



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

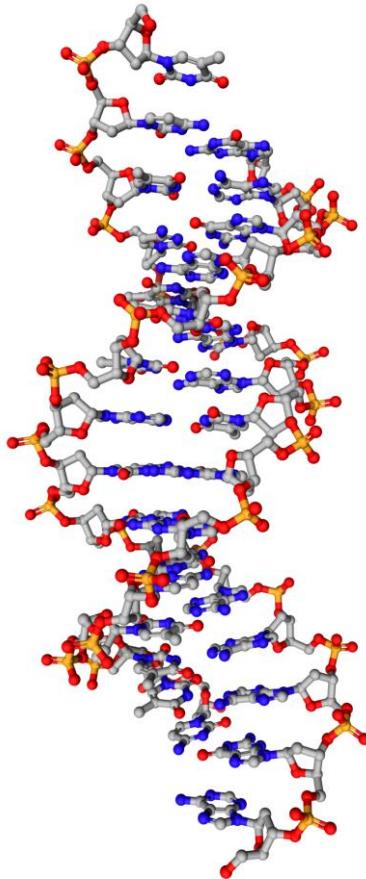
Agencia Nacional de
Investigación y Desarrollo

Aprendizaje Profundo para la Electrostática de Macromoléculas: Resolviendo la Ecuación de Poisson-Boltzmann a partir de Redes Neuronales Informadas por la Física

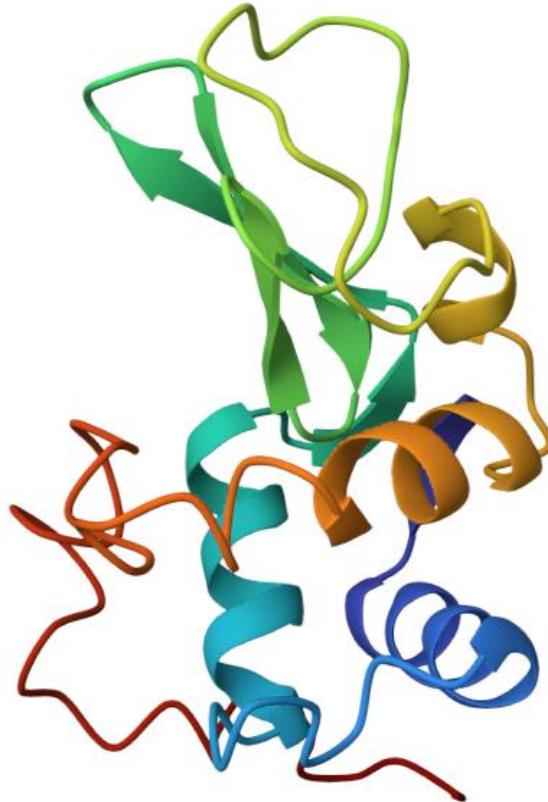
Martín Achondo Mercado
Universidad Técnica Federico Santa María, Chile

21 de Agosto, 2024

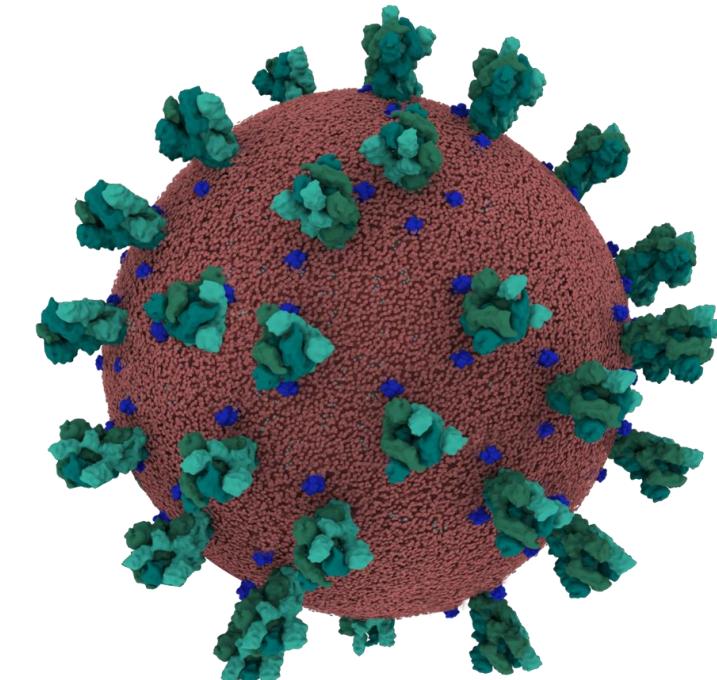
Electrostática Molecular



ADN



Lisozima



SARS-CoV-2

Electrostática Molecular

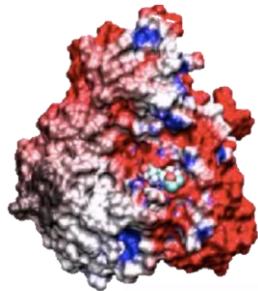
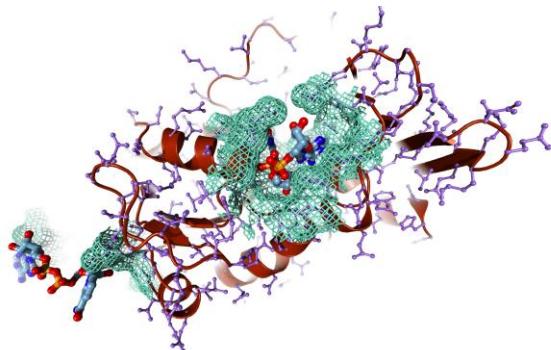
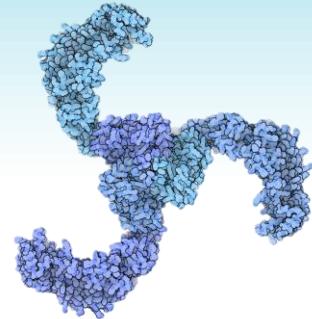
¿Variables de interés?

- Potencial electrostático.
- Fuerzas eléctricas / enlace.
- Energía de solvatación / enlace.
- Cálculos de pKa.
- Interacción de macromoléculas con superficies y otras macromoléculas.



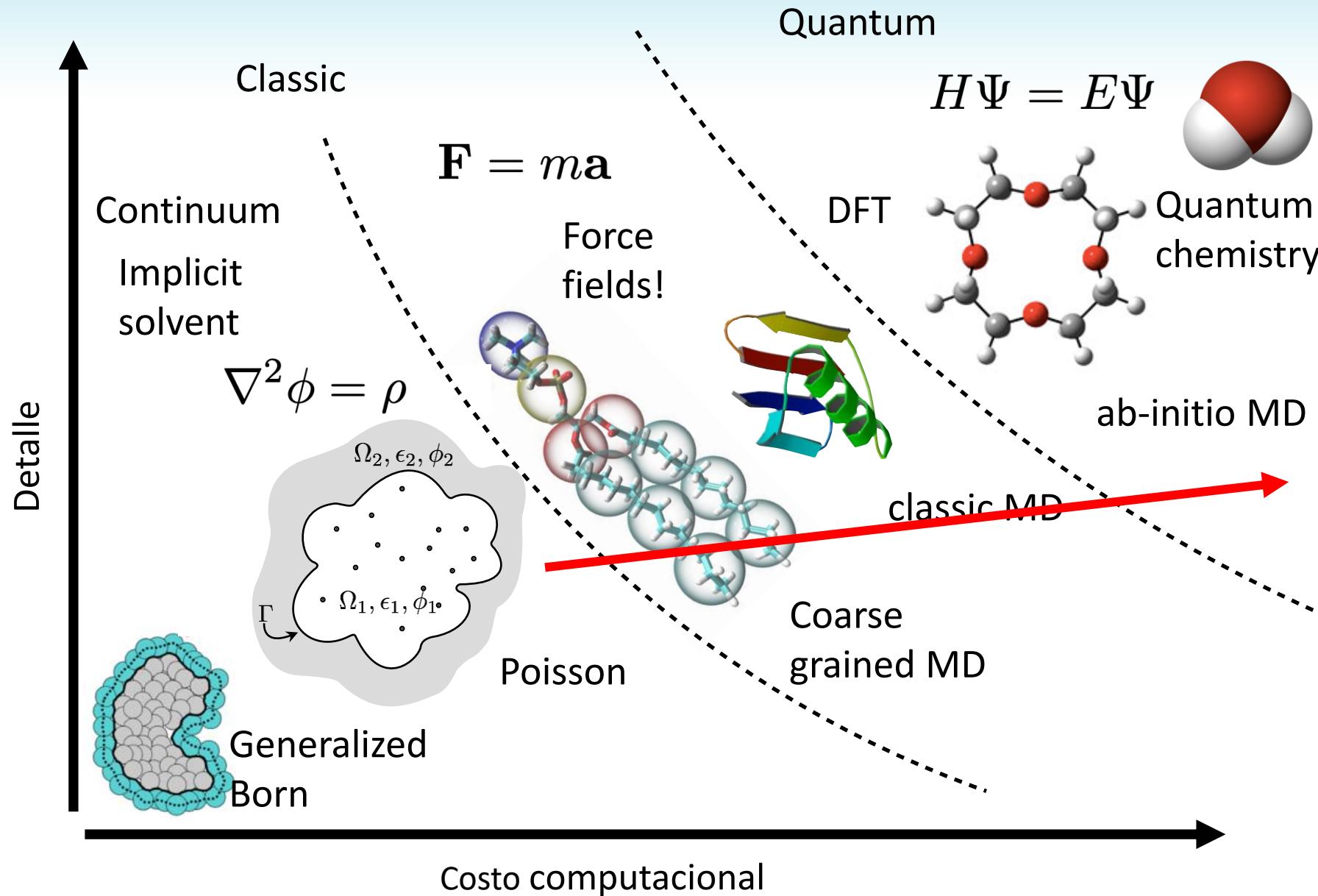
¿Aplicaciones?

- Entender la física.
- Diseño de Fármacos, su interacción y absorción.
- Diseño de biosensores, nanotecnología.
- Producción de H₂ verde.
- Proceso de flotación de Cu.



Le, et al. PLoS (2010)

¿Modelo?



Termodinámica al equilibrio:

- Configuración más probable.
- Reduce el costo computacional.

Motivación

Modelo Implícito lleva a la ecuación de Poisson-Boltzmann: Es una PDE!

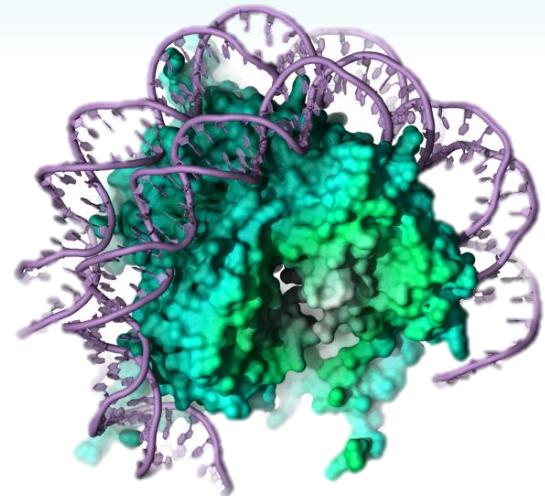
$$\nabla^2 \phi = \rho_f + \rho_m$$

Métodos tradicionales

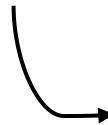
- FDM
- FEM
- BEM
- Entre otros



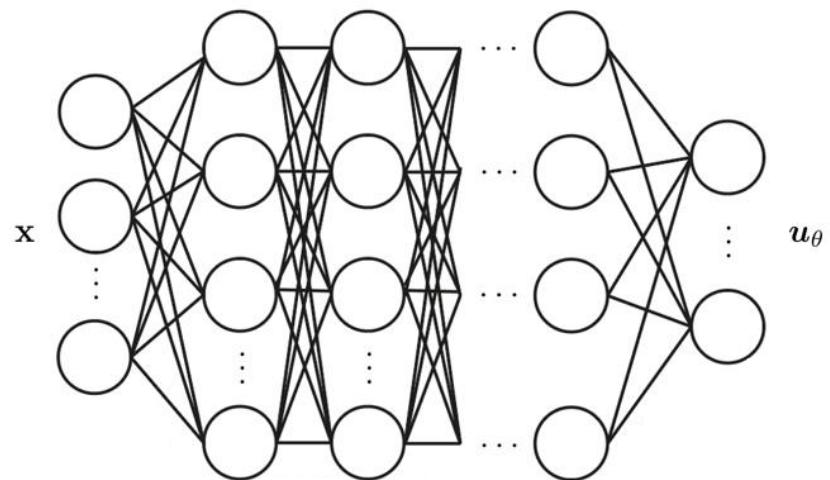
Discretizan el dominio (o la frontera) en elementos, para resolver sistemas lineales de ecuaciones.



PINNs: Physics Informed Neural Networks



- Utiliza una red neuronal para aproximar la solución.
- Se minimiza una función de pérdida.
- No necesita una discretización.



Motivación

- ¿PINNs será útil para **Electrostática Molecular**? → Aplicación directa en Biofísica

Se resuelve la ecuación de **Poisson-Boltzmann**, a partir del modelo del **Solvente Implícito** (simplificación).



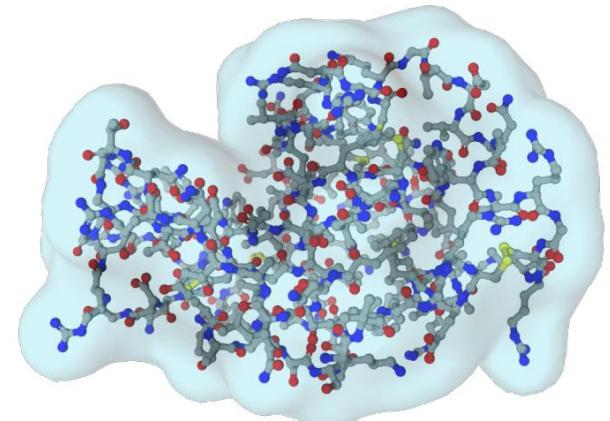
Se puede incluir:

- Mediciones experimentales.
- Dinámica Molecular.
- Estructura de la sal y el solvente.
- Leyes físicas adicionales.

Resolver problemas inversos.

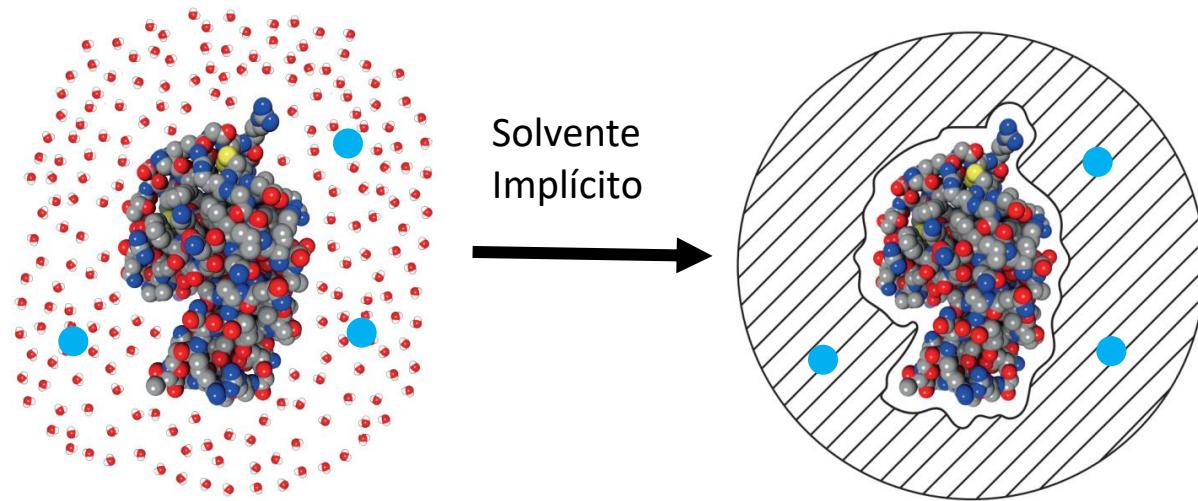
Estudio:

- Verificar que PINNs puede resolver la PBE (lineal).
 - Problema 3D.
 - Se tienen 2 dominios (sólido y solvente).
 - Singularidades por cargas puntuales.
 - Geometría en la interfaz compleja.
- Análisis de sensibilidad en arquitectura, puntos colocación, etc.
- Efecto de datos experimentales. No linealidad.



Ecuación de Poisson-Boltzmann

- Aplicable a macromoléculas en medios polarizables.
- Obtenida a partir del modelo de Solvente Implícito.



- Solvente se modela como un continuo.
- 2 regiones, soluto y solvente.
- Constante dieléctrica fija en cada región.

