

UNIVERSIDAD DE CÓRDOBA

Facultad de Ciencias

Grado de Física

Trabajo Fin de Grado

Introducción a la cromodinámica cuántica en el retículo

Código del TFG: **FS23-35-FSC**

Tipo de TFG: **Trabajo teórico-práctico general**

Autor: Amelia Gómez Barco



07/06/2024

Agradecimientos

Este trabajo simboliza el fin de mi etapa universitaria, es por eso que quiero dar mis agradecimientos a aquellos que me han acompañado en este camino.

En primer lugar me gustaría agradecer la labor de mi tutor: José Manuel Alcaraz Pelegrina, puesto que sin él este trabajo no habría sido posible. Además de todos los años de enseñanza que me ha dado.

Agradecer a mi familia por darme la oportunidad de estudiar la carrera que siempre quise, en especial a mi madre, por todas las horas de llamadas telefónicas escuchando mis quejas.

Por último, agradecer también a mis amigos, tanto de la carrera como de fuera de esta, pues sin su apoyo no podría haber recorrido este camino. Muchas gracias por ser el pilar de mi vida.

Índice general

Índice general	3
Índice de figuras	5
Índice de tablas	6
Resumen. Palabras clave	7
Abstract. Keywords	8
1. Introducción	9
1.1. Integrales de camino	10
1.1.1. Lagrangiana del sistema	13
1.2. Cromodinámica cuántica en el retículo	15
1.2.1. Construcción del retículo.	16
1.2.2. Lagrangiana de Wilson	17
2. Objetivos	23
3. Metodología	24
3.1. Integrales de camino	25
3.1.1. Propagador cuántico	25
3.1.2. Variación de energía	26
3.2. Lattice QCD	28
3.2.1. Monte Carlo para gluones	29
3.2.2. Potencial de interacción quark-antiquark	30
4. Resultados y discusión	34
4.1. Integrales de camino	35
4.1.1. Propagador cuántico	35
4.1.2. Variación de energía	36
4.2. Lattice QCD	38

4.2.1. Monte Carlo para gluones	38
4.2.2. Potencial de interacción quark-antiquark	38
Conclusiones	40
Conclusions	41
Bibliografía	42
Anexo: Definición de funciones en Python	44

Índice de figuras

1.1. Red de dos dimensiones con condiciones de contorno.	16
1.2. Camino arbitrario en una red (G. Lepage, <i>Lattice QCD for novice</i> , página 28)	18
3.1. Diagrama de flujo de un programa en Python para el cálculo de la variación de energía para una acción S dada.	27
3.2. Diagrama de flujo de un programa en Python para el cómputo de un Wilson loop de lados $a \times a$ y $a \times 2a$	30
3.3. Diagrama de flujo de un programa en Python para calcular el potencial de interacción entre un quark y un antiquark estáticos.	32
4.1. Propagador de un oscilador armónico para $T=4$, usando <i>vegas</i> [17] (cruces) para la integración numérica, evaluando una integral de camino en 8 dimensiones. Se ha computado también el resultado exacto (línea), el cual se ha aproximado por el cuadrado de la función de onda del estado fundamental, multiplicado por $e^{-E_0 T}$	35
4.2. Energía del primer estado excitado respecto el estado fundamental 1.17 frente a t para un oscilador armónico. Los parámetros en uso son: $N = 20$, $a = 1/2$, $N_{cf} = 10000$, $N_{cor} = 20$ y $\epsilon = 1,4$	36
4.3. Energía del primer estado excitado respecto el estado fundamental 1.17 frente a t para un oscilador anarmónico. Los parámetros son: $N = 20$, $a = 1/2$, $N_{cf} = 10000$, $N_{cor} = 20$ y $\epsilon = 1,4$	37

4.4. Potencial de interacción entre un quark y un antiquark, ambos estáticos y separados una distancia r . Los valores de los parámetros de la expresión 1.43 se han obtenido mediante una librería de Python <i>scipy</i> , concretamente la función <i>curve.fit</i> [18]. Dichos valores son $\sigma = (0,71 \pm 0,01)GeV/fm$; $\gamma = (0,71 \pm 0,03)GeV \cdot fm$ y $\mu = (1,21 \pm 0,05)GeV$	39
--	----

Índice de tablas

4.1. Valores del Wilson loop para una acción simple, y una mejorada. Los parámetros fueron $\beta = 5, 5$, $a = 0,25\text{fm}$, $N=8$, $N_{cor} = 50$, $N_{cf} = 10$. Se han aportado los resultados obtenidos en [1] para comparación.	38
--	----

Resumen

En este trabajo se realiza una introducción a la cromodinámica cuántica en el retículo usando la formulación mediante integrales de camino. Se han realizado simulaciones para obtener algunas magnitudes en sistemas simples como el oscilador armónico. Entre estas magnitudes se encuentra el propagador cuántico, la diferencia de energía del oscilador armónico entre su estado fundamental y excitado, así como el potencial de interacción entre un quark y un antiquark estáticos, separados una distancia r . El potencial obtenido se ajusta bien a un potencial que combina un término de Coulomb más un término lineal. Dichas simulaciones se han implementado en programas escritos en Python 3.11.7.

Palabras clave: Retículo; cromodinámica cuántica; Python

Abstract

In this work, an introduction to lattice quantum chromodynamics is provided using the path integral formulation. Simulations have been performed to obtain some quantities in simple systems such as the harmonic oscillator. Among these quantities are the quantum propagator, the energy difference of the harmonic oscillator between its ground and excited states, as well as the interaction potential between a static quark and antiquark separated by a distance r . The obtained potential fits well with a potential that combines a Coulomb term and a linear term. These simulations have been implemented in programs written in Python 3.11.7.

Keywords: Lattice; quantum chromodynamics; Python

Introducción

En este trabajo se realiza una introducción a la cromodinámica cuántica (QCD¹), viendo los principales pasos a seguir en pos de conseguir una primera simulación de la misma en el retículo (de ahora en adelante, LQCD²). QCD es una teoría que describe la interacción entre las partículas elementales, los quarks. Esta teoría se propuso a principios de los años 70 con el objetivo de entender la física de los hadrones³. Una de las principales formas de trabajar con QCD es lo que se conoce como LQCD, que permite hacer cálculos de manera no perturbativa.

En este capítulo se realiza una primera aproximación teórica, viendo como se puede formular QCD en función de integrales de camino. Para ello, se dará una introducción a qué son las integrales de camino, además se buscará la expresión de la Lagrangiana de QCD, así como su representación en LQCD. Se mostrará la estructura del retículo, desde su creación hasta el cálculo de algunas magnitudes físicas que se pueden obtener mediante LQCD. En el capítulo 3 se definen una serie de algoritmos, empezando desde algunos más básicos, hasta concluir en una simulación de un potencial de interacción entre un quark y un antiquark. En este trabajo, los algoritmos se implementan en Python 3.11.7. En el capítulo 4, se muestran resultados para el propagador cuántico del oscilador armónico, así como el cálculo de su energía. También se muestra la forma del potencial de interacción

¹QCD viene de las siglas en inglés de Quantum-Chromodynamics.

²LQCD viene de las siglas en inglés de la palabra Lattice Quantum-Chromodynamics, que significa cromodinámica cuántica en el retículo

³Partícula conformada por tres quarks (barión) o por un quark y un antiquark (mesón).

antes mencionado.

1.1. INTEGRALES DE CAMINO

En la década de 1940 el físico Richard Feynman propuso un formalismo, o más bien interpretación, sobre como abordar la mecánica cuántica. Este formalismo se basaba en hacer uso de integrales de camino en vez de hacer uso de la ecuación de *Schrödinger* para hallar las propiedades de un sistema. Lo formuló cómo se presenta en esta sección.

Feynman dijo que una amplitud de probabilidad tenia que estar asociada con cualquier trayectoria que pudiera seguir una partícula. Dicha amplitud tendría que ser la suma de todas las posibles amplitudes, para todas las posibles trayectorias [9].

Para el caso unidimensional, es decir, para una trayectoria $X(t)$ desde $X(t_0) = x_0$ hasta $X(t) = x$, la amplitud será igual a:

$$g(x, t; x_0, t_0) = e^{\frac{i}{\hbar} S[X(t)]} \quad (1.1)$$

siendo S la acción clásica sobre una trayectoria, que viene dada por:

$$S = \int_{t_0}^t L(X, \dot{X}, t) dt \quad (1.2)$$

La amplitud total desde un punto inicial x_0 en un instante inicial t_0 , hasta un punto x en un instante t es lo que se llama *propagador*, y tiene la expresión:

$$G(x, t; x_0, t_0) = \sum_{X_i} e^{\frac{i}{\hbar} S[X_i(t)]}, \quad \text{para } t > t_0 \quad (1.3)$$

que no es más que un sumatorio para todas las trayectorias X_i de la fórmula 1.1. Esta ecuación se puede reescribir para el caso continuo de la forma:

$$G(x, t; x_0, t_0) = \int \mathbf{D}X(t') e^{\frac{i}{\hbar} S[X(t')]} \quad (1.4)$$

Por tanto, la amplitud total de estar en un punto x en un instante t es la suma ponderada de las amplitudes de cada trayectoria desde todos los puntos iniciales, esto es:

$$\Psi(x, t) = \int dx_0 \Psi(x_0, t_0) G(x, t; x_0, t_0) \quad (1.5)$$

Teniendo un potencial sin derivadas temporales $\phi(x)$, la Lagrangiana para una partícula queda de la siguiente forma:

$$L(x, \dot{x}, t) = \frac{m}{2} \left(\frac{dx}{dt} \right)^2 - \phi(x) \quad (1.6)$$

Un dato a tener en cuenta es que no hay números imaginarios, esto es porque se usan las integrales de camino euclídeas, se cambia $it \rightarrow t$, siendo estas mucho más sencillas de resolver numéricamente. Por tanto queda:

$$-H(x, \tau) = -\left[\frac{m}{2} \left(\frac{dx}{d\tau} \right)^2 + \phi(x) \right] \quad (1.7)$$

siendo H el Hamiltoniano. Quedando la acción de la forma:

$$S(X_i) = i \int_{X_{i0}}^{\tau} H(x, \tau') d\tau' \quad (1.8)$$

dicha integral está definida sobre la trayectoria X_i . Si se toman los trozos de esta trayectoria como líneas, espaciadas de manera uniforme por un parámetro ϵ , queda:

$$\frac{1}{\hbar} \int H d\tau \rightarrow \frac{\epsilon}{\hbar} \sum_0^N \left[\frac{m}{2} \left(\frac{x_{j+1} - x_j}{\epsilon} \right)^2 + \phi(x_j) \right] = \frac{\epsilon}{\hbar} \sum E_j \quad (1.9)$$

consiguiendo asociar una energía E con una trayectoria.

