

# Calor Específico del Grafeno

Ricardo López & Donaldo Garrido

Tecnológico de Monterrey, Monterrey, MX

E-mail: A01066515@itesm.mx, A01275416@itesm.mx

Mayo 2021

**Resumen.** Este documento es una reproducción de los resultados presentados por Rasmita Sahoo, en su artículo llamado Lattice Specific Heat of Graphene [1]. En este trabajo se comenzó por definir la matriz dinámica basada en el modelo de masa resorte para el grafeno, considerando solamente primeros vecinos. Una vez definida la matriz dinámica, se obtuvieron los eigenvalores  $M\omega^2$  de forma numérica, utilizando el software MATLAB, dichos eigenvalores fueron utilizados para graficar las seis ramas de la relación de dispersión y hacer el cálculo del calor específico del grafeno.

## 1. Introducción

El grafeno fue el primer cristal de dos dimensiones en ser descubierto[1]. Esta capa de grafito es la base para obtener nanotubos de carbono (NTC), los cuales siguen siendo de gran interés en la comunidad científica. Las relaciones de dispersión así como el calor específico del grafeno han sido calculados por métodos sofisticados, sin embargo en este trabajo se busca aproximar los resultados por un modelo sencillo de masa resorte de las vibraciones de la red cristalina considerando únicamente las interacciones de primeros vecinos. Rasmita Sahoo muestra en su artículo Lattice Specific Heat of Graphene, la comparación entre las relaciones de dispersión obtenidas con el modelo de masa resortes y datos experimentales, la comparación mostró que la exactitud aumenta al incluir más interacciones de vecinos cercanos.[2]

Este documento está organizado de la siguiente manera. En la segunda sección se presenta el método utilizado para realizar los cálculos de las relaciones de dispersión. La tercera sección muestra las gráficas generadas para las relaciones de dispersión considerando interacciones de primeros vecinos. Finalmente, en la cuarta sección presentamos el método y los resultados obtenidos del cálculo del calor específico de forma numérica.

## 2. Cálculo de las relaciones de dispersión

Para el cálculo de todas las cantidades que se presentarán en este trabajo, se realizó un acercamiento numérico en su mayoría. Este tratamiento numérico se decidió por la facilidad y velocidad de cómputo que nos brindan los métodos numéricos, así como por la belleza de su simplicidad.

Se partió de la definición plasmada en el artículo *Lattice Specific Heat of Graphene* [3] de la matriz dinámica:

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}, \quad (1)$$

con elementos

$$D_{11} = C_1 + C_2 + C_3 \quad (2)$$

$$D_{12} = -e^{\frac{i}{3}\vec{k}\cdot(\vec{a}_1+\vec{a}_2)} \left( C_1 + C_2 e^{-i\vec{k}\cdot\vec{a}_2} + C_3 e^{-i\vec{k}\cdot\vec{a}_1} \right) \quad (3)$$

$$D_{21} = -e^{-\frac{i}{3}\vec{k}\cdot(\vec{a}_1+\vec{a}_2)} \left( C_1 + C_2 e^{i\vec{k}\cdot\vec{a}_2} + C_3 e^{i\vec{k}\cdot\vec{a}_1} \right) \quad (4)$$

$$D_{22} = C_1 + C_2 + C_3 \quad (5)$$

$$(6)$$

Donde  $C_1$ ,  $C_2$  y  $C_3$  no son constantes, sino matrices de tres elementos por tres elementos. Al introducir estas matrices directamente en la expresión que tenemos para  $D$  obtenemos una matriz de  $6 \times 6$  elementos con dependencia en  $k(k_x, k_y) = [k_x, k_y]$ . Según el tratamiento numérico, definimos un espacio para  $k_x$  y otro para  $k_y$  e ir procesándolos en la matriz dinámica. Una vez expresado cada vector con componentes  $k_x, k_y$ , se evaluaron los valores propios, con la función de matlab  $\text{eig}(v)$  de la matriz dinámica para obtener cada uno de los resultados para  $M\omega^2$ . Con esto, obtuvimos 6 arreglos con los distintos valores de  $\omega$ , uno por cada vecino cercano.

