

PUNTO 7. Obtención de \hbar y la Función de Trabajo de diversos metales

Toma de medidas con el applet del efecto Fotoeléctrico

Procedimiento

Se toman como valores experimentales la recogida de datos a través del applet de [Phet](#) sobre el efecto fotoeléctrico. Este applet permita la visualización del efecto fotoeléctrico a través del movimiento de electrones y el voltaje detectado en la placa secundaria. También permite aplicar un voltaje a la corriente detectada para poder determinar el voltaje de frenado de una longitud de onda para un metal de terminado y una intensidad lumínica determina.

Para el procedimiento de recogida de datos se determinó la intensidad lumínica de la fuente a 100% y se han definido una serie de valores fijos en el control de voltaje de frenado aplicado. Estos valores van desde -0,20V hasta -1,8 en incrementos de -0,20V.

Para cada uno de estos valores de contravoltaje para el frenado y partiendo del espectro ultravioleta se ha incrementado el valor de longitud de onda hasta que el amperímetro ya no detecta ningún electrón que provoque paso de corriente. De esta manera se determina las tuplas de longitud de onda y su correspondiente voltaje de frenado.

Los valores obtenidos son:

Voltaje	Cobre	Zinc	Sodio	Platino	Calcio
- 0,20 V	241 nm	262nm	466nm	183nm	378nm
- 0,40 V	235 nm	254nm	440nm	179nm	361nm
- 0,60 V	228 nm	246nm	415nm	175nm	344nm
- 0,80 V	221 nm	238nm	390nm	171nm	327nm
- 1,00 V	214 nm	230nm	368nm	167nm	312nm
- 1,20 V	207 nm	222nm	349nm	163nm	298nm
- 1,40 V	201 nm	215nm	331nm	159nm	285nm
- 1,60 V	195 nm	208nm	314nm	155nm	272nm
- 1,80 V	189 nm	201nm	299nm	152nm	261nm

Tratamiento de los valores obtenidos

Después de la obtención de los datos experimentales se procede al tratamiento de estos valores para hallar la constante reducida de Planck y el valor de la función de trabajo para cada metal.

Obtención expresiones matemáticas de para cálculo \hbar y ϕ

Ambos datos se calculan a partir del ajuste a una recta de los valores obtenidos. Al ser una recogida de valores experimentales se aplica un algoritmo de regresión con el objetivo de obtener una aproximación para una modelación de la relación entre las variables que conforman esta recogida de resultados.

Como ya se indicó en el apartado anterior, la expresión $eV = 2\pi\hbar f - \phi$ representa la igualdad de las fuerzas estudiadas, se puede reescribir de la siguiente manera :

$$V = \frac{2c\pi\hbar}{e} \cdot \frac{1}{\lambda} - \frac{\phi}{e} \quad (1)$$

Este modelo de relación entre variables se puede representar por una recta con el algoritmo de regresión (aunque en nuestro caso experimental los puntos sean prácticamente coincidentes); basándonos en la fórmula de la recta $y = m \cdot x \pm b$ podemos inferir que la variable independiente sea la longitud de onda λ , la variable dependiente sea el voltaje de frenado V . La fórmula de la recta $y = m \cdot x \pm b$ se podría utilizar en la igualdad (1) para obtener las siguientes correspondencias:

$$m = \frac{2c\pi\hbar}{e} \quad x = \frac{1}{\lambda} \quad b = \frac{\phi}{e}$$

Si despejamos los valores de las variables \hbar y ϕ obtenemos las siguientes expresiones :

$$\hbar = \frac{m \cdot e}{2 \cdot c \cdot \pi} \quad y \quad \phi = b \cdot e$$

Las únicas variables desconocidas son la **pendiente (m)** y el **origen de ordenadas (b)**; el cálculo de las mismas supondrán la obtención de la constante de Planck reducida y la función de trabajo de los diferentes metales.

Cálculo de pendiente (m) y ordenada en el origen (b)

A continuación se utiliza código Javascript para la realización del cálculo requerido. La librería `regresion.js` que contiene la función `regression.linear` que permite el cálculo de esta recta. También se fijan los valores de los voltajes de frenado comunes utilizados en la recogida experimental de datos.

```
[1] var regression = require('./regression.js');
var voltajes_medidos =
[-0.2,-0.4,-0.6,-0.8,-1.0,-1.2,-1.4,-1.6,-1.8]; // en voltios
(voltaje de frenado)
```

El siguiente código Javascript utiliza los valores obtenidos experimentalmente para calcular a través de la regresión lineal tanto la pendiente como la ordenada en el origen.

