

Blackbody radiation

December 16, 2015

1 La radiación de cuerpo negro - Fenómeno cuántico

Juan José Cadavid Muñoz - Semillero de Física Teórica y Experimental - Universidad EAFIT 2015.

1.1 Sobre el origen del fenómeno

La radiación de cuerpo fue un problema que surge en un contexto donde la física clásica está en auge. El mundo podía ser resumido por los deterministas en: La mecánica, termodinámica y electromagnetismo. Tras el éxito del modelo electromagnético de Maxwell, se intentó dar respuesta a cómo los cuerpos calientes emitían radiación.

Una de las primeras observaciones fue realizada en 1912 por Thomas Wedgwood, quien observó que al tomar diferentes cuerpos sólidos de diferente composición, todos se tornaban de color rojizo a la misma temperatura. Más de 50 años después, los avances en la espectroscopía permitieron detallar que todo cuerpo luminoso al calentarse emitía un espectro continuo.

1.2 La idea de absorción y emisión

En 1859, Gustav Kirchhoff demostró un teorema derivado de la termodinámica, en el que define que si un cuerpo está en equilibrio térmico con la radiación, la potencia de emisión (E_f [W/m^2]) era proporcional a la potencia de absorción, estableciendo la expresión:

$$E_f = J(f, T)A_f \quad (1)$$

donde $J(f, T)$ es una función de emisión universal que depende de la frecuencia de la radiación f y la temperatura T del cuerpo. A_f es la capacidad de absorción del cuerpo. Cuando el cuerpo tiene toda la absorción, $A_f = 1$. El teorema describe que si un cuerpo a una temperatura fija, éste cuerpo emite a una cierta frecuencia que es exacta a la que puede ser absorbida. Éste planteamiento, permitió reversar el problema del cuerpo negro para estudiar no cuánto emite sino cuánto absorbe. De ésta manera se plantea la idea de una cavidad de cuerpo negro.

1.3 La cavidad de cuerpo negro

Un cuerpo es un objeto idealizado en la capacidad de absorción. De ahí deriva su nombre, al tener capacidad absoluta de absorción $A_f = 1$ no hay luz que pueda ser percibida y por ende es negro. Pero al ser un absorbente absoluto, por el teorema de Kirchhoff, también es un emisor absoluto un cuerpo blanco, por lo que ambos problemas son equivalentes.

Una cavidad de cuerpo negro, es un sólido hueco que tiene las mismas propiedades del cuerpo negro, y en la superficie posee un orificio. Bajo la idea de Kirchhoff, toda la radiación que emite un cuerpo caliente saldrá de la cavidad, y por ende toda la radiación que incide sobre la cavidad será absorbida de una manera particular. Kirchhoff retó a experimentalistas y teóricos a entender y encontrar la forma de emisión/absorción en lo que él denominó como *hohlraumstrahlung*. Retó que guiaría posteriormente al inicio de la teoría cuántica.

Entonces la nueva pregunta que se plantea es sobre ¿Cómo la cavidad absorbe toda la radiación? La respuesta a ésta pregunta sería también equivalente a ¿Cómo la cavidad emite toda la radiación? Más adelante se discutirá las aproximaciones que se realizaron para resolver ésta pregunta.

1.4 Absorción de la radiación y naturaleza cuántica

En el contexto del problema, se tenía en conocimiento que el calor podía generar vibraciones en las moléculas y átomos de un sólido, y también se conocía que los átomos eran un conjunto de cargas eléctricas. Sin embargo los efectos atómicos se desconocía, principalmente porque los modelos clásicos fallaban en predecir éstos efectos, por lo que en sí del átomo se conocía parte de su naturaleza eléctrica. Del análisis en las ecuaciones de Maxwell se explicaba que si se tienen portadores de cargas oscilantes, campos electromagnéticos variables en el tiempo eran generados.

Con éstas dos ideas, en el panorama pintaba una idea de que si un cuerpo era calentado, las vibraciones térmicas en la escala atómica, era capaz de que los portadores de carga oscilaban y por ende eran los generadores de perturbaciones ondulatorias electromagnéticas. Las herramientas matemáticas permitían una aproximación en la combinación de la teoría electromagnética y la termodinámica clásica, y por su parte Kirchhoff había planteado un problema que llevaría la combinación de éstas nociones para entender la emisión de la cavidad. Lo anterior era el panorama que tenían los físicos a finales del siglo XIX para éste problema.