## 3. Gráficas de las relaciones de dispersión: primeros vecinos

Para poder graficar las relaciones de dispersión en las distintas zonas  $\Gamma, K, M$  que se ven en la figura 1 y que fue tomada del artículo *Analytical calculation of electron group velocity surfaces in uniform strained graphene* [1] de Gómez-Arias et. al. se tomó la misma función que se describió en la sección anterior y que calcula los valores propios dado un vector  $\vec{k}$ .

Para la primera parte, que es del punto  $\Gamma$  al  $M$ , la componente  $k_y$  se mantuvo como 0, y  $k_x \in \left[0, \frac{2\pi}{\sqrt{3}a}\right]$ . Para pasar del punto  $M$  al  $K$ , se mantuvo la componente  $k_x = \frac{2\pi}{\sqrt{3}a}$  y  $k_y \in \left[0, \frac{2\pi}{3a}\right]$ . Por último, para pasar de  $K$  a  $\Gamma$ , bastó con ir reduciendo sobre la línea recta correspondiente del valor máximo alcanzado para cada componente hasta 0.

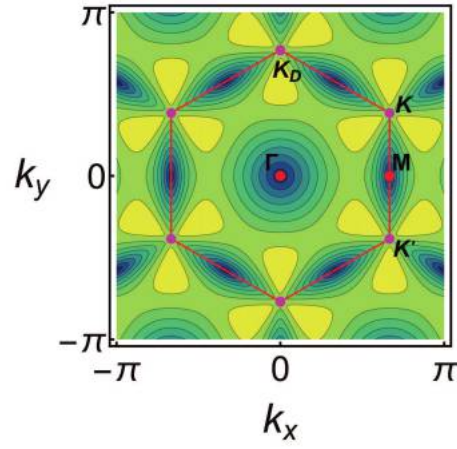


Figura 1: Direcciones en las que se debe variar  $\vec{k} = \vec{k}(k_x, k_y)$ . Imagen tomada de Gómez-Arias et. al.

Posteriormente, tras la graficación de cada uno de los tres segmentos de la curva, obtuvimos la curva completa, que es muy parecida a la que muestra Sahoo en el mismo artículo del 2012. La curva obtenida se ve en la figura 2.

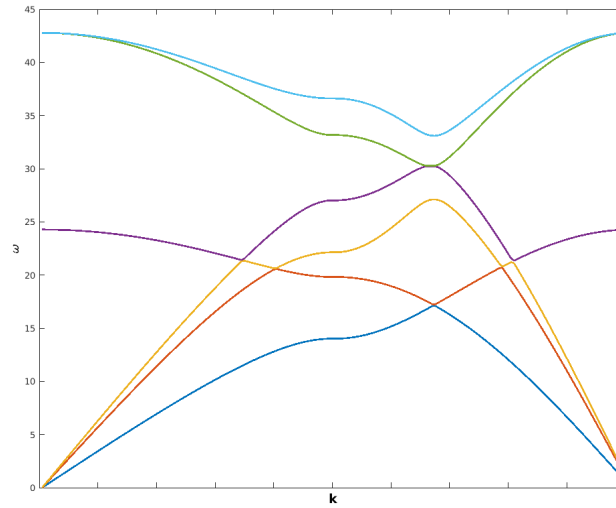


Figura 2: Seis relaciones de dispersión en concordancia a las 6 soluciones que se obtienen de una matriz de  $6 \times 6$ . Cada solución se varió en las direcciones requeridas.