En teoría de campos se utilizan el potencial escalar $\phi(x)$ y el potencial vector $A_\mu(x)$ como coordenadas, siendo x un punto espacio-temporal $x = (t, \vec{x})$. El análogo al estado fundamental será el vacío, y los estados excitados serán estados con varias partículas creadas en el vacío.

Se verá como ejemplo el caso de un oscilador armónico.

Oscilador armónico

En mecánica cuántica se usa el propagador para medir la amplitud de probabilidad de encontrar una partícula en el punto x , en el instante t , si esta ha partido del punto x_0 , en el instante t_0 , teniendo en cuenta todos los posibles caminos que esta haya podido tomar, de una manera ponderada. Partiendo de la expresión 1.4 se llega a:

$$\langle x_f | e^{-\tilde{H}(t_f - t_i)} | x_i \rangle = \int \mathbf{D}x(t) \cdot e^{-S[x]} \quad (1.10)$$

siendo $\int \mathbf{D}x(t)$ la suma para todos los caminos posibles. La acción es la misma que la acción clásica.

$$S[x] \equiv \int_{t_f}^{t_i} dt L(x, \dot{x}) \equiv \int_{t_f}^{t_i} dt \left[\frac{m\dot{x}(t)^2}{2} + V(x(t)) \right] \quad (1.11)$$

con

$$V(x(t)) = \frac{x(t)^2}{2} \quad (1.12)$$

Por tanto, ha de buscarse la forma de resolver el propagador mediante una integral de camino.

Discretizando el problema, las ecuaciones 1.10 y 1.11 quedarían como [1]:

$$\langle x | e^{-\tilde{H}T} | x \rangle = \int_{-\infty}^{\infty} dx_1 \dots dx_{N-1} \cdot e^{-S_{lat}[x]} \quad (1.13)$$

$$S_{lat}[x] \equiv \sum_{j=0}^{N-1} \left[\frac{m}{2a} (x_{j+1} - x_j)^2 + aV(x_j) \right] \quad (1.14)$$

siendo las $x_0 = x_N$ y $a = T/N$. Con ello ya se puede resolver el problema mediante integración numérica. Desarrollando esto numéricamente las expresiones anteriores se transforman en:

$$\langle \langle \Gamma[\phi] \rangle \rangle \equiv \frac{1}{\mathbf{Z}} \int e^{-S[\phi]} \Gamma[\phi] \prod_{x_j \in grid} d\phi(x_j) \quad (1.15)$$

$$\mathbf{Z} \equiv \int e^{-S[\phi]} \prod_{x_j} d\phi(x_j) \quad (1.16)$$

El cambio entre trabajar con el estado fundamental a trabajar también con los estados excitados no es trivial. Se puede obtener la energía de estos estados a partir del cálculo de la siguiente magnitud [1]:

$$\log \frac{G(t)}{G(t+a)} \rightarrow (E_1 - E_0)a \quad (1.17)$$

siendo $G(t)$ el propagador, E_1 y E_0 son los valores de energía del primer estado excitado y del estado fundamental respectivamente.

Se necesita la Lagrangiana del sistema para poder describir dicho sistema en QCD, la cual ha de cumplir distintas características. Estas son que sea local, simétrica con respecto al intercambio de ejes y que sea invariante con el gauge.

1.1.1. LAGRANGIANA DEL SISTEMA La Lagrangiana de QCD viene dada por la siguiente expresión [13]:

$$\mathcal{L} = \sum_q \bar{\psi}_{q,a} (i\gamma^\mu \partial_\mu \delta_{ab} - g_s \gamma^\mu t_{ab}^C \mathcal{A}_\mu^C - m_q \delta_{ab}) \psi_{q,b} - \frac{1}{4} \mathcal{F}_{\mu\nu}^A \mathcal{F}^{A\mu\nu} \quad (1.18)$$

siendo γ^μ las matrices de Dirac, los $\psi_{q,a}$ son espinores de los campos, de sabor q y masa m_q , y con color "a". Hay un total de 3 colores, por lo cual, se encuentran en la representación del grupo de SU(3) de color. La \mathcal{A}_μ^C son los campos generados por los gluones, siendo C el índice que nos indica cual de los 8 generadores se usa.

Las matrices de Dirac son matrices 4×4 convencionales que encuentran especial utilidad en presencia de espinores, ya que se usan para representarlos. Son 4 matrices que tienen la siguiente forma:

$$\begin{aligned} \gamma^0 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}; \gamma^1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \\ \gamma^2 &= \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ -i & 0 & 0 & 0 \end{pmatrix}; \gamma^3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{aligned} \quad (1.19)$$

Los generadores \mathcal{A}_μ^C del espacio SU(3) son también matrices, estas se conocen por el nombre de matrices de *Gell-Mann*. Como antes se mencionó son 8, pues hay un número igual a $n^2 - 1$ generadores en este espacio, siendo $n = 3$. Estas matrices tienen la forma:

$$\begin{aligned}
\lambda^1 &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}; \lambda^2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}; \lambda^3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
\lambda^4 &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}; \lambda^5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix}; \lambda^6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \\
\lambda^7 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}; \lambda^8 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}
\end{aligned} \tag{1.20}$$

que, como se puede ver a simple vista, son análogas a las matrices de Pauli.

El tensor de campo viene dado por la expresión:

$$\mathcal{F}_{\mu\nu} \equiv \partial_\mu A_\nu - \partial_\nu A_\mu + ig[A_\mu, A_\nu] \tag{1.21}$$

En la expresión 1.18 se pueden apreciar 4 términos, la interpretación de cada uno de ellos es la siguiente:

- El primero es el término $\bar{\psi}_{q,a}(i\gamma^\mu\partial_\mu\delta_{ab})\psi_{q,b}$ se puede interpretar como la energía cinética de los quarks.
- El segundo es el término $\bar{\psi}_{q,a}(-g_s\gamma^\mu t_{ab}^C A_\mu^C)\psi_{q,b}$ que surge de la interacción entre quarks y gluones.
- El tercero es el término $\bar{\psi}_{q,a}(-m_q\delta_{ab})\psi_{q,b}$ corresponde a la energía en reposo de los quarks.
- El último es el término $-\frac{1}{4}\mathcal{F}_{\mu\nu}^A\mathcal{F}^{A\mu\nu}$ de interacción entre gluones.

Puesto que este trabajo tiene por objetivo el ser una introducción, se propondrá una primera acción para cromodinámica cuántica que solo tenga en cuenta gluones, esto es, que solo aparezca el cuarto término de la expresión 1.18. Esta acción en el continuo se escribe de la siguiente forma:

$$S = \int d^4x \frac{1}{2} \sum_{\mu,\nu} Tr F_{\mu\nu}^2(x) \tag{1.22}$$

Para obtener resultados de esto se puede hacer uso de una discretización del espacio-tiempo, lo que se conoce como QCD en el retículo, es decir, LQCD antes mencionado. En la siguiente sección se verá como se formula LQCD, desde la creación de la red de puntos, hasta los cálculos correspondientes para obtener resultados.

1.2. CROMODINÁMICA CUÁNTICA EN EL RETÍCULO

La interacción de color es una fuerza fundamental un tanto peculiar, en tanto que surgen 2 fenómenos de especial interés. El primero de ellos es la *Libertad asintótica*, que es una propiedad de algunas partículas, en este ensayo de los quarks, por la cual las interacciones entre ellos se van debilitando conforme la distancia disminuye. El otro fenómeno de interés es el *confinamiento de QCD*, una propiedad por la cual la interacción de color crece conforme aumenta la distancia de separación entre partículas⁴.

Kenneth Wilson formuló lo que hoy conocemos como LQCD. Wilson quería comprender la *Libertad Asintótica* y la propiedad confinante de QCD. Para ello, usó un mallado espacio-temporal, en cuyos nodos se definen los quarks, y en las uniones de estos los gluones. Con esto, consiguió asemejar la teoría cuántica de campos a un sistema estadístico clásico.

Para obtener las propiedades, de manera numérica, de los hadrones se requiere de un cálculo no perturbativo, siendo LQCD una muy buena herramienta para ello. Gracias a este método se han podido determinar varias propiedades como la constante de acoplamiento fuerte, uno de los parámetros fundamentales del modelo estándar, así como las masas de los quarks u otras constantes.

En la aproximación de un retículo, el espacio y el tiempo son discretos. Los nodos, en los que se representan los quarks, están separados por una distancia a llamada *espaciado del retículo*, o comúnmente conocido como *lattice spacing*. La longitud total del retículo

⁴Partículas con carga de color.

se denota por L . En las conexiones entre estos puntos se definen las transiciones entre un estado y otro.

La construcción de dicho retículo no es algo trivial, por lo que se necesitan asentar las bases de este. En el próximo apartado se hace un resumen de como construir una red para resolver cálculos en LQCD. Un análisis más extenso se puede encontrar en el artículo [15].

1.2.1. CONSTRUCCIÓN DEL RETÍCULO. Un retículo es una discretización del espacio y del tiempo, como se comentó anteriormente. Esta discretización no son más que puntos representantes de los ejes, manteniendo el tiempo como un continuo [15]. Estos puntos se definen de la siguiente forma:

$$x_\mu = n_\mu a \quad (1.23)$$

y se les llama puntos del retículo, o en inglés *Lattice sites*.

El parámetro a es el *Lattice spacing* antes mencionado. Este es una magnitud dimensional, generalmente con unidades de fm . Es uno de los parámetros más importantes de esta teoría. En la figura 1.1 se puede ver un ejemplo de mallado.

Estos puntos se unen mediante los llamados enlaces o *links*, que conectan los puntos x y $x + a\hat{\mu}$. El menor cuadrado posible es el denominado operador plaqueta⁵, del cual se hablará más adelante.

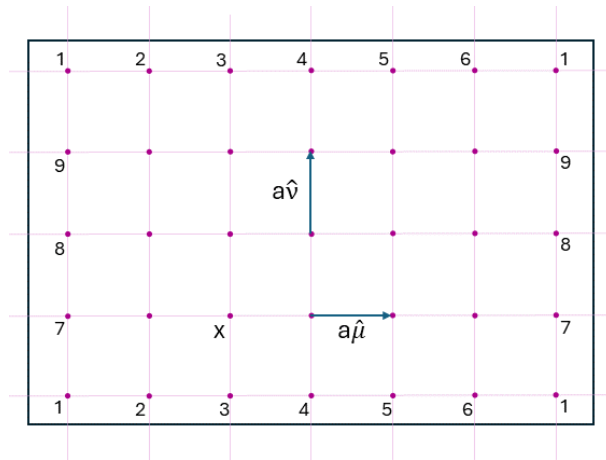


Figura 1.1: Red de dos dimensiones con condiciones de contorno.

⁵Traducción de su nombre habitual *Plaquette operator*.