La primera aproximación fue experimental. Obtener una curva característica de un cuerpo negro, que con la ayuda de la ya desarrollada espectroscopia, permitiría establecer el comportamiento de los datos, por lo que se conocía una forma experimental del espectro de emisión de diferentes longitudes de onda, para algunas temperaturas.

El planteamiento entonces vendría en encontrar una forma analítica que ajustara los datos, por lo que con los hallazgos experimentales surgieron dos observaciones importantes:

1.4.1 Ley de Stefan - Boltzmann

Una de las contribuciones importantes sobre el estudio del cuerpo negro, fue la ley de Stefan propuesta por Josef Stefan, quien experimentalmente encontró una forma para calcular la cantidad de potencia total emitida por unidad de área en un cuerpo negro. La relación describe:

$$E_{total} = \int_0^{\infty} E_f df = \sigma T^4 \quad (2)$$

Posteriormente, Ludwig Boltzmann llega a éste mismo resultado, combinando la termodinámica y las ecuaciones de Maxwell, por lo que ésta ley se le conoce también como la ley de Stefan-Boltzmann. La estrategia de Boltzmann fue considerar la densidad de energía de un campo electromagnético en un volumen y unirla con la densidad de entropía. En éste caso Boltzmann estaba resolviendo el problema termodinámico de un pistón con un gas, pero consideraba la radiación como el gas en el pistón. El detalle de la deducción puede encontrarse en éste vínculo: [Black Body Radiation](#).

1.4.2 Ley de desplazamiento de Wien

La segunda contribución al problema fue propuesta por Wilhelm Wien, siguiendo en parte a la idea de tratar la densidad de energía de la cavidad. La idea de Wien era describir el corrimiento del pico espectral mayor respecto al aumento de temperatura y de ésta manera observar cómo el cuerpo cambiaba el color que era emitido, comenzando por el rojo, que sería el primer color que se vería temperaturas de $900K$ hasta llegar a temperaturas de $10000K$ para observar el azul. La ley establece:

$$f_{max} \propto T \therefore f_{max} = 5.879 \times 10^{10} T \quad (3)$$

donde $5.879 \times 10^{10} [Hz T^{-1}]$ es la constante de proporción experimental propuesta, tras el barrido de temperaturas en el experimento de Wien. Para 1893, él deduce analíticamente la conjetura de la Ley de radiación de Wien estableciendo la relación entre la densidad de energía como:

$$\rho(f, T) = \alpha f^3 e^{-\beta f T^{-1}} \quad (4)$$

1.5 El modelo teórico y los hallazgos experimentales

En 1895, Wien y Lummer realizaron un experimento en la universidad de Berlín, con el que contaban con un horno virtualmente cerrado, salvo por el pequeño orificio para dejar salir la radiación y de ésta forma obtener una aproximación a la cavidad de cuerpo negro.

En el experimento, se instaló una red de difracción a la salida de la cavidad para observar las diferentes longitudes de ondas emitidas por la cavidad. Para determinar la densidad de energía un detector de intensidad era desplazado en cada longitud difractada para determinar su valor de intensidad. En el experimento sólo se midió la radiación visible, eliminando las altas frecuencias a través de reflexiones en diferentes cristales, de ésta manera tener fiabilidad en la energía medida correspondía al espectro visible.

La medición del detector no era directamente sobre la densidad de energía $\rho(f, T)$ sino sobre una fracción de ésta. Debido a que hay una cantidad de entrada y salida de densidad, sólo la mitad salía de la cavidad, al igual que una porción atravesaba el área transversal A , implicando que la intensidad registrada dependía del orificio.

Otro factor era que la radiación que era emitida en la cavidad, no necesariamente pasaba a través del orificio, sino que era un tipo de emisión polar que dependía de un factor $\cos(\theta)$, siendo θ el ángulo entre el eje del orificio y un punto de emisión. Al integrar la forma de emisión se obtiene un factor de $1/4$. Por consiguiente la distribución de energía vista a la salida del horno correspondía a:

$$J(f, T) = \frac{Ac}{4} \rho(f, T) \quad (5)$$

Otro experimento realizado por el espectroscopista alemán Friedrich Paschen, con cuerpos a $1500K$ permitió realizar un barrido espectral de pequeñas longitudes de onda hasta el espectro del infrarrojo en longitudes de onda entre $1 - 4\mu m$; principalmente en las regiones de la máxima frecuencia de emisión. Con los valores experimentales obtenidos y la ley de Wien, se encontró buena correspondencia en el modelo y los datos experimentales en ésta zona.