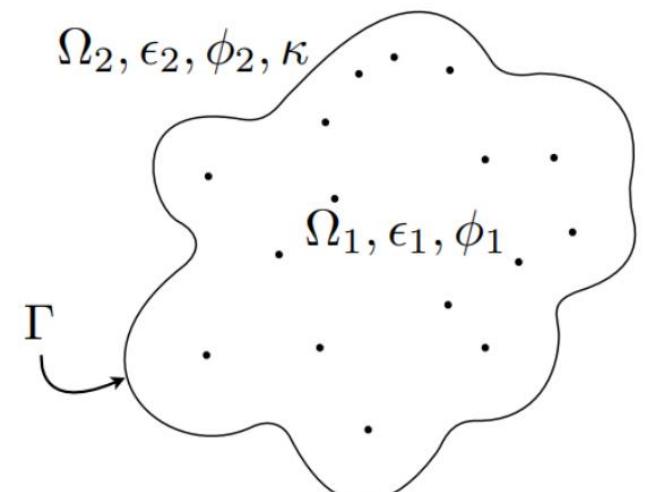
Ecuación de Poisson-Boltzmann

Densidad de carga:

- Cargas puntuales en soluto.
- Iones móviles en solvente (siguen distribución de Boltzmann).

$$\begin{cases} \nabla^2 \phi_1 = -\frac{1}{\epsilon_1} \sum_k q_k \delta(\mathbf{x} - \mathbf{x}_k) & \mathbf{x} \in \Omega_1 \\ \nabla^2 \phi_2 = \frac{2c^\infty q_e}{\epsilon_2} \sinh\left(\frac{\phi_2 q_e}{k_b T}\right) & \mathbf{x} \in \Omega_2 \end{cases}$$

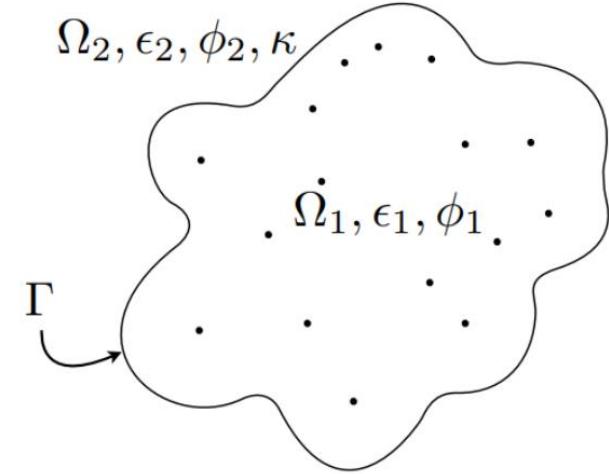
Forma Lineal: $\nabla^2 \phi_2 = \kappa^2 \phi_2$ $\kappa^2 = \frac{2c^\infty q_e^2}{\epsilon_2 k_b T}$



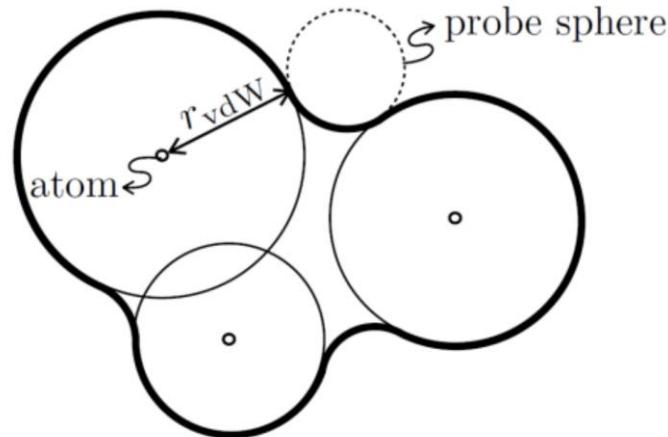
Ecuación de Poisson-Boltzmann

- Relaciones en la interfaz (SES):

$$\begin{cases} \phi_1 = \phi_2 & \mathbf{x} \in \Gamma \\ \epsilon_1 \frac{\partial \phi_1}{\partial n} = \epsilon_2 \frac{\partial \phi_2}{\partial n} & \mathbf{x} \in \Gamma \end{cases}$$



- Superficie SES:



Regularización de Poisson-Boltzmann

- Descomponer en componente **singular** (potencial de Coulomb) y **regular** (potencial de reacción).
- Descomposición **solo en el soluto** (singularidades).

$$\phi_1 = \psi_1 + \mathcal{G} \quad \mathbf{x} \in \Omega_1 \quad \nabla^2 \mathcal{G} = -\frac{1}{\epsilon_1} \sum_k q_k \delta(\mathbf{x} - \mathbf{x}_k)$$
$$\left\{ \begin{array}{l} \nabla^2 \phi_1 = -\frac{1}{\epsilon_1} \sum_k q_k \delta(\mathbf{x} - \mathbf{x}_k) \quad \mathbf{x} \in \Omega_1 \\ \nabla^2 \phi_2 = \kappa^2 \phi_2 \quad \mathbf{x} \in \Omega_2 \end{array} \right.$$
$$\mathcal{G}(\mathbf{r}) = \frac{1}{4\pi\epsilon_1} \sum_k \frac{q_k}{|\mathbf{r} - \mathbf{r}_k|}$$

Regularización de Poisson-Boltzmann

- Descomponer en componente **singular** (potencial de Coulomb) y **regular** (potencial de reacción).
- Descomposición **solo en el soluto** (singularidades).

$$\phi_1 = \psi_1 + \mathcal{G} \quad \mathbf{x} \in \Omega_1 \quad \nabla^2 \mathcal{G} = -\frac{1}{\epsilon_1} \sum_k q_k \delta(\mathbf{x} - \mathbf{x}_k)$$

$$\begin{cases} \nabla^2(\psi_1 + \mathcal{G}) = -\frac{1}{\epsilon_1} \sum_k q_k \delta(\mathbf{x} - \mathbf{x}_k) & \mathbf{x} \in \Omega_1 \\ \nabla^2 \phi_2 = \kappa^2 \phi_2 & \mathbf{x} \in \Omega_2 \end{cases}$$

Regularización de Poisson-Boltzmann

- Descomponer en componente **singular** (potencial de Coulomb) y **regular** (potencial de reacción).
- Descomposición **solo en el soluto** (singularidades).

$$\phi_1 = \psi_1 + \mathcal{G} \quad \mathbf{x} \in \Omega_1$$

Se evitan las singularidades

$$\begin{cases} \nabla^2 \psi_1 = 0 & \mathbf{x} \in \Omega_1 \\ \nabla^2 \phi_2 = \kappa^2 \phi_2 & \mathbf{x} \in \Omega_2 \end{cases}$$

$$\begin{cases} \psi_1 + \mathcal{G} = \phi_2 & \mathbf{x} \in \Gamma \\ \epsilon_1 \frac{\partial}{\partial n} (\psi_1 + \mathcal{G}) = \epsilon_2 \frac{\partial \phi_2}{\partial n} & \mathbf{x} \in \Gamma \end{cases}$$

Potencial adimensional $\phi^* = \frac{\phi \epsilon_0 r_c}{q_e}$

Red Neuronal Artificial (ANN)

Corresponde a una composición de funciones no lineales que operan sobre un conjunto de parámetros θ y una variable de entrada x .

$$\mathbf{u} \approx \mathbf{u}_\theta = \mathcal{N}(\mathbf{x}; \theta)$$

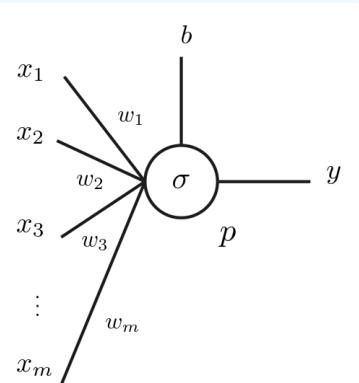
La composición de funciones (agrupadas en capas f_θ^i), detallan la arquitectura de la red neuronal.

$$\mathbf{u}_\theta = f_\theta^{L+1} \circ f_\theta^L \cdots \circ f_\theta^1(\mathbf{x})$$

Para encontrar los parámetros óptimos, se minimiza una función de pérdida que depende del problema.

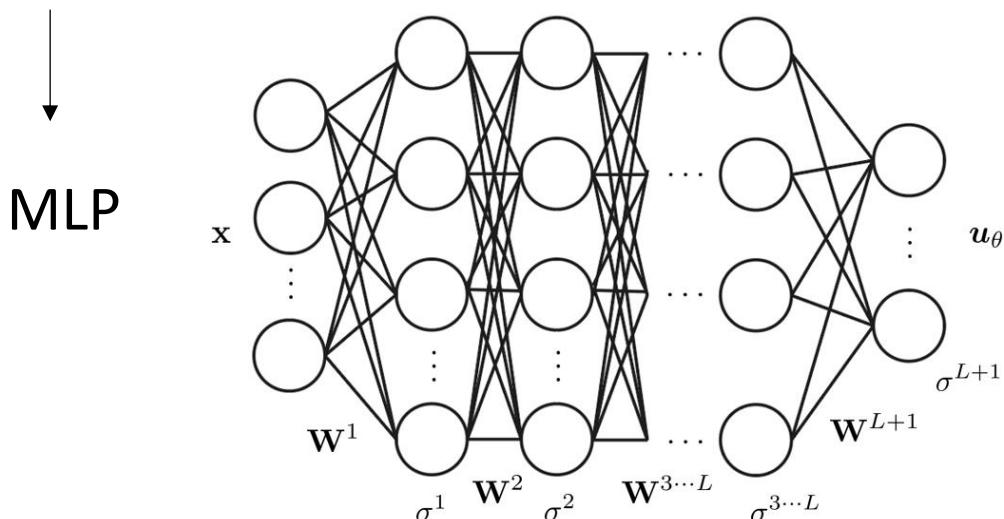
$$\theta^* = \operatorname{argmin}_{\theta} \mathcal{L}(\theta)$$

Ejemplo MLP: Multi Layer Perceptron



$$y = \sigma \left(\sum_{j=1}^m w_j x_j + b \right)$$

Varios perceptrones (neuronas) interconectados entre sí:



Conjunto de parámetros:

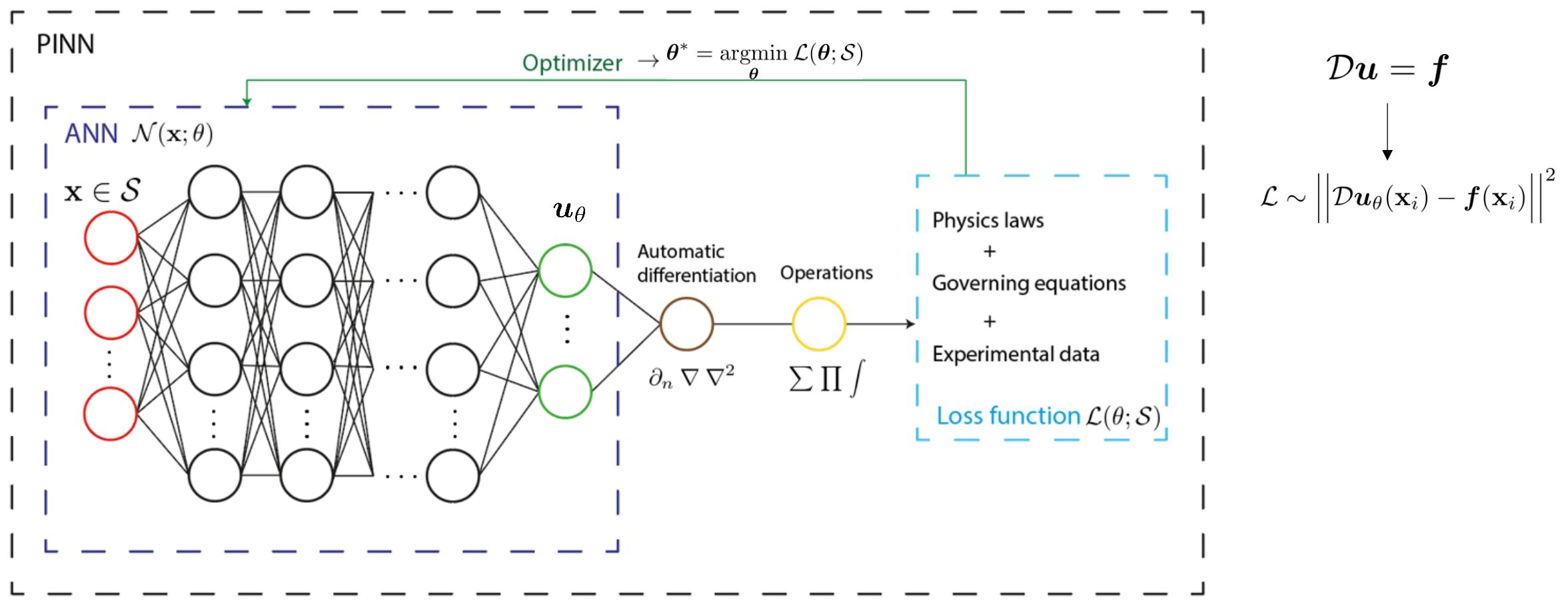
$$\theta = \{\mathbf{W}^1, \mathbf{b}^1, \dots, \mathbf{W}^{L+1}, \mathbf{b}^{L+1}\}$$

Capa de perceptrones:

$$f_\theta^i(\mathbf{x}) = \sigma^i(\mathbf{W}^i \mathbf{x} + \mathbf{b}^i)$$

PINNs (Physics Informed Neural Networks)

- Se incluye la ecuación a resolver (residuales, condiciones de borde, leyes físicas, etc.) en la **función de pérdida** de la red neuronal \mathcal{L} .



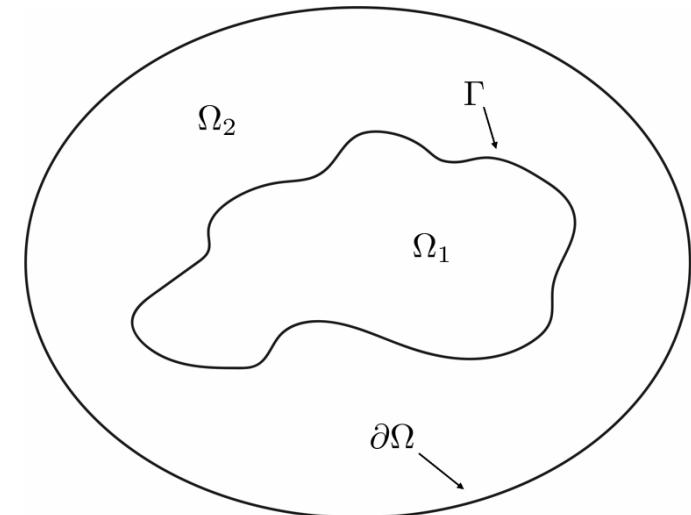
Contexto de PINNs aplicado a PDEs Elípticas

Existen trabajos que utilizan PINNs para resolver PDEs elípticas.

$$\begin{aligned} -\nabla \cdot (a_1 \nabla u) + b_1 u &= f_1(\mathbf{x}) \quad \mathbf{x} \in \Omega_1 \\ -\nabla \cdot (a_2 \nabla u) + b_2 u &= f_2(\mathbf{x}) \quad \mathbf{x} \in \Omega_2 \\ [u] &= \varphi(\mathbf{x}) \quad \mathbf{x} \in \Gamma \\ [a \partial_n u] &= \omega(\mathbf{x}) \quad \mathbf{x} \in \Gamma \\ u &= g(\mathbf{x}) \quad \mathbf{x} \in \partial\Omega \end{aligned}$$

- 2 redes neuronales, una por cada dominio.

$$u \approx u_\theta = \begin{cases} \mathcal{N}_1(\mathbf{x}; \boldsymbol{\theta}^1) & \mathbf{x} \in \Omega_1 \\ \mathcal{N}_2(\mathbf{x}; \boldsymbol{\theta}^2) & \mathbf{x} \in \Omega_2 \end{cases}$$



- Existen estudios de la convergencia y las cotas del error de generalización.
- Estudios “simples”. Aplicaciones en PB. → Se necesita una metodología.

PINNs aplicado a PB regularizada

- **1 ANN con 2 ramificaciones**, una para cada región:

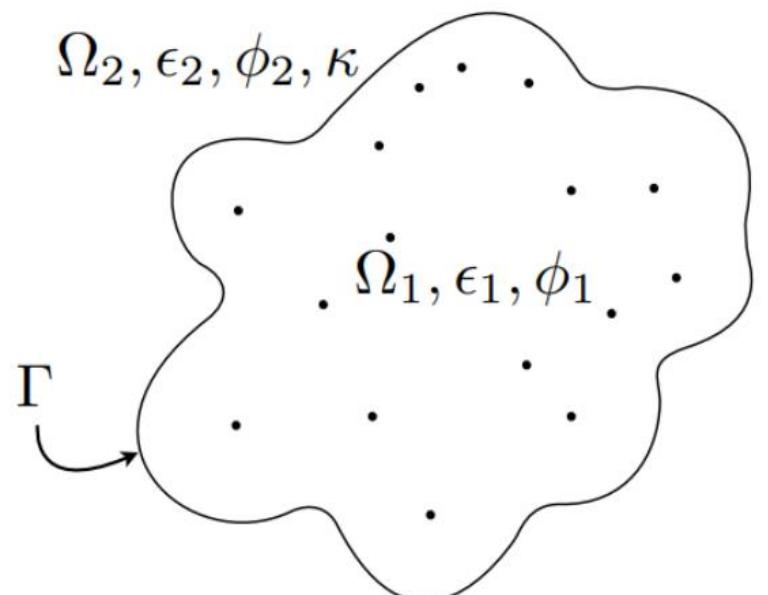
- N°1: Soluto.
- N°2: Solvente.

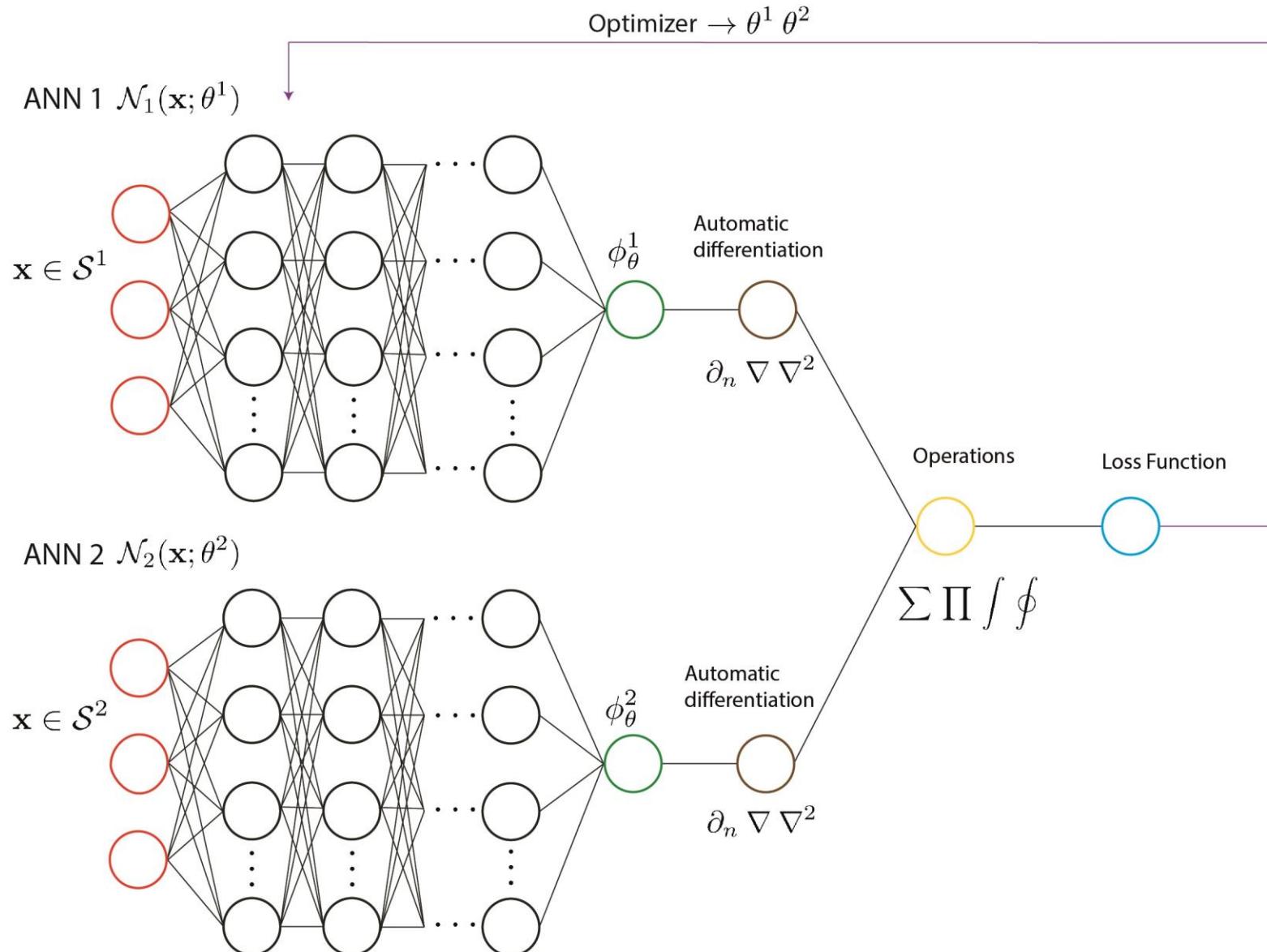
$$\mathcal{N}(\mathbf{x}; \boldsymbol{\theta}) = \begin{cases} \mathcal{N}_1(\mathbf{x}; \boldsymbol{\theta}^1) & \mathbf{x} \in \Omega_1 \\ \mathcal{N}_2(\mathbf{x}; \boldsymbol{\theta}^2) & \mathbf{x} \in \Omega_2 \end{cases}$$