Se utilizan diferentes variables para guardar todos los datos obtenidos con la intención de representar gráficamente todos los datos relevantes.

```
[2] // ##### COBRE #####
var longitudes_de_onda_medidas_cobre =
[241,235,228,221,214,207,201,195,189].map(x => x*1e-9); // para
pasar a metros

var inversa_de_longitudes_de_onda_cobre =
longitudes_de_onda_medidas_cobre.map(lambda => 1/lambda );

var datos_a_ajustar_cobre =
voltajes_medidos.reduce((tuplas_longitud_onda_voltaje_cobre, b,
i) => {

    tuplas_longitud_onda_voltaje_cobre.push([inversa_de_longitudes_d
e_onda_cobre[i], b]);
    return tuplas_longitud_onda_voltaje_cobre;
}, []);

var result = regression.linear(datos_a_ajustar_cobre, {
precision: 15 });
var m_cobre = result.equation[0]; // pendiente
var b_cobre = result.equation[1]; // ordenada en origen

/// ##### ZINC #####
var longitudes_de_onda_medidas_zinc =
[262,254,246,238,230,222,215,208,201].map(x => x*1e-9); // para
pasar a metros

var inversa_de_longitudes_de_onda_zinc =
longitudes_de_onda_medidas_zinc.map(lambda => 1/lambda );

var datos_a_ajustar_zinc =
voltajes_medidos.reduce((tuplas_longitud_onda_voltaje_zinc, b, i)
=> {

    tuplas_longitud_onda_voltaje_zinc.push([inversa_de_longitudes_de
_onda_zinc[i], b]);
    return tuplas_longitud_onda_voltaje_zinc;
}, []);

var result = regression.linear(datos_a_ajustar_zinc, { precision:
15 });
var m_zinc = result.equation[0]; // pendiente
var b_zinc = result.equation[1]; // ordenada en origen
```

```

/// ##### SODIO #####
var longitudes_de_onda_medidas_sodio =
[466,440,415,390,368,349,331,314,299].map(x => x*1e-9); // para
pasar a metros

var inversa_de_longitudes_de_onda_sodio =
longitudes_de_onda_medidas_sodio.map(lambda => 1/lambda );

var datos_a_ajustar_sodio =
voltajes_medidos.reduce((tuplas_longitud_onda_voltaje_sodio, b,
i) => {

    tuplas_longitud_onda_voltaje_sodio.push([inversa_de_longitudes_d
e_onda_sodio[i], b]);
    return tuplas_longitud_onda_voltaje_sodio;
}, []);

var result = regression.linear(datos_a_ajustar_sodio, {
precision: 15 });
var m_sodio = result.equation[0]; // pendiente
var b_sodio = result.equation[1]; // ordenada en origen

/// ##### PLATINO #####
var longitudes_de_onda_medidas_platino =
[183,179,175,171,167,163,159,155,152].map(x => x*1e-9); // para
pasar a metros

var inversa_de_longitudes_de_onda_platino =
longitudes_de_onda_medidas_platino.map(lambda => 1/lambda );

var datos_a_ajustar_platino =
voltajes_medidos.reduce((tuplas_longitud_onda_voltaje_platino, b,
i) => {

    tuplas_longitud_onda_voltaje_platino.push([inversa_de_longitudes
_de_onda_platino[i], b]);
    return tuplas_longitud_onda_voltaje_platino;
}, []);

var result = regression.linear(datos_a_ajustar_platino, {
precision: 15 });
var m_platino = result.equation[0]; // pendiente
var b_platino = result.equation[1]; // ordenada en origen

/// ##### CALCIO #####
var longitudes_de_onda_medidas_calcio =
[378,361,344,327,312,298,285,272,261].map(x => x*1e-9); // para
pasar a metros

var inversa_de_longitudes_de_onda_calcio =
longitudes_de_onda_medidas_calcio.map(lambda => 1/lambda );