En 1900 se realizaron dos nuevos experimentos. Lummer y Pringsheim aumentaron el rango del espectro de emisión hasta la longitud de onda de $18\mu m$. Posteriormente Rubens y Kurlbaum llevaron hasta longitudes de onda de $60\mu m$. Éstas regiones en el experimento, a diferencia de la zona de la versión de Paschen, estaban mucho más alejadas de la frecuencia de máxima emisión. Al contrastar los valores con el modelo de Wien, observaron que no había un ajuste correcto para éstas regiones.

La parte experimental demostró que todavía no había un análisis completo en los modelos teóricos. Sin embargo éste paso fue pionero en lo que llevaría a la solución, al menos analítica del enigma del cuerpo negro. Para ésta época surgen dos prominentes figuras de la física en la búsqueda de la respuesta: John William Strutt (Baron Rayleigh) y Max Planck.

1.6 El comportamiento discreto de la materia en la equipartición de energía

La mayoría de los experimentalistas y teóricos habían centrado su enfoque en la forma de la radiación emitida. Rayleigh había centrado su idea en el comportamiento electromagnético en el interior de la cavidad, y para 1900 ya comenzado a plantear el comportamiento de la radiación ondulatoria y la interacción con las paredes internas.

1.6.1 La deducción electromagnética de una cavidad resonante de Rayleigh

La propuesta de Rayleigh partía de entender la radiación como un conjunto de ondas estacionarias encerradas en una cavidad cúbica de longitud a . Por consiguiente un oscilador electromagnético. La pregunta a resolver sería ¿Cuántas diferentes ondas estacionarias hay en la cavidad? La idea era similar al problema que él había resuelto 25 años antes, sobre las ondas sonoras en una cavidad cúbica.

El oscilador de Rayleigh consideró que la radiación electromagnética y sus componentes, se podrían describir como un oscilador clásico unidimensional. La cantidad y tipo de ondas posibles debían satisfacer

el criterio de los modos normales posibles en la cavidad y las frecuencias permitidas de oscilación. Ésta restricción se establece, respectivamente, como:

$$(k_x, k_y, k_z) = \frac{\pi}{a}(l, m, n) \therefore \frac{c}{2\pi}(k_x^2 + k_y^2 + k_z^2)^{1/2} \quad (6)$$

Lo que Rayleigh encontró fue una expresión para la densidad de estados de radiación en el horno; la cantidad de modos permisibles en la cavidad:

$$N(f)df = \frac{8a^3\pi f^2}{c^3}df \quad (7)$$

1.6.2 La deducción termodinámica y de la mecánica estadística de Planck

Planck por su parte, no tomó una aproximación tan directa como Rayleigh. En lugar de retomar el electromagnetismo, Plack desde hacía unos años trabajaba en un complejo y abstracto modelo conformado por la termodinámica, las leyes de la entropía, la mecánica estadística y parte del electromagnetismo.

Planck conocía la correspondencia de la ley de Wien para altas frecuencias por lo que intentó deducir ésta ley partiendo principalmente de la mecánica estadística y las leyes de Maxwell. Finalmente Planck encuentra un modelo analítico que tenía correspondencia en los experimentos de Rubens. Sin embargo la interpretación de Planck sobre el fenómeno le daría más peso y ayudará a plantear ideas del comportamiento.