- Cada red entrega un potencial distinto:

$$\psi_\theta^1 = \mathcal{N}_1(\mathbf{x}; \boldsymbol{\theta}^1) \quad \phi_\theta^2 = \mathcal{N}_2(\mathbf{x}; \boldsymbol{\theta}^2)$$

$$\boxed{\phi_\theta = \begin{cases} \phi_\theta^1 = \mathcal{G}(\mathbf{x}) + \mathcal{N}_1(\mathbf{x}; \boldsymbol{\theta}^1) & \mathbf{x} \in \Omega_1 \\ \phi_\theta^2 = \mathcal{N}_2(\mathbf{x}; \boldsymbol{\theta}^2) & \mathbf{x} \in \Omega_2 \\ \frac{\phi_\theta^1 + \phi_\theta^2}{2} & \mathbf{x} \in \Gamma \end{cases}}$$





PINNs aplicado a PB:

- 2 ramificaciones.
- Función de pérdida que depende de ambas ramificaciones.

$$\mathcal{N}(x; \theta) = \begin{cases} \mathcal{N}_1(x; \theta^1) & x \in \Omega_1 \\ \mathcal{N}_2(x; \theta^2) & x \in \Omega_2 \end{cases}$$

$$\theta = \theta^1 \cup \theta^2$$

PINNs aplicado a PB regularizada

Residual soluto

$$\mathcal{L}_{\Omega_1}(\boldsymbol{\theta}^1; \mathcal{S}_{\Omega_1})$$

Residual solvente

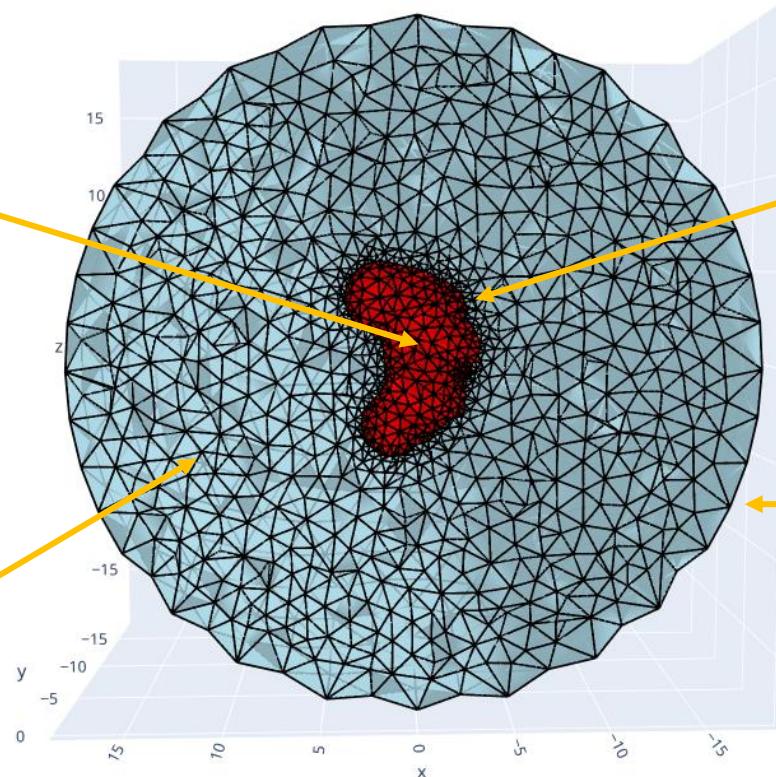
$$\mathcal{L}_{\Omega_2}(\boldsymbol{\theta}^2; \mathcal{S}_{\Omega_2})$$

Continuidad interfaz

$$\mathcal{L}_\Gamma(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2; \mathcal{S}_\Gamma)$$

Condición de borde

$$\mathcal{L}_{\partial\Omega}(\boldsymbol{\theta}^2; \mathcal{S}_{\partial\Omega})$$



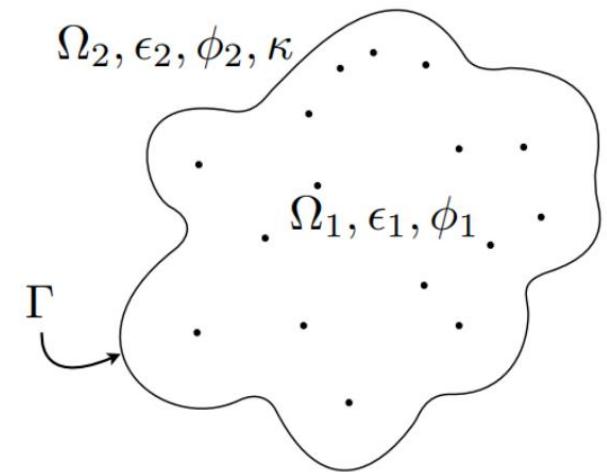
$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}; \mathcal{S}) = & w_{\Omega_1} \mathcal{L}_{\Omega_1}(\boldsymbol{\theta}^1; \mathcal{S}_{\Omega_1}) + w_{\Omega_2} \mathcal{L}_{\Omega_2}(\boldsymbol{\theta}^2; \mathcal{S}_{\Omega_2}) + w_{\partial\Omega} \mathcal{L}_{\partial\Omega}(\boldsymbol{\theta}^2; \mathcal{S}_{\partial\Omega}) \\ & + w_{\Gamma_D} \mathcal{L}_{\Gamma_D}(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2; \mathcal{S}_\Gamma) + w_{\Gamma_N} \mathcal{L}_{\Gamma_N}(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2; \mathcal{S}_\Gamma)\end{aligned}$$

PINNs aplicado a PB

Residuales:

$$\mathcal{L}_{\Omega_1}(\boldsymbol{\theta}^1; \mathcal{S}_{\Omega_1}) = \frac{1}{N_{\Omega_1}} \sum_{\mathbf{x}_i \in \mathcal{S}_{\Omega_1}} \left[\nabla^2 \psi_{\theta}^1(\mathbf{x}_i) \right]^2$$

$$\mathcal{L}_{\Omega_2}(\boldsymbol{\theta}^2; \mathcal{S}_{\Omega_2}) = \frac{1}{N_{\Omega_2}} \sum_{\mathbf{x}_i \in \mathcal{S}_{\Omega_2}} \left[\nabla^2 \phi_{\theta}^2(\mathbf{x}_i) - \kappa^2 \phi_{\theta}^2(\mathbf{x}_i) \right]^2$$

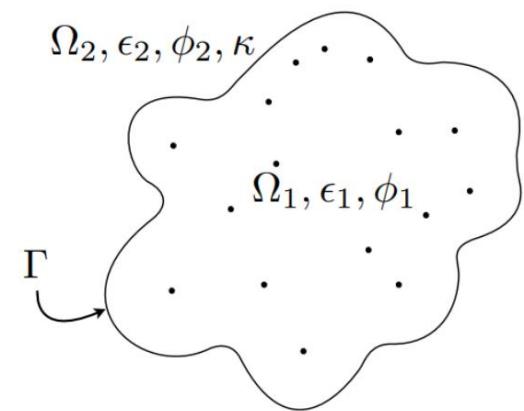


PINNs aplicado a PB

Condiciones en la interfaz:

$$\mathcal{L}_{\Gamma_D}(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2; \mathcal{S}_\Gamma) = \frac{1}{N_\Gamma} \sum_{\mathbf{x}_i \in \mathcal{S}_\Gamma} \left[\phi_\theta^2(\mathbf{x}_i) - \psi_\theta^1(\mathbf{x}_i) - \mathcal{G}(\mathbf{x}_i) \right]^2$$

$$\mathcal{L}_{\Gamma_N}(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2; \mathcal{S}_\Gamma) = \frac{1}{N_\Gamma} \sum_{\mathbf{x}_i \in \mathcal{S}_\Gamma} \left[\epsilon_2 \frac{\partial}{\partial n} \left(\phi_\theta^2(\mathbf{x}_i) \right) - \epsilon_1 \frac{\partial}{\partial n} \left(\psi_\theta^1(\mathbf{x}_i) + \mathcal{G}(\mathbf{x}_i) \right) \right]^2$$



PINNs aplicado a PB

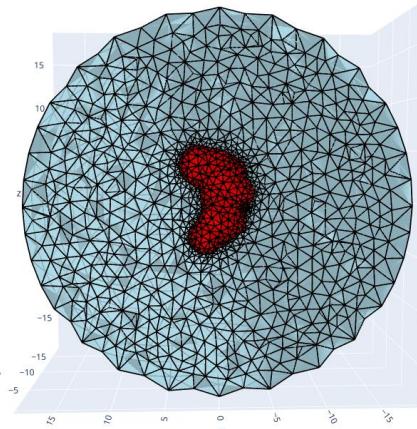
Condición de borde :

- Potencial de Yukawa.
- Buena aproximación lejos de la molécula.

$$\phi(\mathbf{x}_i) \approx \frac{1}{4\pi\epsilon_2} \sum_k \frac{q_k e^{-\kappa|\mathbf{x}_i - \mathbf{x}_k|}}{|\mathbf{x}_i - \mathbf{x}_k|}$$

Forma débil (término en función de pérdida)

$$\mathcal{L}_{\partial\Omega}(\boldsymbol{\theta}^2; \mathcal{S}_{\partial\Omega}) = \frac{1}{N_{\partial\Omega}} \sum_{\mathbf{x}_i \in \mathcal{S}_{\partial\Omega}} \left[\phi_\theta^2(\mathbf{x}_i) - \frac{1}{4\pi\epsilon_2} \sum_k \frac{q_k e^{-\kappa|\mathbf{x}_i - \mathbf{x}_k|}}{|\mathbf{x}_i - \mathbf{x}_k|} \right]^2$$



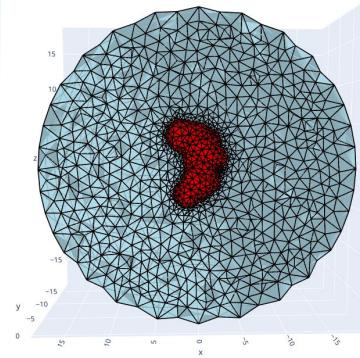
“Solución conocida” (se aplicará solo en solvente).

$$\mathcal{L}_{data}(\boldsymbol{\theta}^2; \mathcal{S}_{data}) = \frac{1}{N_{data}} \sum_{\mathbf{x}_i \in \mathcal{S}_{data}} \left[\phi_\theta^2(\mathbf{x}_i) - \phi^\dagger(\mathbf{x}_i) \right]^2$$

Construcción de Puntos de Colocación

Se necesita un método rápido para generar un muestreo aleatorio en las 4 regiones ($\Omega_1, \Omega_2, \partial\Omega, \Gamma$).

- Se generan **4 mallas** (2 superficiales y 2 volumétricas).
- Se obtienen **muestras aleatorias de estas mallas** (puntos aleatorios dentro de cada elemento)

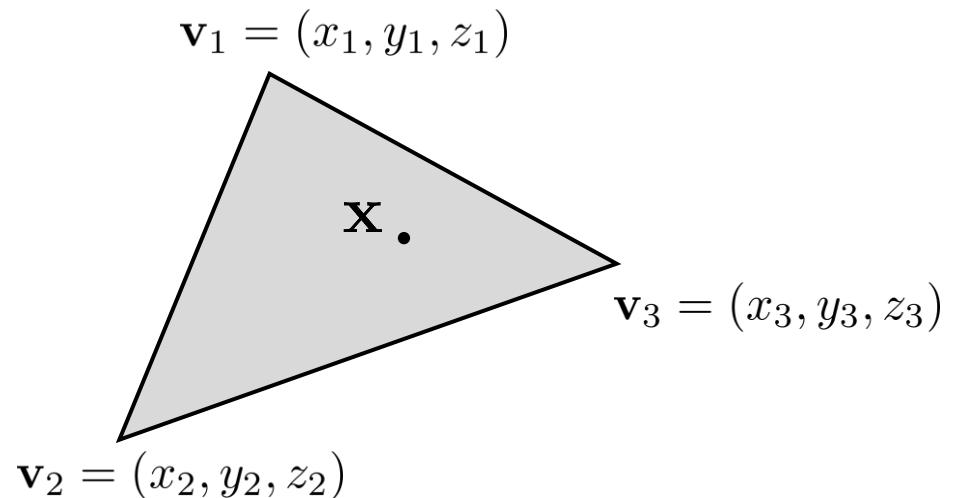


Para cada elemento (triangulares N=3 o tetraédricos N=4)

1) N números aleatorios: $\lambda_j \sim \text{Unif}(0, 1)$

2) Se normalizan: $\hat{\lambda}_j = \frac{\lambda_j}{\sum_k^N \lambda_k}$

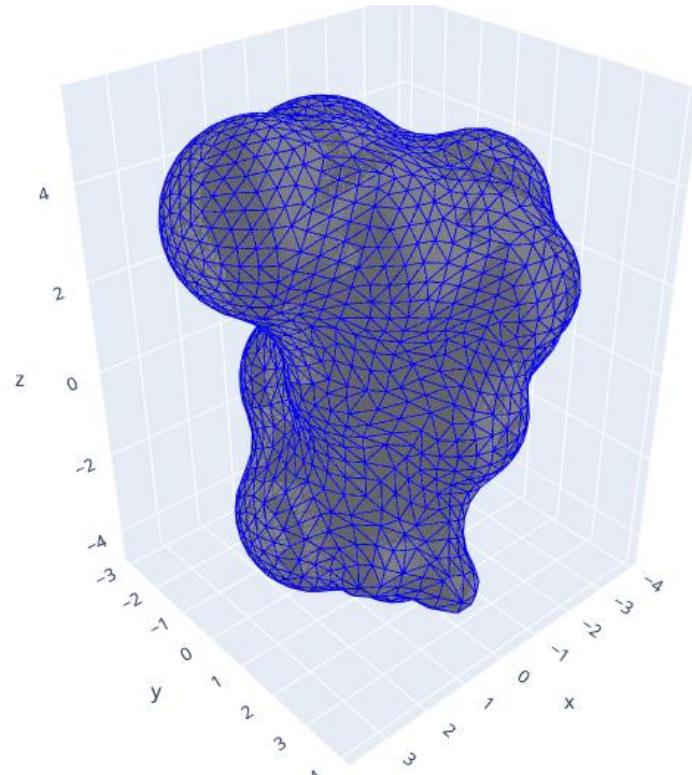
3) Se obtiene el punto aleatorio (combinación lineal de los vértices):
$$\mathbf{x} = \sum_{j=1}^N \lambda_j \mathbf{v}_j$$



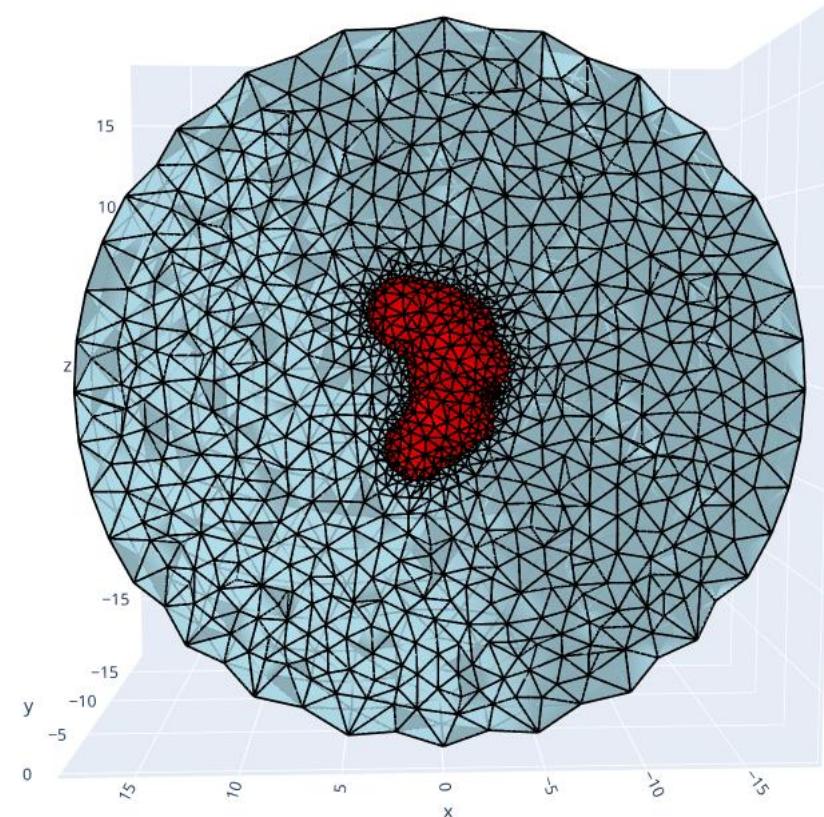
Construcción de Puntos de Colocación

- Construcción de mallas (softwares de MSMS, NanoShaper, Tetgen).

Superficie SES a partir de info. molecular

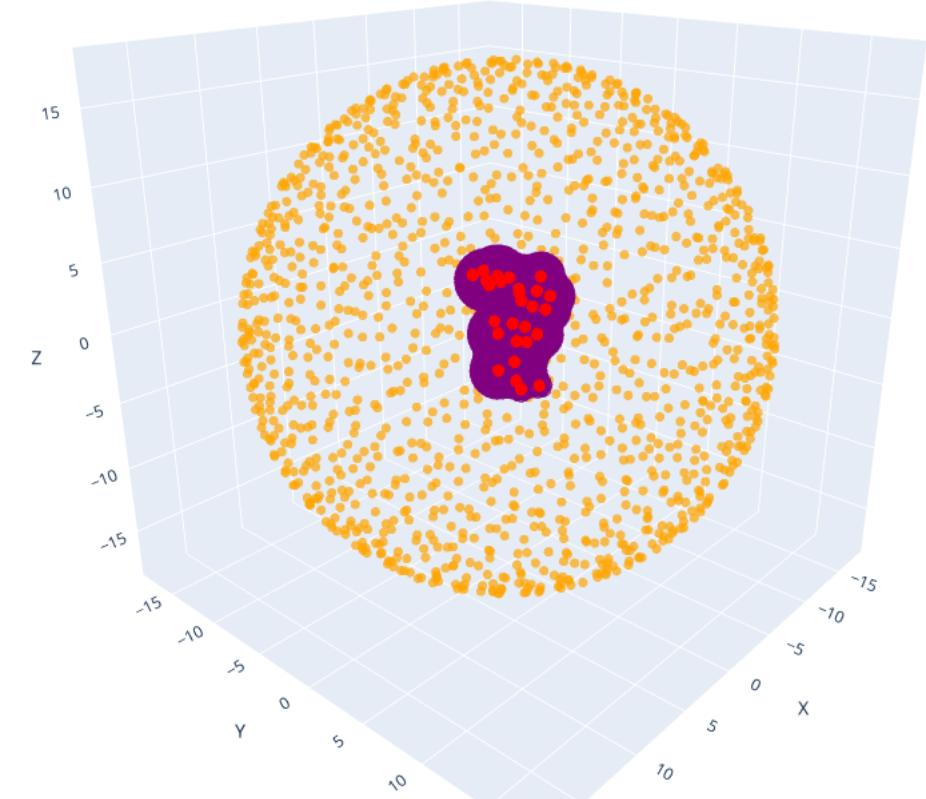
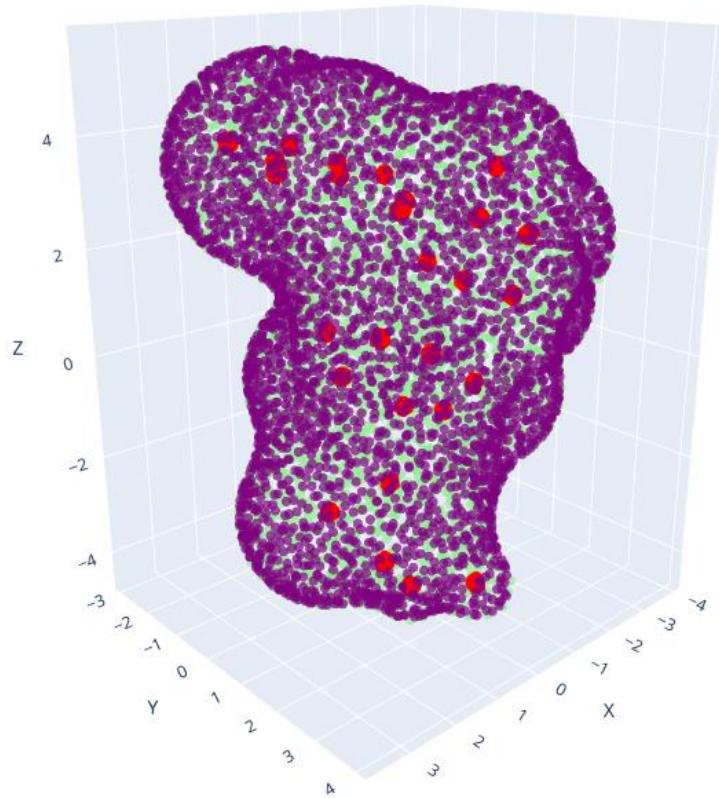


Generación del resto de las mallas



Construcción de Puntos de Colocación

- Ejemplos de puntos de colocación.