Es importante ver que una lattice es algo finito, por lo que surgen algunos problemas, el principal son los posibles efectos de borde, que en ciertas situaciones puede regir el comportamiento del sistema. Para paliar dicha consecuencia, se imponen condiciones de contorno periódicas.

El reto en sí es buscar qué espaciado del retículo es el adecuado, pues uno muy pequeño haría el programa muy lento, y uno muy grande resultaría en errores mayores. Al principio se pensaba que $\sim 0,05 - 0,1 fm$ era necesario, pero se concluyó que para que LQCD funcionase bien tenía que ser del orden de $0,3 - 0,4 fm$.

Otro reto a asumir es el denominado *UV cutoff* [12], pues los quarks y gluones con un momento $p = 2\pi/\lambda$ mayor a π/a son excluidos por los propios cálculos, pues la longitud de onda más pequeña que se puede generar en una red de este tipo es de $\lambda = 2a$. En otras teorías esto no es un problema, pero en QCD si, pues los modos ultravioleta afectan, en gran medida, a los infrarrojos.

Con la red definida, el siguiente paso será expresar una Lagrangiana válida para LQCD.

1.2.2. LAGRANGIANA DE WILSON En LQCD se necesita obtener una característica muy importante, la invariancia frente a transformaciones de gauge. Esto se puede conseguir haciendo transformaciones usando matrices de SU(3).

Para ello, un cambio fundamental en el tensor de campo $\mathcal{F}_{\mu\nu}$, dado en 1.21 es no hacer uso de los potenciales vectoriales A_μ , en su defecto, se usarán unos nuevos que sean invariantes con el gauge. Una posibilidad son los operadores U_μ , estos son matrices del grupo SU(3), que sí cumplen esta necesidad. Por tanto, se representa el campo mediante estos en los enlaces en vez de en los puntos del retículo. Los operadores U_μ se definen como:

$$U_\mu(x) \equiv \mathcal{P}e^{-i \int_x^{x+a\hat{\mu}} gA \cdot dy} \quad (1.24)$$

siendo A el vector potencial de QCD. La letra \mathcal{P} representa el orden en el que actúan, dependiendo pues del camino seguido. Por tanto, estos operadores tendrán el orden del propio camino, algo importante a tener en cuenta porque son matrices que no conmutan.

El producto de diversos operadores U sigue siendo parte del grupo $SU(3)$, pues ellos mismos son elementos de este grupo.

Los operadores U van seguidos de un subíndice μ o ν ⁶, indicando la dirección en la que se establecen. Por tanto, para tomar el sentido negativo del eje de coordenadas, se usa el conjugado del operador.

Para un camino cerrado \mathcal{C} , como el que se encuentra en la figura 1.2, se puede definir un operador, de manera que la traza de dicho elemento sea independiente del punto inicial del camino, además de ser invariante ante transformaciones gauge. Este operador se denomina lazo de Wilson⁷, y tiene la siguiente forma:

$$W(C) \equiv \frac{1}{3} \text{Tr} \mathcal{P} e^{-i \oint_C g A \cdot dx} \quad (1.25)$$

donde la integral de la exponencial es sobre el camino cerrado \mathcal{C} en vez de entre sitios del retículo, como en la expresión 1.24

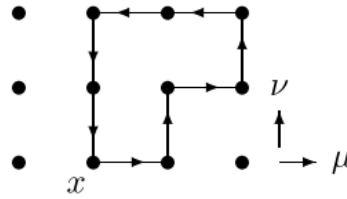


Figura 1.2: Camino arbitrario en una red (G. Lepage, *Lattice QCD for novice*, página 28)

Dichos lazos de Wilson tienen especial interés pues se pueden obtener magnitudes físicas a partir de ellos, por ejemplo la energía del estado fundamental. Un ejemplo de esto sería el siguiente.

Obtención de la energía

El valor medio de los Wilson loops está relacionado con el potencial de interacción de un quark y un antiquark estáticos, separados una distancia r . Para un caso general se tiene el siguiente procedimiento.

⁶En una Lattice se definen las orientaciones como μ (horizontal) y ν (vertical)

⁷En inglés sería *Wilson Loop* y se ha traducido como lazo, o bucle, de Wilson.

Siendo $\Psi_{ij}(T)$ los autovectores del Hamiltoniano usado, se tiene:

$$W(\mathcal{C}_{r,t}) = \langle \frac{1}{N} \text{Tr} \Psi(0) \Psi^\dagger(T) \rangle \quad (1.26)$$

En este punto, podemos hacer uso de la relación de cierre:

$$\sum_n |n\rangle \langle n| = 1 \quad (1.27)$$

Obteniendo así:

$$W(\mathcal{C}_{r,t}) = \sum_n \frac{1}{N} \langle \Psi_{ij}(0) | n \rangle \langle n | \Psi_{ji}^\dagger(T) \rangle = \sum_n \frac{1}{N} |\langle \Psi_{ij}(0) | n \rangle|^2 e^{-E_n T} \quad (1.28)$$

siendo E_n la energía del estado de excitación n . Puesto que hay una exponencial negativa, es directo ver que a mayor T los valores van a ser despreciable, que para la aproximación en la que $T \rightarrow \infty$ solo va a quedar el estado fundamental, es por tanto:

$$W(\mathcal{C}_{r,t}) \longrightarrow e^{-E_0(r)T} \quad (1.29)$$

Quedando definida la energía de esta manera. A modo de ejemplo, en el caso de la figura 1.2 se obtiene el siguiente lazo de Wilson:

$$W(C) = \frac{1}{3} \text{Tr} (U_\mu(x) U_\nu(x + a\hat{\mu}) U_\mu(x + a\hat{\mu} + a\hat{\nu}) \dots U_\nu^\dagger(x)) \quad (1.30)$$

El siguiente paso será buscar una Lagrangiana a partir de los operadores U_μ . En [1] se puede ver un estudio sobre dicha Lagrangiana. Esta debe de tener invariancia con el gauge, además de ser local y simétrica con respecto al intercambio de ejes. El menor loop posible, es decir, el operador plaqueta, tiene la siguiente forma:

$$P_{\mu\nu}(x) = \frac{1}{3} \text{Tr} (U_\mu(x) U_\nu(x + a\hat{\mu}) U_\mu^\dagger(x + a\hat{\mu} + a\hat{\nu}) U_\nu^\dagger(x)) \quad (1.31)$$

Haciendo uso de este operador se puede encontrar una acción de Wilson aproximada, la cual tiene la siguiente forma:

$$S_{Wil} = \beta \sum_{x, \mu > \nu} (1 - P_{\mu\nu}(x)) \quad (1.32)$$

con $\beta = 6/g^2$. Una mejora a dicha acción sería [1], por lo que la expresión mejorada quedaría como:

$$\begin{aligned} S_{Wil} &\equiv -\beta \sum_{x, \mu > \nu} \left\{ \frac{5\mathcal{P}_{\mu\nu}}{3} - \frac{\mathcal{R}_{\mu\nu} + \mathcal{R}_{\nu\mu}}{12} \right\} + const \\ &= \int d^4x \sum_{\mu, \nu} \frac{1}{2} Tr \mathcal{F}_{\mu\nu}^2 + \mathcal{O}(a^4) \end{aligned} \quad (1.33)$$

siendo esta la acción que se usará en este trabajo.

Por último, usando una técnica de filtrado conocida como *smearing* [1], se transforman los operadores enlace U_μ de la siguiente forma:

$$\tilde{U}_\mu(x) \equiv (1 + \epsilon a^2 \Delta^{(2)})^n U_\mu(x) \quad (1.34)$$

con

$$\Delta^{(2)} U_\mu(x) \equiv \sum_{\rho} \Delta_{\rho}^{(2)} U_\mu(x) \quad (1.35)$$

siendo $\Delta_{\rho}^{(2)}$ de la siguiente forma:

$$\begin{aligned} \Delta_{\rho}^{(2)} U_\mu(x) &\equiv \frac{1}{u_0^2 a^2} (U_{\rho}(x) U_{\mu}(x + a\hat{\rho}) U_{\rho}^{\dagger}(x + a\hat{\mu}) - 2u_0^2 U_{\mu}(x) \\ &\quad + U_{\rho}^{\dagger}(x - a\hat{\rho}) U_{\mu}(x - a\hat{\rho}) U_{\rho}(x - a\hat{\rho} + a\hat{\mu})) \end{aligned} \quad (1.36)$$

Haciendo uso de esta técnica los gluones con un momento elevado se excluyen directamente del cálculo, además, se reducen los errores estadísticos del método de Monte Carlo.

Una de las aplicaciones de estos métodos es el cálculo del potencial de interacción.

Potencial de interacción quark-antiquark

El potencial entre un quark y un antiquark $q\bar{q}$ se define mediante la energía de los campos *gauge* en presencia de dos fuentes de color estáticas, separadas por una distancia r . Debido a la *libertad asintótica* antes mencionada, se puede esperar que a cortas distancias este potencial sea coulombiano, y por tanto, viable a la hora de tratarlo perturbativamente. Sin embargo, debido a la propiedad confinante de los quarks, a largas distancias no se puede tratar de esta forma. LQCD se puede usar para describir estos comportamientos de manera no perturbativa, con lo que se puede estudiar las propiedades de un potencial $q\bar{q}$ y su concordancia con la teoría perturbativa.

Se llaman "*observables*" a aquellas funciones que muestran invariancia con el gauge. Por ejemplo, las trazas de ciertos operadores alrededor de caminos cerrados, que dados para un retículo son en la forma $Tr_j U(\mathcal{C})$, siendo Tr la traza, y $U(\mathcal{C})$ el producto de los operadores enlace del retículo para una curva cerrada \mathcal{C} . El valor estimado de estos caminos viene dado de la siguiente forma:

$$W(\mathcal{C}) = \langle Tr U(\mathcal{C}) \rangle \quad (1.37)$$

que no son más que los antes mencionados bucles de Wilson.

Un caso concreto es el de un camino rectangular, de lados $r \times t$, pues está relacionado con el potencial de un quark estático, para valores de t grandes. Se corresponde con la siguiente expresión:

$$V(r) \equiv - \lim_{t \rightarrow \infty} \frac{1}{t} W(\mathcal{C}_{r,t}) \quad (1.38)$$

tal que

$$W(\mathcal{C}_{r,t}) \sim C e^{-tV(r)} \quad (1.39)$$

El lazo de Wilson puede ser interpretado de la siguiente manera:

$$W(\mathcal{C}_{r,t}) = \langle \Psi | e^{-\tilde{H}T} | \Psi \rangle = \sum_n |\langle \Psi^{(n)} | \Psi \rangle|^2 e^{-tE_n} \sim_{t \rightarrow \infty} |\langle \Psi^{(0)} | \Psi \rangle|^2 e^{-tV(r)} \quad (1.40)$$

siendo $\Psi^{(n)}$ un set completo de autovalores del Hamiltoniano, y es un estado del tipo:

$$\Psi_{\alpha\beta}[U'] = V_{\alpha\gamma}(\vec{x})V_{\delta\beta}^{-1}(\vec{y})\Psi_{\gamma\delta}[U] \quad (1.41)$$

que es una función de onda, invariante con el gauge, que describe el estado de un quark en \vec{x} y un antiquark en \vec{y} , ambos estáticos.

