

# Notas Teoria cuantica de campos

Daniel Vazquez Lago

11 de septiembre de 2024

---

---

# Índice general

|  |          |
|--|----------|
| <b>Introducción</b>  | <b>5</b> |
| <b>1. Introducción a Fortran</b>                               | <b>7</b> |
| 1.1. Compilación y ejecución . . . . .                         | 7        |
| 1.2. Formato código fuente . . . . .                           | 7        |
| 1.3. Tipos de datos . . . . .                                  | 8        |
| 1.3.1. Parámetros. Variables. Declaración. Asignación. . . . . | 8        |
| 1.3.2. Arrays, subíndices, substrings . . . . .                | 8        |
| 1.4. Operadores y expresiones . . . . .                        | 8        |
| 1.4.1. Aritméticas . . . . .                                   | 8        |
| 1.4.2. Relación y expresiones lógicas . . . . .                | 9        |



# Introducción

Usaremos  $N = 500$  partículas y una densidad de  $0.5 \text{ } N/V^3$ . La variación máxima de energía permitida es  $1/1000$ .

Usaremos la aproximación de Lennard-Jones, hya que es suave, supone interacciones debiles ideales para los gases nobles.

$$v_{ij}(r_{ij}) = 4\epsilon \left[ (\sigma/r_{ij})^{12} - (\sigma/r_{ij})^6 \right] \quad (1)$$

“Usar una suma doble para luego dividirlo por dos es para pegarle en la cara”. La parte de los sumatorios debe estar libre de polvo y paja para que corra veloz.

$$t_p = \frac{1}{2} \sum \sum v_{ij} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N v_{ij}$$



# 1

## Introducción a Fortran

### 1.1. Compilación y ejecución

### 1.2. Formato código fuente

El formato de código fuente puede ser libre o fijo, y no deben mezclarse ambos en un fichero de código. El código fijo se considera obsoleto en Fortran95. En cualquier caso existen ciertas normas básicas y típicas de Fortran, algunas obligatorias, que todavía se mantienen, por lo que es importante mencionarlas. Estas son:

- Las sentencias de un programa se escriben en diferentes líneas.
- La posición de los caracteres dentro de las líneas es significativa.
- Columnas:
  - 1-5. Número de etiqueta (de 1 a 5 dígitos, se usan números usualmente).
  - 6. Carácter de continuación de línea.
  - Resto. Sentencia.
- Comentarios:
  - Las líneas en blanco se ignoran. Hacen más legible el programa.
  - Si el primer carácter de una línea es \*, c o C la línea es de comentario.
  - Si aparece el carácter ! en una línea (salvo en la columna 6) lo que sigue es un comentario.
- Una línea puede contener varias sentencias separadas por punto y coma (;), el cual no puede estar en la columna 6. Sólo la primera de estas sentencias podría llevar etiqueta.
- Los espacios en blanco son significativos: `IMPLICIT NONE`, `DO WHILE` (obsoleto), `CASE DEFAULT`. Son opcionales en:
  - Palabras clave dobles que comienzan por `END` o `ELSE`.

- DOUBLE PRECISION, GO TO, IN OUT, SELECT CASE.
- El indicador de continuación de una línea es el carácter &.

## 1.3. Tipos de datos

Fortran tiene los siguientes tipos de datos:

- Enteros (INTEGER)
- Reales (REAL, DOUBLE PRECISION)
- Complejos (COMPLEX)
- Lógicos (LOGICAL)
- Caracteres (CHARACTER, CHARACTER(LEN=n), CHARACTER\*n)

### 1.3.1. Parámetros. Variables. Declaración. Asignación.

- Un parámetro tiene un valor que no se puede cambiar (PARAMETER).
- Una variable puede cambiar su valor cuantas veces sea necesario.
- Por defecto, todas las variables que empiecen por `i, j, k, l, m` o `n` son enteras y las demás reales. Es muy recomendable declarar las variables que se utilicen (la sentencia `IMPLICIT NONE` obliga a declarar todas las variables).

### 1.3.2. Arrays, subíndices, substrings

- Un array se define mediante su nombre y dimensiones (cantidad y límites).
- Por defecto el primer índice es 1. En otro caso hay que indicar el rango `i1:i2`.
- Los elementos del array se acceden por sus índices entre paréntesis.

## 1.4. Operadores y expresiones

### 1.4.1. Aritméticas

- Los operadores aritméticos son `+`, `-`, `*`, `/`, `**`.
- El orden de prioridades es el mismo que en el álgebra.
- No puede ver operadores seguidos (incorrecto `a*-b`, correcto `a*(-b)`).



|           |       |       |      |       |        |
|-----------|-------|-------|------|-------|--------|
| Operador  | .NOT. | .AND. | .OR. | .EQV. | .NEQV. |
| Prioridad | 1     | 2     | 3    | 4     | 4      |

### 1.4.2. Relacion y expresiones lógicas

- Los operadores de expresiones son:

|      |      |      |      |      |      |
|------|------|------|------|------|------|
| .EQ. | .NE. | .LT. | .LE. | .GT. | .GE. |
| ==   | /=   | <    | <=   | >    | >=   |

- Los operadores lógicos son:
- Se pueden relacionar expresiones aritméticas con expresiones lógicas y expresiones de caracteres.
- Es recomendable utilizar paréntesis y/o sustituir las expresiones complicadas por combinaciones de expresiones más simples.