La idea de que en las paredes de la cavidad habían osciladores y cada uno a diferentes frecuencias. La radiación que emitían era igual en frecuencia a la frecuencia de vibración. Al aplicar la restricción del equilibrio térmico en la cavidad, encuentra una forma general de la densidad de potencia espectral:

$$\rho(f, T)df = \bar{E}N(f)df \quad (8)$$

Ésta forma describe que la densidad volumétrica de energía depende de la forma y cantidad de oscilaciones en los resonadores en las paredes de la cavidad, $N(f)$, evaluada entre las frecuencias f y Δf . El segundo factor de dependencia es la energía promedio \bar{E} en cada resonador. Planck demuestra con las restricciones termodinámicas que el número de oscilaciones posibles en la cavidad unitaria es:

$$N(f)df = \frac{8\pi f^2}{c^3}df \quad (9)$$

Tanto Rayleigh como Planck acuerdan en la forma en que el comportamiento ondulatorio debe comportarse. Aunque ambos desconocían que habían llegado al mismo resultado y por diferentes razonamientos. Al sustituir el número de modos, la densidad de energía en la cavidad quedaría de la forma:

$$\rho(f, T)df = \bar{E} \frac{8\pi f^2}{c^3}df \quad (10)$$

Lo que quedaría pendiente sería la distribución de energía en los resonadores de la cavidad. Como se verá a continuación sobre la energía hay dos percepciones: Energía continua y energía discreta.

1.6.3 La catástrofe ultravioleta y la ley de Rayleigh-Jeans

Rayleigh y el astrónomo-físico inglés James Jeans, determinaron el término de la energía utilizando la distribución de Maxwell-Boltzaman. El modelo describe cuál es la probabilidad P de encontrar un sistema individual en una energía E , cuando éste hace parte de un conjunto de sistemas a una temperatura T . Planteando:

$$P(E) = P_0 e^{-(E-E_0)/k_B T} \quad (11)$$

donde P_0 es la probabilidad de encontrar el sistema en la energía inicial E_0 . Por consiguiente la energía promedio normalizada en cada sistema sería:

$$\bar{E} = \frac{\sum E P(E)}{\sum P(E)} \quad (12)$$

Con éste planteamiento Rayleigh y Jeans consideraron una forma clásica en que cualquier resonador podría tener ésta energía a cualquier frecuencia. Por consiguiente se planteó una forma continua de la energía:

$$\bar{E} = \frac{\int_0^{\infty} E e^{-E/k_B T} dE}{\int_0^{\infty} e^{-E/k_B T} dE} = k_B T \quad (13)$$

Con ésta consideración, encontraron que ésta energía en cada oscilador era dependiente de la temperatura e independiente de la frecuencia. Por lo que cada resonador tendría la misma energía independiente de las oscilaciones. Con éste término, la densidad de energía de la cavidad sería:

$$\rho(f, T) df = k_B T \frac{8\pi f^2}{c^3} df \quad (14)$$

Sin embargo al evaluar éste modelo para altas frecuencias, en cuerpos a más de 1000K, el modelo diverge a medida que la longitud de onda se aproxima a 0, prediciendo energía infinita en la región del ultravioleta. Posteriormente Paul Ehrenfest se refirió a éste como la catástrofe-ultravioleta. El modelo para bajas frecuencias tiene correspondencia.

1.6.4 La energía discreta del oscilador y la “suerte” de Planck

Planck por su parte, seguía trabajando en las consideraciones de la energía de los resonadores, apoyado de la mecánica estadística. Su idea principal era en mantener los estados discretos. Planck plantea la energía de manera discreta:

$$\bar{E} = \frac{\sum_{n=0}^{\infty} (nhf)(P_0 e^{-nhf/k_B T})}{\sum_{n=0}^{\infty} P_0 e^{-nhf/k_B T}} = \frac{hf(1 - e^{-hf/k_B T})(e^{-hf/k_B T})}{(1 - e^{-hf/k_B T})^2} = \frac{(hf)(e^{-hf/k_B T})}{(1 - e^{-hf/k_B T})} = \frac{hf}{e^{-hf/k_B T} - 1} \quad (15)$$

El modelo considera restricciones no unicamente en la frecuencia sino en una cantidad de energía. El sistema no podría tener más energía que hf . Cada elemento del sistema debería tener una energía asociada que depende; a diferencia del modelo de Rayleigh-Jeans, de la frecuencia y la temperatura. Planck con ésta forma de la energía, interpola la información experimental de Rubens y Kurlbaum del *Restrahlen* (experimento de los rayos residuales) junto con el modelo de las altas frecuencias de Wien.

Un domingo de octubre de 1900, Planck escribe en una carta dirigida Rubens en la que contenía la formula que se conoce como la ley de emisión de Planck:

$$\rho(f, T) df = \frac{8\pi f^2}{c^3} \frac{hf}{e^{-hf/k_B T} - 1} df \quad (16)$$

El modelo, matemáticamente, resuelve la catástrofe ultravioleta al atenuar el término f^2 , para las altas frecuencias. Rubens verifica el total acuerdo con los datos experimentales.