Algoritmo de Ponderación de Pesos de \mathcal{L}

- La idea es balancear la contribución de los distintos términos de la función de pérdida para la modificación del conjunto θ .

$$\mathcal{L}(\theta; \mathcal{S}) = \sum_j w_j \mathcal{L}_j(\theta; \mathcal{S}_j) \longrightarrow C = w_j \|\nabla_{\theta} \mathcal{L}_j\| \quad \forall j \quad \text{Se balancea el gradiente de } \mathcal{L}$$

$$\hat{w}_j = \frac{\sum_i \|\nabla_{\theta} \mathcal{L}_i\|}{\|\nabla_{\theta} \mathcal{L}_j\|} \quad \left. \begin{array}{c} \uparrow \\ C \end{array} \right\}$$

Se calculan los nuevos pesos

$$w_{j,\text{new}} = \alpha w_{j,\text{old}} + (1 - \alpha) \hat{w}_j$$

- El algoritmo funciona bien aplicándose cada 1000 o 2000 iteraciones, con $\alpha=0.7$

Implementaciones

- **Capa de Fourier:** Mapeo de señales de distinta frecuencia.

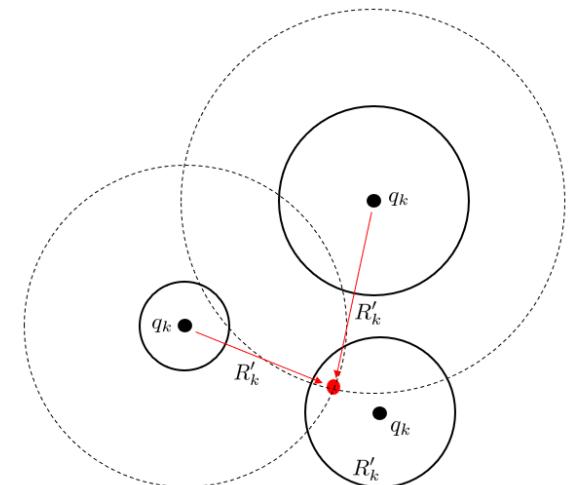
$$\mathbf{h} = \begin{bmatrix} \cos(\mathbf{Bx}) \\ \sin(\mathbf{Bx}) \end{bmatrix} \quad \mathbf{B}: \text{Distribución normal (no entrenable)}$$

- **Capas de escalamiento:** (Valores de entrada y salida entre -1 y 1)

Input: $\mathbf{h} = 2 \frac{(\mathbf{x} - \mathbf{x}_{min})}{(\mathbf{x}_{max} - \mathbf{x}_{min})} - 1$

Output: $\mathbf{y} = \frac{\mathbf{h} + 1}{2} (\mathbf{y}_{max} - \mathbf{y}_{min}) + \mathbf{y}_{min}$

Se necesita estimar el valor máximo y mínimo del potencial



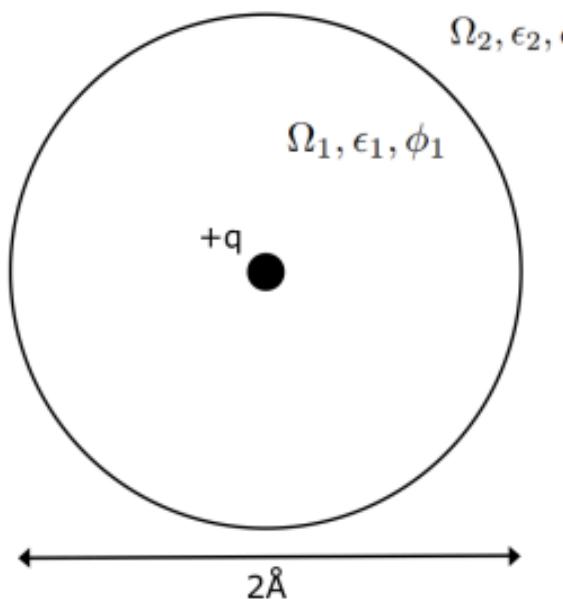
$$\psi_1(r) = \frac{q}{4\pi} \left(\frac{1}{\epsilon_2(1 + \kappa R)R} - \frac{1}{\epsilon_1 R} \right)$$

Caso de prueba: Ion de Born

- Molécula esférica, 1 carga puntual.

$$\phi_1(r) = \frac{q}{4\pi} \left(\frac{1}{\epsilon_1 r} - \frac{1}{\epsilon_1 R} + \frac{1}{\epsilon_2 (1 + \kappa R) R} \right)$$

$$\phi_2(r) = \frac{q}{4\pi} \frac{\exp(-\kappa(r - R))}{\epsilon_2 (1 + \kappa R) r}$$



$$\begin{aligned}\epsilon_1 &= 1 \\ \epsilon_2 &= 80 \\ \kappa &= 0.125\end{aligned}$$

Implementación

- Arquitectura MLP.
- 4 capas ocultas.
- 200 neuronas por capa.
- Activación: tanh entrenable.
- Método optimización: ADAM.
- 20,000 iteraciones.

Experimentos Numéricos

- Cálculo de la energía de solvatación.

$$\Delta G_{solv}^\theta = \frac{1}{2} \sum_k q_k \psi_\theta(\mathbf{x}_k) \quad \mathcal{E}_{G_{solv}} = \left| \frac{\Delta G_{solv}^\theta - \Delta G_{solv}^\dagger}{\Delta G_{solv}^\dagger} \right|$$

- Cálculo del error del potencial en la interfaz.

$$\mathcal{E}_\psi = \sqrt{\frac{\sum_{\mathbf{x}_i \in \mathcal{S}_\Gamma} [\bar{\psi}_\theta(\mathbf{x}_i) - \psi^\dagger(\mathbf{x}_i)]^2}{\sum_{\mathbf{x}_i \in \mathcal{S}_\Gamma} [\psi^\dagger(\mathbf{x}_i)]^2}}$$

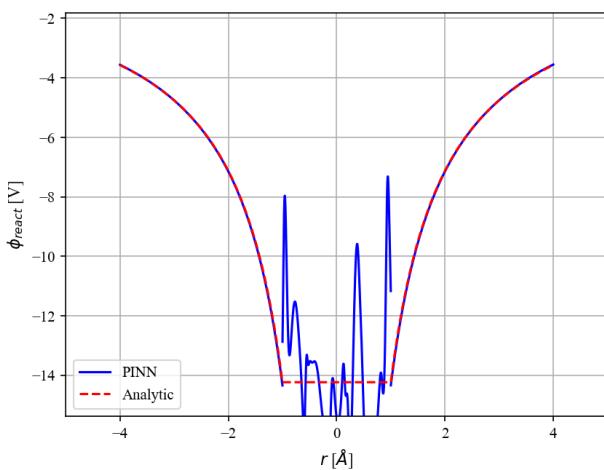
Comparación con:

- Solución analítica.
- BEM.
- FDM.

- Loss de entrenamiento y validación.

Puntos de colocación

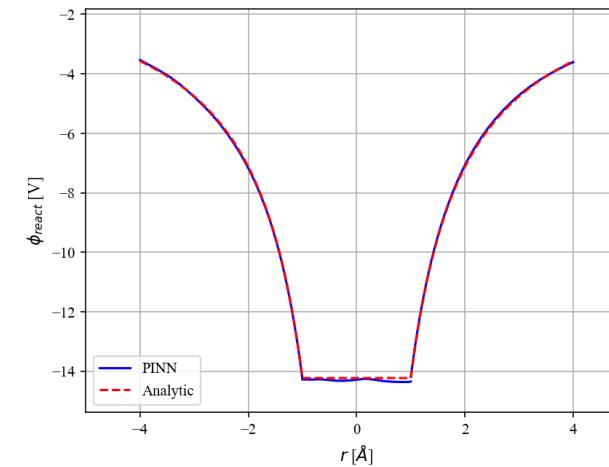
Muestreo fijo



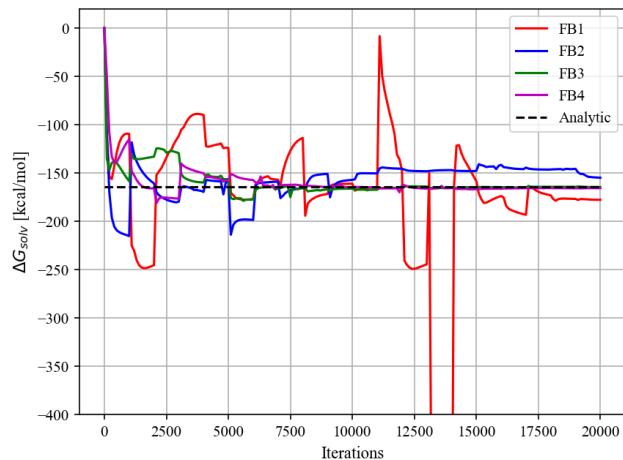
$$\begin{aligned}\mathcal{E}_G &= 8\text{E-}02 \\ \mathcal{E}_\psi &= 2\text{E-}01 \\ \mathcal{L}_V &= 2\text{E+}04\end{aligned}$$

Mejora

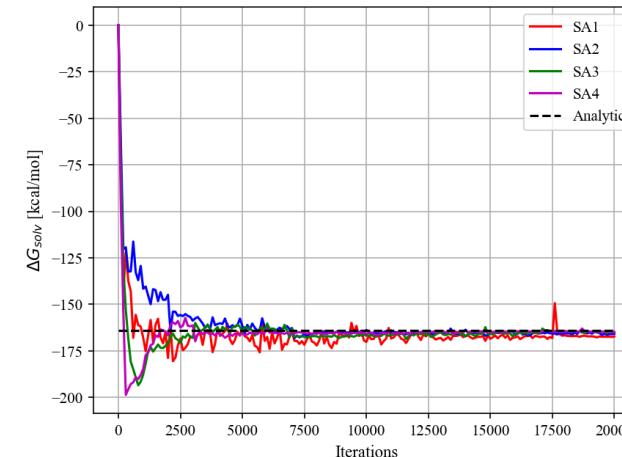
Muestreo aleatorio



$$\begin{aligned}\mathcal{E}_G &= 4\text{E-}03 \\ \mathcal{E}_\psi &= 6\text{E-}03 \\ \mathcal{L}_V &= 4\text{E-}04\end{aligned}$$



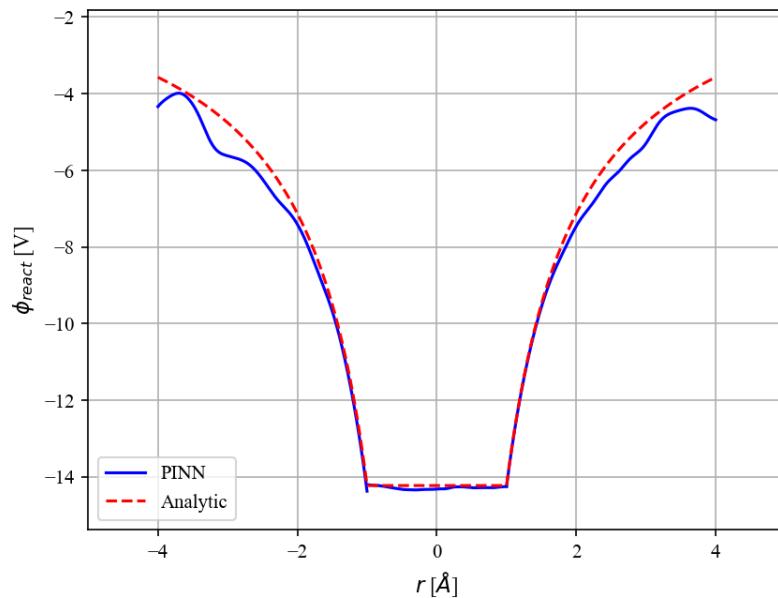
Mejora



Muestras de mayor tamaño mejoran la convergencia.

Algoritmo de ponderación de pesos

Sin ponderación de pesos

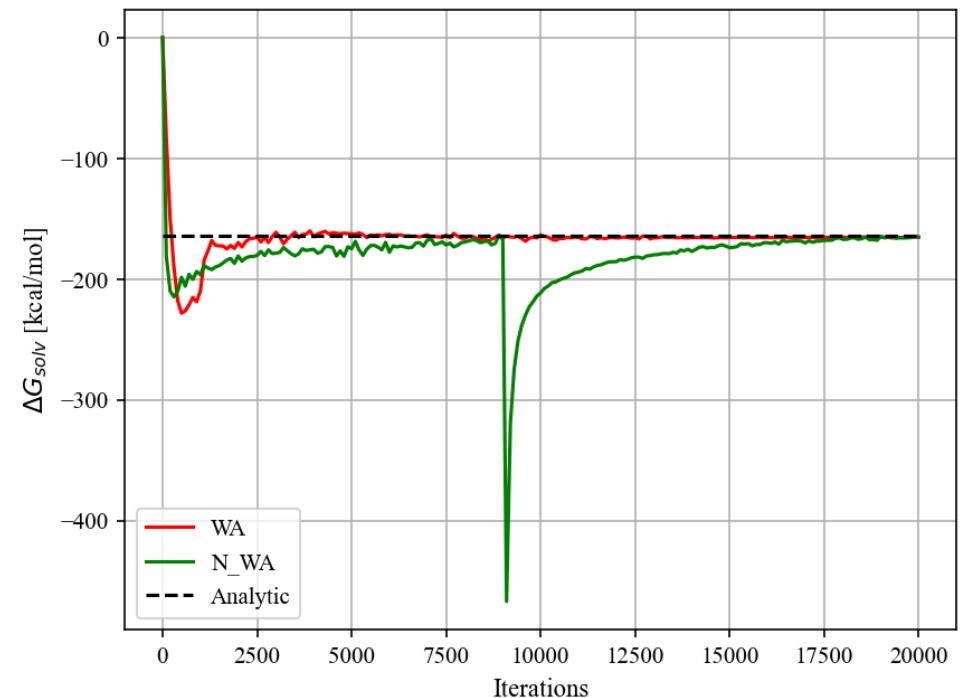


- No utilizar algoritmo empeora la solución.
- Muchos términos se minimizan simultáneamente.

$$\hat{w}_j = \frac{i}{\|\nabla_{\theta} \mathcal{L}_j\|} \sum \|\nabla_{\theta} \mathcal{L}_i\|$$

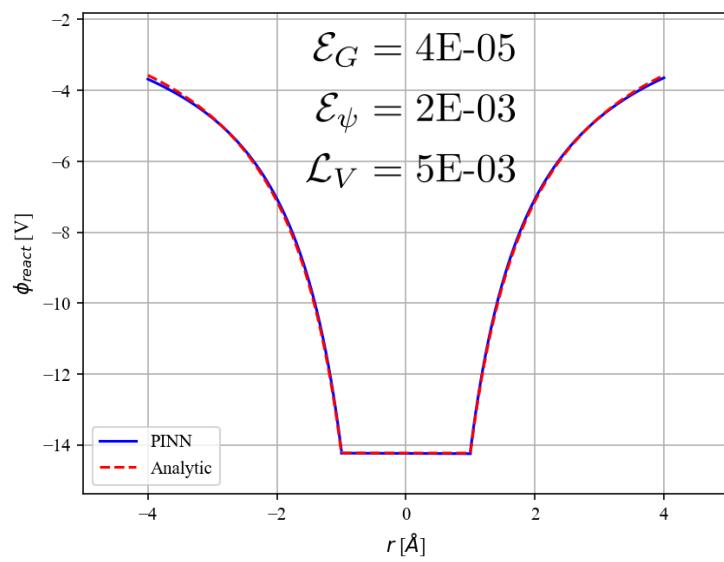
$$w_{j,\text{new}} = \alpha w_{j,\text{old}} + (1 - \alpha) \hat{w}_j$$

$$\begin{aligned}\mathcal{E}_G &= 5\text{E-}03 \\ \mathcal{E}_{\psi} &= 5\text{E-}03 \\ \mathcal{L}_V &= 1\text{E-}02\end{aligned}$$

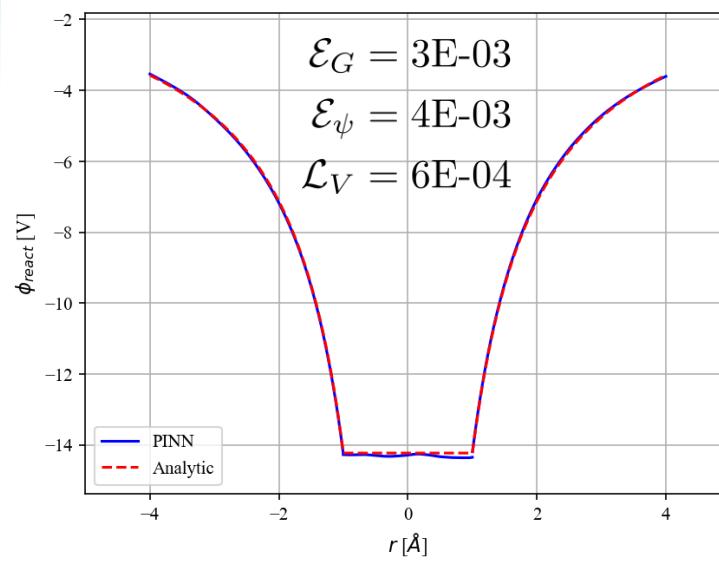


Capa de Fourier

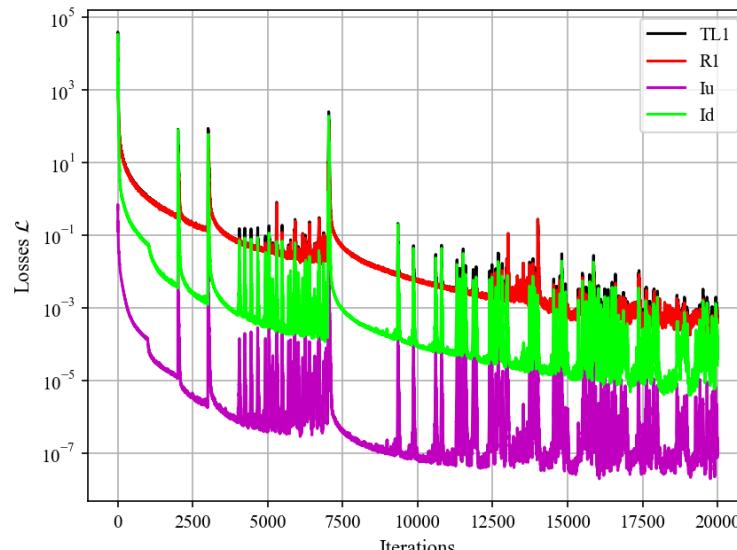
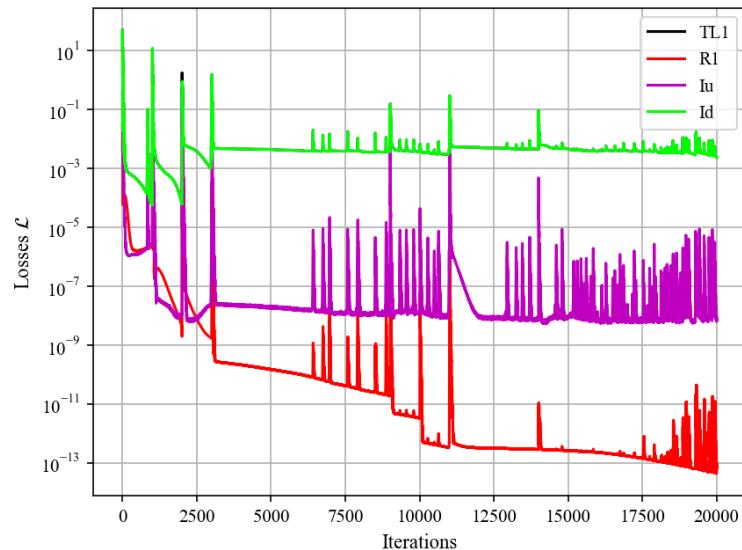
Sin FF



