkin	12
3. Modelo de Solvente Implícito	13
3.1. Formulación Integral	13
3.2. Modelos de Múltiple Superficie	17
3.3. Energía de Solvatación	18
3.4. Campos de Fuerza	18
3.5. Superficie Excluida de Solvente	18
4. Bempp	20
4.1. Espacios de funciones	20
4.2. Operadores	20
4.3. Función de Malla	21
5. Campo de Fuerza Polarizable	22
5.1. Formulación Matemática	22
5.2. Efectos de Polarización	23
5.3. Cálculo del Campo Eléctrico	23
5.3.1. Campo Eléctrico Debido al Solvente	23
5.3.2. Campo Eléctrico Debido a los Multipolos	24
5.4. Esquema iterativo autoconsistente	24
5.5. Energía de Solvatación con Multipolos	25
6. Ejemplos de implementación: Resultados	29
6.1. Esfera con multipolos puntuales	29
6.2. Geometrías Reales	31
6.2.1. Proteína 1pgb	31
6.2.2. Lisozima 1lyz	36
6.2.3. Citocina 1a7m	38
6.2.4. Proteína 1x1u	39
7. Conclusiones	41

Índice de figuras

3.1. Representación gráfica del modelo de solvente implícito ¹⁹	13
3.2. Definición gráfica de los distintos tipos de superficie ¹⁹	19
5.1. Representación gráfica en dos dimensiones del proceso de cálculo de derivadas por diferencias finitas	24
5.2. Representación gráfica en dos dimensiones del proceso de cálculo de segundas derivadas por diferencias finitas	27
6.1. Representación gráfica de los multipolos en el interior de la esfera ² utilizada para la validación del código realizado	29
6.2. Malla utilizada en caso de esfera con multipolos.	30
6.3. Gráfico de error vs N° de elementos de la malla para la esfera con multipolos	31
6.4. Ejemplo de malla utilizada en la modelación de la proteína 1pgb	32
6.5. Comparación entre el código desarrollado con Bempp y Pygbe para la proteína 1pgb	33
6.6. Comparación entre el código desarrollado con Bempp y Pygbe para la proteína 1pgb sin considerar cuadrupolos	33
6.7. Comparación entre el código desarrollado con Bempp y Pygbe para la proteína 1pgb sin considerar polarizabilidad	34
6.8. Comparación entre el código desarrollado con Bempp y Pygbe para la proteína 1pgb sin considerar dipolos ni polarizabilidad	34
6.9. Comparación entre un campo de fuerza polarizable y un campo de fuerza clásico para la proteína 1pgb, considerando $\epsilon_{prot} = 1$	35
6.10. Comparación entre un campo de fuerza polarizable y un campo de fuerza clásico para la proteína 1pgb, considerando $\epsilon_{prot} = 2$ para el campo de fuerza clásico.	36
6.11. Una de las mallas utilizadas para la simulación de la lisozima 1lyz	37
6.12. Comparación entre el código desarrollado con Bempp y Pygbe para la lisozima 1lyz	37
6.13. Una de las mallas utilizadas para la simulación de la Citocina 1a7m	38
6.14. Comparación entre el código desarrollado con Bempp y Pygbe para la citocina 1a7m	38
6.15. Una de las mallas utilizadas para la simulación de la Proteína 1x1u	39
6.16. Comparación entre el código desarrollado con Bempp y Pygbe para la Proteína 1x1u	40

Índice de tablas

6.1. Resumen de comparación entre las distintas biomoléculas simuladas	40
--	----

Lista de Simbolos

Simbolo	Descripción
δ	Función Delta de Dirac
d_k	Componente permanente del dipolo k
$\vec{\mathbf{E}}$	Campo Eléctrico
$\vec{\mathbf{E}}_{solv}$	Campo Eléctrico debido al solvente
$\vec{\mathbf{E}}_{mult}$	Campo Eléctrico debido a los multipolos
G_L	Función de espacio libre de Green de la ecuación de Laplace
G_Y	Función de espacio libre de Green de la ecuación de Poisson-Boltzmann linealizada
G_{solv}	Energía de solvatación
κ	Inverso de la longitud de Debye
K_L^Γ	Operador potencial de capa doble de la ecuación de Laplace evaluado en Γ
K_Y^Γ	Operador potencial de capa doble de la ecuación de Poisson-Boltzmann evaluado en Γ
\mathbf{n}	Vector normal unitario
μ_k	Componente polarizable del dipolo
q_k	Magnitud de la carga puntual k
Q_k	Cuadrupolo puntual k representado por un tensor 3×3
p_k	Dipolo total k representado por un vector
ϕ	Potencial electrostático
ϕ_{coul}	Potencial de coulomb
ϕ_{reac}	Potencial de reacción debido al solvente
ρ	Densidad o distribución de carga
V_L^Γ	Operador potencial de capa simple de la ecuación de Laplace evaluado en Γ
V_Y^Γ	Operador potencial de capa simple de la ecuación de Poisson-Boltzmann evaluado en Γ
x	Vector posición
Γ	Superficie
ω	Coefficiente de relajación
Ω	Volumen
W_{pol}	Energía de polarización de los multipolos

Siglas

BEM Metodo de elementos de frontera. 9

BIE Ecuación Integral de frontera. 9

GMRES Método generalizado de mínimos residuales. 30

SES Superficie excluida de solvente. 13

Abstract: The continuous models have an important role in the modeling of biomolecular systems, obtaining results with lower execution times and computational resources. Using a Poisson-Boltzmann and the boundary element method, the nature of the problem is accurately represented and the calculations are significantly simplified. In this context, the polarizable force field AMOEBA is implemented in order to achieve a more realistic model using higher degree multipoles, this is, a dipole and Quadrupole. The dipole has two components, the permanent dipole, given by the forcefield, and the polarizable dipole, who depends on the electric field, therefore, it is calculated using an iterative scheme. The multipolar formulation requires the calculation of the first and second degree derivative from the electrostatic potential to compute the electric field and the solvation energy. The derivative of the multipoles component for the electrostatic potential is compute by analytical functions, the derivative of the reaction component are calculated by finite difference scheme. The formulation and implementation codes are validated by direct comparison with the Software Pygbe. In this report, the Python library, Bempp is used, to compute simulation results using the implicit solvent model with AMOEBA polarizable force field. The results obtained show differences of less than 1 % for each simulated case, this implies that the formulated models are consistent and Bempp library is a good option to implicit solvent models using boundary element method.

Resumen: Los modelos continuos juegan un papel importante en la modelación de sistemas biomoleculares, ya que permiten obtener resultados con tiempos de ejecución bajos y pocos recursos computacionales, específicamente, los modelos continuos permiten un rápido cálculo de la energía de solvatación, la cual puede resultar complicada de obtener por medio de otros modelos. En las simulaciones se utilizo un modelo de Poisson- Boltzmann y el método de elementos de frontera, el cual representa de manera precisa la naturaleza del problema y simplifica considerablemente los cálculos. Este trabajo consiste en la implementación del campo de fuerza polarizable AMOEBA, el cual considera componente multipolares de grado superior, tales como dipolos y cuadrupolos, para lograr una representación más realista de la situación. El dipolo se divide a su vez en dos componentes, el dipolo permanente, dado por el campo de fuerza, y una componente de dipolo polarizable, la cual depende del campo eléctrico en la ubicación de los multipolos, debido a esto, la componente polarizable del dipolo se calcula por medio de un esquema iterativo autoconsistente. Para los distintos cálculos, se utilizo la librería de Python, Bempp, la cual resulta conveniente para para aplicar y desarrollar el método de elementos de frontera. La formulación para campos de fuerza polarizables requiere del cálculo de derivadas de primer y segundo grado del potencial electrostático, en el caso de la componente del potencial debido a los multipolos, esta se calculo mediante funciones analíticas, en cambio, para la componente de reacción debido al solvente, Bempp no ofrece métodos para su cálculo, por lo tanto, esta se aproximo mediante diferencias finitas. Con toda la formulación presentada a lo largo de este informe, se desarrolló un código de Python en formato de librería, capaz de calcular la energía de solvatación de distintas biomoléculas, para su validación se comparó los resultados obtenidos con el Software Pygbe. Los resultados obtenidos con el código desarrollado presentan diferencias menores al 1 % para cada caso simulado, lo cual significa que los modelos desarrollados e implementados son consistentes y el código desarrollado utilizando Bempp representa una herramienta confiable para simulaciones de modelo de solvente implícito utilizando el método de elementos de frontera.

1. Introducción

En la actualidad, resulta de interés generar modelos confiables capaces de simular el proceso de disolución de biomoléculas, en este contexto, existen distintas formas capaces de abarcar este problema.

Por un lado, existe el modelo atómico, que intenta describir la situación en base a las relaciones presentes entre los átomos de la biomolécula y los iones del solvente, luego, se posee también los modelos de dinámica molecular que estudian la evolución dinámica de sistemas moleculares en el tiempo, basándose en la segunda ley de Newton. El principal inconveniente de este tipo de modelos es que frente a sistemas con gran número de elementos, se requiere de muchos recursos computacionales para llevar a cabo simulaciones, además de tomar un tiempo considerable en estas, también se debe considerar, la dificultad para determinar ciertos parámetros por medio de dinámica molecular, como por ejemplo, la energía de solvatación, de la cual se hablará más adelante.

Debido a lo anterior, como una alternativa muy eficiente y complementaria a los métodos de modelación ya comentados, se encuentran los modelos continuos, cuya principal ventaja es la facilidad para representar modelos de gran tamaño sin utilizar muchos recursos computacionales, tiempos de simulación cortos en comparación a otros modelos y además, obtener fácilmente valores de energía de solvatación que resultan muy complejo determinar con dinámica molecular.

Se debe considerar también que en la comunidad no existe un acuerdo respecto de los modelos a utilizar, en este contexto, los modelos continuos juegan un rol muy importante, debido a que logra una descripción promedio del comportamiento de la molécula y, como se menciono antes, permite calcular de forma rápida, la energía libre de solvatación.

Buscando métodos que permitan una representación más realista de la situación se encuentran los campos de fuerza polarizables, los cuales utilizando una distribución de carga con componentes multipolares de orden superior logran una modelación más certera del proceso de disolución de las biomoléculas. Para la formulación matemática de los modelos continuos se tiene el método de elementos de frontera (BEM), como principales ventajas, brinda una descripción fiel de la molécula y se obtiene un potencial cero al infinito, lo cual es difícil de lograr con otros métodos.

Los objetivos de este trabajo son:

- Realizar la formulación matemática del modelo de solvente implícito para una biomolécula utilizando el método de elementos de frontera para un campo de fuerzas polarizable.
- Desarrollar un código en el lenguaje de programación Python, que permita calcular energías de solvatación de biomoléculas utilizando el campo de fuerza polarizable AMOEBA.
- Estudiar las diferencias en los cálculos realizados en comparación a un campo de fuerza clásico.
- Acoplar el código realizado a una librería de Python que permita ejecutar y realizar cálculos de manera rápida utilizando la interfaz de Jupyter Notebooks.

2. Marco Teórico

En esta sección se abordarán distintos aspectos matemáticos que se utilizarán en derivaciones posteriores.

2.1. Función de Green

La función de Green es una función matemática utilizada como núcleo de un operador lineal de forma integral, se utiliza en la resolución de ecuaciones diferenciales con condiciones de contorno específicas. Por ejemplo, sea L un operador diferencial lineal, tal que:

$$L[f(x)] = g(x) \quad x \in \Omega \quad (2.1)$$

La función de Green cumple con:

$$L[G(x, x')] = -\delta(x - x') \quad (2.2)$$

Donde δ es la función delta de Dirac. Por ejemplo, para las ecuaciones que serán tratadas a lo largo de este trabajo, se tiene la ecuación de Poisson la cual se define como:

$$\nabla^2 \phi = f \quad (2.3)$$

En tal caso, se define la función de Green G_L como:

$$G_L(x, x') = \frac{1}{4\pi|x - x'|} \quad (2.4)$$

En cambio, para la ecuación de Poisson-Boltzman, descrita como:

$$(\nabla^2 - \kappa^2)\phi = 0 \quad (2.5)$$

La función de Green G_Y es:

$$G_Y(x, x') = \frac{\exp(-\kappa|x - x'|)}{4\pi|x - x'|} \quad (2.6)$$

2.2. Operadores de Borde

- **Operador potencial de capa simple:** Se define el operador potencial de capa simple, aplicado sobre una distribución $\psi(x)$ como:

$$V[\psi(x)] = \int G(x, x')\psi(x')dx' \quad (2.7)$$

Donde $G(x, x')$ es la función de Green. Físicamente, el operador V representa el potencial debido a la distribución de carga superficial $\psi(x)$.

- **Operador potencial de capa doble:** En este caso, el operador potencial de capa doble, aplicado sobre una distribución $\psi(x)$ es:

$$K[\psi(x)] = \int \frac{\partial}{\partial n}(G(x, x'))\psi(x')dx' \quad (2.8)$$

En este caso, el operador K representa el potencial debido a una distribución superficial de dipolos $\psi(x)$.

Notar que tanto K como V dependen de la función de Green, por lo tanto, estos serán distintos según la ecuación que se trabaja.

2.3. Segunda Identidad de Green

Sean ψ y ϕ dos funciones diferenciables en un espacio Ω limitado por una superficie Γ , entonces se tiene:

$$\int_{\Omega} (\nabla^2 \psi \phi - \psi \nabla^2 \phi) d\Omega = \int_{\Gamma} \phi \frac{\partial \psi}{\partial n} - \psi \frac{\partial \phi}{\partial n} d\Gamma \quad (2.9)$$

Donde n es la dirección normal a la superficie Γ , apuntando hacia el exterior.

2.4. Método de Elementos de Frontera

El método de elementos de frontera (BEM, por sus siglas en ingles) resuelve ecuaciones integrales de borde (BIE) de forma numérica. Resulta muy útil en la resolución de EDP cuando estas pueden escribirse como BIE.