El estudio de la forma de este potencial va más allá de este trabajo, por lo que se hará un breve resumen. El potencial $V(r)$ se comporta asintóticamente de forma lineal con r , a este comportamiento se le llama *confinamiento lineal quark*.

$$V(r) \sim_{r \rightarrow \infty} \sigma r \quad (1.42)$$

En el límite en el que r es muy grande, el valor esperado del bucle de Wilson se puede predecir por la teoría de cuerdas bosónicas efectivas [2]. Gracias a este estudio se llega a que, para un r grande, el potencial tiene la siguiente expansión asintótica:

$$V(r) = \sigma r + \mu - \frac{\gamma}{r} \quad (1.43)$$

siendo σ la tensión entre quarks⁸, μ es una constante de masa y γ es un efecto cuántico característico de las cuerdas bosónicas relativistas.[4]

En los próximos capítulos de esta obra se pondrá en manifiesto como implementar esto para obtener resultados numéricos.

⁸Esta expresión viene de la traducción en inglés de *string tension*.

Objetivos

El objetivo general de este trabajo es hacer una primera aproximación a la cromodinámica cuántica en el retículo.

Como objetivos específicos se plantean los siguientes:

- Realizar una introducción a la formulación mediante integrales de camino de la teoría cuántica.
- Llevar a cabo una introducción a los métodos de trabajo de la cromodinámica cuántica en el retículo.
- Realizar una simulación que permita obtener el potencial de interacción quark-antiquark.

Metodología

En este capítulo se muestra como obtener un algoritmo que aporte resultados de un potencial de interacción a partir de LQCD.

En primer lugar se presentan métodos capaces de resolver integrales de camino. Por tanto, se va a implementar el método Monte Carlo en un algoritmo del tipo Metropolis¹. Se aplicará a un oscilador armónico.

Se va a desarrollar un algoritmo que incluya las ecuaciones 1.13 y 1.14 en pos de resolver el propagador cuántico mediante integrales de camino. Una vez creado, se puede implementar en el lenguaje de programación que se quiera. En este trabajo se implementa en Python 3.11.7. Se especificará para un oscilador armónico, para dar ejemplo del potencial de dicho programa.

Aprovechando el caso del oscilador armónico, se extiende el algoritmo para hacer el cálculo de los incrementos de energía 1.17, ampliando también para un oscilador anarmónico. Se hace uso del primero de ellos pues se conocen sus soluciones, por lo que se puede comprobar la certeza de los resultados.

Se aplicarán los cambios necesarios a dicho programa para poder trabajar con la me-

¹Algoritmo que usa el método Monte Carlo para obtener un camino y poder hallar el valor esperado de ciertas integrales.

cánica cuántica, es decir, con LQCD. Se implementará el código resultante para calcular el valor esperado de una acción de Wilson para caminos de longitudes $a \times a$ y $a \times 2a$.

Por último, se buscará un método eficiente para calcular la forma del potencial de interacción quark-antiquark estáticos y separados una distancia r . Dicho potencial se espera tenga la forma 1.43.

$$V(r) = \sigma r + \mu - \frac{\gamma}{r} \quad (3.1)$$

3.1. INTEGRALES DE CAMINO

El primer objetivo para proceder con este apartado es el poder calcular el propagador cuántico, viendo la forma de este y como se ajusta a algunos resultados ya conocidos.

3.1.1. PROPAGADOR CUÁNTICO Para realizar esta tarea se requiere de un algoritmo que describa la acción de un oscilador armónico, para computar la integral de camino resultante en N dimensiones. Por último, el algoritmo representará dichos resultados mediante una gráfica.

Si se parte de la expresión 1.13

$$\langle x | e^{-\tilde{H}T} | x \rangle = \int_{-\infty}^{\infty} dx_1 \dots dx_{N-1} \cdot e^{-S_{lat}[x]} \quad (3.2)$$

se tiene que escribir primero el término de la exponencial, para posteriormente evaluarlo mediante otras técnicas. Por tanto, se escribe de forma directa la acción de un oscilador armónico 1.14

$$S_{lat}[x] \equiv \sum_{j=0}^{N-1} \left[\frac{m}{2a} (x_{j+1} - x_j)^2 + a \frac{x_j^2}{2} \right] \quad (3.3)$$

y se calcula su valor para cada punto del camino. El número de puntos se le aporta al programa en la propia definición de la función. Por último, se pide a la función que nos devuelva el valor $\exp(-S_{lat}[x])$.

Hay múltiples formas de evaluar integrales de camino en Python. El procedimiento seguido para este cálculo será el siguiente.

- En primer lugar se define el integrando, en este ensayo, el $\exp(-S_{lat}[x])$.
- Se define el operador integración, en el que se define el número de grados de libertad.
- Por último, se aplica dicho operador a la función definida, que puesto que usa métodos estadísticos (Monte Carlo) se le pide una cantidad de repeticiones para que estime el resultado.
- El valor estimado será la media ponderada entre todos los resultados estimados del apartado anterior.

Este algoritmo viene descrito en la librería *vegas* [17], la cual tiene gran utilidad matemática. Concretamente, se hará uso de la función *integrator*, que procede de esa forma.

Con esto se procede a la graficación de los resultados obtenidos, además de la solución exacta para comparar.

Esto se puede aplicar a múltiples procesos, por ejemplo, para calcular el intervalo de energía entre varios estados de un oscilador armónico.

3.1.2. VARIACIÓN DE ENERGÍA Para el cálculo de la variación de energía se usa un algoritmo del tipo Metropolis. El proceso a seguir se describe de la siguiente forma.

El primer paso es la creación del camino, para ello se usará la función *update*. Esta usa la siguiente lógica:

- Se genera un valor aleatorio R , con probabilidad uniforme entre $-\epsilon$ y ϵ , que se aportará al programa.
- Se reemplaza el x_j por un $x_j + R$ y se computa el cambio en la acción.
- Si dicho incremento es negativo, se deja el valor de x_j obtenido y se procede con el siguiente punto.
- Si dicho incremento no es negativo, se vuelve a generar un valor aleatorio del 0 al 1 y se comprueba que $\exp(-\Delta S)$ sea mayor que dicho número, si es así, se guarda este valor.

- Se repite este proceso hasta tener todos los puntos que se requieran.

Con esto se consigue generar un camino arbitrario que va desde x_0 hasta x_f , en función de una acción dada.

La primera configuración no suele tener una aleatoriedad suficiente, por lo que puede haber valores repetidos, y de manera general, no se tiene un buen equilibrio estadístico. Para solucionar esto se *termaliza* el camino, esto es descartar las primeras configuraciones² en pos de encontrar este equilibrio estadístico.

En este punto se hace uso de una función *ComputeG* que calcula el valor medio de la integral de camino, para cada uno de los puntos N del camino. Una vez hecho el cálculo se procede a realizar la media aritmética de estos.

Para continuar se calcula la variación de energía, esto se hace de forma directa aplicando la ecuación 1.17 en el código. Los valores obtenidos se representan en una gráfica y se concluye dicho programa. En la figura 3.1 se encuentra el diagrama de flujo del programa.

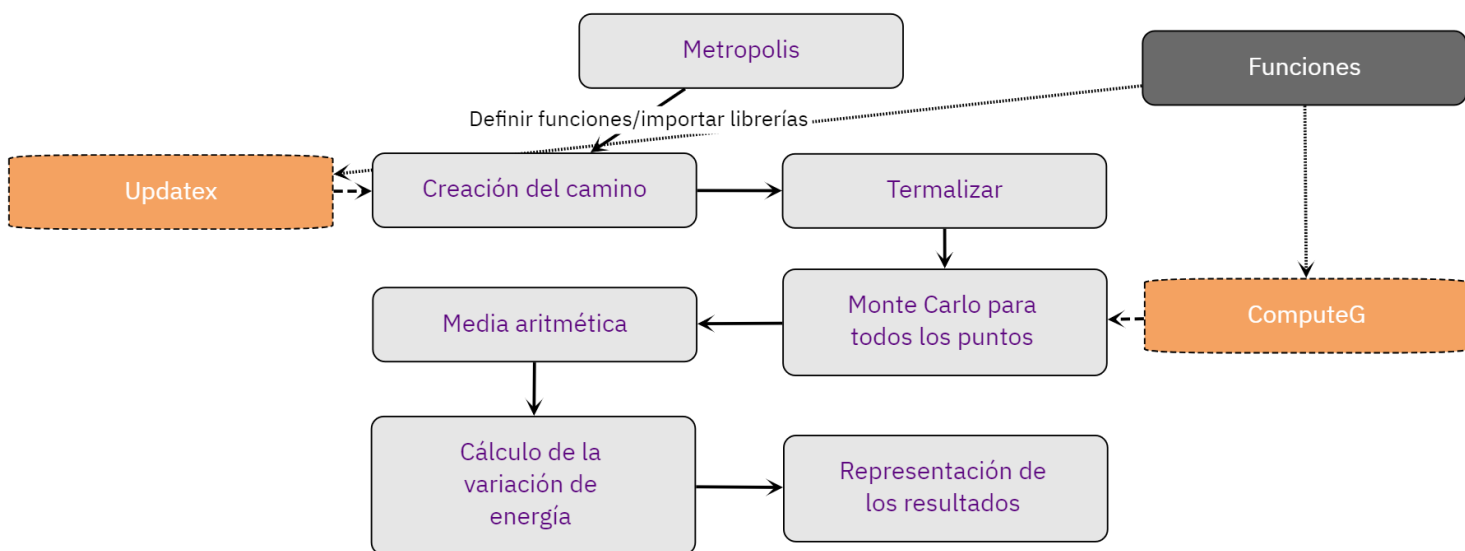


Figura 3.1: Diagrama de flujo de un programa en Python para el cálculo de la variación de energía para una acción S dada.

Los distintos tipos de líneas implican diferentes cosas. Las líneas sólidas son el recorrido natural del programa. Las líneas punteadas hacen referencia a las funciones definidas

²A un camino creado se le llama configuración.

en el capítulo 4.2.2 de este trabajo. Por último, las restantes hacen referencia a donde se implementan dichas funciones, las cuales están representadas en naranja.

Este algoritmo se ha implementado en un programa escrito en Python 3.11.7

3.2. LATTICE QCD

En primer lugar, se darán las pautas a seguir para la creación de un programa funcional para dos cálculos esenciales. El primero el cálculo del valor de la acción de Wilson, haciendo uso de distintos Wilson loops. El segundo algoritmo, y más importante, será capaz de calcular y representar un potencial de interacción $q\bar{q}$ estáticos, a una distancia r , con el que se podrá apreciar la forma del mismo. Pero antes, un par de mejoras al código anterior.