Con FF



$$\mathbf{h} = \begin{bmatrix} \cos(\mathbf{Bx}) \\ \sin(\mathbf{Bx}) \end{bmatrix}$$

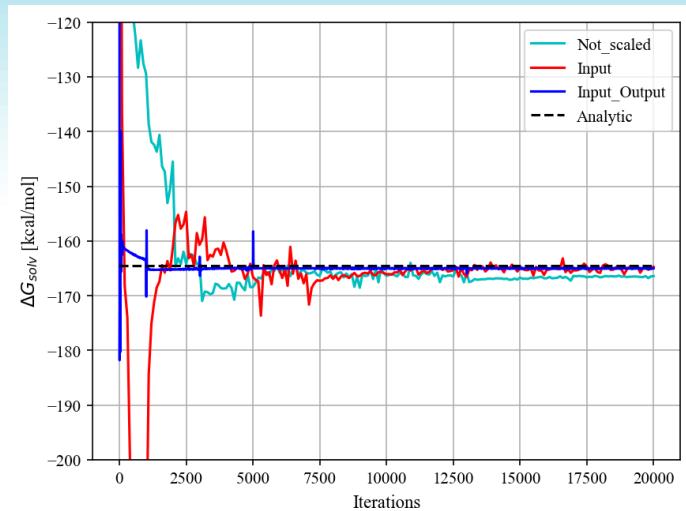


- FF agrega desviaciones en el potencial.
- Existe un cambio en el comportamiento de los losses.

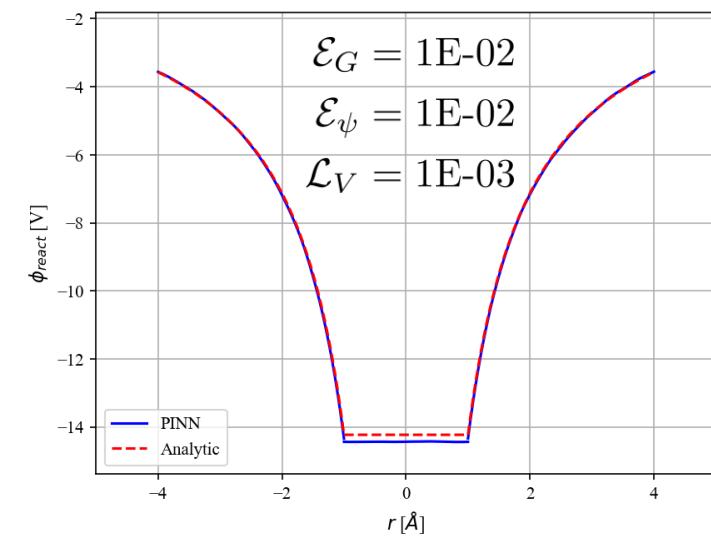
Escalamientos

Input: $h = 2 \frac{(\mathbf{x} - \mathbf{x}_{min})}{(\mathbf{x}_{max} - \mathbf{x}_{min})} - 1$

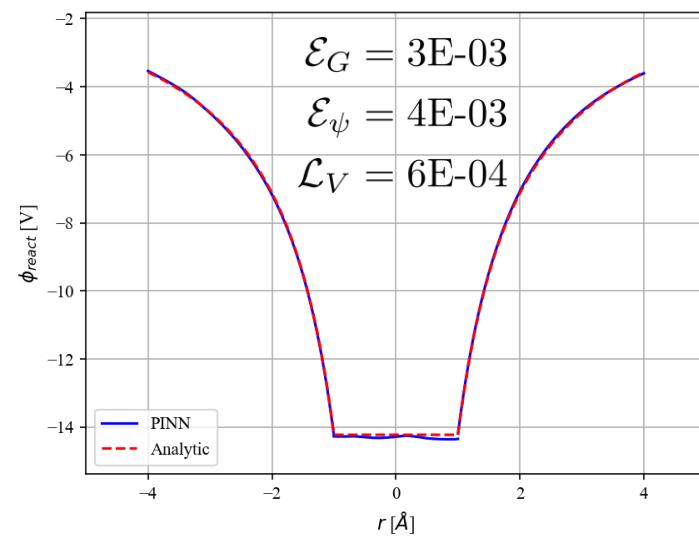
Output: $y = \frac{h + 1}{2} (y_{max} - y_{min}) + y_{min}$



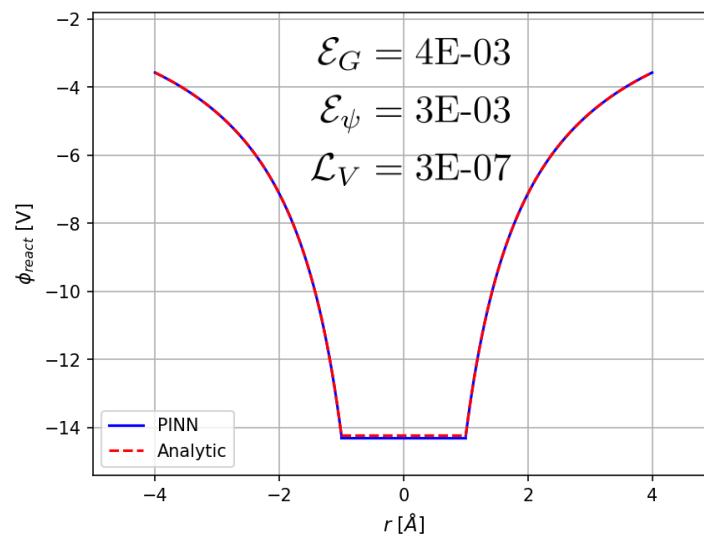
Sin escalamientos



Input

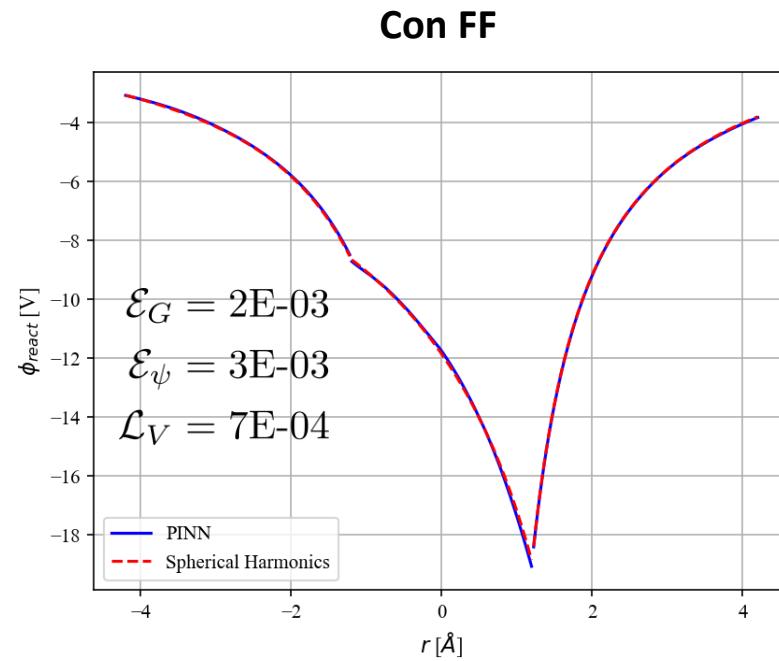
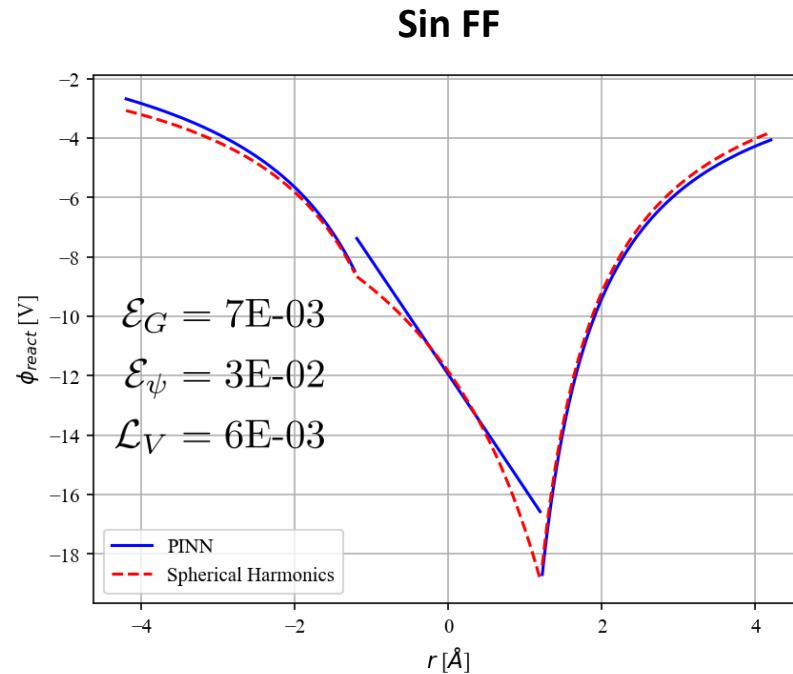
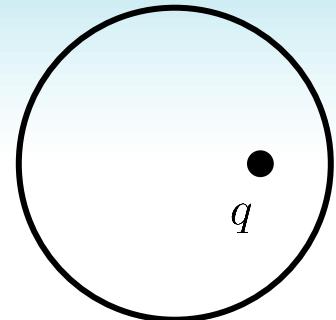


Input + Output



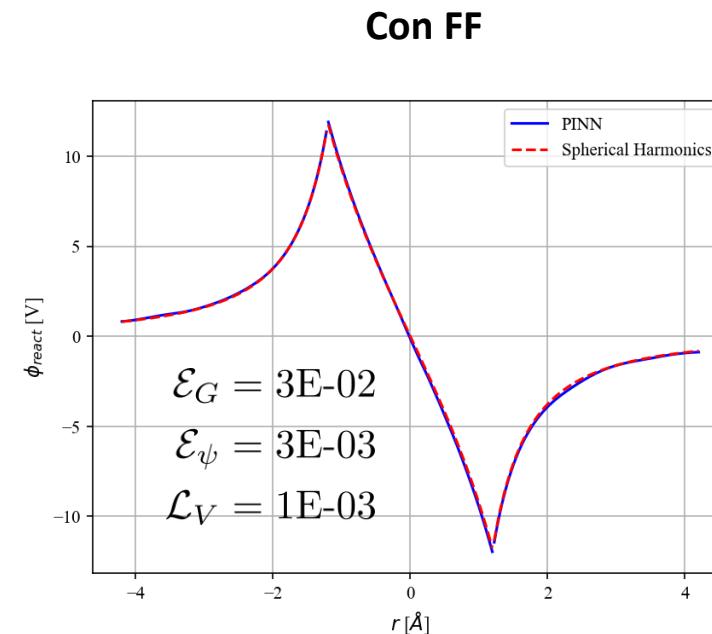
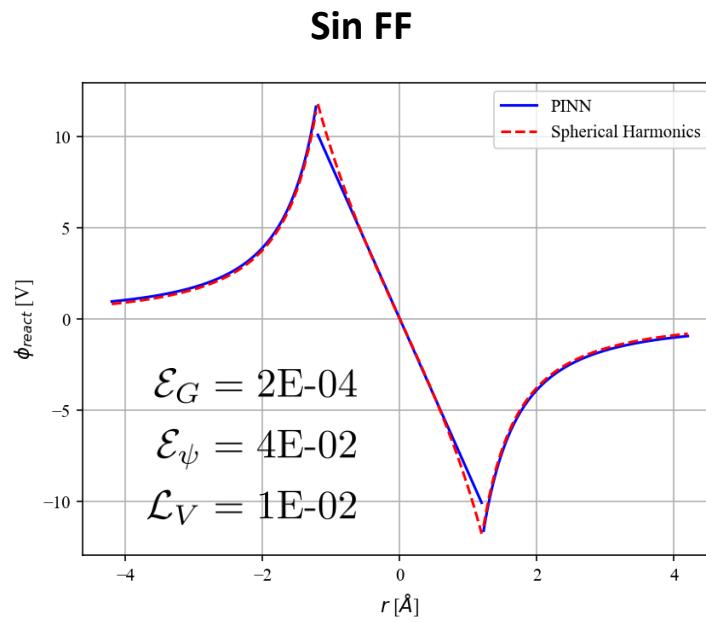
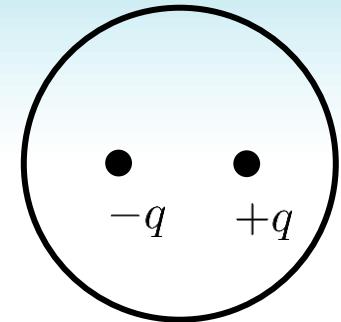
Carga Descentralizada

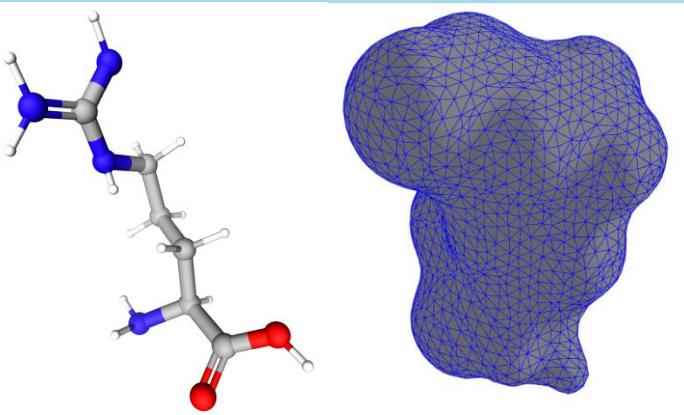
- Solución analítica disponible (armónicos esféricos).
- Capa de Fourier es esencial para replicar la curvatura.



Cargas de signos opuestos

- Solución analítica disponible (armónicos esféricos).
- Capa de Fourier es esencial para replicar la curvatura.



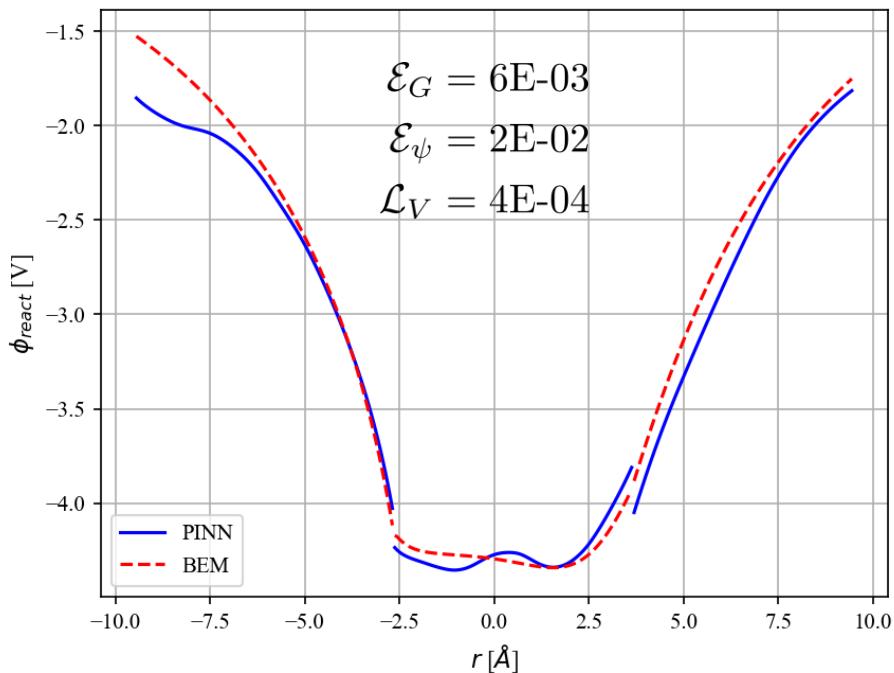


Arginina

- Molécula con 27 cargas.
- Se nota que no es necesario utilizar datos conocidos.
- Influyen en la minimización de los residuales.