Ejemplo: Ecuación de Laplace. Una ecuación que puede ser resuelta con BEM es la ecuación de Laplace, la ecuación de Laplace es un caso especial de la ecuación de Poisson (2.3), donde $f = 0$. Para una función escalar ϕ en un dominio Ω con superficie Γ y con condiciones de borde conocidas, pudiendo ser estas de Dirichlet o Neumann, la ecuación es la siguiente:

$$\begin{aligned} \nabla^2 \phi &= 0 \text{ en } \Omega \\ \phi &= \phi_0 \text{ o} \\ \frac{\partial \phi}{\partial n} &= \frac{\partial \phi_0}{\partial n} \text{ en } \Gamma \end{aligned} \quad (2.10)$$

Primero, se “debilita” la ecuación, multiplicándola por una función peso w e integrando sobre todo el dominio, obteniendo la siguiente ecuación integral (IE):

$$\int_{\Omega} \nabla^2 \phi(x'_{\Omega}) w(x'_{\Omega}) d\Omega' = 0 \quad (2.11)$$

Como función de peso se utilizará la función de Green (descrita en la ecuación (2.4)), por lo tanto la ecuación (2.11) queda como:

$$\int_{\Omega} \nabla^2 \phi(x'_{\Omega}) G_L(x_{\Omega}, x'_{\Omega}) d\Omega' = 0 \quad (2.12)$$

Antes de continuar, se debe definir una identidad de la divergencia de funciones vectoriales que resultará útil para el desarrollo, sean f y g dos funciones vectoriales, se tiene entonces que:

$$\nabla(f \cdot g) = \nabla f \cdot g + f \cdot \nabla g \quad (2.13)$$

La ecuación (2.13) es importante ya que, considerando $f = \nabla \phi$ y $g = G_L$ y despejando el primer termino del lado derecho, es posible reescribir la ecuación (2.12) como:

$$\int_{\Omega} \nabla(\nabla \phi(x'_{\Omega}) G_L(x_{\Omega}, x'_{\Omega})) - \nabla \phi(x'_{\Omega}) \nabla G_L(x_{\Omega}, x'_{\Omega}) d\Omega' = 0 \quad (2.14)$$

Es posible seguir desarrollando el segundo termino de esta ecuación, utilizando nuevamente la ecuación (2.13) y considerando esta vez $f = \phi$ y $g = \nabla G_L$, se llega a:

$$\int_{\Omega} \nabla(\nabla \phi(x'_{\Omega}) G_L(x_{\Omega}, x'_{\Omega})) - (\nabla(\phi(x'_{\Omega}) \nabla G_L(x_{\Omega}, x'_{\Omega})) - \phi(x'_{\Omega}) \nabla^2 G_L(x_{\Omega}, x'_{\Omega})) d\Omega' = 0 \quad (2.15)$$

Notar que, en la ecuación (2.15), el operador divergencia esta aplicado sobre los primeros dos términos, por lo tanto, se puede usar el teorema de la divergencia o teorema de Gauss, el cual relaciona el flujo de un campo vectorial a través de una superficie, con la integral de su divergencia en el espacio cerrado por la superficie, es decir:

$$\int_{\Gamma} F \cdot d\Gamma = \int_{\Omega} \nabla \cdot F d\Omega \quad (2.16)$$

Utilizando la ecuación (2.16) para los primeros dos términos, y la ecuación (2.2) para el tercer término, la ecuación (2.15) se reduce a:

$$\int_{\Gamma} \nabla \phi(x'_{\Gamma}) G_L(x_{\Omega}, x'_{\Gamma}) d\Gamma' - \int_{\Gamma} \phi(x'_{\Gamma}) \nabla G_L(x_{\Omega}, x'_{\Gamma}) d\Gamma' - \int_{\Omega} \phi(x'_{\Omega}) \delta(r_{\Omega} - r'_{\Omega}) d\Omega' = 0 \quad (2.17)$$

Por definición, el tercer termino es igual a $\phi(x_{\Omega})$, por lo tanto, reorganizando términos:

$$\phi(x_{\Omega}) = \int_{\Gamma} \frac{\partial \phi}{\partial n}(x'_{\Gamma}) G_L(x_{\Omega}, x'_{\Gamma}) d\Gamma' - \int_{\Gamma} \frac{\partial G_L}{\partial n}(x_{\Omega}, x'_{\Gamma}) \phi(x'_{\Gamma}) d\Gamma' \quad (2.18)$$

O también, escrito en función de los operadores de borde vistos en la sección 2.2:

$$\phi(x_{\Omega}) = V_L \left[\frac{\partial \phi}{\partial n} \right] - K_L[\phi] \quad (2.19)$$

Notar que x_{Ω} puede ser un punto cualquiera dentro del dominio Ω , mientras que x'_{Γ} puede estar únicamente en la superficie, por lo tanto, existe una singularidad cuando x_{Ω} se aproxima

a Γ , es decir, cuando $x = x'$. Para evitar este problema, la integral se resolverá sobre una superficie Γ' , idéntica a Γ en todas partes excepto en x , ya que contiene una semiesfera de radio ε centrada en este punto, la integral sobre Γ' se separa entre las integrales sobre la región donde Γ y Γ' coinciden más la integral sobre la semiesfera, se denotará esta superficie como Γ'_{esf} . Notar que al tomar el límite cuando $\varepsilon \rightarrow 0$ se recupera la superficie original. Por lo tanto, se deben calcular las integrales sobre la semiesfera:

$$\int_{\Gamma'_{\text{esf}}} \frac{\partial \phi}{\partial n}(x') G_L(x, x') d\Gamma' = \frac{1}{4\pi} \int_{\Gamma'_{\text{esf}}} \frac{\partial \phi}{\partial n}(x') \frac{1}{|x - x'|} d\Gamma' \quad (2.20)$$

Notar que, al integrar sobre la semiesfera, la magnitud $|x - x'|$ es constante e igual a ε por lo tanto, reescribiendo la ecuación (2.20) y tomando el límite cuando $\varepsilon \rightarrow 0$, se tiene:

$$\lim_{\varepsilon \rightarrow 0} \frac{1}{4\pi\varepsilon} \int_{\Gamma'_{\text{esf}}} \frac{\partial \phi}{\partial n}(x') d\Gamma' = \lim_{\varepsilon \rightarrow 0} \frac{1}{4\pi\varepsilon} \frac{\partial \phi}{\partial n}(x) \int_{\Gamma'_{\text{esf}}} d\Gamma' \quad (2.21)$$

La integral $\int_{\Gamma'_{\text{esf}}} d\Gamma'$ corresponde a la superficie de la semiesfera, por lo tanto, la ecuación (2.21) se reduce a:

$$\begin{aligned} \lim_{\varepsilon \rightarrow 0} \frac{1}{4\pi\varepsilon} \frac{\partial \phi}{\partial n}(x') 2\pi\varepsilon^2 &= \lim_{\varepsilon \rightarrow 0} \frac{1}{2\pi} \frac{\partial \phi}{\partial n}(x') \pi\varepsilon \\ &= 0 \end{aligned} \quad (2.22)$$

Se debe seguir el mismo procedimiento para la segunda integral, pero antes se debe calcular la derivada de la función de Green en la semiesfera, esta es:

$$\frac{\partial G_L}{\partial n}(x, x') = -\frac{1}{4\pi} \frac{(x - x') \cdot n}{|x - x'|^3} \quad (2.23)$$

Siguiendo la misma derivación que las utilizadas desde la ecuación (2.20) hasta (2.22), la segunda integral se reduce a:

$$\begin{aligned} \lim_{\varepsilon \rightarrow 0} -\frac{1}{4\pi\varepsilon^2} \phi(x) \int_{\Gamma'_{\text{esf}}} d\Gamma' &= \lim_{\varepsilon \rightarrow 0} -\frac{1}{4\pi\varepsilon^2} \phi(x) 2\pi\varepsilon^2 \\ &= -\frac{\phi(x)}{2} \end{aligned} \quad (2.24)$$

Con los resultados obtenidos en (2.22) y (2.24), es posible calcular el límite de la ecuación (2.18) cuando $x \rightarrow \Gamma$:

$$\frac{\phi(x_\Gamma)}{2} - \int_\Gamma \frac{\partial \phi}{\partial n}(x'_\Gamma) G_L(x_\Gamma, x'_\Gamma) d\Gamma' + \int_\Gamma \frac{\partial G_L}{\partial n}(x_\Gamma, x'_\Gamma) \phi(x'_\Gamma) d\Gamma' = 0 \quad (2.25)$$

O, de forma equivalente, escrito en función de operadores de borde:

$$\frac{\phi(x_\Gamma)}{2} - V_L^\Gamma \left[\frac{\partial \phi}{\partial n} \right] + K_L^\Gamma[\phi] = 0 \quad (2.26)$$

La cual corresponde a una ecuación integral de borde.

2.5. Método de Galerkin

El método de Galerkin corresponde a a la conversión de un problema continuo a uno discreto, a continuación se observa un ejemplo de discretización utilizando este método para la ecuación de Laplace.

Para este caso, se supone que la información de borde v es conocida, y se busca obtener t , como se mostró en la sección 2.4, la ecuación para este caso es:

$$V[t] = \left(\frac{1}{2}I + K \right) [v] \quad (2.27)$$

Se tiene que en $L^2(\Gamma)$ el producto interno esta denotado por:

$$\langle f, g \rangle := \int_{\Gamma} \bar{f}(x)g(x)d\Gamma \quad (2.28)$$

Se puede mostrar la variación de la ecuación (2.27) como se ve a continuación. Se debe encontrar t tal que:

$$\langle \psi, Vt \rangle = \left\langle \psi, \left(\frac{1}{2}I + K \right) v \right\rangle \quad (2.29)$$

Con lo que se obtiene la formulación discretizada de Galerkin, que tiene forma matricial:

$$Vt = \left(\frac{1}{2}M + K \right) v \quad (2.30)$$

Donde las matrices están definidas por:

$$\begin{aligned} V_{i,j} &= \int_{\Gamma} \psi_i^{(0)}(x) \int_{\Gamma} G(x,y)\phi_j^{(0)}(y)d\Gamma(y)d\Gamma(x) \\ M_{i,j} &= \int_{\Gamma} \psi_i^{(0)}(x)\phi_j^{(1)}(x)d\Gamma(x) \\ K_{i,j} &= \int_{\Gamma} \psi_i^{(0)}(x) \int_{\Gamma} G(x,y)\phi_j^{(1)}(y)d\Gamma(y)d\Gamma(x) \end{aligned} \quad (2.31)$$

3. Modelo de Solvente Implícito

El modelo de solvente implícito utiliza dos regiones para representar una biomolécula en un medio acuoso (pudiendo ser más, esto se explicará más adelante), una región denominada Ω_1 que representa la biomolécula, esta región posee una constante dieléctrica baja (ϵ_1) y una distribución de carga discreta. La segunda región (Ω_2) corresponde al solvente, generalmente agua con sal, el cual se modela como un medio dieléctrico continuo de alta constante (ϵ_2). La separación entre ambas regiones corresponde a la superficie excluida de solvente (SES), denotada por Γ , y representa lo más cerca que puede estar una molécula de agua de la biomolécula. La teoría electrostática continua permite plantear un sistema de ecuaciones acoplado para el potencial ϕ , donde la región de la biomolécula esta gobernada por una ecuación de Poisson, mientras que la región del solvente esta gobernada por una ecuación de Poisson-Boltzmann linealizada (también conocida como ecuación de Helmholtz modificada), tal que:

$$\begin{aligned} \nabla^2 \phi_1(x) &= - \sum_k \frac{q_k}{\epsilon_1} \delta(x, x_k) && \text{en } \Omega_1 \\ (\nabla^2 - \kappa^2) \phi_2(x) &= 0 && \text{en } \Omega_2 \\ \phi_1 &= \phi_2 && \text{en } \Gamma \\ \epsilon_1 \frac{\partial \phi_1}{\partial n} &= \epsilon_2 \frac{\partial \phi_2}{\partial n} && \text{en } \Gamma \end{aligned} \quad (3.1)$$

Donde κ es el inverso de la longitud de Debye y n es la dirección normal a la SES, apuntando hacia la región del solvente.

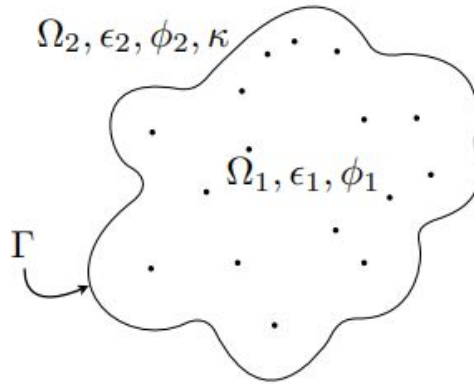


Figura 3.1: Representación gráfica del modelo de solvente implícito¹⁹

En la figura 3.1, la región Ω_1 representa la región de la biomolécula mientras que Ω_2 es la región del solvente.