```

```

var datos_a_ajustar_calcio =
voltajes_medidos.reduce((tuplas_longitud_onda_voltaje_calcio, b,
i) => {

    tuplas_longitud_onda_voltaje_calcio.push([inversa_de_longitudes_
de_onda_calcio[i], b]);
    return tuplas_longitud_onda_voltaje_calcio;
}, []);

var result = regression.linear(datos_a_ajustar_calcio, {
precision: 15 });
var m_calcio = result.equation[0]; // pendiente
var b_calcio = result.equation[1]; // ordenada en origen

```

Se opta por emplear la función *console.table* para la visualización tabulada de los datos utilizando la creación de objetos de datos para una correcta visualización. Para cada metal se crea un objeto material que incluye todos los datos relevantes de cálculo y posterior representación gráfica.

A continuación se explica cada uno de los atributos de los objetos.

- Material.Pendiente: pendiente calculada con función de regresión
- Material.H_Redu_Calc: constante de Planck reducida calculada con función de regresión
- Material.Error_H_Redu: Porcentaje de error entre el valor experimentado y el valor real
- Material.OrdenadaOrigen: ordenada en el origen calculada con función de regresión.
- Material.FuncionTrabajoCalc: función de trabajo calculado de cada material (mismo valor que OrdenadaOrigen en eV)
- Material.FuncionTrabajoReal: función de trabajo REAL de cada material recogido de [aquí](#)
- Material.Error_FuncionTrabajo: Porcentaje de error entre el valor experimentado y el valor real

No se calcula la función de trabajo de los metales porque después veremos que es igual a la ordenada en el origen medida en electronvoltios (eV).

```

[3] // function material(Pendiente_Vm,
OrdenadaOrigen_V,H_Reducida_Js,FuncionTrabajo_J){
//    this.Pendiente_Vm = Pendiente_Vm;
//    this.OrdenadaOrigen_V = OrdenadaOrigen_V;
//    this.H_Reducida_Js = H_Reducida_Js;
//    this.FuncionTrabajo_J = FuncionTrabajo_J;
//}
function material(Pendiente, H_Redu_Calc,
Error_H_Redu,OrdenadaOrigen,FuncionTrabajoCalc,FuncionTrabajoReal
,Error_FuncionTrabajo){
    this.Pendiente = Pendiente;
    this.H_Redu_Calc = H_Redu_Calc;
    this.Error_H_Redu = Error_H_Redu;

```

```

        this.OrdenadaOrigen = OrdenadaOrigen;
        this.FuncionTrabajoReal = FuncionTrabajoReal;
        this.FuncionTrabajoCalc = FuncionTrabajoCalc;
        this.Error_FuncionTrabajo = Error_FuncionTrabajo;
    }

    var c_electron = (-1.602766E-19); // Carga del electron

    // Multiplicamos previamente los valores fijos del calculo de la
    // constante del Planck reducida para optimizacion
    var aux_h = c_electron/(2*2.998E8*Math.PI);
    // Calculamos previamente h reducida para optimizar posteriores
    // calculos
    var h_redu = 6.63E-34/(2*Math.PI);

    var materiales = {};
    materiales.Cobre = new material(m_cobre.toExponential(5)+ " Vm",
    (aux_h*m_cobre).toExponential(5) + " Js",
    Math.round(Math.abs(((aux_h*m_cobre)-h_redu)/h_redu)*100)+"
    %",Math.round(b_cobre*100)/100 + " V",Math.round(b_cobre*100)/100
    + " eV",5.1 + " eV", Math.round(Math.abs(((b_cobre-5.1)/5.1)*100))
    + " %");
    materiales.Zinc = new material(m_zinc.toExponential(5) + " Vm",
    (aux_h*m_zinc).toExponential(5) + " Js",
    Math.round(Math.abs(((aux_h*m_zinc)-h_redu)/h_redu)*100)+"
    %",Math.round(b_zinc*100)/100 + " V",Math.round(b_zinc*100)/100 +
    " eV",3.63 + " eV", Math.round(Math.abs(((b_zinc-
    3.63)/3.63)*100)) + " %");
    materiales.Sodio = new material(m_sodio.toExponential(5) + " Vm",
    (aux_h*m_sodio).toExponential(5) + "
    Js",Math.round(Math.abs(((aux_h*m_sodio)-h_redu)/h_redu)*100)+"
    %",Math.round(b_sodio*100)/100 + " V",Math.round(b_sodio*100)/100
    + " eV",2.36 + " eV", Math.round(Math.abs(((b_sodio-
    2.36)/2.36)*100)) + " %");
    materiales.Platino = new material(m_platino.toExponential(5) + "
    Vm", (aux_h*m_platino).toExponential(5) + "
    Js",Math.round(Math.abs(((aux_h*m_platino)-h_redu)/h_redu)*100)+"
    %",Math.round(b_platino*100)/100 + "
    V",Math.round(b_platino*100)/100 + " eV",5.64 + " eV",
    Math.round(Math.abs(((b_platino-5.64)/5.64)*100)) + " %");
    materiales.Calcio = new material(m_calcio.toExponential(5) + "
    Vm", (aux_h*m_calcio).toExponential(5) + "
    Js",Math.round(Math.abs(((aux_h*m_calcio)-h_redu)/h_redu)*100)+"
    %",Math.round(b_calcio*100)/100 + "
    V",Math.round(b_calcio*100)/100 + " eV",2.87 + " eV",
    Math.round(Math.abs(((b_calcio-2.87)/2.87)*100)) + " %");
    console.log('Aproximación de cálculo de Constante de Planck
    reducida y % error experimentado');
    console.table(materiales,
    ["Pendiente","H_Redu_Calc","Error_H_Redu"]);