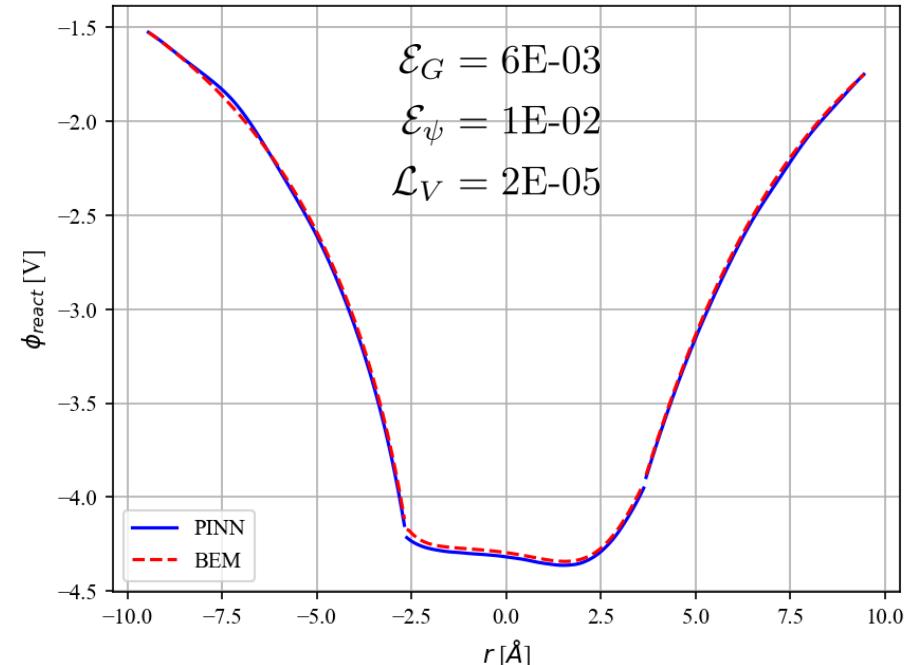
$$\mathcal{L}_{data}(\theta^2; \mathcal{S}_{data}) = \frac{1}{N_{data}} \sum_{\mathbf{x}_i \in \mathcal{S}_{data}} \left[\phi_\theta^2(\mathbf{x}_i) - \phi^\dagger(\mathbf{x}_i) \right]^2$$

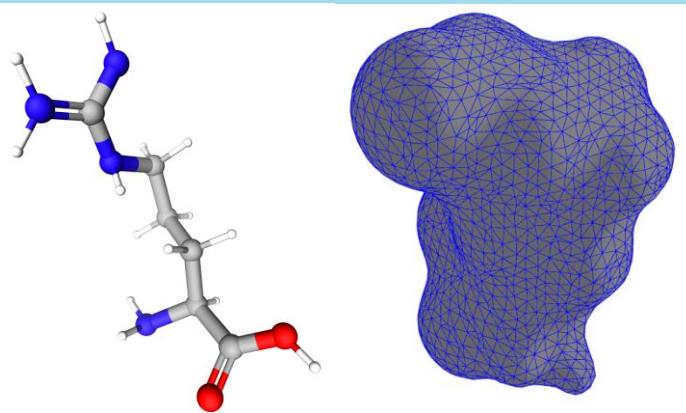
Uso de datos conocidos



Mejora

Sin datos conocidos

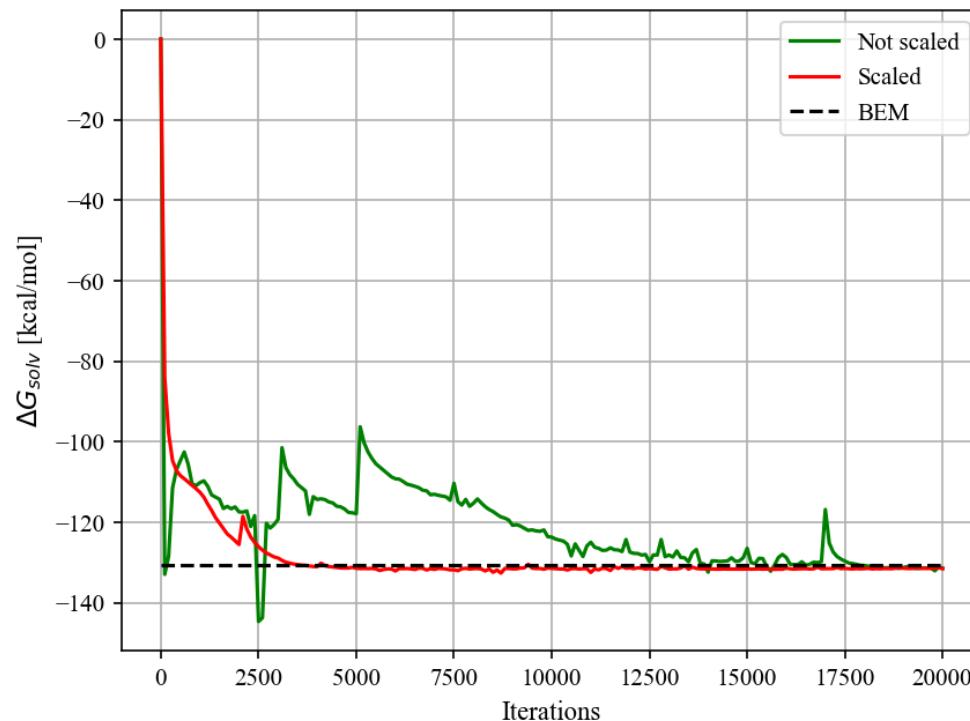




Arginina

¿Cómo afecta el **escalamiento** de la red neuronal?

- Se nota la necesidad del escalamiento.
- Rápida convergencia de la energía de solvatación.



$$\mathcal{E}_G = 6\text{E-}03$$

$$\mathcal{E}_\psi = 1\text{E-}02$$

$$\mathcal{L}_V = 2\text{E-}05$$



$$\mathcal{E}_G = 6\text{E-}03$$

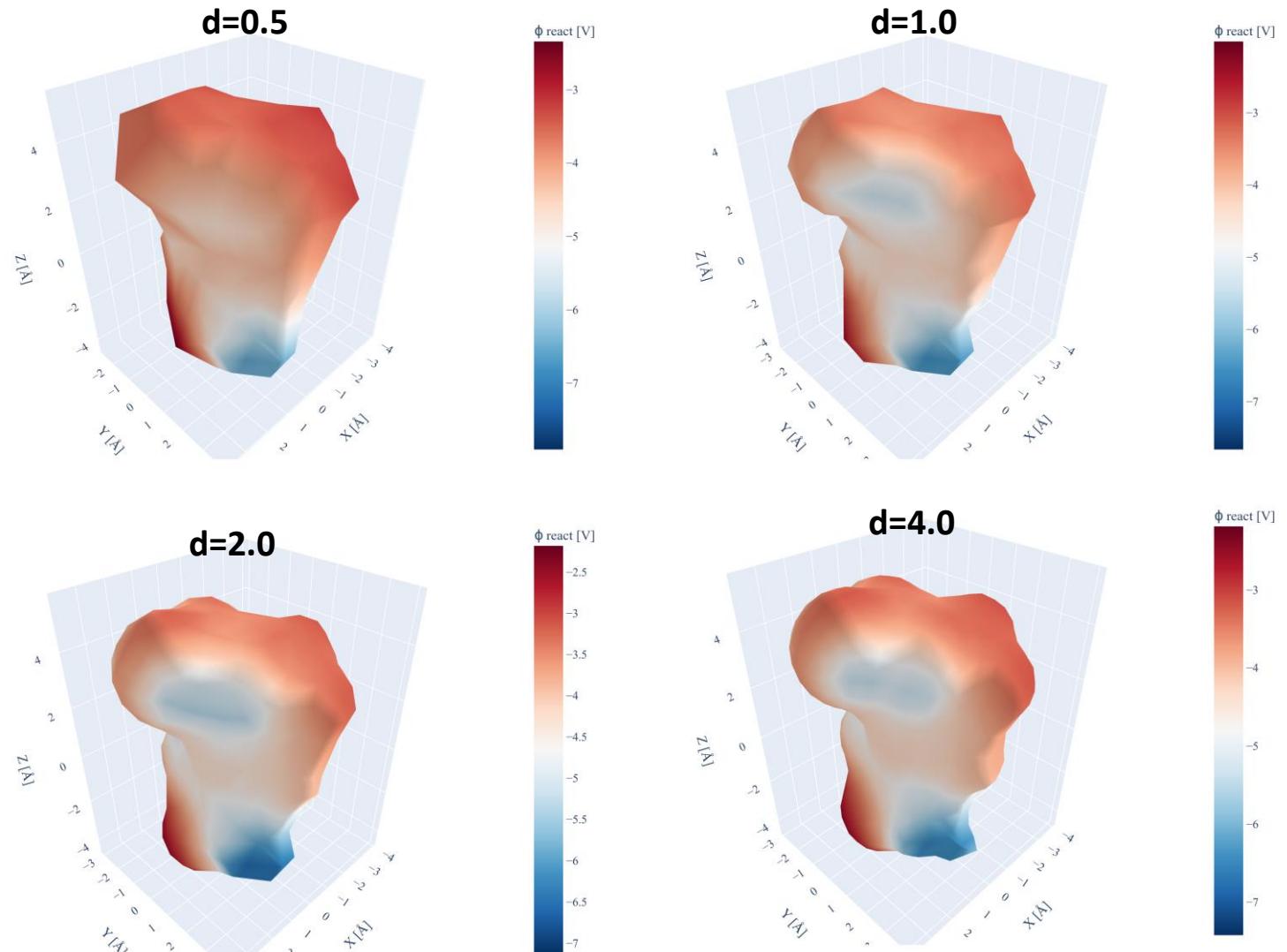
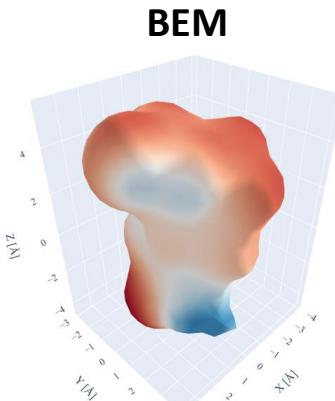
$$\mathcal{E}_\psi = 4\text{E-}04$$

$$\mathcal{L}_V = 6\text{E-}06$$

Estudio de malla

Convergencia respecto a la malla superficial utilizada:

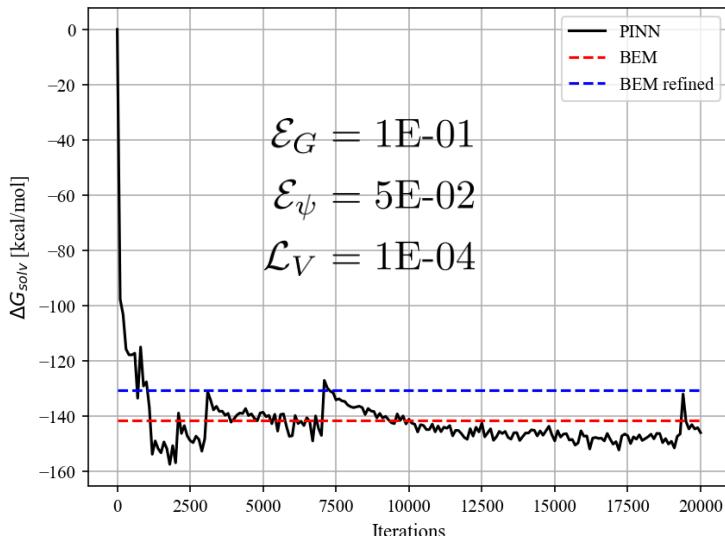
- 4 densidades de malla distinta ($0.5, 1, 2, 4$).
- Geometría no es bien representada con las mallas gruesas.
- Se ven comportamientos similares.



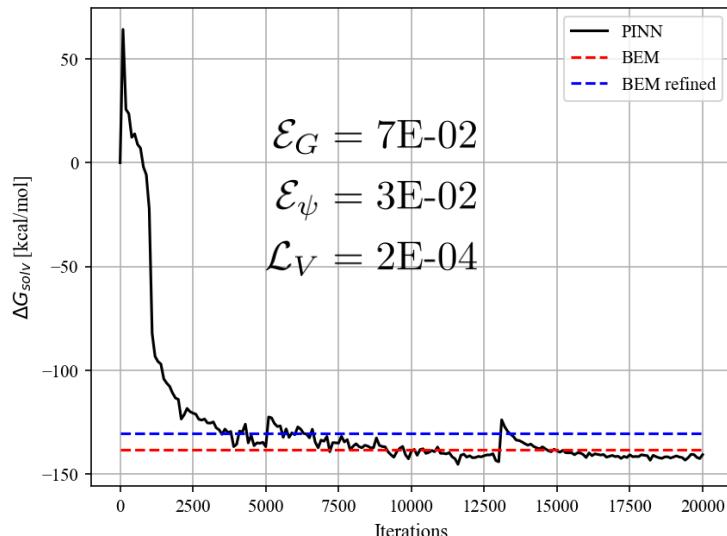
Argentina

- Mallas gruesas no convergen a la solución de BEM.
- Refinando la malla, PINNs se aproxima a BEM.

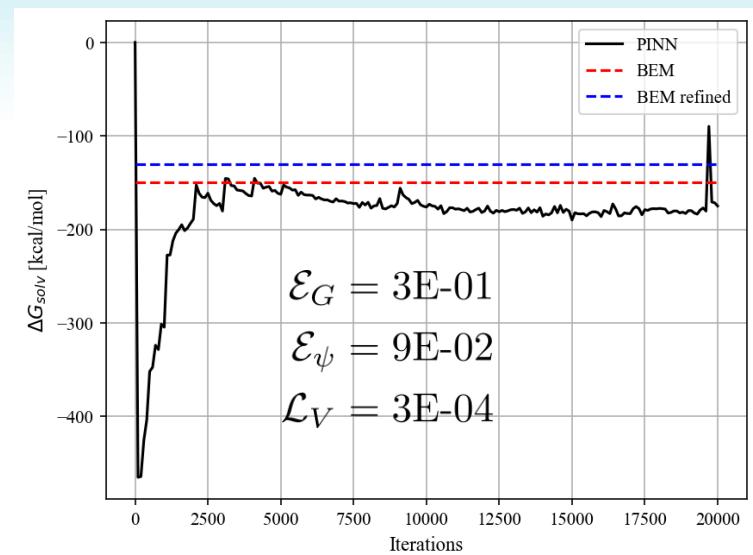
d=1.0



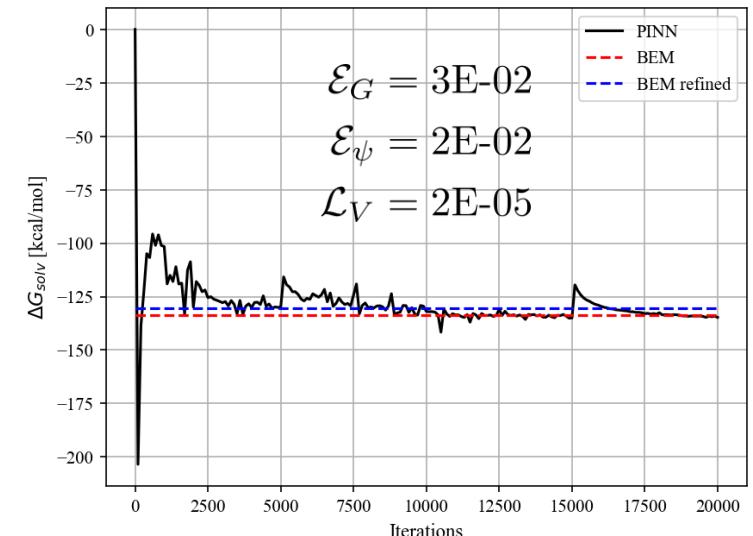
d=2.0



d=0.5



d=4.0

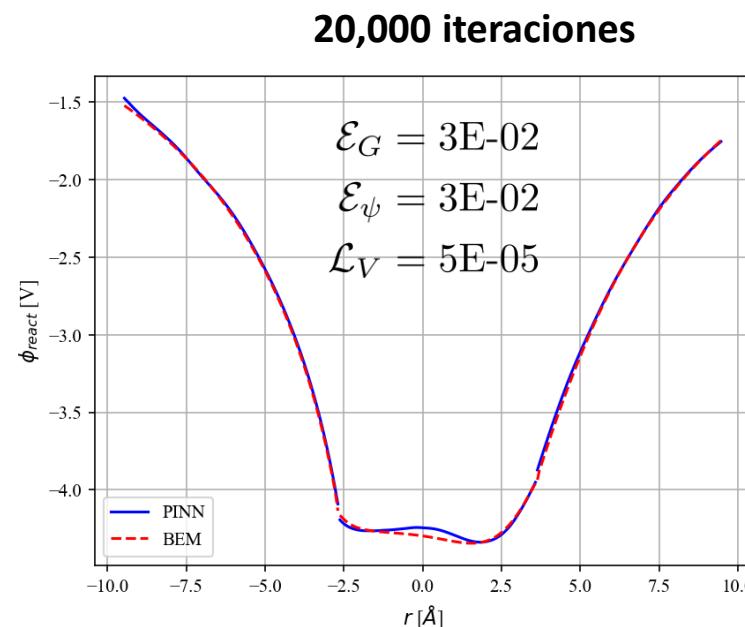


Arginina

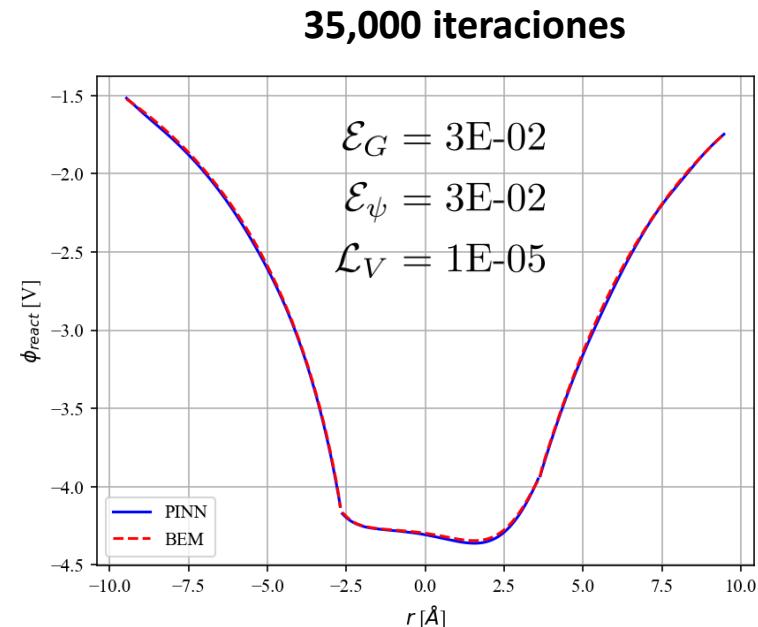
PINNs permite utilizar una menor cantidad de puntos en las mallas volumétricas.

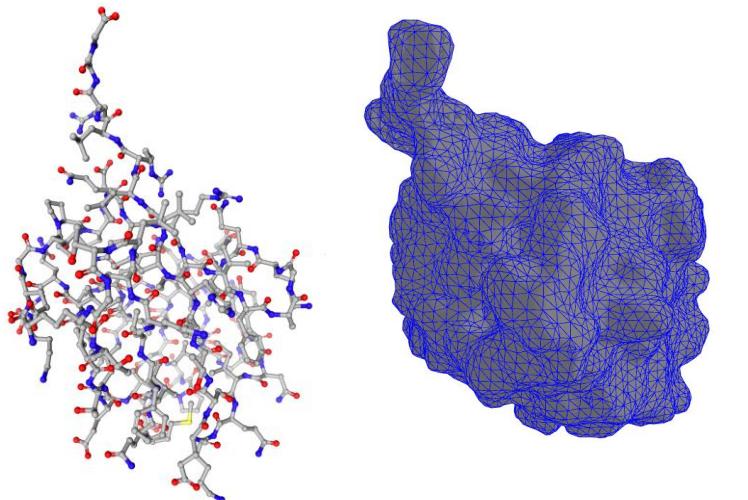
- Se debe mantener fija la malla superficial (lo más fina posible).
- Se debe realizar un mayor número de iteraciones.

Ejemplo: Muestreo del 60% de ambas mallas volumétricas.