3.1. Formulación Integral

El sistema de ecuaciones (3.1) puede ser resuelto utilizando el método de elementos de frontera, detallado en la sección 2.4 y así obtener de forma numérica el valor de ϕ_1 en distintos

puntos dentro de la región Ω_1 . Para esto, se debe transformar el problema a un sistema de ecuaciones integrales, se comenzará por debilitar las ecuaciones, similar al ejemplo realizado para la ecuación de Laplace, en el caso de la primera ecuación del sistema (3.1) se tiene:

$$\int_{\Omega_1} \nabla^2 \phi_1(x'_{\Omega_1}) G_L(x_{\Omega_1}, x'_{\Omega_1}) d\Omega' = - \int_{\Omega_1} \sum_k \frac{q_k}{\epsilon_1} \delta(x'_{\Omega_1}, x_k) G_L(x_{\Omega_1}, x') d\Omega' \quad (3.2)$$

Por definición, el lado derecho de la ecuación queda como:

$$\int_{\Omega_1} \sum_k \frac{q_k}{\epsilon_1} \delta(x'_{\Omega_1}, x_k) G_L(x_{\Omega_1}, x') d\Omega' = \sum_k \frac{q_k}{\epsilon_1} G_L(x, x_k) \quad (3.3)$$

Para el lado izquierdo de la misma ecuación, se tiene:

$$\begin{aligned} \int_{\Omega_1} \nabla^2 \phi_1(x'_{\Omega_1}) G_L(x_{\Omega_1}, x'_{\Omega_1}) d\Omega' &= \int_{\Omega_1} (\nabla^2 \phi_1(x'_{\Omega_1}) G_L(x_{\Omega_1}, x'_{\Omega_1}) - \phi_1(x'_{\Omega_1}) \nabla^2 G_L(x_{\Omega_1}, x'_{\Omega_1}) \\ &\quad + \phi_1(x'_{\Omega_1}) \nabla^2 G_L(x_{\Omega_1}, x'_{\Omega_1})) d\Omega' \end{aligned} \quad (3.4)$$

Notar que, se puede agrupar los dos primeros términos y aplicar la segunda identidad de Green (Eq. 2.9) para obtener lo siguiente:

$$\begin{aligned} \int_{\Omega_1} (\nabla^2 \phi_1(x'_{\Omega_1}) G_L(x_{\Omega_1}, x'_{\Omega_1}) - \phi_1(x'_{\Omega_1}) \nabla^2 G_L(x_{\Omega_1}, x'_{\Omega_1})) d\Omega' &= \int_{\Gamma} (G_L(x_{\Omega_1}, x'_{\Gamma}) \frac{\partial \phi_1}{\partial \mathbf{n}}(x'_{\Gamma}) \\ &\quad - \phi_1(x'_{\Gamma}) \frac{\partial G_L}{\partial \mathbf{n}}(x_{\Omega_1}, x'_{\Gamma})) d\Gamma' \end{aligned} \quad (3.5)$$

Adicionalmente, para el tercer termino en la ecuación (3.4), por definición de la función de Green (2.2), se tiene:

$$\begin{aligned} \int_{\Omega_1} \phi_1(x'_{\Omega_1}) \nabla^2 G_L(x_{\Omega_1}, x'_{\Omega_1}) d\Omega' &= - \int_{\Omega_1} \phi_1(x'_{\Omega_1}) \delta(x_{\Omega_1}, x'_{\Omega_1}) d\Omega' \\ &= -\phi_1(x_{\Omega_1}) \end{aligned} \quad (3.6)$$

Finalmente, utilizando las ecuaciones (3.3), (3.5) y (3.6), además de aplicar los operadores de borde, vistos en la sección 2.2, a la ecuación (3.5), la primera ecuación del sistema (3.1) queda como:

$$\phi_1(x_{\Omega_1}) + K_L[\phi_1] - V_L \left[\frac{\partial \phi_1}{\partial \mathbf{n}} \right] = \sum_k \frac{q_k}{4\pi\epsilon_1 |x - x_k|} \quad (3.7)$$

Notar que el procedimiento para llevar la ecuación desde (3.2) hasta (3.7) es equivalente al utilizado en el ejemplo de la ecuación de Laplace desde (2.11) hasta (2.18).

Luego, se debe realizar el mismo procedimiento para la segunda ecuación del sistema (3.1), la formulación débil de esta ecuación es:

$$\int_{\Omega_2} (\nabla^2 - \kappa^2)\phi_2(x'_{\Omega_2})G_Y(x_{\Omega_2}, x'_{\Omega_2})d\Omega' = 0 \quad (3.8)$$

La ecuación (3.8) puede expandirse de manera similar a lo realizado en la ecuación (3.4), en este caso se obtiene:

$$\begin{aligned} \int_{\Omega_2} (\nabla^2 - \kappa^2)\phi_2(x'_{\Omega_2})G_Y(x_{\Omega_2}, x'_{\Omega_2})d\Omega' &= \int_{\Omega_2} (\nabla^2\phi_2(x'_{\Omega_2})G_Y(x_{\Omega_2}, x'_{\Omega_2}) - \phi_2(x'_{\Omega_2})\nabla^2G_Y(x_{\Omega_2}, x'_{\Omega_2}) \\ &+ \phi_2(x'_{\Omega_2})\nabla^2G_Y(x_{\Omega_2}, x'_{\Omega_2}) - \kappa^2\phi_2(x'_{\Omega_2})G_Y(x_{\Omega_2}, x'_{\Omega_2}))d\Omega' \end{aligned} \quad (3.9)$$

Luego, es posible agrupar los dos primeros términos del lado derecho y aplicar la segunda identidad de Green, tal que:

$$\begin{aligned} \int_{\Omega_2} (\nabla^2\phi_2(x'_{\Omega_2})G_Y(x_{\Omega_2}, x'_{\Omega_2}) - \phi_2(x'_{\Omega_2})\nabla^2G_Y(x_{\Omega_2}, x'_{\Omega_2}))d\Omega' &= \int_{\Gamma} (-G_Y(x_{\Omega_2}, x'_{\Gamma})\frac{\partial\phi_2}{\partial\mathbf{n}}(x'_{\Gamma}) \\ &+ \phi_2(x'_{\Gamma})\frac{\partial G_Y}{\partial\mathbf{n}}(x_{\Omega_2}, x'_{\Gamma}))d\Gamma' \end{aligned} \quad (3.10)$$

Notar que se han invertido los signos al aplicar la segunda identidad de Green, esto se debe a que $\frac{\partial}{\partial\mathbf{n}}$ es una derivada direccional, y como se menciono en su respectiva sección, \mathbf{n} es la dirección normal a la superficie Γ , apuntando hacia el exterior, sin embargo, el exterior para Ω_2 corresponde a la región Ω_1 , por lo tanto, al ser una dirección opuesta (180°) se agrega un signo negativo.

Los dos términos restantes, es posible agruparlos para formar el operador diferencial de la ecuación de Poisson-Boltzmann y aplicar la definición de la ecuación de Green, es decir:

$$\begin{aligned} \int_{\Omega_2} \phi_2(x'_{\Omega_2})\nabla^2G_Y(x_{\Omega_2}, x'_{\Omega_2}) - \kappa^2\phi_2(x'_{\Omega_2})G_Y(x_{\Omega_2}, x'_{\Omega_2})d\Omega' &= \int_{\Omega_2} \phi_2(x'_{\Omega_2})(\nabla^2 - \kappa^2)G_Y(x_{\Omega_2}, x'_{\Omega_2})d\Omega' \\ &= - \int_{\Omega_2} \phi_2(x'_{\Omega_2})\delta(x_{\Omega_2}, x'_{\Omega_2})d\Omega' \\ &= -\phi_2(x_{\Omega_2}) \end{aligned} \quad (3.11)$$

Reemplazando lo anterior, la ecuación (3.8) queda como:

$$\phi_2(x_{\Omega_2}) - K_Y[\phi_2] + V_Y \left[\frac{\partial\phi_2}{\partial\mathbf{n}} \right] = 0 \quad (3.12)$$

Tomando las ecuaciones (3.7) y (3.12), es posible reescribir el sistema (3.1) como:

$$\begin{aligned}\phi_1(x_{\Omega_1}) + K_L[\phi_1] - V_L \left[\frac{\partial \phi_1}{\partial \mathbf{n}} \right] &= \sum_k \frac{q_k}{4\pi\epsilon_1|x - x_k|} \\ \phi_2(x_{\Omega_2}) - K_Y[\phi_2] + V_Y \left[\frac{\partial \phi_2}{\partial \mathbf{n}} \right] &= 0\end{aligned}\quad (3.13)$$

Luego, para obtener ecuaciones integrales de borde se toman los límites cuando $x_{\Omega_1} \rightarrow \Gamma$ y $x_{\Omega_2} \rightarrow \Gamma$, como existe una singularidad cuando $x = x'$, la integral se resolverá sobre una superficie Γ' idéntica a la detallada en la sección 2.4. El procedimiento para la primera ecuación del sistema es idéntico al realizado desde la ecuación (2.20) hasta (2.24), por lo tanto, es posible utilizar este resultado. Para la segunda ecuación se tiene:

$$\begin{aligned}\lim_{\epsilon \rightarrow 0} \int_{\Gamma_{esf}} G_Y(x_\Gamma, x'_\Gamma) \frac{\partial \phi_2}{\partial \mathbf{n}}(x'_\Gamma) d\Gamma' &= \lim_{\epsilon \rightarrow 0} \int_{\Gamma_{esf}} \frac{\partial \phi_2}{\partial \mathbf{n}}(x'_\Gamma) \frac{e^{-\kappa|x-x'|}}{4\pi|x-x'|} d\Gamma' \\ &= \lim_{\epsilon \rightarrow 0} \frac{e^{-\kappa\epsilon}}{4\pi\epsilon} \frac{\partial \phi_2}{\partial \mathbf{n}}(x_\Gamma) \int_{\Gamma_{esf}} d\Gamma' \\ &= \lim_{\epsilon \rightarrow 0} \frac{e^{-\kappa\epsilon}}{4\pi\epsilon} \frac{\partial \phi_2}{\partial \mathbf{n}}(x_\Gamma) 2\pi\epsilon^2 \\ &= 0\end{aligned}\quad (3.14)$$

Esto, ya que la magnitud $|x - x'|$ es constante e igual a ϵ . Para el cálculo del límite, en el caso del operador potencial de capa doble, se debe calcular previamente la derivada de la función de Green, esta es:

$$\frac{\partial G_Y}{\partial \mathbf{n}}(x, x') = \frac{-\kappa|x - x'|e^{-\kappa|x-x'|}}{4\pi|x - x'|} + \frac{e^{-\kappa|x-x'|}(x - x') \cdot \mathbf{n}}{4\pi|x - x'|^3}\quad (3.15)$$

Luego, es posible calcular el límite del operador potencial de capa doble como:

$$\begin{aligned}\lim_{\epsilon \rightarrow 0} \int_{\Gamma_{esf}} \phi_2(x'_\Gamma) \frac{\partial G_Y}{\partial \mathbf{n}}(x_\Gamma, x'_\Gamma) d\Gamma' &= \lim_{\epsilon \rightarrow 0} \int_{\Gamma_{esf}} \phi_2(x'_\Gamma) \left(\frac{-\kappa|x - x'|e^{-\kappa|x-x'|}}{4\pi|x - x'|} + \frac{e^{-\kappa|x-x'|}(x - x') \cdot \mathbf{n}}{4\pi|x - x'|^3} \right) d\Gamma' \\ &= \lim_{\epsilon \rightarrow 0} \phi_2(x_\Gamma) \left(\frac{-\kappa e^{-\kappa\epsilon}}{4\pi} + \frac{e^{-\kappa\epsilon}}{4\pi\epsilon^2} \right) \int_{\Gamma_{esf}} d\Gamma' \\ &= \lim_{\epsilon \rightarrow 0} \phi_2(x_\Gamma) \left(\frac{-\kappa e^{-\kappa\epsilon}}{4\pi} + \frac{e^{-\kappa\epsilon}}{4\pi\epsilon^2} \right) 2\pi\epsilon^2 \\ &= \lim_{\epsilon \rightarrow 0} \phi_2(x_\Gamma) \left(\frac{-\kappa\epsilon^2 e^{-\kappa\epsilon}}{2} + \frac{e^{-\kappa\epsilon}}{2} \right) \\ &= \frac{\phi_2(x_\Gamma)}{2}\end{aligned}\quad (3.16)$$

Así, se obtiene un sistema de ecuaciones integrales de borde:

$$\begin{aligned} \frac{\phi_1(x_\Gamma)}{2} + K_L^\Gamma[\phi_1] - V_L^\Gamma \left[\frac{\partial \phi_1}{\partial \mathbf{n}} \right] &= \sum_k \frac{q_k}{4\pi\epsilon_1|x_\Gamma - x_k|} \\ \frac{\phi_2(x_\Gamma)}{2} - K_Y^\Gamma[\phi_2] + V_Y^\Gamma \left[\frac{\partial \phi_2}{\partial \mathbf{n}} \right] &= 0 \end{aligned} \quad (3.17)$$

Es posible dejar ambas ecuaciones en términos de ϕ_1 utilizando las condiciones de borde enunciadas en (3.1):

$$\begin{aligned} \frac{\phi_1(x_\Gamma)}{2} + K_L^\Gamma[\phi_1] - V_L^\Gamma \left[\frac{\partial \phi_1}{\partial \mathbf{n}} \right] &= \sum_k \frac{q_k}{4\pi\epsilon_1|x_\Gamma - x_k|} \\ \frac{\phi_1(x_\Gamma)}{2} - K_Y^\Gamma[\phi_1] + \frac{\epsilon_1}{\epsilon_2} V_Y^\Gamma \left[\frac{\partial \phi_1}{\partial \mathbf{n}} \right] &= 0 \end{aligned} \quad (3.18)$$

O equivalente, escrito de forma matricial:

$$\begin{bmatrix} \frac{1}{2} + K_L^\Gamma & -V_L^\Gamma \\ \frac{1}{2} - K_Y^\Gamma & \frac{\epsilon_1}{\epsilon_2} V_Y^\Gamma \end{bmatrix} \begin{bmatrix} \phi_1 \\ \frac{\partial \phi_1}{\partial \mathbf{n}} \end{bmatrix} = \begin{bmatrix} \sum_k \frac{q_k}{4\pi\epsilon_1|x_\Gamma - x_k|} \\ 0 \end{bmatrix} \quad (3.19)$$

3.2. Modelos de Múltiple Superficie

Como se menciono al inicio de esta sección, el modelo de solvente implícito usa la teoría electrostática continua para representar el solvente alrededor de una proteína, sin embargo la ecuación (3.1) corresponde a un modelo de única superficie, en el cual solo se tienen las dos regiones mencionadas. Es posible representar múltiples superficies con el modelo de solvente implícito, las cuales pueden ser cavidades llenas de solvente o una región en la interfaz entre el solvente y la proteína, conocida como “*Capa de Exclusión Iónica*”. Las cavidades llenas de solvente, al igual que la región Ω_2 en (3.1) se rigen por la ecuación de Poisson-Boltzmann, es decir:

$$\nabla^2 \phi_i = \kappa \phi_i \quad (3.20)$$

La *Capa de Exclusión Iónica* en cambio, se modela como una región llena de agua, pero sin iones de sal, por lo tanto, tendrá la misma constante dieléctrica de la región del solvente, pero con κ igual a 0, por lo tanto, estará gobernada por una ecuación de Laplace, tal que:

$$\nabla^2 \phi_i = 0 \quad (3.21)$$

La *Capa de Exclusión Iónica* se utiliza para compensar la sobre estimación de cargas en la interfaz entre regiones de la SES, esta se produce debido a que existe un máximo de cargas que pueden ubicarse en la interfaz, dado por el tamaño de la superficie y el diámetro de los átomos, al considerar cargas puntuales, se sobrepasa el valor máximo real.