Para empezar, se generan configuraciones aleatorias de la misma manera, pero se adaptaran para que funcionen con matrices de $SU(3)$. El campo gluónico se crea a partir de los operadores $U_\mu(x)$. Para actualizar los enlaces del retículo se multiplicaran por una matriz aleatoria de $SU(3)$. Al principio de la simulación se generan estas de la siguiente forma:

- Se genera un set de matrices hermiticas H de elementos aleatorios entre -1 y 1.
- Se transforman matrices de $SU(3)$ haciendo la operación $1 + i\epsilon H$.
- Las matrices resultantes se normalizan.
- Se guardan las matrices inversas de estas.

Como recomendación, se actualizaran varias veces cada U_μ , en vez de hacerlo una única vez. Con esto se consigue un equilibrio estadístico con sus operadores consecutivos. Esto apenas aumenta el coste del algoritmo, por lo que no es un mayor problema.

Con esto, ya se puede crear un algoritmo para hacer los primeros cálculos.

3.2.1. MONTE CARLO PARA GLUONES Se va a hacer prueba de un programa para calcular los valores de un bucle de Wilson, de la forma 1.32, haciendo uso de la acción siguiente:

$$S_{Wil} = -\tilde{\beta} \sum_{x, \mu > \nu} \frac{\mathcal{P}_{\mu\nu}}{u_0^4} \quad (3.4)$$

Siendo $\tilde{\beta}$ el valor de β antes visto dividido por u_0^4 , y $\mathcal{P}_{\mu\nu}$ el operador plaqueta. u_0 es el valor medio de los operadores U_μ , el cual se introduce para mejorar los resultados.

El valor de u_0 depende del espaciado del retículo únicamente, y es comúnmente conocido como el valor esperado del Landau-gauge, pero a nivel numérico, es decir, el de este trabajo, podemos definirlo como la raíz cuarta del valor esperado del operador plaqueta.

Se hará uso de esta acción en concreto pues es la más sencilla que permite formular LQCD.

Lo primero que se realiza al ejecutar el programa será la definición de parámetros y arrays necesarios, así como las definiciones y librerías que se necesitan en este. Los parámetros usados son: $N = 8$ puntos en el retículo, $\beta = 5,5$, que al ser $u_0 = 0,797$ queda $\tilde{\beta} = 1,719$. Con estos parámetros se estima un espaciado del retículo $a = 0,25 fm$. Se saltarán 50 configuraciones entre cada cálculo de Monte Carlo, y se generan un total de $N_{cf} = 10$ configuraciones.

El siguiente paso será generar las matrices de $SU(3)$, de la forma que se ha explicado en este mismo apartado.

Una vez los preparativos están hechos, empieza a generarse el camino, se termaliza. En este caso, estamos haciendo uso de matrices, por lo que la generación del camino es un tanto diferente. El proceso es análogo al anterior, pero requiere el paso extra de computar las funciones γ . Estas se encuentran en los anexos 4.2.2.

Para conseguir esto se usan las funciones *ComputeWL* y *ComputeWLax2a*. Que proceden de la siguiente forma:

- Se localizan todos los puntos del retículo.
- Se computan los valores de los operadores U_μ , generados previamente, en dichos puntos.
- Se calcula la traza del productorio de estos operadores, en el orden del camino.

Generando un bucle en todas las dimensiones se obtienen los valores para un Wilson loop de lados $a \times a$ y otro de lados $a \times 2a$.

Se procesan dichos valores para dejar únicamente el valor medio, así como el cálculo de sus errores. En la figura 3.2 se puede ver el diagrama de flujo de dicho algoritmo.

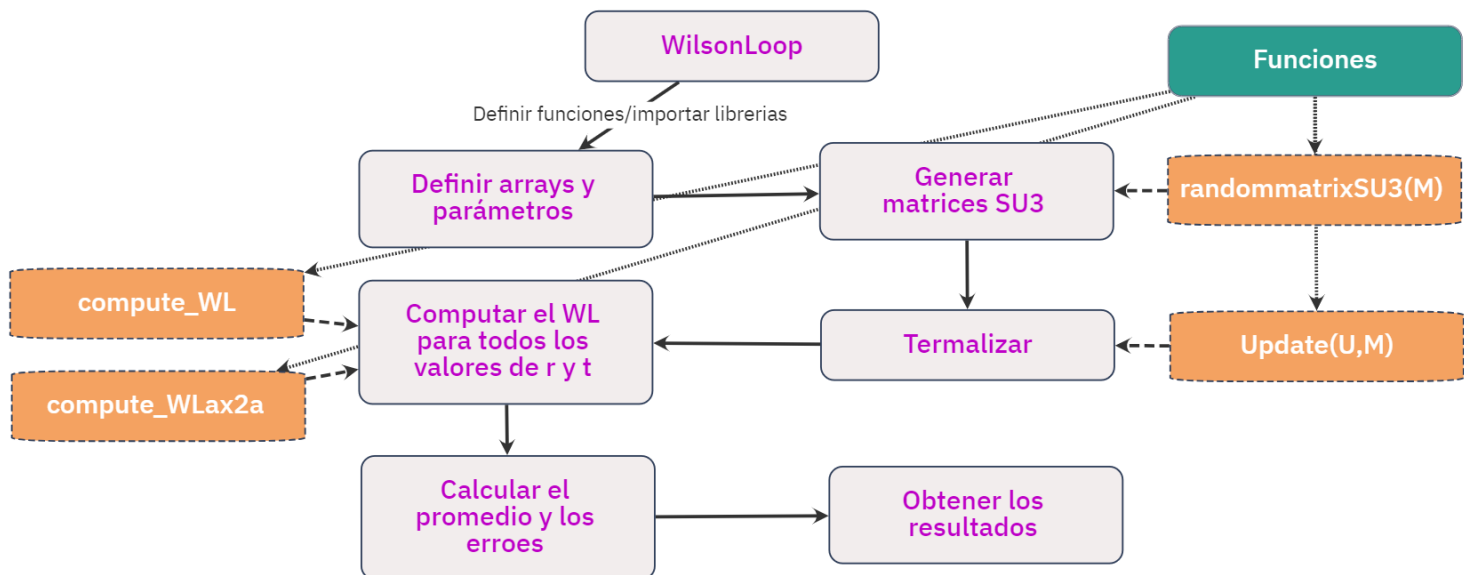


Figura 3.2: Diagrama de flujo de un programa en Python para el cómputo de un Wilson loop de lados $a \times a$ y $a \times 2a$.

En este caso, se ha pedido al programa que muestre los resultados además del tiempo de cómputo que fue necesario para obtenerlos. El tiempo es de especial interés, pues en simulaciones avanzadas es un factor a tener en cuenta.

3.2.2. POTENCIAL DE INTERACCIÓN QUARK-ANTIQUARK Una primera simulación puede ser una que solo involucre gluones, por ejemplo, el potencial entre un quark y un antiquark, ambos estáticos y separados una distancia r . Este potencial tiene especial interés por demostrar el confinamiento de los quarks. A cortas distancias este potencial

será coulombiano, pasando a ser lineal conforme esta aumenta.

El propagador para un quark no relativista en un campo A_μ , escrito en LQCD es:

$$G(x, t) = U_t^\dagger(x, t - a)U_t^\dagger(x, t - 2a)\dots U_t^\dagger(x, 0) \quad (3.5)$$

El propagador para un antiquark sería el complejo conjugado de este último $G^\dagger(x, t)$.

Los parámetros usados son: $N = 8$ puntos en el retículo, $N_r = 4$ puntos de extensión máxima en r y $N_t = 4$ puntos de extensión máxima en t ³. Un espaciado del retículo $a = 0,25 fm$. Se saltarán $N_{cor} = 10$ configuraciones entre cada cálculo de Monte Carlo, y se generan un total de $N_{cf} = 10$ configuraciones.

Como es evidente, tanto el punto de la generación de matrices como el punto de termalizar sigue el mismo proceso que el comentado anteriormente. El cuarto paso es el de computar el valor de Monte Carlo para el propagador que se le da al programa. Este proceso usa la técnica de *smearing*, comentada en el apartado de introducción, integrada en la función *SubstitutionU*.

Esta función realizará la siguiente tarea:

- Se generan los operadores U_μ como hasta ahora.
- Se realiza una función que calcule la expresión 1.36, para todos los puntos y direcciones.
- Se actualizan los operadores U_μ con esta derivada.

Se realiza un bucle para todos estos, calculando el valor del Wilson loop. Se repetirá dicho computo N_{cf} veces, parámetro que se le aporta al programa. Puesto que son programas con una elevada duración, se le pedirá que muestre un mensaje al terminar cada cálculo.

³En el programa anterior se definía N como el numero de puntos del retículo. En este se define una N por cada dimensión.

Una vez terminado este bucle, se computara el Wilson loop para todos los valores de r y t de la lattice. Se usa el programa *ComputeWL* para esta labor.

Por último, se representan los puntos y se realiza una regresión lineal de estos, con ello obteniendo los valores de los parámetros de un potencial del tipo 1.43. Para los puntos se usa la función *gendata*.

El diagrama de flujo del programa creado para el cálculo de dicho potencial se encuentra en la figura 3.3.

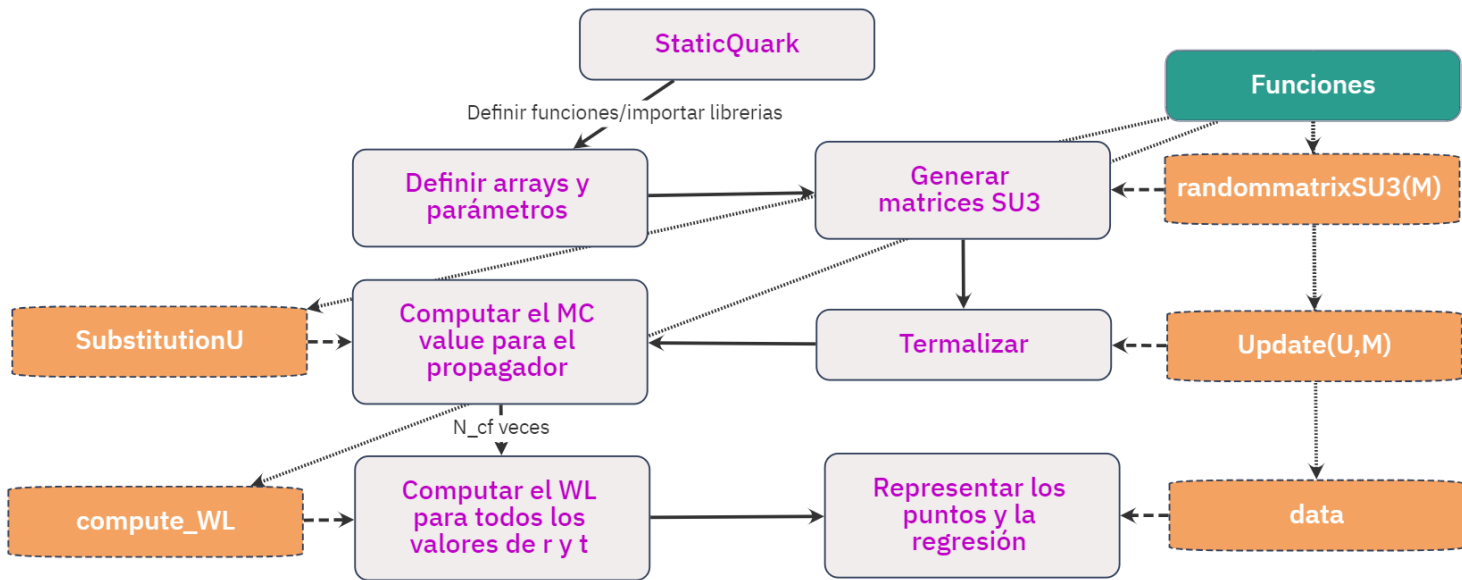


Figura 3.3: Diagrama de flujo de un programa en Python para calcular el potencial de interacción entre un quark y un antiquark estáticos.