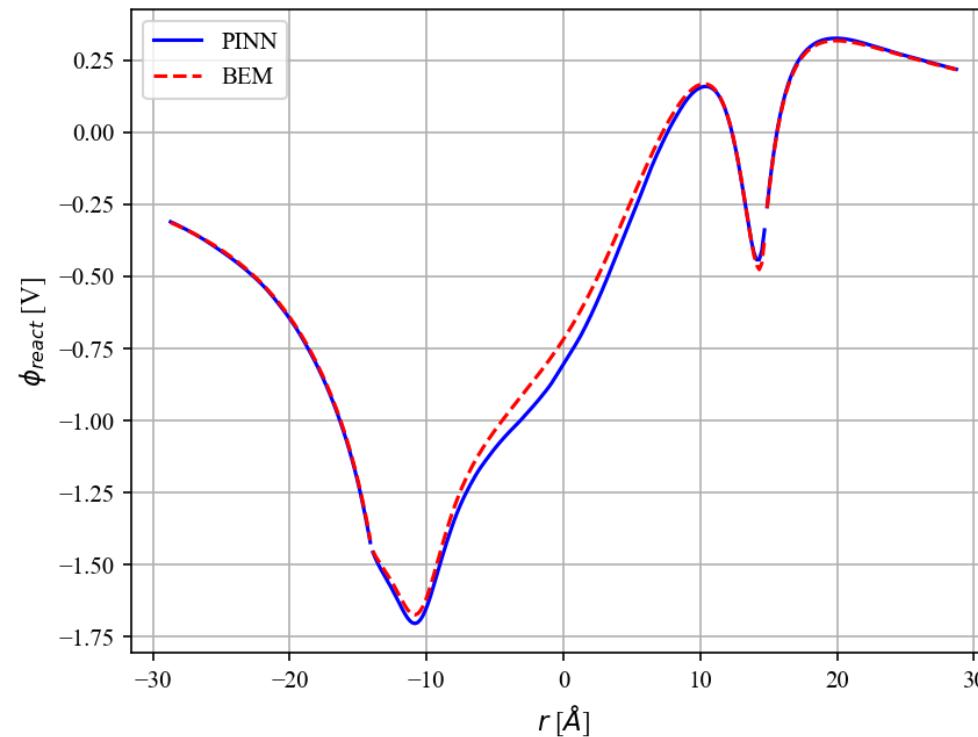
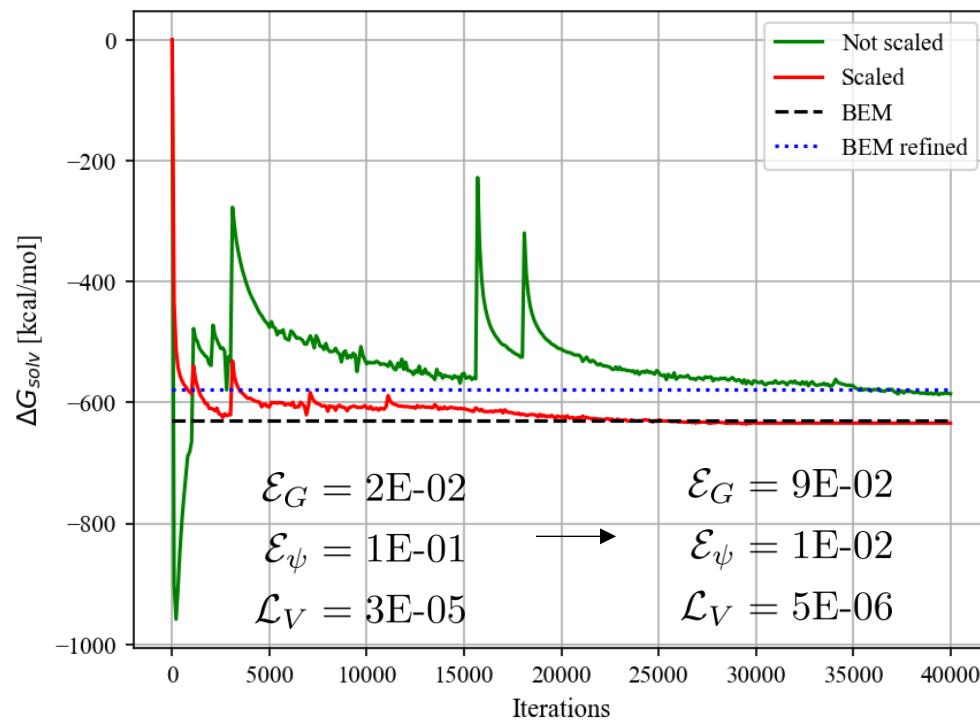
→
Mejora





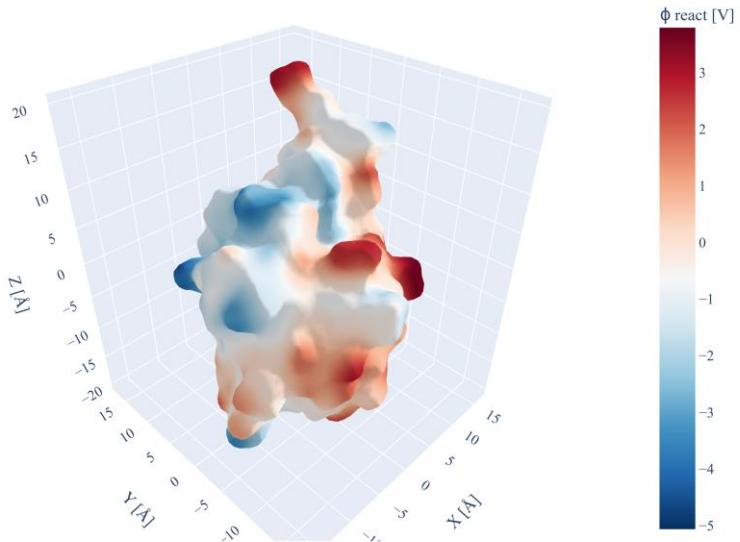
Ubicuitina

- Molécula con 1200 cargas aprox.
- Se visualiza la necesidad del escalamiento (moléculas grandes).
- Leves desviaciones en el potencial.

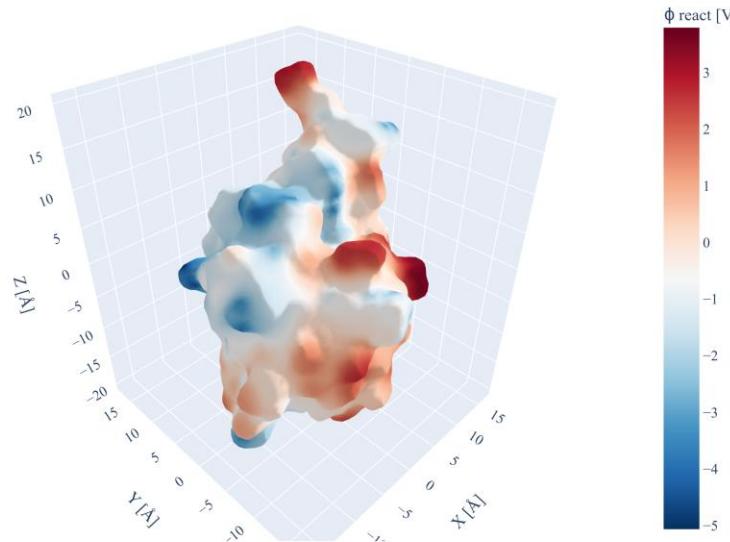


Ubicuitina

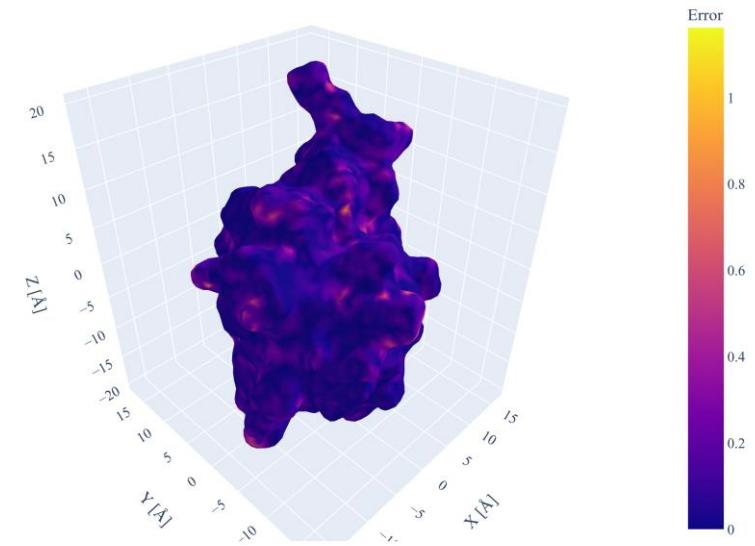
PINN



BEM



Error absoluto



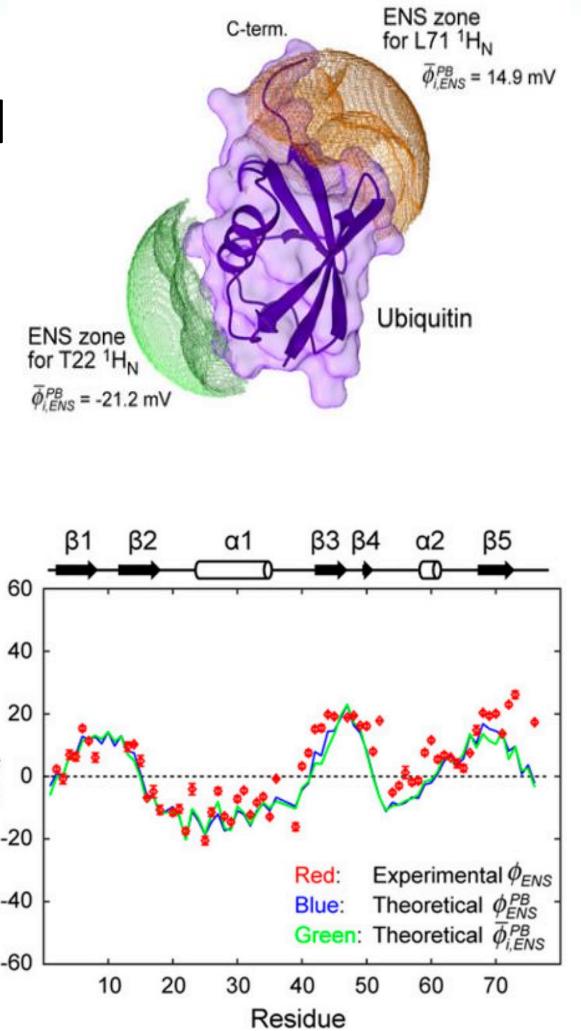
Mediciones Experimentales ϕ_{ENS}

Resultados para el “effective near surface potential” experimental
Medido con NMR (Resonancia magnética nuclear).

- Para cada átomo de hidrógeno H :

$$\phi_{ENS}(\mathbf{r}_H) = \frac{-k_b T}{2q_e} \ln \left(\frac{\int_{\Omega^*} |\mathbf{r} - \mathbf{r}_h|^{-6} e^{-\frac{q_e \phi(\mathbf{r})}{k_b T}} dV}{\int_{\Omega^*} |\mathbf{r} - \mathbf{r}_h|^{-6} e^{\frac{q_e \phi(\mathbf{r})}{k_b T}} dV} \right)$$

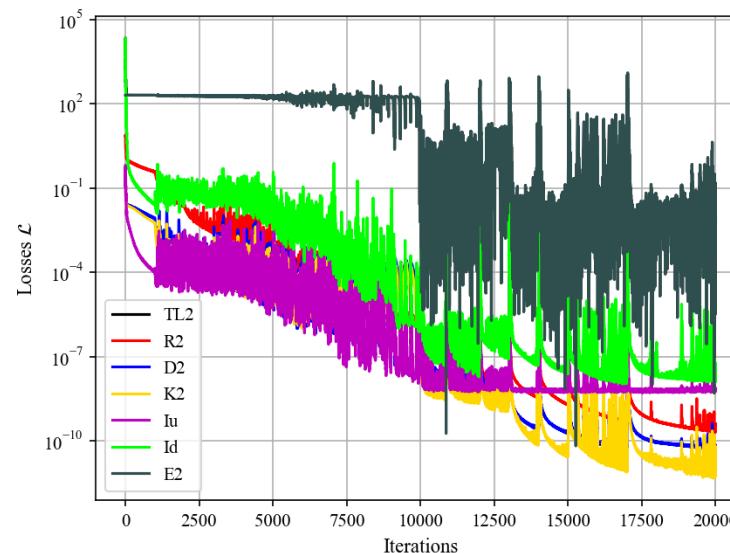
$$\mathcal{L}_E(\boldsymbol{\theta}^2; \mathcal{S}_H) = \frac{1}{N_H} \sum_{\mathbf{x}_h \in \mathcal{S}_H} \left[\phi_{ENS,\theta}(\mathbf{x}_h) - \phi_{ENS}^\dagger(\mathbf{x}_h) \right]^2$$



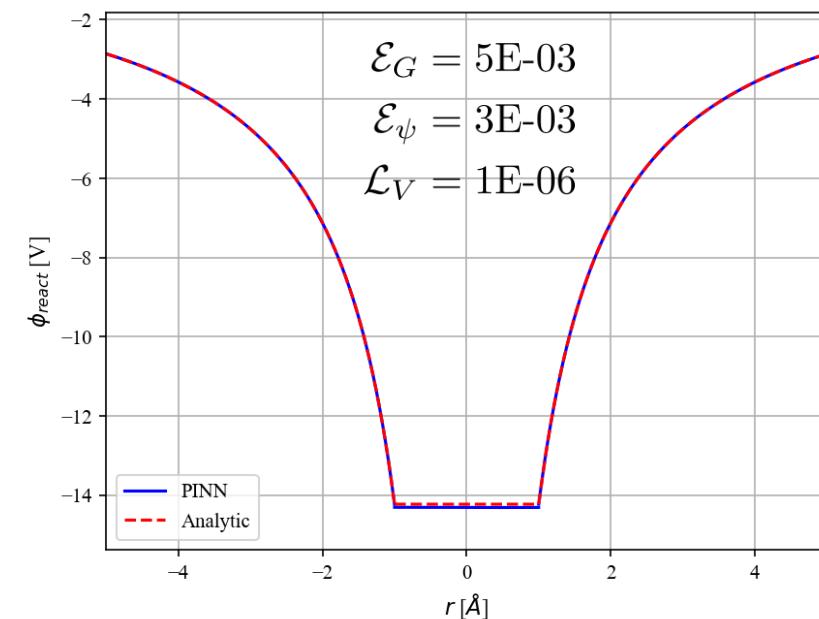
Mediciones experimentales

Ion de Born:

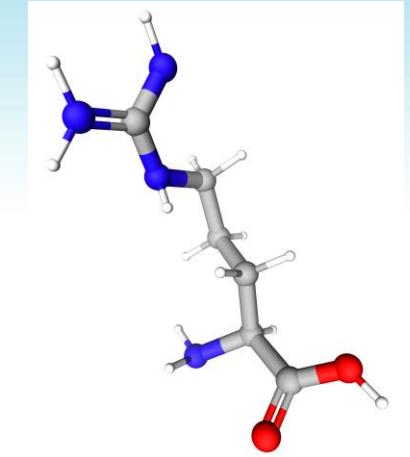
- Exponenciales se aproximan con polinomios de Taylor.
- Comportamiento caótico del loss experimental.
- No se evidencia problema en el potencial y en la energía de solvatación.



$$\phi_{ENS}(\mathbf{r}_H) = \frac{-k_b T}{2q_e} \ln \left(\frac{\int_{\Omega^*} |\mathbf{r} - \mathbf{r}_h|^{-6} e^{-\frac{q_e \phi(\mathbf{r})}{k_b T}} dV}{\int_{\Omega^*} |\mathbf{r} - \mathbf{r}_h|^{-6} e^{\frac{q_e \phi(\mathbf{r})}{k_b T}} dV} \right)$$

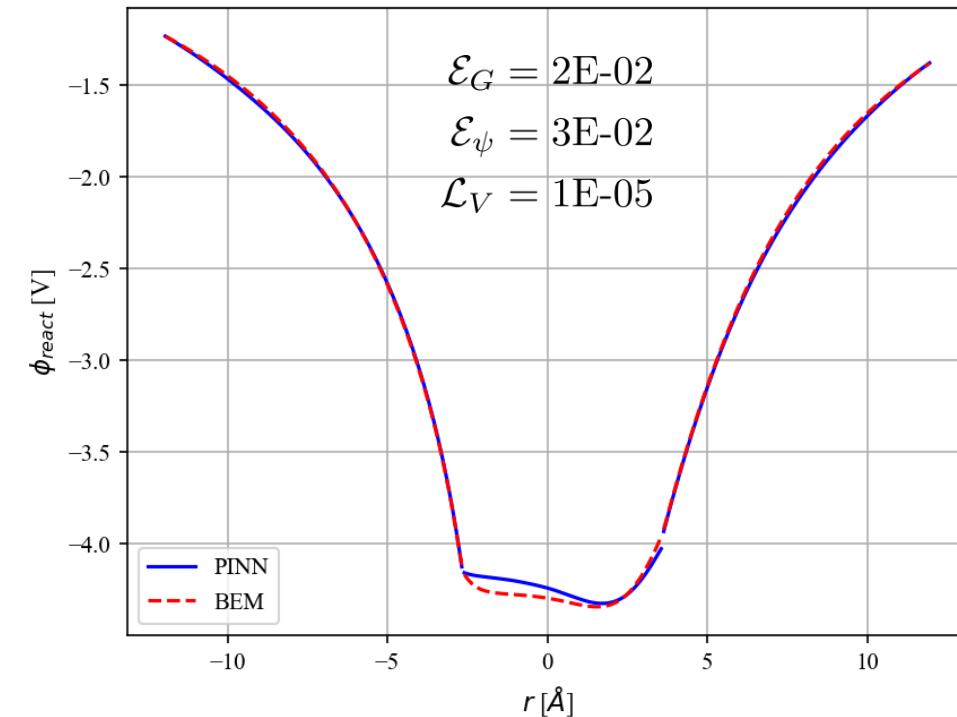
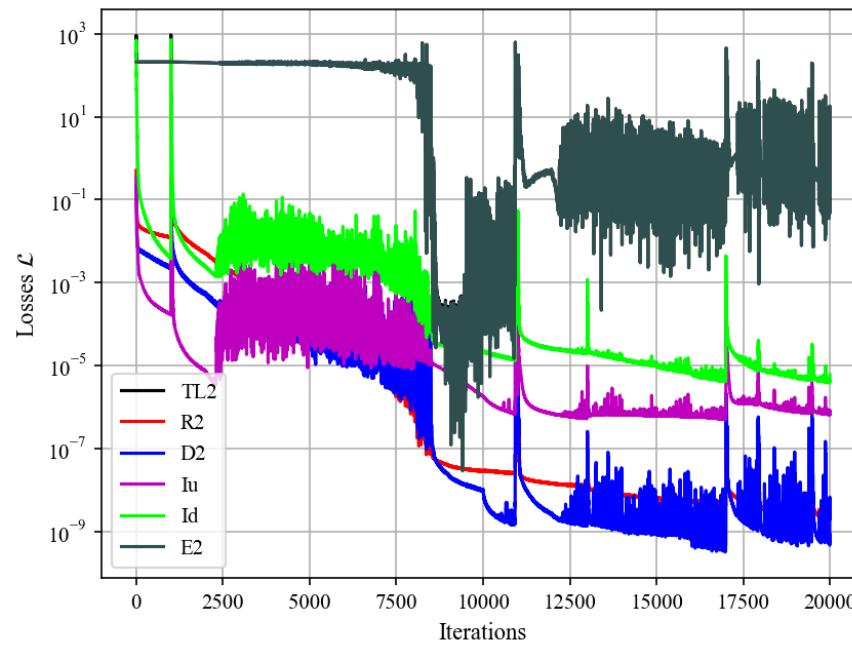


Mediciones experimentales

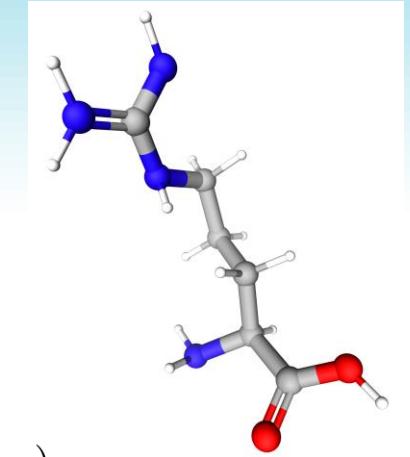


Arginina:

- Comportamiento similar en el loss experimental.
- Curva del potencial evidencia desviaciones.