Finalmente, para los modelos de múltiples superficies, las condiciones de borde entre regiones serán:

$$\begin{aligned}\phi_i &= \phi_j \\ \epsilon_i \frac{\partial \phi_i}{\partial \mathbf{n}} &= \epsilon_j \frac{\partial \phi_j}{\partial \mathbf{n}}\end{aligned}\quad (3.22)$$

3.3. Energía de Solvatación

El sistema (3.19) permite obtener el valor numérico de ϕ_1 y $\frac{\partial \phi_1}{\partial \mathbf{n}}$ en la superficie de la biomolécula, sin embargo, interesa calcular la energía de solvatación de esta, denotada por ΔG_{solv} . Se conoce como energía de solvatación a la energía requerida para crear la cavidad en el solvente.

Para obtener la energía de solvatación, primero se debe calcular el potencial de reacción (ϕ_{reac}) en la ubicación de los átomos, el cual es igual al potencial ϕ_1 quitándole la contribución de las cargas, es decir:

$$\phi_{reac} = \phi_1 - \phi_{coulomb} \quad (3.23)$$

Escrito en términos de operadores de borde, la ecuación (3.23) queda como:

$$\phi_{reac} = V_L \left[\frac{\partial \phi_1}{\partial \mathbf{n}} \right] - K_L[\phi_1] \quad (3.24)$$

Finalmente, se obtiene la energía de solvatación multiplicando el potencial de reacción en la ubicación de los átomos por su respectiva carga, sobre todos los átomos presentes en la biomolécula, es decir:

$$\Delta G_{solv} = \sum_k^{Nq} q_k \phi_{reac}(x_k) \quad (3.25)$$

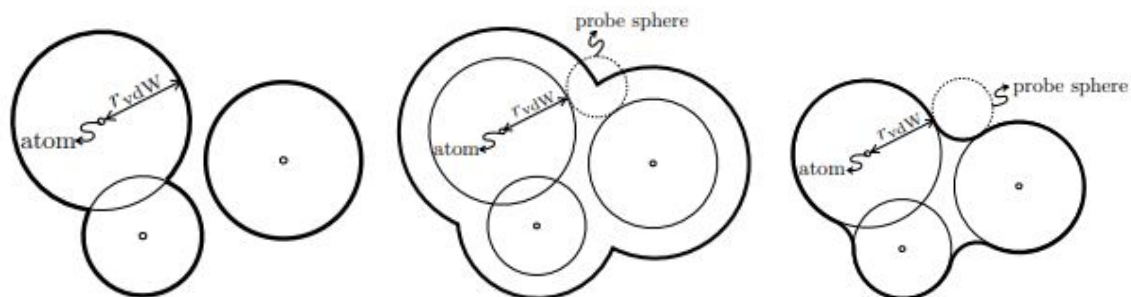
3.4. Campos de Fuerza

Como se observa tanto en la ecuación (3.19) como en (3.25), es necesario conocer de antemano la ubicación y magnitud de las cargas, el cual esta dado principalmente por los **Campos de Fuerza**. A partir de la información de la estructura cristalina de las proteínas (la cual puede encontrarse en el banco de datos de proteínas, PDB por sus siglas en ingles) los distintos **Campos de Fuerza** calculan los radios de Van der Waals y la magnitud de la carga puntual en la ubicación de cada átomo.

3.5. Superficie Excluida de Solvente

Para el modelo de solvente implícito, es necesario generar esta separación entre el solvente y la proteína, existen diferentes tipos de superficies, para este trabajo se utiliza la superficie excluida de solvente. La superficie excluida de solvente recibe este nombre, ya que representa lo más cerca que puede estar una molécula de agua de la proteína. Para generar esta superficie,

es necesario de antemano conocer la estructura cristalina de la proteína en cuestión y el radio de Van der Waals de los átomos, la superficie se genera haciendo rodar una esfera de prueba del tamaño de una molécula de agua ($\sim 1,4\text{\AA}$), trazando todo su recorrido alrededor de la proteína.



(a) Superficie de Van de Waals (b) Superficie accesible al solvente(c) Superficie excluida de solvente

Figura 3.2: Definición gráfica de los distintos tipos de superficie¹⁹

4. Bempp

Bempp es una plataforma de elementos de frontera que permite resolver problemas acústicos, electrostáticos y electromagnéticos. El programa y todas sus funciones se utilizan a través de Python, resulta útil para el caso de la proteína ya que es posible reducir considerablemente la cantidad de líneas de código a emplear en la resolución, debido a que solo se debe plantear las ecuaciones de manera correcta, mientras que el Bempp realiza los cálculos complejos.

A continuación se describirán de forma breve las funciones que resultan útiles para resolver las ecuaciones planteadas en la sección 3.

4.1. Espacios de funciones

Los espacios de funciones³ en Bempp se utilizan para representar el tipo de expresión aproximada que se está utilizando en el dominio discretizado (Malla) para representar los datos, la función puede ser continua o discontinua por partes y tener un grado desde 0 hasta 10. Los argumentos necesarios para la función son: la malla sobre la cual se está trabajando, el tipo y el grado de la función. Esta se llama de la siguiente manera:

```
bempp.api.function_space(grid, kind, degree)
```

4.2. Operadores

En Bempp existen dos tipos de operadores,⁴ de borde y potenciales. Los operadores de borde corresponden a los vistos en la sección 2.2 y se utilizan para mapear un espacio, denominado espacio de dominio, hacia un “espacio de rango”. Su principal función está en la resolución de los sistemas planteados por el método BEM, un ejemplo de cómo llamar las funciones es:

```
bempp.api.operators.boundary.laplace.single_layer(domain, range, dual_to_range)
```

Notar que en este caso, corresponde al operador de capa simple para la ecuación de Laplace (o Poisson), si se desea utilizar en cambio, un operador para la ecuación de Poisson-Boltzmann, basta con reemplazar “laplace” por “modified_helmholtz”, de manera similar, para utilizar el operador de capa doble, se debe cambiar “single” por “double”, por ejemplo, el operador de capa doble para la ecuación de Poisson-Boltzmann sería:

```
bempp.api.operators.boundary.modified_helmholtz.double_layer(domain, range, dual_to_range, kappa)
```

Es importante destacar que en el caso de los operadores de borde para la ecuación de Poisson-Boltzmann, se debe ingresar también, como cuarto parámetro, el inverso de la longitud de Debye, κ .

Luego, se tiene los operadores potenciales,⁵ los cuales calculan los valores del potencial en todo el dominio (no únicamente en la frontera), en el caso particular del problema de la proteína, sirve para calcular el potencial en los puntos de interés, es decir, en la ubicación de los átomos. La función tiene como argumentos obligatorios el espacio de funciones y los puntos sobre los que calcular el potencial, esta llama de la siguiente manera:

```
bempp.api.operators.potential.laplace.single_layer(space, points)
```

Al igual que con los operadores de borde, es posible cambiar el nombre de la función para el caso de Poisson-Boltzmann o en caso de utilizar el operador de capa doble.

4.3. Función de Malla

Frecuentemente, sera necesario definir una función que actué como el lado derecho de la ecuación a resolver, para esto se utiliza GridFunction de bempp, la cual permite representar funciones en una malla, los argumentos de entrada necesarios son el espacio funcional sobre el cual actuará la función, y la función como tal definida previamente, tal que:

```
bempp.api.GridFunction(space, fun= Funcion)
```

5. Campo de Fuerza Polarizable

El principal objetivo de este trabajo consiste en la implementación del campo de fuerza polarizable AMOEBA a los modelos desarrollados y explicados en la sección 3. Un campo de fuerza polarizable busca una representación más realista de los modelos y consiste en una variación en la distribución de carga utilizada en el interior de las biomoléculas. Mientras que los campos de fuerza clásicos utilizan una distribución de carga con cargas puntuales, los campos de fuerza polarizables utilizan componentes multipolares de orden superior, es decir, se tiene una carga puntual q , un dipolo p representado por un vector, un cuadrupolo Q representado por un tensor 3x3 y una polarizabilidad α representada por un tensor 3x3 cuyo significado será explicado más adelante.

5.1. Formulación Matemática

Para aplicar un campo de fuerza polarizable al modelo de solvente implícito, se debe realizar primero la derivación matemática del problema, considerando esta vez la presencia de multipolos. Tomando como punto de inicio el sistema de ecuaciones (3.1), se puede notar que el lado izquierdo de ambas ecuaciones permanece inalterable, y que solo se debe trabajar el lado derecho de la ecuación de Poisson, la cual representa el potencial debido a las cargas en el borde, para que incluya las componentes multipolares. En general, el potencial debido a una distribución de carga $\rho(\mathbf{r})$ es igual a:

$$\Phi = \frac{1}{4\pi\epsilon} \int_{\Omega} \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' \quad (5.1)$$

Cuando la distribución de carga $\rho(\mathbf{r})$ corresponde a un conjunto de N cargas puntuales, se obtiene la expresión que se observa en el lado derecho de la ecuación de Poisson en (3.19). Si se considera un multipolo puntual compuesto de una carga, un dipolo y un cuadrupolo, la ecuación (5.1) queda como:

$$\Phi(\mathbf{r}) = \frac{1}{4\pi\epsilon} \left[\frac{1}{|\mathbf{r}|} \int_{\Omega} \rho(\mathbf{r}') d\mathbf{r}' + \frac{r_i}{|\mathbf{r}|^3} \int_{\Omega} r'_i \rho(\mathbf{r}') d\mathbf{r}' + \frac{r_i r_j}{2|\mathbf{r}|^5} \int_{\Omega} \rho(\mathbf{r}') (3r'_i r'_j - |\mathbf{r}'|^2 \delta_{ij}) d\mathbf{r}' \right] \quad (5.2)$$

Donde la primera integral corresponde al potencial debido a la carga puntual, la segunda integral es el potencial debido al dipolo y por ultimo, la tercera integral, el potencial debido al cuadrupolo. Si la distribución de carga corresponde a un conjunto discreto de cargas, dipolos y cuadrupolos puntuales, el nuevo lado derecho de la ecuación (3.19) queda como:

$$\frac{1}{4\pi\epsilon} \sum_{k=1}^{N_q} \frac{q_k}{|\mathbf{r}_{\Gamma} - \mathbf{r}_k|} + p_i \frac{(r_{\Gamma,i} - r_{k,i})}{|\mathbf{r}_{\Gamma} - \mathbf{r}_k|^3} + Q_{ij} \frac{(r_{\Gamma,i} - r_{k,i})(r_{\Gamma,j} - r_{k,j})}{2|\mathbf{r}_{\Gamma} - \mathbf{r}_k|^5} \quad (5.3)$$

Por lo tanto, la nueva situación queda representada por la ecuación (3.19) pero con la ecuación (5.3) como lado derecho de la ecuación de Poisson, es decir:

$$\begin{bmatrix} \frac{1}{2} + K_L^{\Gamma} & -V_L^{\Gamma} \\ \frac{1}{2} - K_Y^{\Gamma} & \epsilon_1 V_Y^{\Gamma} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \frac{\partial \phi_1}{\partial \mathbf{n}} \end{bmatrix} = \begin{bmatrix} \frac{1}{4\pi\epsilon} \sum_{k=1}^{N_q} \frac{q_k}{|\mathbf{r}_{\Gamma} - \mathbf{r}_k|} + p_i \frac{(r_{\Gamma,i} - r_{k,i})}{|\mathbf{r}_{\Gamma} - \mathbf{r}_k|^3} + Q_{ij} \frac{(r_{\Gamma,i} - r_{k,i})(r_{\Gamma,j} - r_{k,j})}{2|\mathbf{r}_{\Gamma} - \mathbf{r}_k|^5} \\ 0 \end{bmatrix} \quad (5.4)$$

5.2. Efectos de Polarización

Si bien, el problema esta bien representado por la ecuación (5.4), aun no es posible resolverlo, esto se debe a que, como indica el nombre del campo de fuerza, existen efectos de polarización que no han sido considerados. Cada multipolo posee una componente de dipolo inducido dada por:

$$\mu = \alpha \vec{\mathbf{E}} \quad (5.5)$$

Donde μ es el dipolo inducido, α es la polarizabilidad y $\vec{\mathbf{E}}$ es el campo eléctrico en la ubicación del multipolo. Luego el dipolo p total utilizado en la ecuación (5.3) y el sistema (5.4) está dado por:

$$p = \mu + d \quad (5.6)$$

Donde d es la componente constante del dipolo o dipolo permanente, la cual está dada por el campo de fuerza polarizable.

5.3. Cálculo del Campo Eléctrico

Como se puede observar en la ecuación (5.5), es necesario el cálculo del campo eléctrico en la ubicación de los multipolos, sabiendo que:

$$\vec{\mathbf{E}} = -\nabla \phi \quad (5.7)$$

Por simplicidad es conveniente separar el potencial ϕ (y por ende, el campo eléctrico) en dos componentes para el cálculo del dipolo inducido, una debido a la reacción del solvente, denominada ϕ_{solv} y otra debido al resto de los multipolos ϕ_{mult} . Se tiene entonces, en cada ubicación:

$$\phi_k = \phi_{solv,k} + \phi_{mult,k} \quad (5.8)$$

5.3.1. Campo Eléctrico Debido al Solvente

Si bien el cálculo del potencial debido al solvente se realiza utilizando la librería Bempp, esta no ofrece formas ni métodos para el cálculo de su derivada, por lo tanto, se utilizó una aproximación por medio de diferencias finitas. Para cada multipolo puntual, se definió un espacio de puntos de la forma $x_i \pm h$ donde x_i representa cada coordenada del multipolo y h el espaciamiento. Una vez se han definido los puntos, utilizando Bempp se calcula el potencial ϕ_{solv} sobre ellos, finalmente se obtiene el campo eléctrico en la ubicación del multipolo como:

$$E_{solv,i} = -\frac{\partial \phi_{solv}}{\partial x_i} \approx -\frac{\phi_{solv}(x_i + h) - \phi_{solv}(x_i - h)}{2h} \quad (5.9)$$

Considerando que el potencial debido al solvente ϕ_{solv} se obtiene utilizando la expresión (3.24) en los nuevos puntos definidos para realizar las diferencias centradas.