En 1918 el comité del premio Nobel decide que Max Planck es merecedor del premio nobel por “contribuciones en el avance de la física y el descubrimiento de la energía cuantizada”. Planck en 1920 en su discurso menciona su insatisfacción. Si bien el modelo ajusta el experimento perfectamente, no fue más que una “conjetura con suerte”. La interpretación física sobre el cuerpo negro, Planck aún la desconocía.

1.7 El gas de radiación de Einstein.

En 1905, Einstein promueve una de las revoluciones más grandes de la física. En éste año, Einstein nuevamente retoma el estudio sobre el cuerpo negro, tras casi 5 años de silencio en el tema. Tomando los resultados de Rayleigh para regiones de bajas frecuencias y los resultados de Planck para las altas frecuencias, se percata de una analogía con la densidad de energía en la distribución clásica de los gases:

$$f(E) = \left[4\pi \left(\frac{m}{2\pi k_B T} \right)^{3/2} v^2 \right] E e^{-E/k_B T} \quad (17)$$

donde m es la masa de las partículas y v es la velocidad. En el caso de las altas frecuencias de la ley de Planck, ésta tiene la forma:

$$\rho(f, T) = \frac{8\pi f^2 h f}{c^3} e^{-hf/k_B T} \quad (18)$$

Einstein propone que en las altas frecuencias la radiación se podría imaginar como un gas de partículas independientes, cuya energía sería $E = hf$. Reescribiendo la expresión anterior:

$$\rho(f, T) = \left[\frac{8\pi f^2}{c^3} \right] E e^{-hf/k_B T} \quad (19)$$

Por consiguiente la analogía entre las expresiones es cercana. Para éste gas la frecuencia es proporcional al número de onda y al momentum. Para los átomos la velocidad es proporcional al momentum. Por tanto ambas expresiones se encuentran en el espacio del momentum. Ésta observación conllevó a Einstein a afirmar que “la radiación en el interior de la cavidad de por sí está cuantizada”. Por consiguiente la cuantización en la energía no era algo exclusivo de los radiadores de Planck, sino también de la radiación; paquetes de radiación.

2 Métodos numéricos en el problema de cuerpo negro

2.1 Sobre el código

El siguiente código, es un ejercicio sobre métodos numéricos para resolver numéricamente algunos de los planteamientos anteriormente vistos. En éste notebook se utilizan funciones nativas de python y otras funciones importadas del repositorio remoto, por lo que para la ejecución de nuevos parámetros o modificación, es necesario abrir éste archivo en una consola interactiva de Jupyter.

La sección computacional se deriva en cinco subsecciones:

- Ley de Planck de la radiación espectral de cuerpo negro
- Determinación por gradiente de los máximos de emisividad de la distribución
- Regresión linear en la ley de desplazamiento de Wien
- Interpolación y comparación de la ley de Rayleigh-Jeans
- Ley de Stefan-Boltzmann

```
In [1]: # Import Modules
import os
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import time
import numpy as np
import pandas as pd
from math import pi

%matplotlib inline
```

2.2 Ley de Planck de la radiación espectral de cuerpo negro

La ley de Planck permite determinar la densidad de energía emitida por un un cuerpo negro, acorde a la temperatura.

$$\rho(\lambda, T) = \frac{2hc^2}{\lambda^5 \left[\exp\left(\frac{hc}{\lambda k_B T}\right) - 1 \right]} \quad (20)$$

Donde λ corresponde a la longitud de onda, T corresponde a la temperatura del cuerpo, h corresponde a la constante de Planck, c corresponde a la velocidad de la luz y k_B corresponde a la constante de Boltzmann cuyo valor es igual a $1.3805 \cdot 10^{-23}$ J/K.