#### 4. Cálculo del calor específico

El calor específico del grafeno puede ser calculado utilizando la ec.7 la cual debe ser integrada sobre el área de toda la zona de Brillouin. La ec.7 fue evaluada en un cuarto de la zona de Brillouin y posteriormente multiplicada por 4. El cuarto de la zona de Brillouin es un trapecio, dicha zona es mostrada en la fig.3 con el trapecio encerrado en el contorno rojo. El trapecio delimita la zona de integración con  $k_x \in \left[0, \frac{2\pi}{\sqrt{3}a}\right]$  y  $k_y \in \left[0, \frac{4\pi}{3a} - \frac{k_x}{\sqrt{3}}\right]$

$$C_{ph} = 4 \frac{\partial}{\partial T} \sum_s \int_0^{\frac{4\pi}{3a} - \frac{k_x}{\sqrt{3}}} dk_y \int_0^{\frac{2\pi}{\sqrt{3}a}} dk_x \frac{\hbar \omega_s(k_x, k_y)}{(2\pi)^2 e^{\hbar \omega_s(k_x, k_y)/k_B T} - 1} \quad (7)$$

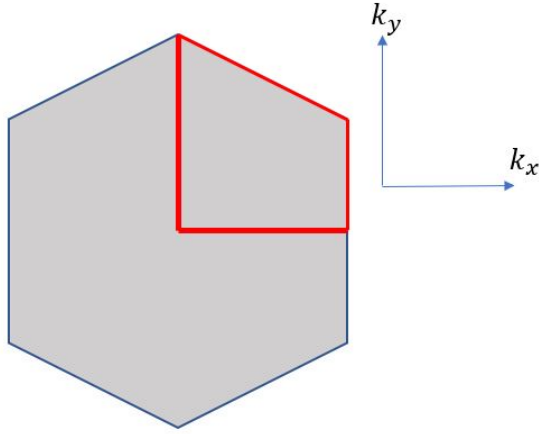


Figura 3: Zona de Brillouin del grafeno

Al igual que los cálculos anteriormente presentados, la integral del calor específico fue obtenida de forma numérica. Para poder realizar esta integración numérica, primero se calculó cada una de las seis ramas de dispersión  $\omega_s(k_x, k_y)$  evaluadas en cada una de las combinaciones de  $k_x$  y  $k_y$  dentro de la región de integración.

El algoritmo utilizado para obtener la curva del calor específico puede ser resumido en los siguientes pasos. En el código de MATLAB, presetado en Anexos, se definió un vector de temperatura  $T$ . Después, de forma iterativa el primer valor de  $T$  se multiplicó por el integrando. La integral fue calculada utilizando un algoritmo trapezoidal, se realizó la sumatoria sobre las seis ramas de dispersión, se derivó numéricamente con respecto a  $T$  y el resultado fue guardado en un vector definido como  $C_{ph}$ , una vez guardado el valor, se repetía el mismo proceso pero ahora multiplicando el integrando por el segundo valor del vector  $T$  y así sucesivamente hasta recorrer todos los valores de la temperatura. El resultado de este método numérico es presentado en la fig.4 donde se puede observar que la curva del calor específico es bastante similar a otras presentadas en la literatura.