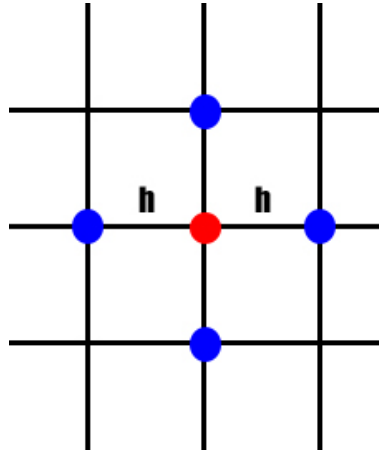


Figura 5.1: Representación gráfica en dos dimensiones del proceso de cálculo de derivadas por diferencias finitas

En la figura 5.1, el punto rojo representa la ubicación de un multipolo, los puntos azules, son ubicaciones generadas a una distancia h del multipolo sobre las cuales se calcula el potencial para luego utilizar la ecuación (5.9)

5.3.2. Campo Eléctrico Debido a los Multipolos

Como el potencial debido a los multipolos es conocido (e igual al potencial de coulomb), es posible obtener el campo eléctrico de forma analítica derivando la expresión para ϕ_{mult} , por lo tanto, el campo eléctrico debido a N_m multipolos en la ubicación del multipolo l es igual a:

$$E_{mult,k} = - \sum_{m=1, m \neq l}^{N_m} \frac{1}{4\pi\epsilon} \frac{\partial}{\partial \mathbf{r}_k^{(l)}} \left[\frac{q^{(m)}}{|\mathbf{r}^{(l)} - \mathbf{r}^{(m)}|} + \frac{r_i^{(l)} - r_i^{(m)}}{|\mathbf{r}^{(l)} - \mathbf{r}^{(m)}|^3} p_i^{(m)} + \frac{(r_i^{(l)} - r_i^{(m)})(r_j^{(l)} - r_j^{(m)})}{2|\mathbf{r}^{(l)} - \mathbf{r}^{(m)}|^5} Q_{ij}^{(m)} \right] \quad (5.10)$$

5.4. Esquema iterativo autoconsistente

Como se puede notar en la derivación tratada a lo largo de esta sección, para realizar los cálculos de potencial es necesario conocer previamente el momento dipolar p , el cual depende del valor del campo eléctrico, sin embargo este a su vez depende del valor que tenga la componente inducida del dipolo. Por lo tanto, la situación planteada se debe resolver con un esquema iterativo autoconsistente, como sigue:

1. Asumir μ (Por ejemplo $\mu = 0$)
2. Obtener ϕ y $\frac{\partial \phi}{\partial \mathbf{n}}$ en Γ con la ecuación (5.4) utilizando Bempp.
3. Obtener ϕ en puntos cercanos a la ubicación de multipolos utilizando operadores potenciales en Bempp.

4. Calcular $\vec{\mathbf{E}}_{solv}$ y $\vec{\mathbf{E}}_{mult}$ utilizando las ecuaciones (5.9) y (5.10) respectivamente.
5. Calcular el campo eléctrico total en la ubicación de los multipolos como: $\vec{\mathbf{E}} = \vec{\mathbf{E}}_{solv} + \vec{\mathbf{E}}_{mult}$.
6. Obtener nuevo valor de μ utilizando la ecuación (5.5).
7. Repetir desde paso 2 hasta la convergencia.

De manera opcional, se puede ponderar el dipolo polarizable μ por un coeficiente ω tal que:

$$\mu = \omega\mu_i + (1 - \omega)\mu_{i-1} \quad (5.11)$$

Donde μ_i es el dipolo polarizado calculado en la iteración actual, μ_{i-1} el dipolo polarizado de la iteración previa y ω el coeficiente de relajación, esto a fin de relajar el esquema iterativo y lograr menores variaciones. Notar que el valor de ω debe estar entre 0 y 1. Como criterio de convergencia se utilizará una comparación directa en el valor de μ para cada multipolo, es decir, se logra la convergencia una vez que cada punto cumpla con:

$$\frac{|\mu_i - \mu_{i-1}|}{|\mu_{i-1}|} < \text{tol} \quad (5.12)$$

Según se requiera mayor o menor precisión, la tolerancia puede variar entre 1×10^{-2} para cálculos rápidos y 1×10^{-8} para cálculos precisos. Una vez que se ha llegado a la convergencia, se puede proceder con el cálculo de energía de solvatación.

5.5. Energía de Solvatación con Multipolos

Como se menciona en la sección 3.3, la energía de solvatación es la energía necesaria para disolver una biomolécula en un solvente, la sección mencionada nos presenta también una expresión (ecuación (3.25)) para la energía de solvatación, si bien esta expresión es válida únicamente al utilizar campos de fuerza clásicos, es posible extenderla al uso de campos de fuerza polarizables. Se puede definir de manera general la energía de solvatación como la diferencia entre la energía libre en estado disuelto y en el vacío, adicionalmente, existe un gasto energético implicado en la polarización de los multipolos, el cual debe ser considerado, es decir:

$$\Delta G_{solv} = G_{diss} - G_{vacc} + W_{pol} \quad (5.13)$$

Donde G_{diss} es la energía libre en estado disuelto, G_{vacc} es la energía libre en el vacío y W_{pol} es la energía utilizada en polarizar los multipolos. La energía libre se puede definir como:

$$G = \frac{1}{2} \int_{\Omega} \rho(\mathbf{r})\phi(\mathbf{r})d\mathbf{r} \quad (5.14)$$

Donde ρ es la distribución de carga y ϕ el potencial. Si se aplica esta definición a la ecuación (5.13) y, además, se separa el potencial para el estado disuelto ϕ_{diss} aplicando la ecuación (5.8), se obtiene:

$$\begin{aligned}\Delta G_{solv} &= \frac{1}{2} \left(\int_{\Omega} \rho_{diss}(\mathbf{r}) \phi_{diss}(\mathbf{r}) d\mathbf{r} - \int_{\Omega} \rho_{vacc}(\mathbf{r}) \phi_{vacc}(\mathbf{r}) d\mathbf{r} \right) + W_{pol} \\ &= \frac{1}{2} \left(\int_{\Omega} \rho_{diss}(\mathbf{r}) \phi_{solvent}(\mathbf{r}) d\mathbf{r} + \int_{\Omega} \rho_{diss}(\mathbf{r}) \phi_{mult}(\mathbf{r}) d\mathbf{r} - \int_{\Omega} \rho_{vacc}(\mathbf{r}) \phi_{vacc}(\mathbf{r}) d\mathbf{r} \right) + W_{pol}\end{aligned}\quad (5.15)$$

Cabe destacar que para el caso de campos de fuerza clásicos, al tener una representación utilizando únicamente cargas puntuales no hay efectos de polarización y por lo tanto se tiene $W_{pol} = 0$, además, debido a la misma razón anterior, la distribución de carga es igual tanto en estado disuelto como en el vacío y, por ultimo, el potencial en el vacío se debe exclusivamente al potencial generado por las cargas, es decir, $\rho_{diss} = \rho_{vacc}$ y $\phi_{mult} = \phi_{vacc}$, por ende, el segundo y tercer término de la ecuación (5.15) se cancelan y se obtiene:

$$\Delta G_{solv} = \frac{1}{2} \int_{\Omega} \rho(\mathbf{r}) \phi_{solvent}(\mathbf{r}) d\mathbf{r} \quad (5.16)$$

Que es equivalente a la expresión (3.25).

En este caso, dado que la componente polarizable del dipolo es distinta en estado disuelto y en el vacío, se tiene: $\rho_{diss} \neq \rho_{vacc}$, por lo tanto, la segunda y tercera integral no se cancelan y deben ser consideradas en el cálculo. Sabiendo que la distribución de carga ρ es un conjunto discreto de multipolos puntuales, la integral pasa a ser una sumatoria, quedando la expresión para la energía libre como:

$$G = \frac{1}{2} \sum_{m=1}^{N_m} q^{(m)} \phi(\mathbf{r}^{(m)}) + p_i^{(m)} \frac{\partial \phi}{\partial r_i}(\mathbf{r}^{(m)}) + \frac{1}{6} Q_{ij}^{(m)} \frac{\partial}{\partial r_i} \frac{\partial}{\partial r_j} \phi(\mathbf{r}^{(m)}) \quad (5.17)$$

Donde m denota el numero del multipolo y N_m el total de multipolos. Notar que los términos variables en (5.17) corresponden al potencial y a la componente inducida del dipolo (y por ende, p_i), al utilizar $\phi = \phi_{solvent}$ y $p_i = p_{i,diss}$ se obtiene la energía libre en estado disuelto aportada por el solvente, mientras que $\phi = \phi_{mult}$ entrega el aporte de los multipolos. Finalmente, si $\phi = \phi_{vacc}$ y $p_i = p_{i,vacc}$ se obtiene la energía libre en el vacío.

Se debe destacar la presencia de derivadas del potencial de primer y segundo orden en la ecuación (5.17), el valor de la primera derivada del potencial corresponde de hecho, al valor numérico del campo eléctrico con signo negativo, por lo tanto no es necesario realizar este calculo nuevamente para la energía libre en estado disuelto. Para la segunda derivada, se siguió un proceso similar al utilizado en las secciones 5.3.1 y 5.3.2. En el caso de la energía libre aportada por el solvente, se aproximó la segunda derivada por medio de diferencias finitas, para esto, de manera similar al cálculo del campo eléctrico, se definió una cantidad apropiada de puntos con un espaciamiento de h sobre los cuales se calculaba el potencial debido al solvente por medio de la expresión (3.24), para luego obtener la segunda derivada

como:

$$\frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} \phi \approx \frac{\phi_{i+1,j+1} + \phi_{i-1,j-1} - \phi_{i+1,j-1} - \phi_{i-1,j+1}}{4h^2} \quad (5.18)$$

La expresión (5.18) es valida únicamente para $i \neq j$, notar además que las derivadas cruzadas son iguales, por lo tanto se debe calcular únicamente la mitad de la matriz de derivadas. Para la diagonal, es decir cuando $i = j$, se utiliza:

$$\frac{\partial^2 \phi}{\partial x_i^2} \approx \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2} \quad (5.19)$$

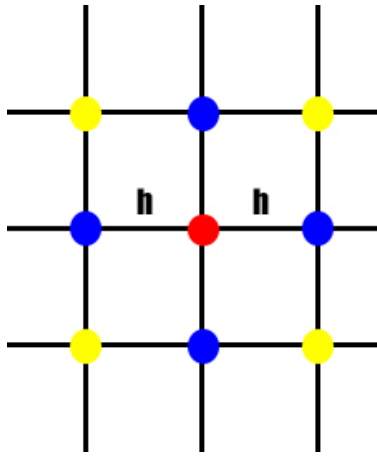


Figura 5.2: Representación gráfica en dos dimensiones del proceso de cálculo de segundas derivadas por diferencias finitas

En la figura 5.2 se observa 4 puntos adicionales con respecto a la figura 5.1, estos puntos señalados en color amarillo, están ubicados a una distancia de h en ambos ejes y también deben ser generados para calcular sobre ellos el potencial de reacción para, de esta manera, utilizar la ecuación 5.18.

En el caso de la segunda derivada para la energía libre relacionada a los multipolos, se utilizó una función analítica, derivando la expresión original para el potencial debido a multipolos puntuales.

Ya se han definido completamente los primeros dos términos de la ecuación (5.13), el ultimo término, correspondiente a la energía de polarización de los multipolos, se define como:

$$\begin{aligned} W_{pol} &= \frac{1}{2} \sum_{m=1}^{N_m} \mu^{(m)} \mathbf{E}_{pol}(\mathbf{r}^{(m)}) - \mu_{vacc}^{(m)} \mathbf{E}_{pol,vacc}(\mathbf{r}^{(m)}) \\ &= -\frac{1}{2} \sum_{m=1}^{N_m} \mu^{(m)} \frac{\partial}{\partial r_i} \phi(\mathbf{r}^{(m)}) - \mu_{vacc}^{(m)} \frac{\partial}{\partial r_i} \phi_{vacc}(\mathbf{r}^{(m)}) \end{aligned} \quad (5.20)$$

Donde μ es la componente inducida del dipolo y \mathbf{E} es el campo eléctrico total que polariza los dipolos. El subíndice *vacc* denota que las cantidades están calculadas en el vacío, estas deben considerarse ya que los dipolos inducidos no siempre comienzan a polarizarse desde $\mu = 0$ sino que, en ocasiones pueden encontrarse multipolos polarizados en el vacío.

Si se compara la ecuación (5.20) con la ecuación (5.17) se puede notar que la energía de polarización se cancela con el dipolo inducido (ecuación (5.6)), por lo tanto, se obtiene:

$$G = \frac{1}{2} \sum_{m=1}^{N_m} q^{(m)} \phi(\mathbf{r}^{(m)}) + d_i^{(m)} \frac{\partial \phi}{\partial r_i}(\mathbf{r}^{(m)}) + \frac{1}{6} Q_{ij}^{(m)} \frac{\partial}{\partial r_i} \frac{\partial}{\partial r_j} \phi(\mathbf{r}^{(m)}) \quad (5.21)$$

Si se utiliza la ecuación (5.21) para el cálculo de energía libre, la energía de polarización ya se encuentra considerada implícitamente y no es necesario calcularla.

6. Ejemplos de implementación: Resultados

En la presente sección se presentarán distintos ejemplos de aplicación donde se utilizan los modelos presentados en las secciones 3 y 5, solucionados utilizando la librería Bempp.

6.1. Esfera con multipolos puntuales

El primer caso donde se implemento los modelos matemáticos presentados corresponde a una esfera con multipolos puntuales en su interior, se realizo un código¹⁰ utilizando la librería Bempp y para su validación se comparó los resultados obtenidos con la solución analítica¹¹ al problema.