El siguiente código ilustrara en una gráfica la relación entre la intensidad de radiación y la longitud de onda, aplicando la ley de Planck. Se analizara la presente gráfica partiendo de la variación de la temperatura en el cuerpo negro, esto con el fin de comprobar computacionalmente la relación inversa que se presenta entre la temperatura y la longitud de onda, por lo que a mayores temperatura menor será el tamaño de la longitud de onda, y mayor la intensidad de radiación.

```
In [2]: # %load https://raw.githubusercontent.com/fisicatyc/numericos-interactivo/master/project_2014_n
        """
        Description: function comptraprule is part of the Composite Integration
        Techniques and can be thought as an upgrated version of Trapezoidal rule
        integration technique that aproximates a defined integral evaluated in [a,b]
        with better presicion in larger intervals. It's truncation error is of order
        h^2 and defined by a second order derivative. Similar to compmidtrule but
        request more operations Presicion is similar to Composite midpoint rule.
        Difference from the other methods is the need of only one integration interval
        therefore the number of subintervals can be even or odd.

        Inputs: lowerlimit - float - defines first limit of integration.
                upperlimit - float - defines last limit of integration.
                redc - integer - integer that reduces space grid in domain.
                function - function type object - evaluates f(x) at x.

        Outputs: integ - float - defined integral aproximation

        Example line: integ = comptraprule(-2, 2, 40, (lambda x: 3*x**4));

        Dependencies: None.

        Version: 1.2 for Python 3.4

        Definitions were taken from:
            Richard L. Burden, J. Douglas Faires. "Numerical Analysis" 9th ed.
            "Chapter 4 - Numerical Differentiation and Integration".
            Cengage Learning. 2010. pp: 153 - 156.

        Author: J.J. Cadavid - SFTC - EAFIT University.
        Contact: jcadav22@eafit.edu.co

        Date: 28/12/2014.
        """

def comptraprule (lowerlimit, upperlimit, redc, function):

    # Input error verification
    if redc % 2 != 0:
        raise Exception('Reduced inverval - redc- must be an even integer');

    test = lambda: None;
```

```

        if isinstance(function,type(test)) == 0:
            raise Exception("function must be lambda object-type");

# Zero initialing
sumf = 0.0;

# Space grid size
h = (upperlimit - lowerlimit)/redc;

# Main loop - Composite trapezoidal rule approximation .
for k in range(1, redc):
    sumf = sumf + 2*function(lowerlimit + k*h);

integ = 0.5*h*(sumf + function(upperlimit) + function(lowerlimit));

return(integ);

In [3]: # Stefan-Boltzmann Constant
SIGMA = 5.670e-8;

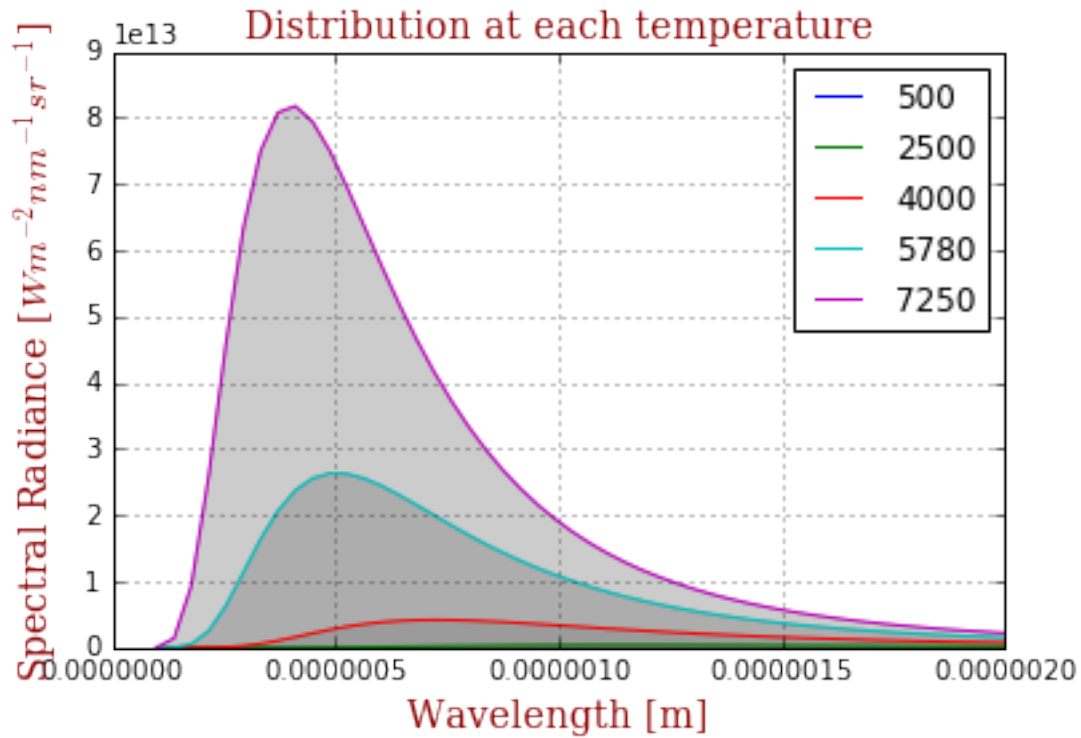
# Planck parameters
h = 6.62607004e-34;
c = 3.0*10**8;
Kb = 1.38064852e-23;
TemperatureVect = [500,2500,4000,5780,7250];
VectSize = len(TemperatureVect);

# Data initialization
dataSize = 50;
IntegSamples = 100 # Defines the number of samples
RadianceMat = [[0 for x in range(dataSize)] for x in range(VectSize)];
MaxDataMat = [[0 for x in range(3)] for x in range(VectSize)];
waveLengthMin = 100e-9;
waveLengthMax = 2000e-9;
waveLengthVect = np.linspace(waveLengthMin,waveLengthMax,dataSize);

# Planck's Law function
for n in range(0,VectSize):
    B = lambda waveLength: (2*h*c**2)/waveLength**5*(1/(np.exp((h*c)/(waveLength*Kb*TemperatureVect[n]))));
    RadianceMat[n][:] = B(waveLengthVect);
    MaxDataMat[n][2]= comptraprule(waveLengthMin, waveLengthMax, IntegSamples, B);

In [4]: # Plot parameters
font = {'family' : 'serif', 'color' : 'darkred', 'weight' : 'normal', 'size' : 14};
for n in range(0,VectSize):
    plt.plot(waveLengthVect,RadianceMat[n][:], label = TemperatureVect[n],);
    plt.fill_between(waveLengthVect,RadianceMat[n][:], alpha=0.05*n, facecolor='k');
plt.axis('auto');
plt.title('Distribution at each temperature', fontdict=font);
plt.xlabel('Wavelength [m]', fontdict=font);
plt.ylabel('Spectral Radiance [ $\text{W m}^{-2}\text{nm}^{-1}\text{sr}^{-1}$ ]', fontdict=font);
plt.grid();
plt.legend();
plt.show();