```

Aproximación de cálculo de Constante de Planck reducida y % error experimentado

(index)	Pendiente	H_Redu_Calc	Error_H_Redu
Cobre	'-1.38094e-6 Vm'	'1.17499e-34 Js'	'11 %'
Zinc	'-1.37311e-6 Vm'	'1.16832e-34 Js'	'11 %'
Sodio	'-1.32397e-6 Vm'	'1.12652e-34 Js'	'7 %'
Platino	'-1.41114e-6 Vm'	'1.20068e-34 Js'	'14 %'
Calcio	'-1.33708e-6 Vm'	'1.13767e-34 Js'	'8 %'

```
[4] console.log("Aproximación a cálculo de Función de Trabajo de
metales y % error experimentado");
console.table(materiales,
["OrdenadaOrigen", "FuncionTrabajoCalc", "FuncionTrabajoReal", "Error_
FuncionTrabajo"]);
```

Aproximación a cálculo de Función de Trabajo de metales y % error experimentado

(index)	OrdenadaOrigen	FuncionTrabajoCalc	FuncionTrabajoReal	Error_FuncionTrabajo
Cobre	'5.48 V'	'5.48 eV'	'5.1 eV'	'7 %'
Zinc	'5 V'	'5 eV'	'3.63 eV'	'38 %'
Sodio	'2.61 V'	'2.61 eV'	'2.36 eV'	'11 %'
Platino	'7.48 V'	'7.48 eV'	'5.64 eV'	'33 %'
Calcio	'3.3 V'	'3.3 eV'	'2.87 eV'	'15 %'

Se entiende que **el valor absoluto** de la ordenada en el origen en voltios es el mismo que la función de trabajo del metal en eV por el siguiente razonamiento:

Verificamos que ϕ viene expresado en medidas de energía:

$$\phi = b(V) \cdot e(C) \quad C = \frac{J}{V} \quad \phi = b(V) \cdot e\left(\frac{J}{V}\right) \rightarrow \phi = (b \cdot e)(J)$$

Teniendo en cuenta la siguiente conversión de energía:

$$1(eV) = 1.602177 \cdot 10^{-19}(J) \quad \rightarrow \quad 1(J) = \frac{1}{1.602177 \cdot 10^{-19}}(eV)$$

En la siguiente igualdad se aprecia que el valor número de la carga del electron $1.602177 \cdot 10^{-19}$ **se anula** con el valor número de la conversión de J en eV.

$$\phi = (b \cdot e)J \rightarrow \phi = b \cdot e \cdot \left(\frac{1}{1.602177 \cdot 10^{-19}} eV \right) \rightarrow \phi = b(eV)$$

Tras lo cual se puede apreciar que el valor de la ordenada en el origen (valor del voltaje (V)) es el mismo valor que el de la función de trabajo en eV.

Representación gráfica de los datos

Para la representación de los datos se ha optado por representar como datos discretos los valores obtenidos a través de la recogida experimental a través del applet; los datos obtenidos a través de la función de regresión se representan como una recta.