Este código fue implementado en Python, parte de estos se pueden encontrar en los anexos de este trabajo 4.2.2. En el enlace [20] se pueden encontrar de manera completa.

Se realizaron todas las simulaciones en un ordenador personal. Las características, relevantes para este ámbito, del mismo son:

- Procesador Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
- Memoria RAM de 16GB

Los parámetros usados en estas simulaciones son los siguientes:

- El parámetro a , o espaciado del retículo, será $a = 0,5 fm$ en los 2 primeros apartados, y será $a = 0,25 fm$ en los dos últimos.
- El número de puntos de la red, N , será $N = 8$ o $N = 20$ dependiendo del ejercicio.
- El parámetro N_{cf} indica cuantas configuraciones se guardan.
- El parámetro N_{cor} indica cada cuantos bucles se guarda una configuración.
- ϵ es un parámetro relevante en la creación del camino. Se usará $\epsilon = 0,24$ y $\epsilon = 1,4$
- β es un parámetro dependiente del espaciado del retículo, para un $a = 0,25 fm$ se corresponde una $\beta = 5,5$.

La salida de los programas serán gráficas que nos aportan la información sobre una magnitud física para distintos valores de x o t , siendo estas una de las coordenadas de la red. Para uno de los programas no se genera una figura, pues los resultados de este se recogen en una tabla.

Los tiempos de ejecución oscilan entre varios minutos hasta varias horas.

Los resultados para los métodos seguidos en este apartado se encuentran en el siguiente capítulo, así como la discusión de los mismos.

Resultados y discusión

En este capítulo se va a dar muestra de los resultados obtenidos para los diversos programas explicados en el capítulo anterior, así como una breve explicación o verificación de los mismos.

En primer lugar se muestra el resultado de aplicar el concepto de integral de camino a un propagador cuántico, obteniendo una serie de puntos que se pueden comparar con el valor exacto que deberían tener. Para ello, se ha usado la solución exacta del caso del oscilador armónico.

Para continuar, se ha aplicado este método de forma iterativa para hallar los valores de energía, o más bien de los intervalos de energía, entre los estados de un par de osciladores, que en este caso, serán armónico y anarmónico.

Se ha realizado un primer acercamiento a LQCD haciendo el cálculo de una acción de Wilson simple, para un camino de longitudes $a \times a$ y otro $a \times 2a$. Se ha realizado una segunda vez para una acción mejorada.

Para finalizar, se muestran los resultados de una primera simulación, hallando la forma de un potencial de interacción entre un quark y un antiquark estáticos. En primera instancia, se puede verificar con esto la *propiedad confinante* y *libertad asintótica* que se mencionaron al principio de este texto.

4.1. INTEGRALES DE CAMINO

4.1.1. PROPAGADOR CUÁNTICO El primer resultado que se muestra será el de un ejercicio para el oscilador armónico, para el cual se ha hecho uso de la librería *vegas*, introducida anteriormente, para evaluar el propagador de un oscilador armónico y compararlo con su solución exacta en mecánica cuántica. En la figura 4.1 se muestra el resultado obtenido.

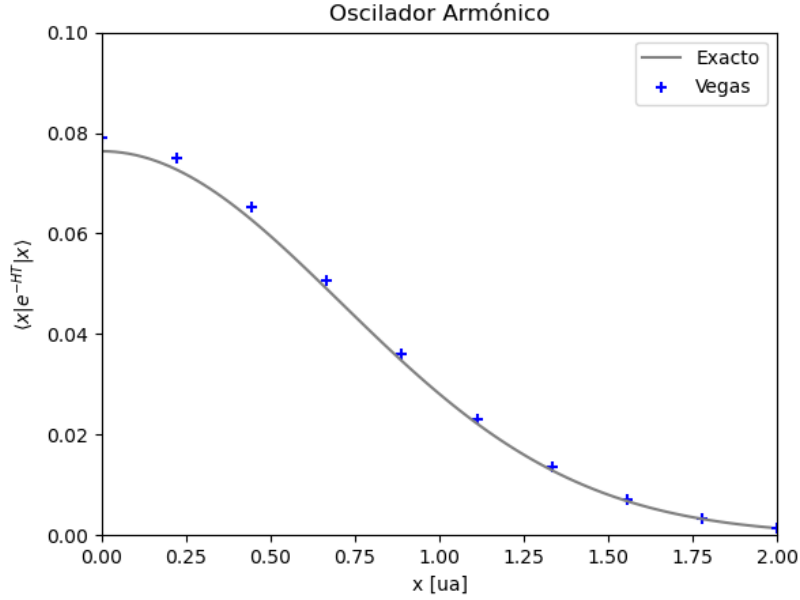


Figura 4.1: Propagador de un oscilador armónico para $T=4$, usando *vegas* [17] (cruces) para la integración numérica, evaluando una integral de camino en 8 dimensiones. Se ha computado también el resultado exacto (línea), el cual se ha aproximado por el cuadrado de la función de onda del estado fundamental, multiplicado por $e^{-E_0 T}$

Para el resultado del propagador N-dimensional, en este caso con $N=8$, se ve claramente la similitud entre ambos resultados. La forma del propagador es la misma para cualquier valor de N , pero cambia el punto inicial, siendo más elevado cuanto menor sea N .

En vista de este resultado, se puede ver que la interpretación mediante integrales de camino puede ser una buena forma de trabajar con mecánica cuántica.

4.1.2. VARIACIÓN DE ENERGÍA Un primer ejemplo muy claro y conocido sería el de un oscilador armónico. Se han aplicado los cálculos anteriormente explicados a un retículo, obteniendo los valores de energía entre los estados para un oscilador de este tipo.

Como variación a este, se muestran también los resultados para un oscilador anarmónico.

Los valores de la energía vistos en la ecuación 1.17, obtenidos mediante un programa Monte Carlo para un oscilador armónico y anarmónico se encuentran en las figuras 4.2 y 4.3 respectivamente.

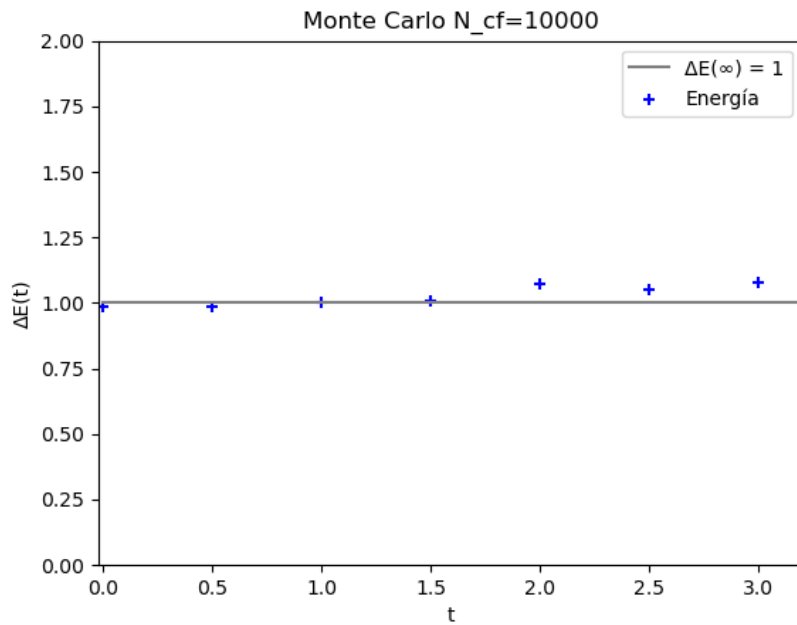


Figura 4.2: Energía del primer estado excitado respecto el estado fundamental 1.17 frente a t para un oscilador armónico. Los parámetros en uso son: $N = 20$, $a = 1/2$, $N_{cf} = 10000$, $N_{cor} = 20$ y $\epsilon = 1,4$.

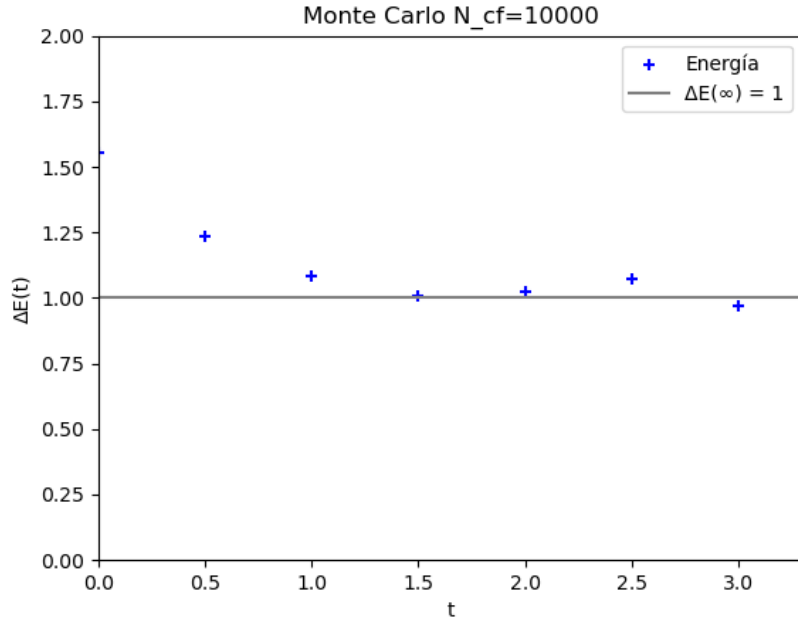


Figura 4.3: Energía del primer estado excitado respecto el estado fundamental 1.17 frente a t para un oscilador anarmónico. Los parámetros son: $N = 20$, $a = 1/2$, $N_{cf} = 10000$, $N_{cor} = 20$ y $\epsilon = 1,4$.

siendo la linea gris la representación de su valor exacto.