```

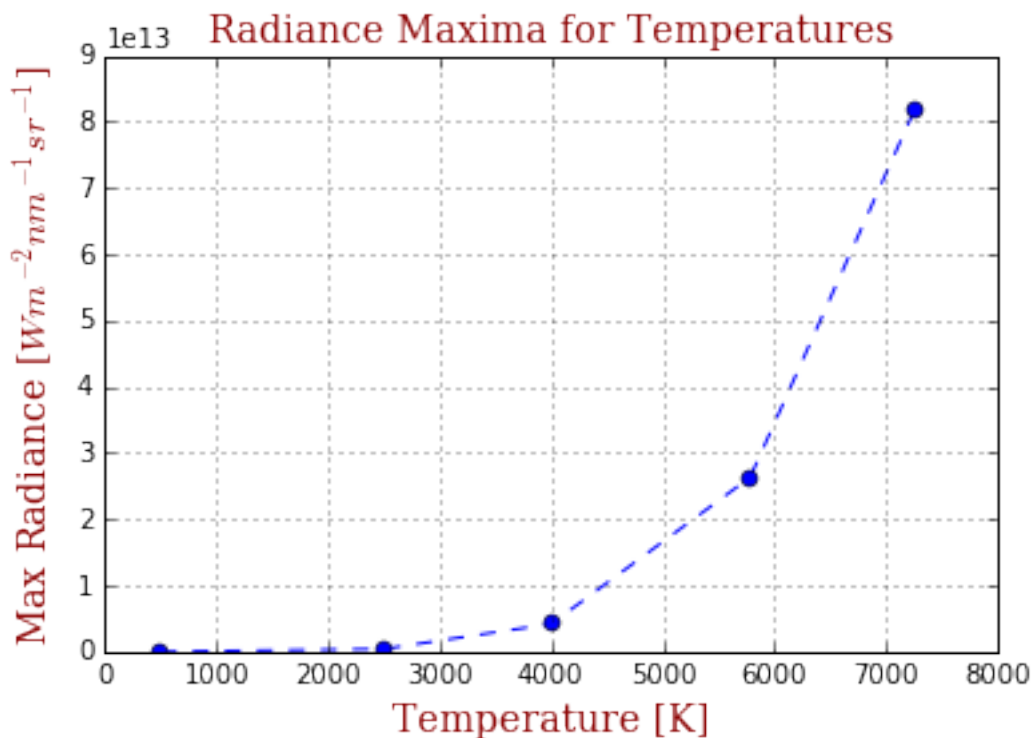



```
In [5]: # Data analysis
for n in range(0,VectSize):
    CurrentData = RadianceMat[n][:];
    MaxDataMat[n][0] = waveLengthVect[CurrentData.index(max(CurrentData))];
    MaxDataMat[n][1] = max(CurrentData);

MaxDataArray = np.asarray(MaxDataMat)
TempArray = np.asarray(TemperatureVect);

plt.plot(TemperatureVect,MaxDataArray[:,1], marker='o', linestyle='--', color='b');
plt.axis('auto');
plt.title('Radiance Maxima for Temperatures', fontdict=font);
plt.xlabel('Temperature [K]', fontdict=font);
plt.ylabel('Max Radiance [ $W\ m^{-2}nm^{-1}sr^{-1}$ ]', fontdict=font);
plt.grid();
plt.show();

Frame = pd.DataFrame(MaxDataMat,index=TemperatureVect, columns=['Max Radiance WL [m]', 'Max Radiance [W m-2nm-1sr-1]', 'Temperature [K]'])
Frame
```