De esta manera se puede

```
[5] function Plotly(data, layout, $$) {
    $$ = $$ || global.$$;
    $.mime({ "application/vnd.plotly.v1+json": { data: data,
    layout: layout } });
}

var voltajes_de_frenado_calculados_sodio =
  inversa_de_longitudes_de_onda_sodio.map(inversa_lambda =>
    inversa_lambda * m_sodio + b_sodio);
var voltajes_de_frenado_calculados_calcio =
  inversa_de_longitudes_de_onda_calcio.map(inversa_lambda =>
    inversa_lambda * m_calcio + b_calcio);
var voltajes_de_frenado_calculados_zinc =
  inversa_de_longitudes_de_onda_zinc.map(inversa_lambda =>
    inversa_lambda * m_zinc + b_zinc);
var voltajes_de_frenado_calculados_cobre =
  inversa_de_longitudes_de_onda_cobre.map(inversa_lambda =>
    inversa_lambda * m_cobre + b_cobre);
var voltajes_de_frenado_calculados_platino =
  inversa_de_longitudes_de_onda_platino.map(inversa_lambda =>
    inversa_lambda * m_platino + b_platino);
```

Para la representación de los datos se ha optado por representar como datos discretos los valores obtenidos a través de la recogida experimental por **puntos** y los datos obtenidos a través de la función de regresión a través de una **recta**.

De esta manera se pueden apreciar visualmente de manera más intuitiva.


```
[6] var datos = [
    { x: inversa_de_longitudes_de_onda_sodio, y: voltajes_medidos,
      name: 'SODIO - Datos experimentales', mode: "markers" },
    { x: inversa_de_longitudes_de_onda_sodio, y:
      voltajes_de_frenado_calculados_sodio, name: 'SODIO - Datos
      calculados tras ajuste', mode: "lines" },
    { x: inversa_de_longitudes_de_onda_calcio, y:
      voltajes_medidos, name: 'CALCIO - Datos experimentales',mode:
      "markers" },
    { x: inversa_de_longitudes_de_onda_calcio, y:
      voltajes_de_frenado_calculados_calcio, name: 'CALCIO - Datos
      calculados tras ajuste',mode: "lines" },
    { x: inversa_de_longitudes_de_onda_zinc, y: voltajes_medidos,
      name: 'ZINC - Datos experimentales', mode: "markers" },
    { x: inversa_de_longitudes_de_onda_zinc, y:
      voltajes_de_frenado_calculados_zinc, name: 'ZINC - Datos
      calculados tras ajuste', mode: "lines" },
    { x: inversa_de_longitudes_de_onda_cobre, y: voltajes_medidos,
      name: ' COBRE - Datos experimentales',mode: "markers" },
    { x: inversa_de_longitudes_de_onda_cobre, y:
      voltajes_de_frenado_calculados_cobre, name: 'COBRE - Datos
      calculados tras ajuste', mode: "lines" },
    { x: inversa_de_longitudes_de_onda_platino, y:
      voltajes_medidos, name: 'PLATINO - Datos experimentales',mode:
      "markers" },
    { x: inversa_de_longitudes_de_onda_platino, y:
      voltajes_de_frenado_calculados_platino, name: 'PLATINO - Datos
      calculados tras ajuste',mode: "lines" }
  ];
```

Se modifican los valores iniciales del rango del eje y para definir los valores entre 0 y -2. Se añade un fondo gris claro al fondo de la gráfica y se indican tanto el dato representado por los ejes de la gráfica como las unidades utilizadas.

```
[7] var layout = {
    title: 'Ajuste Lineal para diferentes metales', plot_bgcolor:
    "rgb(238, 238, 238)",
    xaxis: { title: 'x Inversa de longitudes de onda (1/λ) (1/m)',
      titlefont: { family: 'Courier New, monospace', size: 18,
      color: '#7f7f7f',
      type: 'scatter' }
    },
    yaxis: {
      title: 'y Voltaje de frenado (V)', titlefont: {
```

```

    family: 'Courier New, monospace', size: 18, color:
    '#7f7f7f'},
    range: [0,-2]
  }
};
Plotly(datos, layout);

```

Ajuste Lineal para diferentes metales