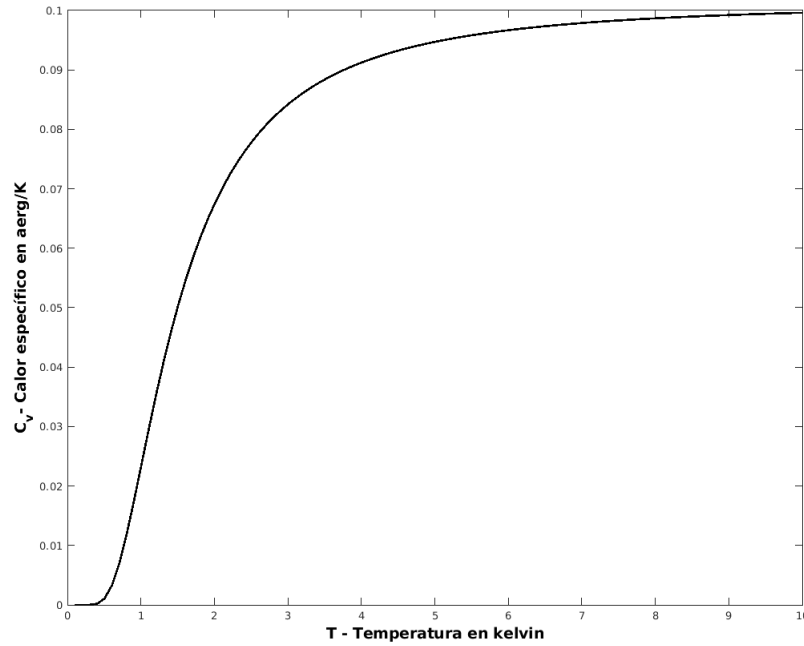


Figura 4: Calor específico del grafeno

## 5. Conclusiones

En la realización de este trabajo, observamos que un modelo de masa resorte para las vibraciones de la red, considerando solo interacciones entre primeros vecinos, se acerca al comportamiento de los resultados mostrados por Rasmita Sahoo[2]. Es importante recordar que a una mayor cantidad de interacciones entre vecinos cercanos, la exactitud de los resultados es mejorada. A pesar de que en este proyecto se obtuvo un comportamiento bastante similar a los presentados en otras fuentes bibliográficas, nuestras unidades difieren, ya que varias de las constantes del modelo fueron ajustadas para tener una mejor visualización de las curvas. Además, los cálculos numéricos de este modelo son bastante sensibles a los cambios de los valores de las constantes, encontrar estos valores en las unidades deseadas no resultó ser una tarea sencilla.

## 6. Referencias

- [1] W. A. Gómez-Arias and G. G. Naumis, “Analytical calculation of electron group velocity surfaces in uniform strained graphene,” *International Journal of Modern Physics B*, vol. 30, no. 03, p. 1550263, 2016.
- [2] R. Sahoo and R. R. Mishra, “Lattice specific heat of graphene,” pp. 186–188, 2018.
- [3] R. Sahoo and R. Ranjan, “Phonon dispersion of graphene revisited,” *Journal of Experimental and Theoretical Physics*, 2012.

## Apéndice A: Código para la integración del calor específico

[https://github.com/Riloro/specific\\_heat\\_of\\_graphene/blob/main/SecondProyect.m](https://github.com/Riloro/specific_heat_of_graphene/blob/main/SecondProyect.m)

```

1 close all
2
3 n = 100; % Número de puntos para integrar y para definir espacios
4
5 %Definición de zona para integrar y tomar kx y ky
6 kxVal = linspace(0,2*pi/(sqrt(3)*a),n);
7 kyVal = linspace(0,4*pi/(3*a),n);
8
9 RelDispersion = zeros(length(kxVal),length(kyVal),6);
10
11 %Cálculo de los eigenvalores para cada combinación de kx y ky
12 %Según los limites de integración que se usarán
13 for ii = 1:length(kxVal)
14     for jj = 1:length(kyVal)
15         kVec = [kxVal(ii), kyVal(jj)];
16         RelDispersion(ii,jj,:) = omegas(kVec);
17     end
18 end
19
20 T = linspace(0,500,n); %Vector de la temperatura
21 h = 1.05457e-34; %Constante de Planck
22 kB = 1.380649e-23; %Constante de Boltzmann
23
24 Cv = zeros(1,n); %Inicialización de Cv
25
26 for count=1:6 %Ciclo sobre cada una de las relaciones de dispersión
27     Integrand = zeros(n,n,n); %Inicialización: matriz del integrando
28
29     %Cálculo de los integrandos a través de una matriz 3D
30     %Cada valor de T, tiene un arreglo de (kx,ky)
31     for kk=1:n
32         if kk ~= 1
33             Integrand(:, :, kk) = (h^2/kB)*(exp(h.*RelDispersion(:, :,
count)/(kB*T(kk))).*RelDispersion(:, :, count).^2)/...
34                 ((exp(h.*RelDispersion(:, :, count)/(kB*T(kk)))-1).^2*T(
kk).^2);
35         else
36             Integrand(:, :, kk) = zeros(n);
37         end
38     end
39     prueba = Integrand(:, :, 50); %Monitoreo
40
41     %Parámetros para delimitación de la integración
42     alpha = pi/3;
43     trig = tan(alpha);
44     %Conversión del rectángulo de integración al trapecio
45     for aa = 1:n
46         for ii = 1:n
47             for jj = 1:n