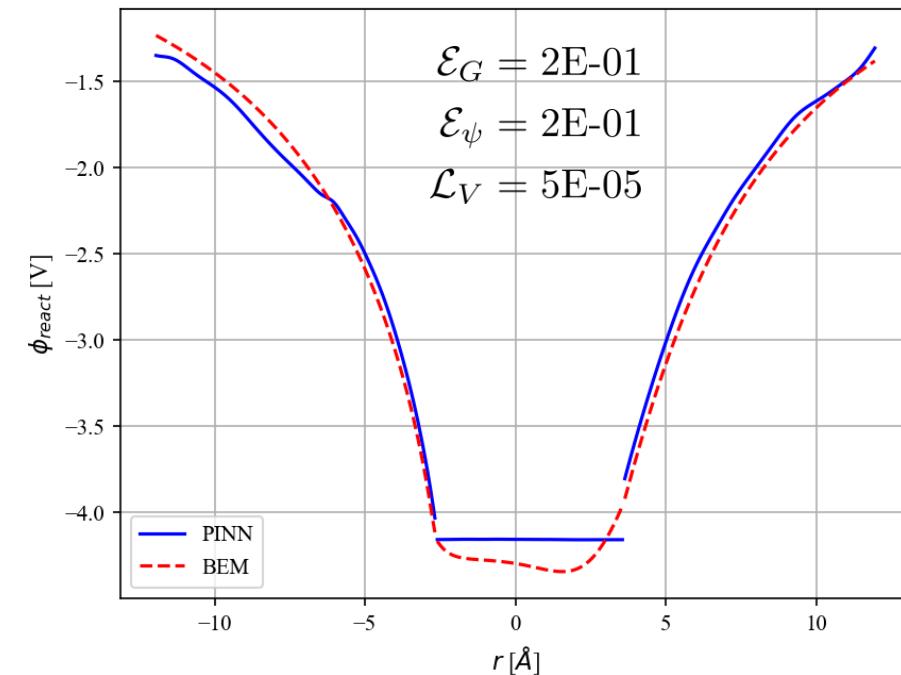
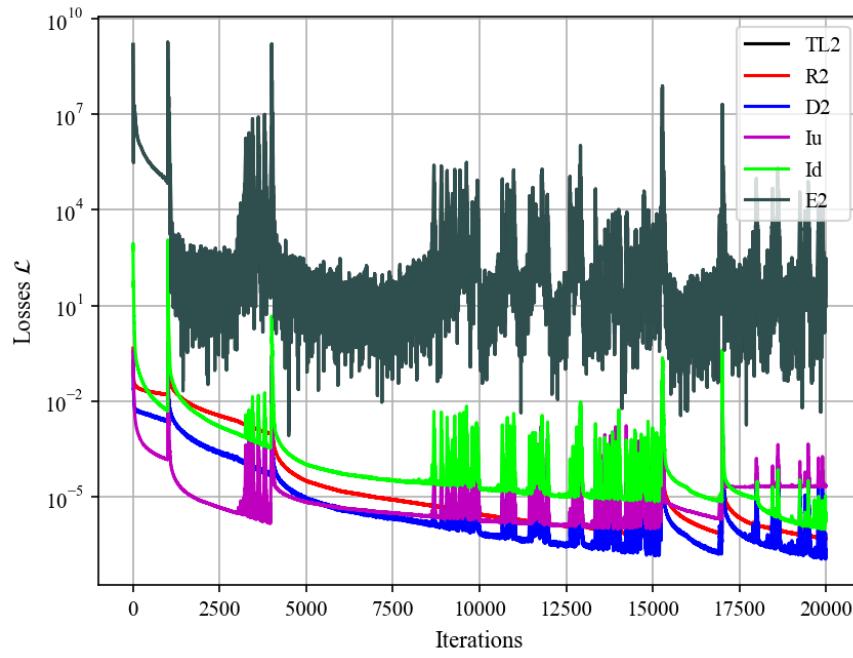
Mediciones experimentales

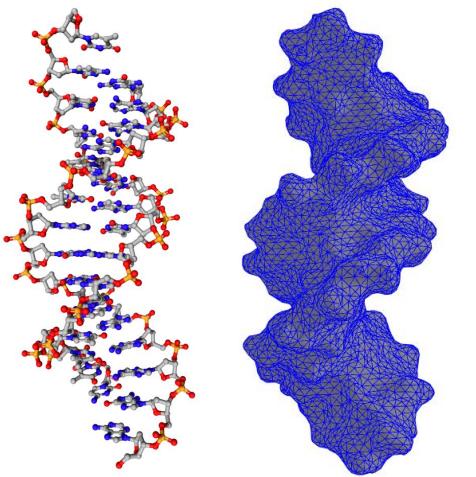


Arginina:

- Definición alternativa de ϕ_{ENS} sin exponenciales..
- Empeora los resultados anteriores

$$\phi_{ENS,\theta}(\mathbf{x}_H) = \frac{1}{N_{\Omega^*}} \sum_{\mathbf{x}_i \in \mathcal{S}_{\Omega^*}} \phi_\theta(\mathbf{x}_i)$$

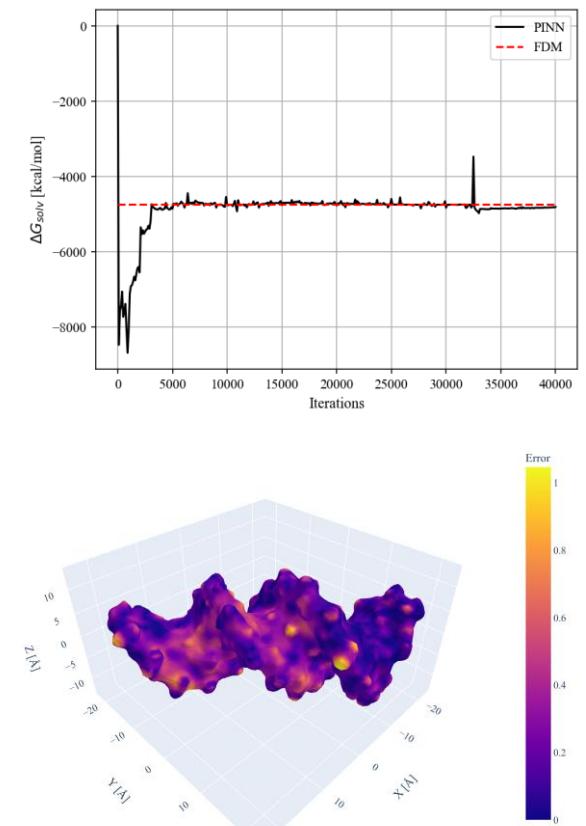
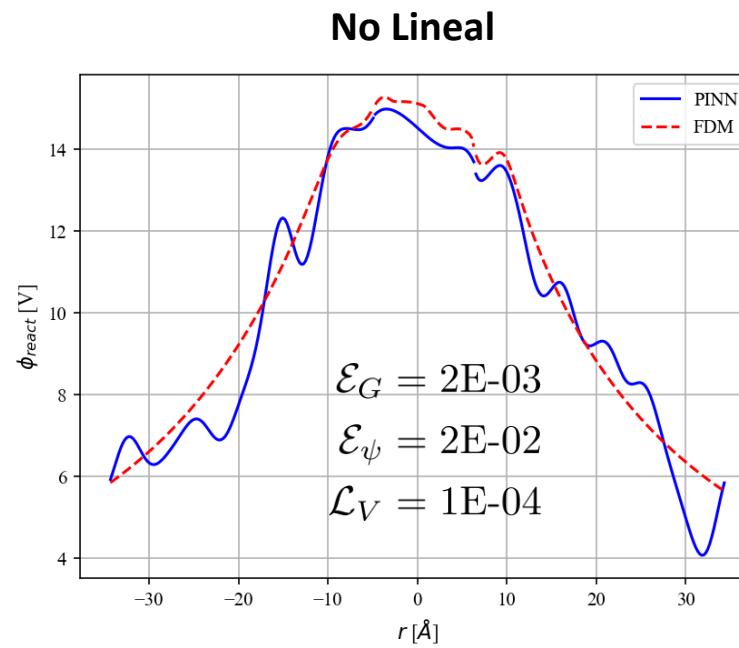
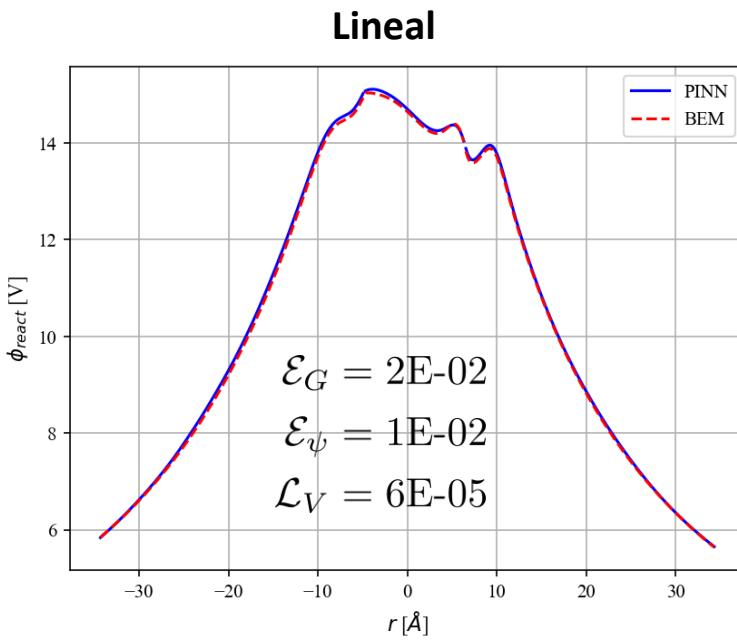




¿PB no lineal?

$$\nabla^2 \phi_2 = \frac{2c^\infty q_e}{\epsilon_2} \sinh\left(\frac{\phi_2 q_e}{k_b T}\right)$$

- Se prueba para el ADN, 950 cargas (molécula altamente cargada).
- Se debe aproximar el seno hiperbólico por polinomios de Taylor.
- Aparecen oscilaciones en la zona del solvente.



Efecto de la adimensionalización

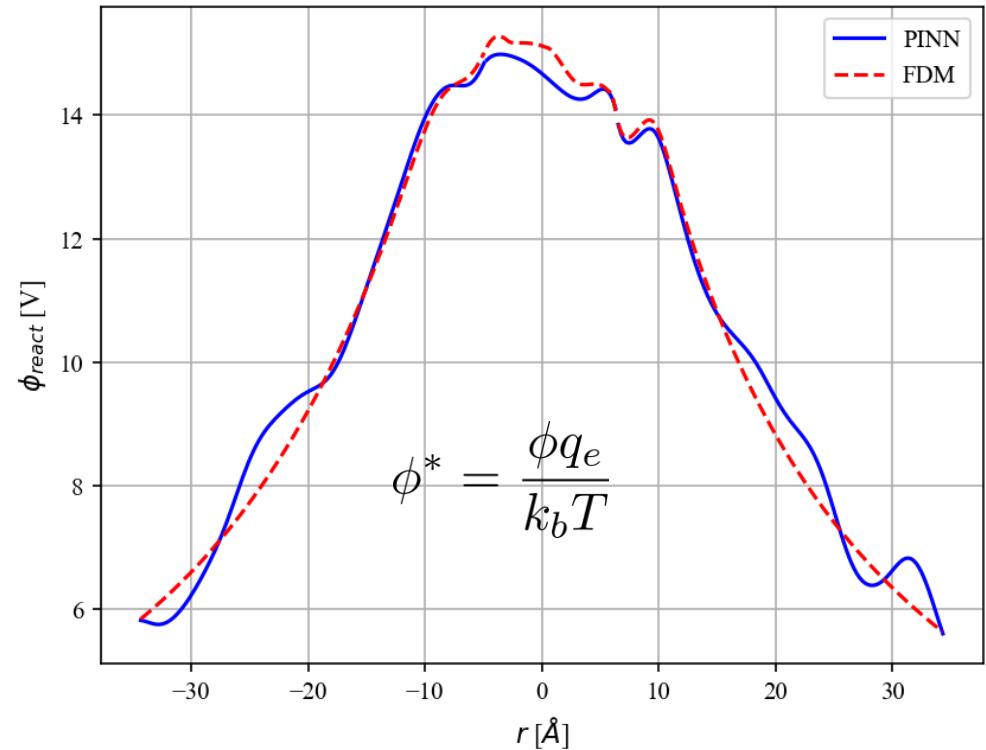
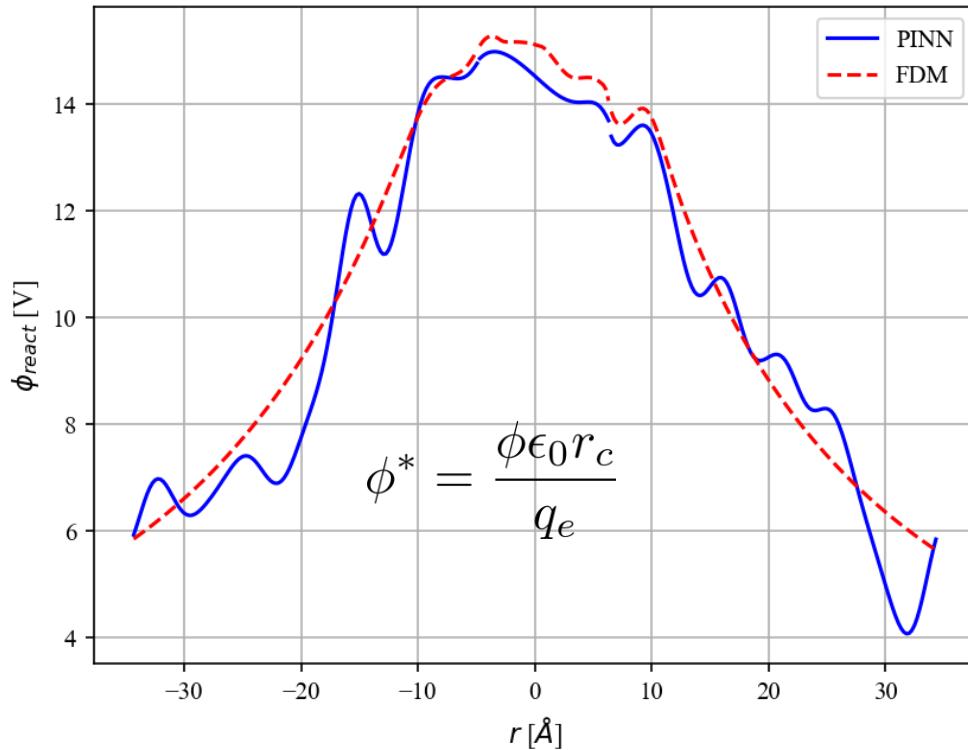
$$\phi^* = \frac{\phi \epsilon_0 r_c}{q_e}$$

$$\phi^* = \frac{\phi q_e}{k_b T}$$

$$-\nabla^{*2} \phi^* + \kappa^{*2} \beta^* \sinh(\phi^*/\beta^*) = \frac{1}{\epsilon_1^*} \sum_k q_k^* \delta^*(\mathbf{r}^* - \mathbf{r}_k^*)$$

$$\beta^* = T \epsilon_0 r_c k_b / q_e^2$$

$$-\nabla^{*2} \phi^* + \kappa^{*2} \sinh(\phi^*) = \frac{1}{\epsilon_1^* \beta^{*-1}} \sum_k q_k^* \delta^*(\mathbf{r}^* - \mathbf{r}_k^*)$$



XPPBE

- Código se implementó como librería en Python: <https://github.com/MartinAchondo/XPPBE>

XPPBE: PINN Solver for 3D Poisson-Boltzmann Equation

XPPBE version 1.0.0 build passing python >=3.9,<3.10

Physics-Informed Neural Network solver for the Poisson-Boltzmann equation applied to real macromolecules in polarizable media.

$$\nabla^2 \phi_1 = -\frac{1}{\epsilon_1} \sum_k q_k \delta(x_k) \quad x \in \Omega_1$$
$$\nabla^2 \phi_2 = \kappa^2 \phi_2 \quad x \in \Omega_2$$

➤ Código basado en Tensorflow

- Uso sencillo:

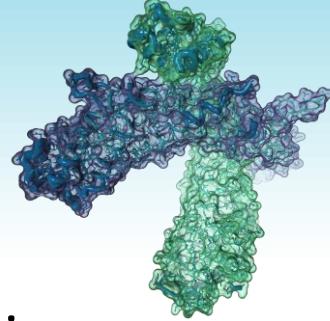
```
from xppbe import Simulation
simulation = Simulation(yaml_path,molecule_dir)
simulation.create_simulation()
simulation.adapt_model()
simulation.solve_model()
simulation.postprocessing(run_all=True)
```



Se definen 2 archivos:
- Archivo .pqr (info. molecular).
- Archivo .yaml (parámetros para el solver).

- Contribución comunidad científica.

Conclusiones



- PINNs puede resolver la ecuación de Poisson-Boltzmann en distintos escenarios.
- La **energía de solvatación** es un indicador bastante robusto (llegando a 1E-03).

Es esencial utilizar:

- Muestreo aleatorio uniforme.
- Capa de Fourier.
- Algoritmo de ponderación.
- Escalamiento de la red neuronal.

Desafíos:

- Incluir mediciones experimentales (losses adicionales integrales).
- Ecuación de PB no lineal.



El software de **XPPBE** creado permite utilizar PINNs de manera sencilla, dando una nueva herramienta a la comunidad científica. —————> Publicación en JCP.

Colaboradores



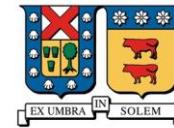
Christopher Cooper

- PhD (Boston University)
Mechanical Engineering.
- Profesor en USM.



Jehanzeb Chaudhry

- PhD (University of Illinois)
Computer Science.
- Profesor en UNM.



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Numerical
modeling of
molecular
electrostatics and
solvation.

Referencias

- 1) B. Roux and T. Simonson, “Implicit solvent models,” *Biophysical chemistry*, vol. 78, no. 1-2, pp. 1–20, 1999.
- 2) Baker, N. A. (2004). Poisson--Boltzmann methods for biomolecular electrostatics. *Methods in enzymology*, 383, 94--118. Elsevier.
- 3) Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- 4) G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.
- 5) M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.
- 6) S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, “Scientific machine learning through physics-informed neural networks: Where we are and what’s next,” *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, 2022.
- 7) S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, “An Expert’s Guide to Training Physicsinformed Neural Networks,” *arXiv preprint arXiv:2308.08468*, 2023.
- 8) S. Wu, A. Zhu, Y. Tang, and B. Lu, “Convergence of physics-informed neural networks applied to linear second-order elliptic interface problems,” *arXiv preprint arXiv:2203.03407*, 2022.
- 9) X. Jiang, Z. Wang, W. Bao, and Y. Xu, “Generalization of PINNs for elliptic interface problems,” *Applied Mathematics Letters*, p. 109175, 2024.
- 10) H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, “The protein data bank,” *Nucleic acids research*, vol. 28, no. 1, pp. 235–242, 2000.

Aprendizaje Profundo para la Electrostática de Macromoléculas: Resolviendo la Ecuación de Poisson-Boltzmann a partir de Redes Neuronales Informadas por la Física

Martín Achondo Mercado
Universidad Técnica Federico Santa María, Chile

21 de Agosto, 2024