El valor de los multipolos, su ubicación y polarizabilidad está retratado en la figura 6.1

$$\begin{array}{lll}
 x_1 = (0, 0, 0)\text{Å} & x_2 = (1, \sqrt{2}, 1)\text{Å} & x_3 = (-1, -1, \sqrt{2})\text{Å} \\
 q_1 = 1e^- & q_2 = -1e^- & q_3 = -1e^- \\
 p_1 = (0, 1, 0)e^- \text{Å} & p_2 = (1, 0, 0)e^- \text{Å} & p_3 = (0, 0, -1)e^- \text{Å} \\
 Q_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} e^- \text{Å}^2 & Q_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} e^- \text{Å}^2 & Q_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} e^- \text{Å}^2 \\
 \alpha_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{Å}^3 & \alpha_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{Å}^3 & \alpha_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{Å}^3
 \end{array}$$

Figura 6.1: Representación gráfica de los multipolos en el interior de la esfera² utilizada para la validación del código realizado

Las simulaciones se realizaron considerando una esfera de radio 3 Å y se utilizaron distintos números de elementos para la malla generada durante las simulaciones. A continuación, en la figura 6.2 se observa un ejemplo de malla utilizada durante las simulaciones:

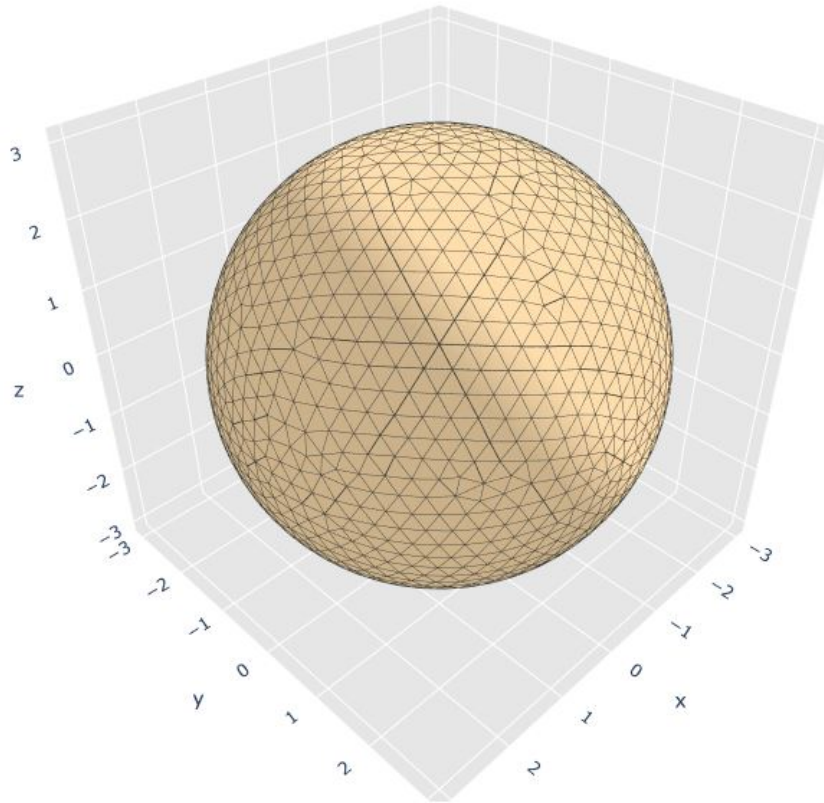


Figura 6.2: Malla utilizada en caso de esfera con multipolos.

Primero, se realizaron simulaciones para distintos valores de altura promedio de los elementos (a menor altura promedio, los elementos son más pequeños y por ende, la malla resultante es más fina), se calculo el error para cada malla como:

$$e(h) = \frac{|G_{solv,bempp}(h) - G_{solv,Analytical}|}{|G_{solv,Analytical}|} \quad (6.1)$$

Donde $G_{solv,Analytical}$ es el valor de la solución analítica al problema, esta no varía ya que no se necesita de una malla para obtener el resultado. Para las simulaciones se utilizo una tolerancia de 1×10^{-6} para la convergencia del dipolo inducido (ecuación (5.12)), para la resolución de cada sistema lineal se utiliza el método generalizado de mínimos residuales (GMRES por sus siglas en ingles) con una tolerancia de 1×10^{-8} y se utilizó un coeficiente de relajación $\omega = 0,7$ para la ecuación (5.11). A continuación en la figura 6.3 se observa un gráfico del error obtenido vs el numero de elementos que presentaba cada malla utilizada:

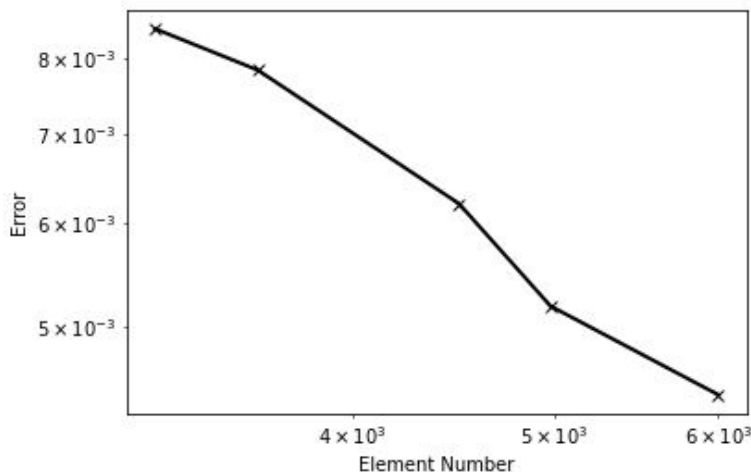


Figura 6.3: Gráfico de error vs N° de elementos de la malla para la esfera con multipolos

Como se puede observar en el gráfico de la figura 6.3, para una malla de 3200 elementos, se obtuvo un error cercano a $8,42 \times 10^{-3}$, mientras que en el caso de la malla de 6000 elementos (cerca del doble de elementos) se obtuvo un error aproximado de $4,44 \times 10^{-3}$, por lo tanto, el código realizado presenta un orden de convergencia aproximadamente lineal, $O(A^1)$.

6.2. Geometrías Reales

Una vez validado el modelo matemático con el caso de la esfera con multipolos puntuales en su interior, se procedió a extender el trabajo realizado para trabajar con geometrías reales, como resultado, se desarrolló un código¹² capaz de obtener los parámetros de cada biomolécula a partir del campo de fuerza polarizable AMOEBA, leer mallas generadas por el software Nanoshaper¹³ y obtener de este manera la energía de solvatación de cada biomolécula.

La principal diferencia entre los casos de geometrías reales y el caso de la esfera que consideraba únicamente 3 multipolos en su interior reside en que AMOEBA utiliza un esquema de polarización basado en grupos donde multipolos del mismo grupo no se polarizan entre ellos, esto resulta sumamente importante al momento del desarrollo y debe ser considerado. A continuación se muestran los ejemplos de biomoléculas modeladas con el código realizado¹² para su validación mediante comparación con el software Pygbe.¹⁴ Para cada caso simulado, se utilizó una constante dieléctrica $\epsilon_{prot} = 1$ para la región de la biomolécula, $\epsilon_{solv} = 78,3$ para la región del solvente y $\kappa = 0,125 \text{ \AA}^{-1}$

6.2.1. Proteína 1pgb

El primer caso de geometría real modelado para validación del código, fue la 1pgb,¹⁵ una proteína de unión a inmunoglobulina con aproximadamente 920 multipolos en su interior.

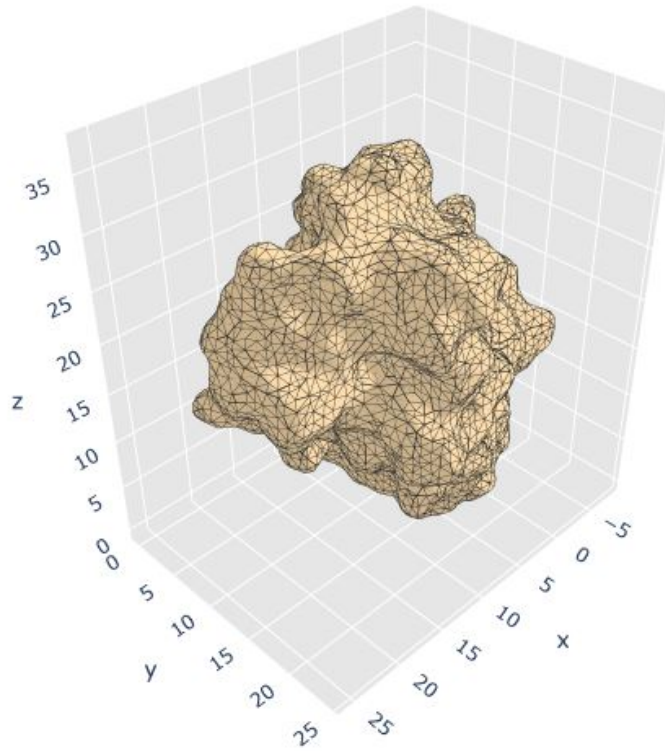


Figura 6.4: Ejemplo de malla utilizada en la modelación de la proteína 1pgb

Para las simulaciones se utilizaron distintas mallas variando su valor de longitud característica promedio, se utilizó una tolerancia de 1×10^{-3} para la convergencia del dipolo inducido y una tolerancia de 1×10^{-5} para el método general de mínimos residuales, a continuación se muestra una comparación entre las simulaciones realizadas con Bempp y las simulaciones utilizando Pygbe mediante un gráfico de la energía de solvatación obtenida vs el inverso de la longitud característica, denominado “Grid Scale”, se muestra también un gráfico logarítmico de error vs el área característica de la malla, la cual se define como:

$$A_c = \frac{L_c^2}{2} \tag{6.2}$$

Donde L_c es el inverso de la longitud característica.

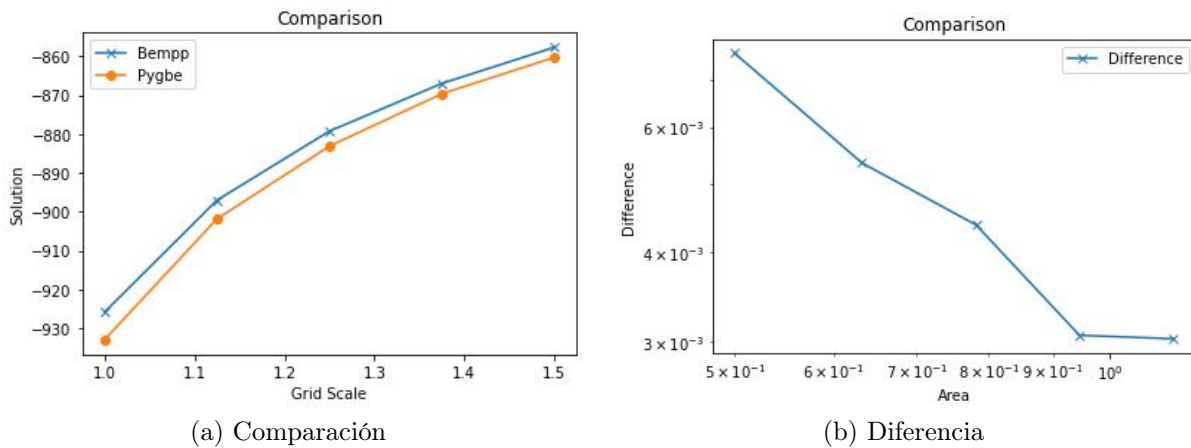


Figura 6.5: Comparación entre el código desarrollado con Bempp y Pygbe para la proteína 1pgb

Adicionalmente, al ser el primer caso de geometría real en estudio, se realizaron diversas pruebas bajo los mismos parámetros que los utilizados para las simulaciones mostradas en 6.5, donde se consideraban de manera parcial las componentes del campo de fuerza polarizable, es decir, se analizaron casos en donde no se consideraban los dipolos, los cuadrupolos, la polarizabilidad o combinaciones de estas. El fin de estas simulaciones fue aislar el aporte de las derivadas numéricas por diferencias finitas al error total. Primero se comparo una modelación donde solo fueron considerados los dipolos y la polarizabilidad, mientras que la contribución de los cuadrupolos se considero como nula, los resultados obtenidos se observan en la figura 6.6

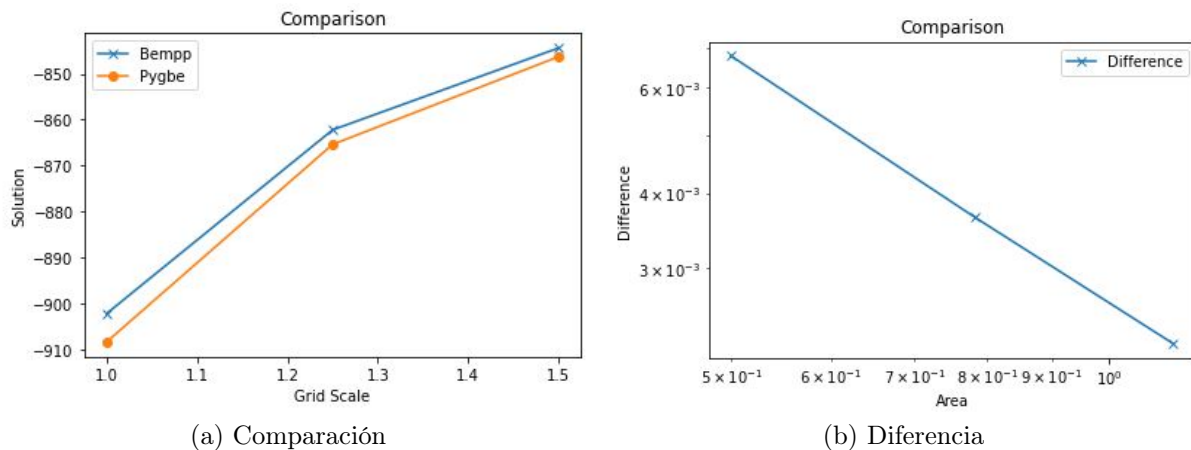


Figura 6.6: Comparación entre el código desarrollado con Bempp y Pygbe para la proteína 1pgb sin considerar cuadrupolos

Con esta simulación se buscaba aislar la contribución de la segunda derivada numérica, presente en la ecuación (5.21), comparando las figuras 6.5b y 6.6b es posible notar que el error, para la malla más gruesa cae desde aproximadamente 0.77% hasta 0.69% mientras

que en el caso de la malla más fina, el error cayó desde aproximadamente 0.3 % hasta 0.22 %. La siguiente simulación se realizó sin considerar la polarizabilidad (y por ende, sin dipolo inducido), los resultados obtenidos se observan a continuación.