```

```

48         comparison = max(kyVal)/2+(max(kxVal)-kxVal(ii))/trig;
49         if kyVal(jj) > comparison
50             Integrand(ii,jj,aa) = 0;
51         end
52     end
53 end
54 end
55 recorte = Integrand(:,:,50); %Monitoreo de correcta acotación
56
57 %Proceso de integración con técnica de trapecio
58 Int1 = zeros(1,n);
59 for nn = 1:n
60     Int1(nn) = trapz(kxVal,trapz(kyVal,Integrand(:,:,nn),2))*10^5;
61     disp(Int1(nn))
62 end
63 %Sumatoria del resultado de cada relación de dispersión
64 Cv = Cv+Int1;
65 end
66
67 %Gráfica de Cv en función de la Temperatura
68 plot(T,4*Cv/(2*pi)^2)
69
70
71 function w = omegas(k)
72 M = 1.9945e-23; %Masa del carobono [KG]
73 a = 2.46e-10; %Distancia entre primeros vecinos
74 a1 = a*[sqrt(3)/2, 1/2];
75 a2 = a*[sqrt(3)/2, -1/2];
76
77 %Constantes de fuerza
78 phi_r = 36.5;
79 phi_ti = 24.5;
80 phi_to = 9.82;
81
82 %Vectores constantes
83 C1 = [phi_r, 0, 0;0, phi_ti, 0; 0, 0, phi_to]; %Matr
84 C2 = (1/4)*[phi_r+3*phi_ti, sqrt(3)*(phi_ti-phi_r), 0; sqrt(3)*(phi_ti-
    -phi_r), 3*phi_r+phi_ti, 0; 0, 0, 4*phi_to];
85 C3 = (1/4)*[phi_r+3*phi_ti, sqrt(3)*(phi_r-phi_ti), 0; sqrt(3)*(phi_r-
    -phi_ti), 3*phi_r+phi_ti, 0; 0, 0, 4*phi_to];
86
87 %Definición de la matriz dinámica
88 D = [C1+C2+C3, -exp(1i*dot(k,(a1+a2)/3))*(C1+C2*exp(-1i*dot(k,a2))+C3
    *exp(-1i*dot(k,a1))); ...
89     -exp(-1i*dot(k,(a1+a2)/3))*(C1+C2*exp(1i*dot(k,a2))+C3*exp(1i*dot(k,
    a1))), C1+C2+C3];
90
91 %Cálculo de eigenvalores
92 w = sqrt(real(eig(D)));
93 end

```

Listing 1: Código en Matlab: Calor Específico de Grafeno.

## Apéndice B: Código para la obtención de las curvas de dispersión

[https://github.com/Riloro/specific\\_heat\\_of\\_graphene/blob/main/SecProy.](https://github.com/Riloro/specific_heat_of_graphene/blob/main/SecProy.m)

m

```

1 close all
2
3 n = 100;
4
5 kxVal = linspace(0,2*pi/(sqrt(3)*a),n);
6 kyVal = linspace(0,4*pi/(3*a),n);
7
8 C = 0;
9
10 RelDispersion = zeros(length(kxVal),length(kyVal),6);
11
12 for ii = 1:length(kxVal)
13     for jj = 1:length(kyVal)
14         kVec = [kxVal(ii), kyVal(jj)];
15         RelDispersion(ii,jj,:) = omegas(kVec);
16     end
17 end
18
19 T = linspace(0,10,n);
20 h = 1;%6.63e-34;
21 kB = 1;%1.38e-23;
22
23 %Cálculo de la integral independizando la temperatura
24 for count=1:6
25
26     Integrand = h.*RelDispersion(:,:,count).^2/(exp(h.*RelDispersion
27 (:,:,count))/(kB))-1);
28
29     alpha = pi/3;
30     trig = tan(alpha);
31
32     for ii = 1:n
33         for jj = 1:n
34             comparison = max(kyVal)/2+(max(kxVal)-kxVal(ii))/trig;
35             if kyVal(jj) > comparison
36                 Integrand(ii,jj) = 0;
37             end
38         end
39     end
40     Int1 = real(trapz(kxVal,trapz(kyVal,Integrand,2)));
41     C = (C+Int1)/(1e19);
42 end
43 %C = C*10^(11);
44 disp(C)
45 %Gráfica de Cv
46 Cv = (h^2/kB).*(exp(h.*C./(kB.*T)).*C.^2)/(((exp(h.*C./(kB*T))-1).^2.*
47 T.^2);
48 figure(1)
49 plot(T,4*Cv/(2*pi)^2,'LineWidth',2,'Color','k')