Se puede apreciar si los valores se acercan a su valor asintótico por encima o por debajo, siendo esta segunda opción incorrecta generalmente. Por lo que habría que tener cuidado a la hora de analizar resultados que pudiesen venir desde abajo. Esto es debido a la acción usada, que puede mejorarse incorporando nuevos términos, así como mejorar el algoritmo de cálculo usado.

Unos resultados por debajo implicaría una norma negativa, lo cual no es posible en la mecánica cuántica.

Para el caso del oscilador anarmónico, se puede apreciar que para " t " bajos no se ajusta del todo bien, mejorando en gran medida conforme esta aumentan. De igual forma, es claro ver que al avanzar los puntos tienden a su valor teórico, por lo que se ajusta bastante bien a lo esperado.

Una vez se ha visto la eficacia de las integrales de camino para la resolución de pro-

blemas, se aplican a LQCD para obtener unos primeros resultados de gran interés físico.

4.2. LATTICE QCD

Siguiendo el esquema propuesto, se verán los resultados obtenidos evaluando integrales de camino gluónicas mediante métodos Monte Carlo.

4.2.1. MONTE CARLO PARA GLUONES La aplicación directa sería calcular los valores de una acción de Wilson del tipo 3.4 para 2 redes arbitrarias. Esto es, evaluar integrales de camino gluónicas, siguiendo un camino concreto aportado al programa.

Para 2 retículos de longitudes $a \times a$ y $a \times 2a$ se han calculado los valores aproximados del *Wilson loop*, estos valores se encuentran en la tabla 4.1.

Wilson loop	Acción simple	[1]	Acción mejorada	[1]
$a \times a$	$0,4990 \pm 0,0016$	$0,50 \pm 0,01$	$0.541 \pm 0,001$	$0,54 \pm 0,01$
$a \times 2a$	$0,2620 \pm 0,0019$	$0,26 \pm 0,01$	0.285 ± 0.001	$0,28 \pm 0,01$

Tabla 4.1: Valores del Wilson loop para una acción simple, y una mejorada. Los parámetros fueron $\beta = 5,5$, $a = 0,25\text{fm}$, $N=8$, $N_{cor} = 50$, $N_{cf} = 10$. Se han aportado los resultados obtenidos en [1] para comparación.

Se puede apreciar que los valores obtenidos en este trabajo son muy parecidos por los obtenidos en el artículo de *Lepage*, por lo que, debido a su similitud, se concluye el apartado con un resultado de acuerdo a lo esperado. *Lepage* no aporta el valor de los errores en su trabajo, por lo que se ha asumido que son los dados en la tabla por el redondeo de los valores.

Por último, se implementa este método para el cálculo del potencial de interacción $q\bar{q}$.

4.2.2. POTENCIAL DE INTERACCIÓN QUARK-ANTIQUARK Una simulación que aporte resultados de interés sería la de un potencial de interacción de la forma 1.43. Se ha hecho uso de un espaciado del retículo de $a = 0.25\text{fm}$. En la figura 4.4 se encuentra el potencial resultante.

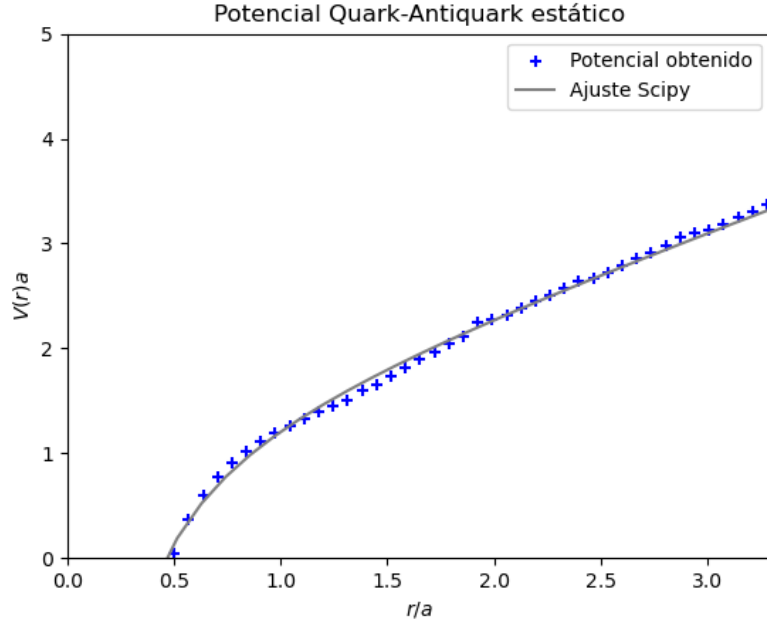


Figura 4.4: Potencial de interacción entre un quark y un antiquark, ambos estáticos y separados una distancia r . Los valores de los parámetros de la expresión 1.43 se han obtenido mediante una librería de Python *scipy*, concretamente la función *curve.fit* [18]. Dichos valores son $\sigma = (0,71 \pm 0,01)GeV/fm$; $\gamma = (0,71 \pm 0,03)GeV \cdot fm$ y $\mu = (1,21 \pm 0,05)GeV$.

Para analizar la veracidad de estos datos hay que fijarse en la forma de la curva. En primer lugar, se aprecia que para valores bajos de r la curva tiene una forma coulombiana, es decir, $V(r) \sim \gamma/r$. Por tanto, está dando información respecto la *libertad asintótica*.

En segundo lugar, se ve que al aumentar los valores de r el potencial sigue en crecimiento, concretamente de una forma lineal $V(r) \sim \sigma r$. Esto confirma el confinamiento quark, es decir, que conforme aumenta la distancia, las partículas están más ligadas.

Por último, la bondad del ajuste se ha calculado mediante la técnica de mínimos cuadrados. El valor que se obtiene de este es $R^2 = 0,998$, por lo que se concluye que dicho ajuste (línea gris) se correlaciona en gran medida con el potencial obtenido (cruces).

En conclusión, se puede apreciar que el resultado está en gran medida de acuerdo con la teoría esperada, confirmando las sospechas sobre estos dos fenómenos.

Conclusiones

Las principales conclusiones que se han sacado de este TFG son:

- La formulación mediante integrales de camino permite una implementación numérica que requiere menor cálculo que el uso de la ecuación de *Schrödinger*, que además se puede adaptar a la teoría de campos.
- La QCD en el retículo permite obtener soluciones a problemas que no podrían obtenerse de otra manera.
- Las simulaciones de sistemas simples pueden realizarse con un coste computacional bajo, obteniendo resultados de interés físico.

Conclusions

The main conclusions drawn from this final project are:

- The path integral formulation allows for a numerical implementation that requires less computation than using the Schrödinger equation, and it can also be adapted to field theory.
- Lattice QCD allows for solutions to problems that could not be obtained otherwise.
- Simulations of simple systems can be performed with a low computational cost, yielding results of physical interest.

Bibliografía

- [1] Lepage, G. P. (2005, June 30). *Lattice QCD for novices*. arXiv.org. <https://arxiv.org/abs/hep-lat/0506036v1>
- [2] Nambu, Y. (1979). *QCD and the string model*. Physics Letters B, 80(4-5), 372-376. [https://doi.org/10.1016/0370-2693\(79\)91193-6](https://doi.org/10.1016/0370-2693(79)91193-6)
- [3] Creutz, M. (2022). *Quarks, gluons and lattices*. <https://doi.org/10.1017/9781009290395>
- [4] Necco, S. (2003, June 5). *The static quark potential and scaling behavior of $SU(3)$ lattice Yang-Mills theory*. arXiv.org. <https://doi.org/10.48550/arXiv.hep-lat/0306005>
- [5] Annema, P. (2019, July). *Determining the String Tension using QCD Lattice Simulations* - Student Theses Faculty of Science and Engineering. <https://fse.studenttheses.ub.rug.nl/20342/>
- [6] Scott, J. L. (2016). *Monte Carlo Applications and Lattice QCD*. https://digitalcommons.lib.uconn.edu/cgi/viewcontent.cgi?article=1502&context=srhonors_theses
- [7] Vege, H. M. M. (2019). *Solving $SU(3)$ Yang-Mills theory on the lattice: a calculation of selected gauge observables with gradient flow*. INSPIRE. <https://inspirehep.net/literature/1750558>
- [8] Wagner, S., Bode, A., Satzger, H., Brehm, M. (2014). *High performance computing in science and engineering Garching/Munich 2016*. In Bayerische Akademie Der Wissenschaften, Muenchen (Germany); Leibniz-Rechenzentrum (LRZ), Garching (Germany). <https://doku.lrz.de/books-with-results-on-lrz-hpc-systems-10745976.html>

- [9] MacKeown, P. K. (1985). *Evaluation of Feynman path integrals by Monte Carlo methods*. American Journal of Physics, 53(9), 880-885. <https://doi.org/10.1119/1.14356>
- [10] Gross, F., Klempt, E., Brodsky, S.J. et al. *50 Years of quantum chromodynamics*. Eur. Phys. J. C 83, 1125 (2023). <https://doi.org/10.48550/arXiv.2212.11107>
- [11] Ukawa, A. (2015). *Kenneth Wilson and Lattice QCD*. Journal of Statistical Physics, 160(5), 1081-1124. <https://doi.org/10.48550/arXiv.1501.04215>
- [12] Carmona, J. M., Cortés, J. L. (2001). *Infrared and ultraviolet cutoffs of quantum field theory*. Physical Review. D. Particles, Fields, Gravitation, and Cosmology/Physical Review. D. Particles and Fields, 65(2). <https://doi.org/10.48550/arXiv.hep-th/0012028>
- [13] Particle Data Group. (n.d.). Particle Data Group. https://pdg.lbl.gov/2023/reviews/contents_sports.html
- [14] Deldar, S. (2000). *Static $SU(3)$ potentials for sources in various representations*. Physical Review. D. Particles, Fields, Gravitation, and Cosmology/Physical Review. D. Particles and Fields, 62(3). <https://doi.org/10.48550/arXiv.hep-lat/9911008>
- [15] Makeenko, Y. M. (2010). *Brief introduction to Wilson loops and large N* . Physics of Atomic Nuclei, 73(5), 878-894. <https://doi.org/10.48550/arXiv.0906.4487>
- [16] Davies, C. (2005, September 15). *Lattice QCD - A guide for people who want results*. arXiv.org. <https://doi.org/10.48550/arXiv.hep-lat/0509046>
- [17] Librería *vegas*. <https://pypi.org/project/vegas/>
- [18] Función *Curve.fit* https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html
- [19] Github Alessandro Cotellucci <https://github.com/AlessandroCotellucci/Lattice-QCD-for-novice>
- [20] Github: https://github.com/Lahelia0/TFG_Codes

Anexo: Definición de funciones en Python

A continuación se encuentran definidas algunas de las funciones de las que se ha hecho uso en los algoritmos de cálculo. Los programas implementados en Python, así como algunas definiciones restantes, se encuentran en [20].