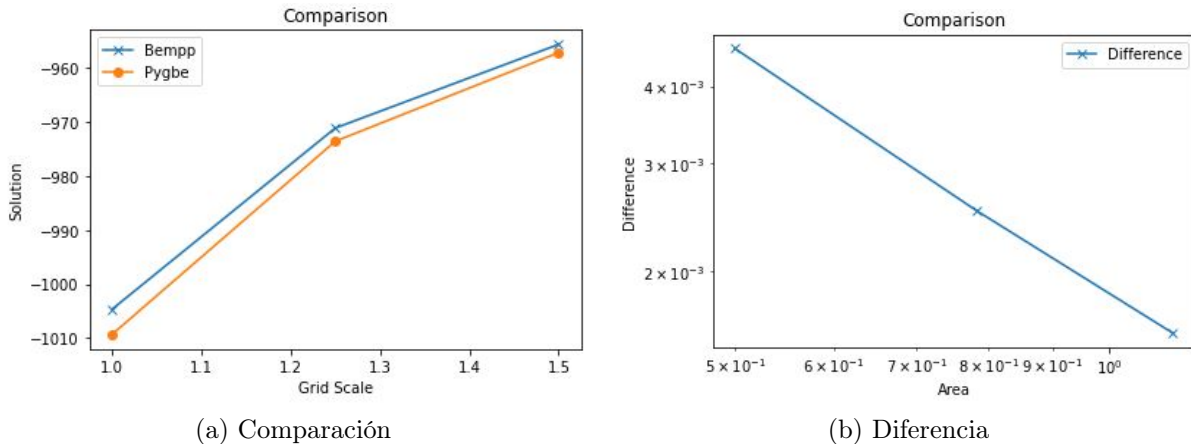


Figura 6.7: Comparación entre el código desarrollado con Bempp y Pygbe para la proteína 1pqb sin considerar polarizabilidad

El objetivo de este caso era cuantificar las diferencias debido al proceso de polarización del dipolo inducido además del error que podría aportar la primera derivada numérica utilizada en la ecuación 5.5, comparando los gráficos 6.5b y 6.7b se puede observar como para la malla más fina, el error cae hasta casi 0.15 %, aproximadamente la mitad del error total de la primera simulación. A modo de confirmar la contribución de la primera derivada al error total, la siguiente simulación se efectuó únicamente considerando los cuadrupolos, es decir, se tomó la contribución de los dipolos como nula y no se consideró ningún efecto de polarización, los resultados obtenidos se muestran a continuación.

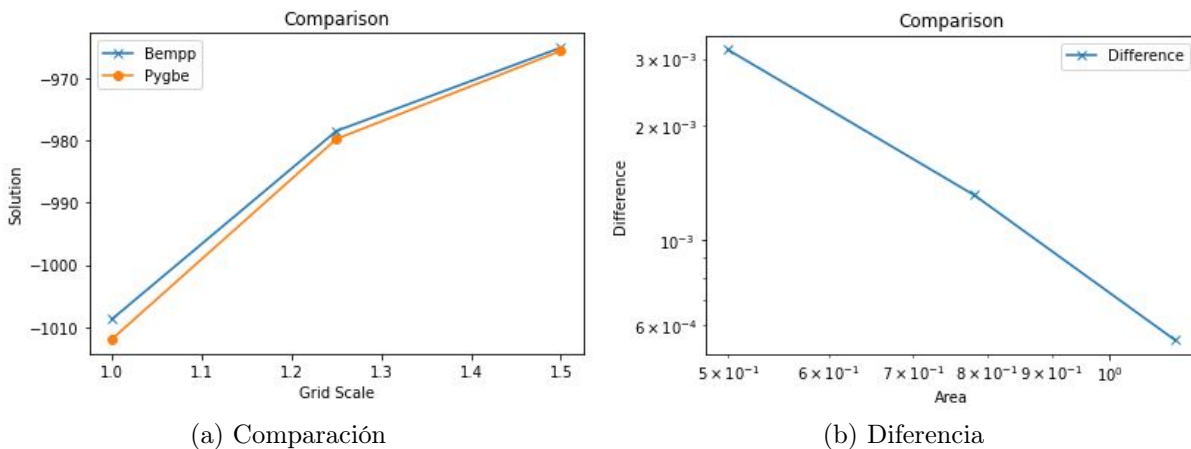


Figura 6.8: Comparación entre el código desarrollado con Bempp y Pygbe para la proteína 1pqb sin considerar dipolos ni polarizabilidad

Mirando el gráfico 6.8b resulta evidente que la principal fuente de error de las simulaciones

realizadas es la primera derivada numérica del potencial de reacción, la cual tiene doble influencia en los cálculos, ya que, como se mencionó con anterioridad, participa en el cálculo del dipolo inducido por medio del cálculo del campo eléctrico en la ecuación (5.9), además, se encuentra presente también en el cálculo de la energía de solvatación para el aporte del solvente, como se observa en la ecuación (5.21). Por último, se realiza una comparación para la energía de solvatación obtenida utilizando el campo de fuerza polarizable AMOEBA y el campo de fuerza clásico PARSE, es decir, considerando únicamente cargas puntuales.

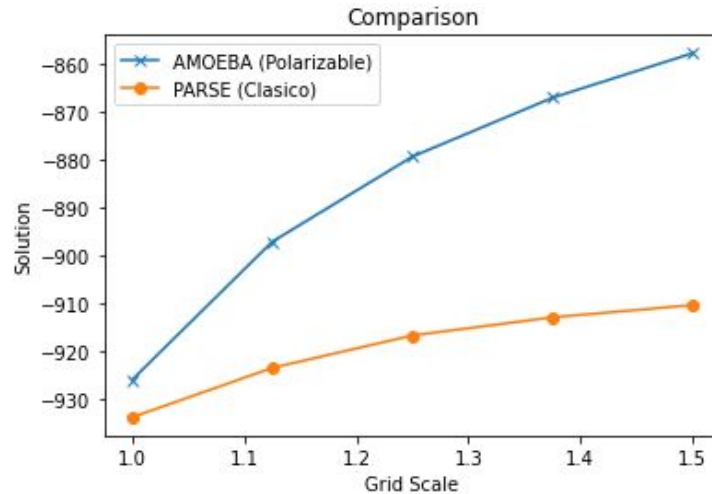


Figura 6.9: Comparación entre un campo de fuerza polarizable y un campo de fuerza clásico para la proteína 1pgb, considerando $\epsilon_{prot} = 1$.

Para ambas simulaciones se utilizaron los mismos valores de permitividad y κ , mencionados al inicio de la sección 6.2, considerando que en campos de fuerza clásicos la permitividad utilizada en la región de la proteína es generalmente 2 o 4, se muestra a continuación la comparación entre ambos campos de fuerza pero utilizando $\epsilon_{prot} = 2$ para las simulaciones con el campo de fuerza PARSE.

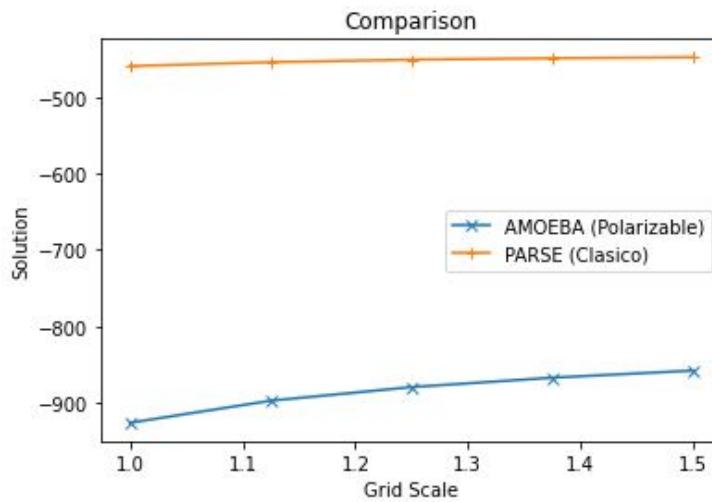


Figura 6.10: Comparación entre un campo de fuerza polarizable y un campo de fuerza clásico para la proteína 1pbg, considerando $\epsilon_{prot} = 2$ para el campo de fuerza clásico.

Analizando las figuras 6.9 y 6.10 se puede observar que los efectos de polarizabilidad no reemplazan a la permitividad efectiva utilizada en campos de fuerza clásicos, debido principalmente a la diferencia de los resultados obtenidos y mostrados en la figura 6.10. De hecho, se estima que para lograr resultados similares a los obtenidos con AMOEBA, utilizando PARSE, se debería utilizar una permitividad cercana a $\epsilon_{prot} = 1,1$

6.2.2. Lisozima 1lyz

Otra biomolécula de estudio fue la 1lyz,¹⁶ una lisozima encontrada en la clara de huevo de gallina, la cual cuenta con un total de 1961 multipolos en su interior.

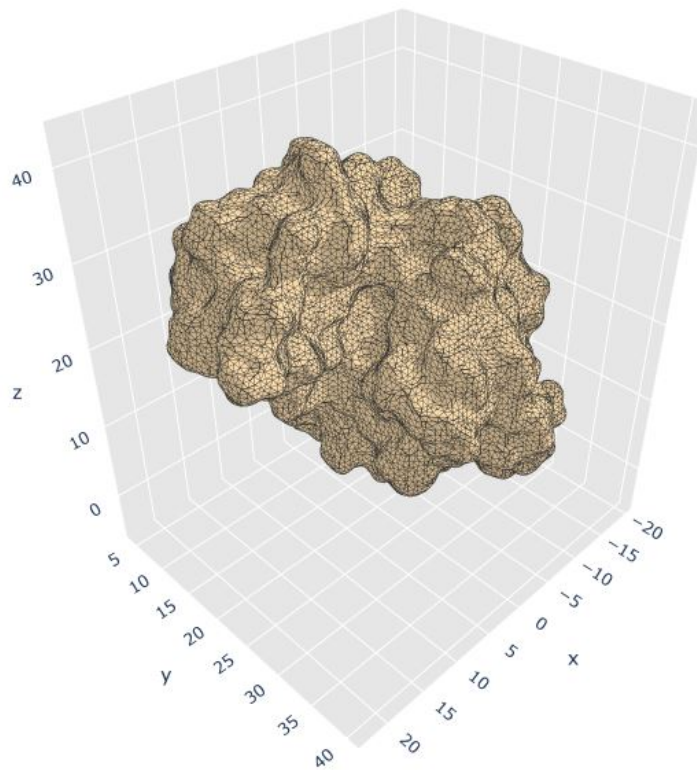


Figura 6.11: Una de las mallas utilizadas para la simulación de la lisozima 1lyz

Se realizó tres simulaciones para distintos valores de la longitud característica de la malla, se utilizó una tolerancia de 1×10^{-3} para el cálculo del dipolo inducido y una tolerancia de 1×10^{-5} para la solución del sistema lineal en cada iteración. Se muestra una comparación entre los resultados obtenidos utilizando Bempp y Pygbe, se muestra también un gráfico de diferencia.

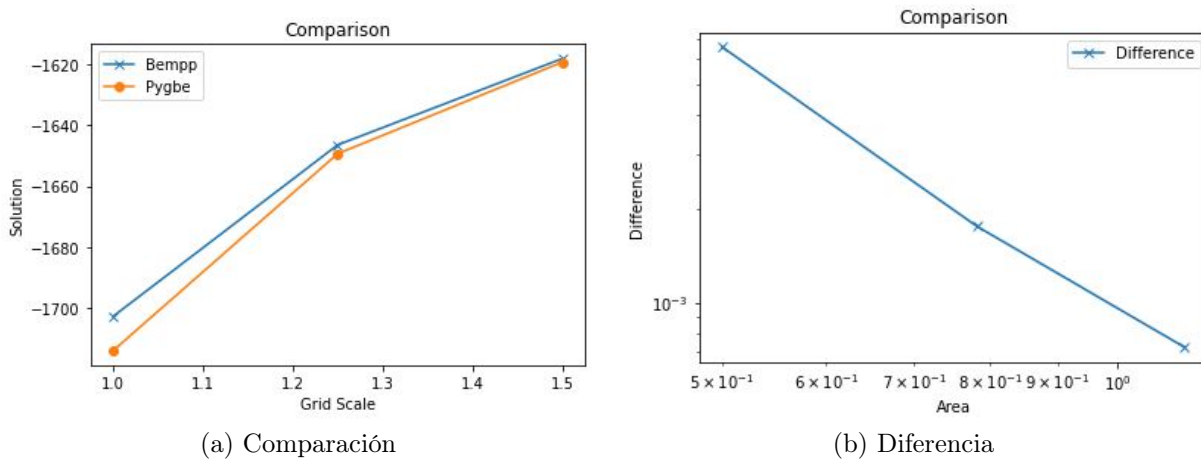


Figura 6.12: Comparación entre el código desarrollado con Bempp y Pygbe para la lisozima 1lyz

6.2.3. Citocina 1a7m

El siguiente caso de estudio fue la citocina 1a7m,¹⁷ un factor inhibidor de la leucemia, la cual cuenta con 2809.

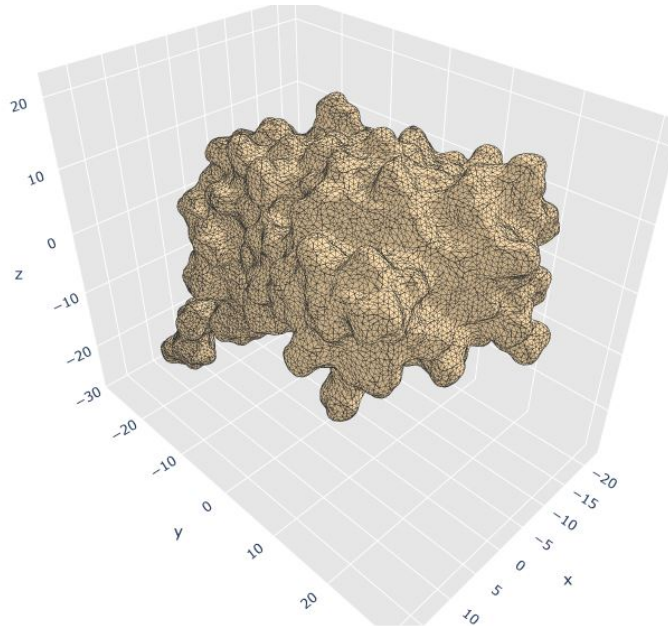


Figura 6.13: Una de las mallas utilizadas para la simulación de la Citocina 1a7m

Nuevamente se realizaron tres simulaciones, para distintos valores del “Grid Scale”, se utilizó una tolerancia de 1×10^{-3} para la convergencia del dipolo inducido y una tolerancia de 1×10^{-5} para la solución del sistema lineal en cada iteración, la comparación de los resultados obtenidos así como un gráfico del error, se muestran a continuación.