```



```

48 xlabel('T - Temperatura en kelvin','FontSize',14,'FontWeight','bold','
    Color','k')
49 ylabel('C_v- Calor específico en aerg/K','FontSize',14,'FontWeight','
    bold','Color','k')
50
51 %=====Variación de direcciones, obtención gráfica
    =====
52 %De G a M
53 GaM_x = kxVal;
54 GaM_y = zeros(1,n);
55
56 GaM = zeros(6,n);
57 for ii = 1:n
58     GaM(:,ii) = omegas([GaM_x(ii),GaM_y(ii)]);
59 end
60 figure(2)
61 plot(GaM_x,real(GaM))
62
63
64 %De M a K
65 MaK_x = 2*pi/(sqrt(3)*a)*ones(1,n);
66 MaK_y = kyVal/0.68;
67
68 MaK = zeros(6,n);
69 for ii = 1:n
70     MaK(:,ii) = omegas([MaK_x(ii),MaK_y(ii)]);
71 end
72 figure(3)
73 plot(MaK_y,real(MaK))
74 lim = max(MaK_y);
75
76
77 %De K a G
78 KaG_x = flip(kxVal);
79 KaG_y = KaG_x/tan(pi/6);
80
81 hh = 1;
82
83 for ii = 1:n
84     if sqrt(KaG_x(ii)^2+KaG_y(ii)^2)>max(KaG_x)
85         KaG(:,hh) = omegas([KaG_x(ii),KaG_y(ii)]);
86         hh=hh+1;
87     end
88 end
89 figure(4)
90 plot(real(KaG(1,:)))
91
92
93
94 figure(5)
95 %Gráfica de las 6 relaciones de dispersión
96 for ii = 1:6
97     DispRelPlotY = [real(GaM(ii,:)),real(MaK(ii,:))];
98     plot(DispRelPlotY,'LineWidth',2)

```

```

99     hold on
100 end
101 xlabel('k', 'FontSize',14, 'FontWeight', 'bold', 'Color', 'k')
102 ylabel('\omega', 'FontSize',14, 'FontWeight', 'bold', 'Color', 'k')
103
104 set(gca, 'xticklabel', [])
105
106 function w = omegas(k)
107 a = 2.46;
108 a1 = a*[sqrt(3)/2, 1/2];
109 a2 = a*[sqrt(3)/2, -1/2];
110 phi_r = 365;
111 phi_ti = 245;
112 phi_to = 98.2;
113
114 C1 = [phi_r, 0, 0; 0, phi_ti, 0; 0, 0, phi_to];
115 C2 = (1/4)*[phi_r+3*phi_ti, sqrt(3)*(phi_ti-phi_r), 0; sqrt(3)*(phi_ti-
    phi_r), 3*phi_r+phi_ti, 0; 0, 0, 4*phi_to];
116 C3 = (1/4)*[phi_r+3*phi_ti, sqrt(3)*(phi_r-phi_ti), 0; sqrt(3)*(phi_r-
    phi_ti), 3*phi_r+phi_ti, 0; 0, 0, 4*phi_to];
117
118 D = [C1+C2+C3, -exp(1i*dot(k, (a1+a2)/3))*(C1+C2*exp(-1i*dot(k, a2))+C3
    *exp(-1i*dot(k, a1))); ...
    -exp(-1i*dot(k, (a1+a2)/3))*(C1+C2*exp(1i*dot(k, a2))+C3*exp(1i*dot(
    k, a1))), C1+C2+C3];
119
120 w = sqrt(real(eig(D)));
121 end

```

Listing 2: Código en Matlab: Curvas de dispersión.