Estos programas han sido una modificación de los propuestos en [19]

```
1 #Actualiza la posicion en funcion de una accion dada
2 #input:-x:vector posicion
3 #inner parameter:-eps:intervalo
4 #           -N:Numero de puntos en la lattice
5 def update(x):
6     N=20
7     eps=1.4
8     for j in range(0,N):
9         old_x = x[j] # save original value
10        old_Sj = S(j,x)
11        x[j] = x[j] + random.uniform(-eps,eps) # update x[j]
12        dS = S(j,x) - old_Sj # change in action
13        if dS>0 and exp(-dS)<random.uniform(0,1):
14            x[j] = old_x # restore old value
15
16 #accion de un oscilador armonico
17 #input:-j:posicion donde se calcula la accion
18 #           -x:vector posicion
19 #inner parameter:-a:lattice spacing
20 #           -N:numero de puntos en la lattice
21 def S(j,x):
22     N=20
23     a=1/2
24     jp = (j+1)%N
25     jm = (j-1)%N
26     return a*x[j]**2/2 +x[j]*(x[j]-x[jp]-x[jm])/a
27
28 #Computar la funcion como un valor medio de la integral de camino
29 #input:-x:vector posicion
30 #           -n:Posicion
31 #inner parameters:-N:numero de puntos en la lattice
32 def compute_G(x,n):
```

```

33     g = 0
34     N=20
35     for j in range(0,N):
36         g = g + (x[j])*(x[(j+n)%N])
37     return g/N
38
39 #Asignacion de arrays y parametros
40 a=1/2
41 N=20
42 N_cf=10000
43 N_cor=20
44 x=ones(N, 'double')
45 G=ones((N_cf,N), 'double')
46 avg_G=ones(N, 'double')
47 avg_Gsquare=ones(N, 'double')
48 errorG=ones(N, 'double')
49
50 #Computar el MC value del propagador para todos los puntos.
51 for j in range(0,N): #inicializar x
52     x[j] = 0
53 for j in range(0,5*N_cor): # termalizar x
54     update(x)
55 for alpha in range(0,N_cf): # Bucle para caminos aleatorios
56     for j in range(0,N_cor):
57         update(x)
58         for n in range(0,N):
59             G[alpha][n] = compute_G(x,n)
60 for n in range(0,N):
61     avg_G[n] = 0
62     avg_Gsquare[n] = 0
63     for alpha in range(0,N_cf):
64         avg_G[n] = avg_G[n] + G[alpha][n]
65         avg_Gsquare[n]=avg_Gsquare[n] + (G[alpha][n])**2
66     avg_G[n] = avg_G[n]/N_cf
67     avg_Gsquare[n]=avg_Gsquare[n]/N_cf
68     errorG[n]=((avg_Gsquare[n]-(avg_G[n])**2)/N_cf)**(1/2)
69     print(n*a,avg_G[n],errorG[n]) #print
70
71 #Asignar parametros y arrays
72 errorE=ones(N-1, 'double')
73 deltaE=ones(N-1, 'double')
74 t=ones(N-1, 'double')
75 exact=ones(N-1, 'double')
76
77 #Computar la variacion de energia para cada par de puntos.
78 for n in range(0,N-1):
79     deltaE[n]=log(abs(avg_G[n]/avg_G[n+1]))/a
80     errorE[n]=(((errorG[n]*avg_G[n+1])/(a*avg_G[n]))**2+
81                ((errorG[n+1]*avg_G[n])/(a*avg_G[n+1]))**2)**(1/2)
82     t[n]=n*a
83     exact[n]=1
84     print(n*a,deltaE[n],errorE[n])
85

```

```

86 #Plot
87 fig, ax = plt.subplots()
88 plt.plot(t, exact, c='gray', label='deltaE(inf) = 1')
89 ax.scatter(t, deltaE, marker='+', c='blue', label='Energia');
90 plt.legend(loc='upper right')
91 plt.xlim(-0.02, 3.2)
92 plt.ylim(0, 2)
93 plt.xlabel('t')
94 plt.ylabel('deltaE(t)')
95 plt.title('Monte Carlo N_cf=10000')
96
97 plt.show()

```

```

1 #matriz compuesta conjugada
2 def dagger(M):
3     N=len(M)
4     H=zeros((N,N),dtype=complex)
5     R=matrix(M)
6     H=R.getH()
7     return H.copy()
8
9 #Calculo regresion
10 #input:-x:parametro
11 #     -r:variable radial
12 #     -y:ordenada
13 def exacto(x,r,y):
14     return x[0]*r-x[1]/r+x[2]-y
15
16 #potencial estatico
17 #input:-r:radio
18 #     -a:string tension
19 #     -b:parametro coulombiano
20 #     -c:constante
21 def data(r,a,b,c):
22     return a*r-b/r+c

```

```

1 #Generador de Matrices de SU(3)
2 #input:-M:Matrix array
3 #output:-M:Matrix array con matrices SU(3)
4 #inner parameters: -Nmatrix: matrices que vamos a generar
5 #                 -eps: parametro
6 def randommatrixSU3(M):
7     identity=eye(3)
8     Nmatrix=100
9     M=zeros((200,3,3),dtype=complex)
10    H=zeros((3,3),dtype=complex)
11    eps=0.24
12    w=cmath.sqrt(-1)
13    for s in range(0,Nmatrix):
14        for j in range(0,3):
15            for i in range(0,3):
16                H[j,i]=complex(random.uniform(-1,1),

```

```

17         random.uniform(-1,1))
18     H=(H.copy()+dagger(H.copy()))/2. #matriz hermitica
19     for n in range(30): #Taylor
20         M[s]=M[s]+(w*eps)**n/math.factorial(n)*
21         linalg.matrix_power(H,n)
22     M[s]=M[s]/linalg.det(M[s])**((1/3)
23
24     M[s+Nmatrix]=dagger(M[s]) #inversa
25     return M.copy()

```

```

1 #Actualizar la posicion en funcion de la accion de wilson para QCD,
2 #usando un algoritmo de tipo metropolis.
3 #input:-U:array de las variables de enlace
4 #       -M:matrices de SU(3)
5 #inner parameters: -N:numero de puntos en la lattice
6 def update(U,M):
7     Nmatrix=100
8     N=8
9     beta=5.5
10    beta_imp=1.719
11    u0=0.797
12    improved=True #true = usar gamma improved, false = usar gamma
13    for x, y, z, t in itertools.product(range(N), repeat=4):
14        for mi in range(0,4):
15            gamma=Gamma(U,mi,x,y,z,t)
16            if improved:
17                gamma_imp=Gamma_improved(U,mi,x,y,z,t)
18            for p in range(10): #numero de iteraciones por punto
19                s=random.randint(2,2*Nmatrix) #elegir matriz
20                if improved:
21                    dS = -beta_imp/(3)*(5/(3*u0**4)*
22                    real(trace(dot((dot(M[s],U[x,y,z,t,mi])
23                    -U[x,y,z,t,mi]),gamma))) -1/(12*u0**6)*
24                    real(trace(dot((dot(M[s],U[x,y,z,t,mi])
25                    -U[x,y,z,t,mi]),gamma_imp))))
26                else:
27                    dS = -beta/(3)*real(trace(dot((dot(M[s].copy(),
28                    U[x,y,z,t,mi].copy())
29                    -U[x,y,z,t,mi].copy()),gamma.copy()))
30                if dS<0 or exp(-dS)>random.uniform(0,1):
31                    U[x,y,z,t,mi] = dot(M[s].copy(),
32                    U[x,y,z,t,mi].copy()) # update U

```

```

1 #Computar el lado del Wilson Loop
2 #input:-x,y,z,t: posiciones
3 #       -U:array de las variables de enlace
4 #       -f:direccion
5 #inner parameter:-N:numero de puntos de la lattice
6 def ProductU(U,x,y,z,t,n,f):
7     N=8
8     I=eye(3)
9     productU=zeros((3,3),dtype=complex) #crear el array productU

```



```

10     productU=I.copy() #inicializar ProductU
11     inc=zeros((4),'int')
12     inc[f]=1 #incre. en la direccion f
13     for i in range(n):
14         x_inc=(x+i*inc[0])%N #incre. en dicha direccion
15         y_inc=(y+i*inc[1])%N #incre. en dicha direccion
16         z_inc=(z+i*inc[2])%N #incre. en dicha direccion
17         t_inc=(t+i*inc[3])%N #incre. en dicha direccion
18         productU=dot(productU,U[x_inc,y_inc,z_inc,t_inc,f])
19     return productU.copy()
20
21 #Computar la inversa del lado del Wilson Loop
22 #input:-x,y,z,t: posiciones
23 #       -U:array de las variables de enlace
24 #       -f:direccion
25 #inner parameter:-N:numero de puntos de la lattice
26 def ProductUdagger(U,x,y,z,t,n,f):
27     N=8
28     I=eye(3)
29     productUdagger=zeros((3,3),dtype=complex) #crear el array
30     productUdagger=I.copy() #inicializar ProductUdagger
31     inc=zeros((4),'int')
32     inc[f]=1 #incre. en la direccion f
33     for i in range(n):
34         x_inc=(x-(i+1)*inc[0])%N #decre. en dicha direccion
35         y_inc=(y-(i+1)*inc[1])%N #decre. en dicha direccion
36         z_inc=(z-(i+1)*inc[2])%N #decre. en dicha direccion
37         t_inc=(t-(i+1)*inc[3])%N #decre. en dicha direccion
38         productUdagger=dot(productUdagger,
39                             dagger(U[x_inc,y_inc,z_inc,t_inc,f]))
40     return productUdagger.copy()

```

```

1 #Smearing
2 #input:-U: array de las variables de enlace
3 #inner parameters:-eps: termino de substitucion
4 #                  -a: lattice spacing
5 #output:-U: U "smeared"
6 def SubstitutionU(U):
7     eps=1./12.
8     a=0.25
9     N=8
10    sec_der=zeros((N,N,N,N,4,3,3),dtype=complex)
11    sec_der=Gauge_cov_deriv(U) #derivada de segundo orden
12
13    for x in range(N):
14        for y in range(N):
15            for z in range(N):
16                for t in range(N):
17                    for mi in range(0,3):
18                        U[x,y,z,t,mi]=U[x,y,z,t,mi].copy()
19                        +eps*a**2*sec_der[x,y,z,t,mi].copy()
20    return U

```