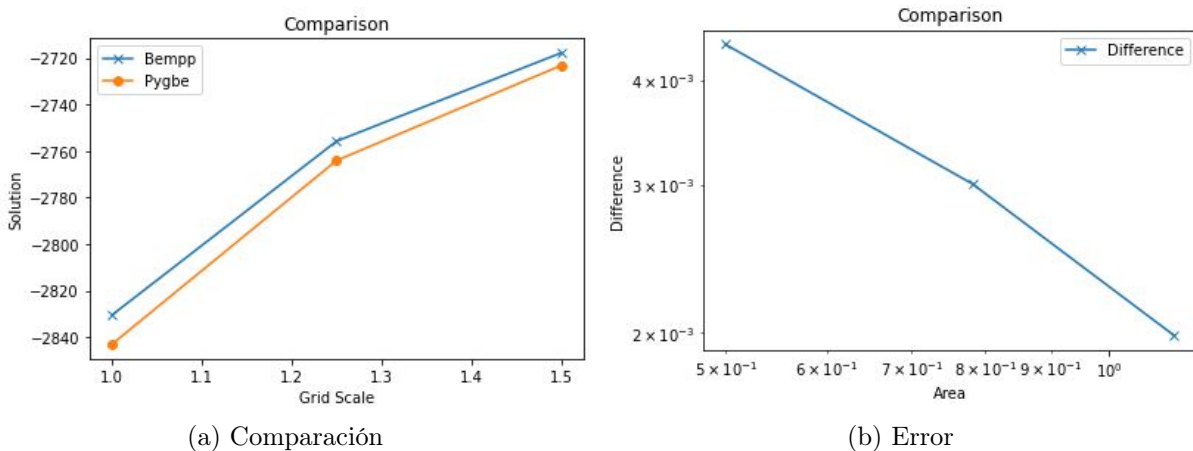


Figura 6.14: Comparación entre el código desarrollado con Bempp y Pygbe para la citocina 1a7m

6.2.4. Proteína 1x1u

El ultimo caso de estudio fue la proteína de unión 1x1u,¹⁸ corresponde al caso de mayor tamaño simulado ya que cuenta con 9476 multipolos.

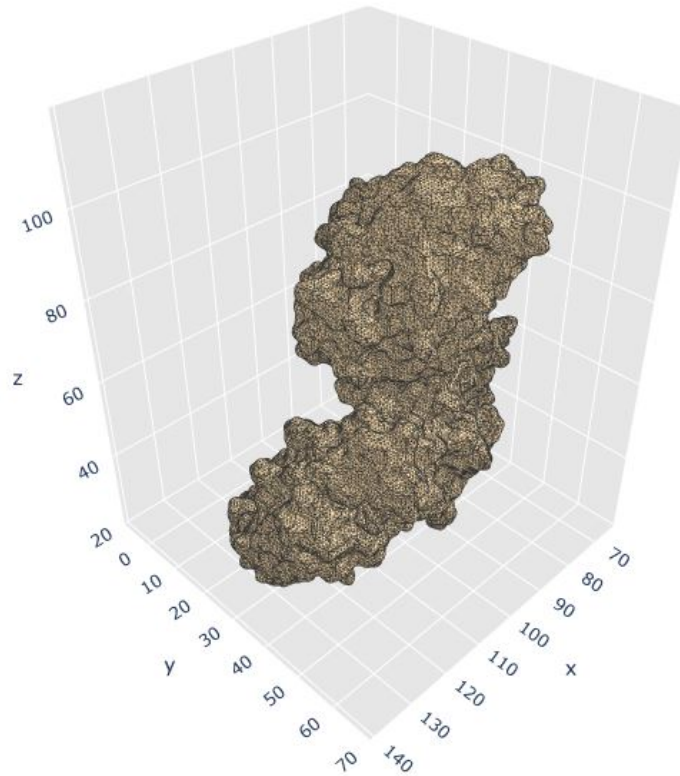


Figura 6.15: Una de las mallas utilizadas para la simulación de la Proteína 1x1u

Debido a lo anterior, se considero una tolerancia menor al momento de comprobar la convergencia del dipolo inducido, utilizando 1×10^{-2} en este caso, la tolerancia para la solución del sistema lineal en cada iteración se mantuvo en 1×10^{-5} y se utilizaron los mismos valores de “Grid Scale” que los casos anteriores, los resultados obtenidos se muestran a continuación.

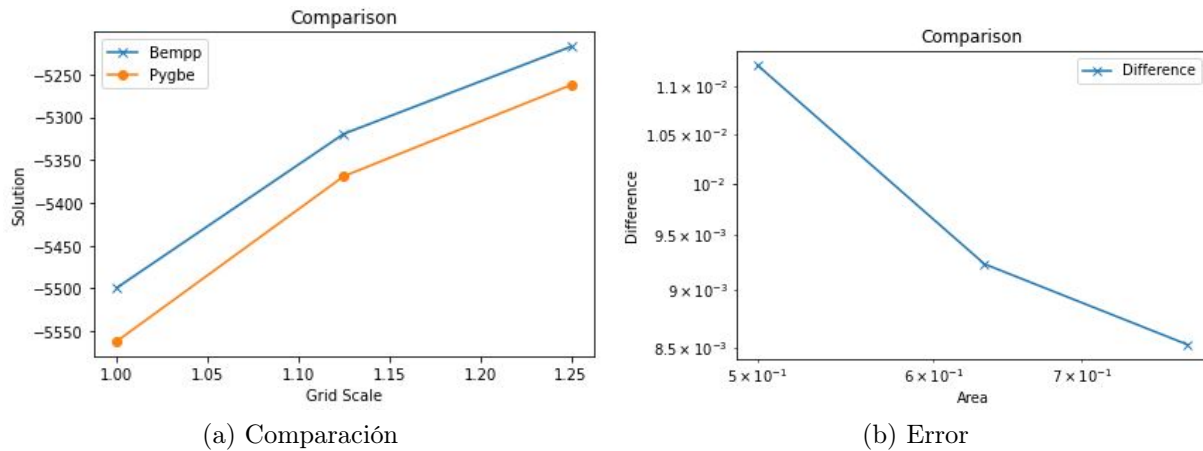


Figura 6.16: Comparación entre el código desarrollado con Bempp y Pygbe para la Proteína 1x1u

Cabe destacar que, debido principalmente a la menor tolerancia que se utilizó para las simulaciones, se obtuvo una diferencia de orden mayor comparada a la de simulaciones anteriores. De hecho, en la figura 6.16 es posible apreciar como existe consistentemente un 1 % de diferencia que tiende a ser constante.

Tabla N° 6.1: Resumen de comparación entre las distintas biomoléculas simuladas

Biomolécula	N_m	$G_{solv} [\frac{kcal}{kmol}]$		Diferencia		Tolerancia	
		Bempp	Pygbe	Mínimo	Máximo	Dipolo	GMRES
1pgb	927	-885.38	-889.59	$3,02 \times 10^{-3}$	$7,62 \times 10^{-3}$	1×10^{-3}	1×10^{-5}
1lyz	1961	-1655.81	-1660.92	$7,21 \times 10^{-4}$	$6,56 \times 10^{-3}$	1×10^{-3}	1×10^{-5}
1a7m	2809	-2768.06	-2776.82	$1,98 \times 10^{-3}$	$4,41 \times 10^{-3}$	1×10^{-3}	1×10^{-5}
1x1u	9476	-5345.42	-5397.74	$8,52 \times 10^{-3}$	$1,12 \times 10^{-2}$	1×10^{-2}	1×10^{-5}

En la tabla 6.1 se pueden observar una comparación entre las distintas biomoléculas simuladas, N_m es el número de multipolos en el interior de la biomolécula y el valor G_{solv} incluido para cada caso corresponde al promedio entre todas las simulaciones realizadas.

7. Conclusiones

A lo largo de este trabajo se realizó la derivación para implementar componentes multipolares de orden superior además de una componente de dipolo polarizable en un modelo de solvente implícito de Poisson-Boltzmann.

Los resultados observados en la tabla 6.1 indican que los modelos matemáticos desarrollados en la sección 5 son consistentes y demuestra que el modelo de solvente implícito es una alternativa eficiente en términos de recursos computacionales y tiempos de ejecución incluso cuando se trabaja con campos de fuerza polarizables.

El código desarrollado constituye una alternativa confiable de simulación para obtener valores de energía de solvatación para biomoléculas de tamaño medio y pequeños utilizando el campo de fuerza polarizable AMOEBA, en particular, se comprobó la eficacia de la librería Bempp para abordar problemas de electrostática utilizando el método de elementos de frontera. Adicionalmente se comprobó que el potencial de reacción en el interior de las biomoléculas es lo suficientemente suave para poder obtener derivadas de este utilizando diferencias finitas con errores del orden del 0.12 % para la primera derivada y 0.8 % para la segunda derivada. Para las biomoléculas con menor número de multipolos tales como la 1lyz y la 1pqb, es decir, biomoléculas con $N_m < 2000$, los tiempos de ejecución del código realizado eran comparables a los obtenidos con Pygbe, por ejemplo, en el caso de la proteína 1pqb, para la malla más gruesa, Pygbe obtuvo la energía de solvatación en aproximadamente 90[s], mientras que Bempp demoró aproximadamente 170[s]. Para las biomoléculas de mayor tamaño, tales como la 1x1u, el tiempo que tomaba Bempp en llegar a la solución era considerablemente mayor en comparación a Pygbe. Específicamente para esta biomolécula, Pygbe necesitaba de aproximadamente 572 [s] para obtener el valor de energía de solvatación, mientras que Bempp demoró cerca de 2[h] para obtener el mismo resultado. A raíz de lo anterior, sumado al hecho de que en cada iteración de dipolo inducido, se necesitaban más iteraciones de GMRES en el caso de Bempp antes de llegar a la tolerancia impuesta, se concluye que una posibilidad de mejora para el código realizado podría ser la implementación de preconditionadores en el sistema lineal presentado en la ecuación (5.4) a fin de mejorar la eficiencia en su resolución, además de considerar formas de aceleración para el código que reduzcan la brecha considerable de tiempo que existe actualmente con Pygbe en simulación de biomoléculas de mayor tamaño.

Finalmente, se organizó el código en forma de librería, lo cual permite un uso fácil y rápido para realizar simulaciones desde una interfaz de Jupyter Notebooks necesitando únicamente la instalación de dependencias obligatorias tales como Bempp, y los archivos necesarios con información acerca de la estructura de la biomolécula. La separación del código en formato de librería facilita el entendimiento de este y la eventual manipulación que se quiera realizar, además, facilita su conexión con otros códigos pensados para simulaciones biomoleculares.

Referencias

- [1] W. Śmigaj, S. Arridge, T. Betcke, J. Phillips & M. Schweiger. *Solving boundary integral problems with BEM++*, *ACM Transactions on Mathematical Software*, 2015
- [2] C. D. Cooper *A Boundary-Integral Approach for the Poisson-Boltzmann Equation with Polarizable Force Fields*, 2019
- [3] Espacios Funcionales en Bempp
http://bempp.com/handbook/api/function_spaces.html
- [4] Operadores de borde en Bempp
http://bempp.com/handbook/api/boundary_operators.html
- [5] Operadores potenciales en Bempp
http://bempp.com/handbook/api/potential_operators.html
- [6] Solucionadores lineales en Bempp
<http://bempp.com/handbook/api/solvers.html>
- [7] Ejemplo Laplace en esfera unitaria
https://nbviewer.jupyter.org/github/bempp/bempp-cl/blob/master/notebooks/laplace/laplace_interior_dirichlet.ipynb
- [8] Banco de datos de proteínas
<https://www.rcsb.org/>
- [9] PDB2PQR Software
<http://server.poissonboltzmann.org/pdb2pqr>
- [10] Código realizado para el caso de la esfera
https://github.com/fvicencioh/Sphere_multipolar/blob/main/sphere_multipolar.ipynb
- [11] Solución analítica al caso de la esfera con multipolos
https://github.com/cdcooper84/pygbe/blob/master/pygbe/util/an_solution.py
- [12] Código final realizado para geometrias reales
https://github.com/fvicencioh/Multipoles_calculations
- [13] S. Decherchi & W. Rocchia *A general and Robust Ray-Casting-Based Algorithm for Triangulating Surfaces at the Nanoscale*
<http://www.plosone.org/article/metrics/info%3Adoi%2F10.1371%2Fjournal.pone.0059744>
- [14] Christopher D. Cooper, Natalia C. Clementi, Gilbert Forsyth & Lorena A. Barba. *Pygbe: Python, GPUs and Boundary elements for biomolecular electrostatics*, 2016

- [15] Proteína 1pgb:
<https://www.rcsb.org/structure/1PGB>
- [16] Lisozima 1lyz:
<https://www.rcsb.org/structure/1LYZ>
- [17] Citocina 1a7m: <https://www.rcsb.org/structure/1A7M>
- [18] Proteína 1x1u: <https://www.rcsb.org/structure/1X1U>
- [19] Christopher Cooper Villagrán. *Biomolecular electrostatics with continuum models: a boundary integral implementation and applications to biosensors. PhD thesis* Boston University, Boston, MA, United States, 2015
- [20] Michael J. Schnieders, Nathan A. Baker, Pengyu Ren & Jay W. Ponder *Polarizable atomic multipole solutes in a Poisson-Boltzmann continuum*, 2007