Out[5]:

	Max Radiance WL [m]	Max Radiance	Total Energy
500	2.000000e-06	2.082200e+06	0.358755
2500	1.146939e-06	3.990157e+11	445878.365127
4000	7.204082e-07	4.184867e+12	3949892.500594
5780	4.877551e-07	2.631612e+13	18902104.924807
7250	4.102041e-07	8.174296e+13	48105063.417381

2.3 Determinación por gradiente de los máximos de emisividad de la distribución

Acorde la gráfica radiancia máxima, se puede observar una relación entre los máximos de cada temperatura. Ésta gráfica muestra el comportamiento que la ley de desplazamiento Wien para diferentes temperaturas. Aunque ésta ley solo predice el máximo para cada temperatura, al tener los diferentes máximos se puede ver un comportamiento exponencial en la densidad de energía asociada. Sin embargo, la forma de encontrarlos fue directamente comparando el máximo valor en el arreglo de datos.

Éstos máximos para cada distribución pueden obtenerse también al entender el comportamiento de los cambios en la función, principalmente en los valores de las pendientes. Si se observa la gráfica de las distribuciones de la densidad de energía, desde las longitudes de onda mayores hasta las pequeñas, la función decrece (menor energía para longitudes de ondas grandes). Por consiguiente ésta región tiene pendiente negativa. Sin embargo, si se observa en la región de las longitudes menores hacia la parte visible del espectro (La transición del ultravioleta al azul), se observa que la pendiente aumenta. (Gran parte de la densidad de energía del espectro está en la región visible).

Los cambios en el signo de la pendiente indican que hay un cruce por cero, o por consiguiente que un máximo (o un mínimo) local está presente. El algoritmo propuesto evalúa localmente las derivadas de toda la función utilizando diferencias finitas de primer orden centrada. El código es el siguiente:

```

In [6]: # %load https://raw.githubusercontent.com/fisicatyc/numericos-interactivo/master/project_2014_n
        #!/usr/bin/python
        """
        Description: class centraldiff defines a set of numerical methods based on
        finite differences of central step. Capable of aproximating derivatives at
        a given point inside an interval. Central step does not evaluate derivatives
        at endpoints. Precision is highly dependant on interval size, bigger intervals
        lead to highter truncation errors. Minor error types are of  $h^2$  order.
        Analitical errors can be calculated using fourth derivatives. Highter
        derivative orders can lead to high truncation errors, so results are likely
        to differ a lot from analytical results, especially with interpolated data.

        FCDA or First Centered Difference Aproximation, calculates the first
        derivative in the interval. Dependencies: testlambda, getarray.

        Method inputs: n - integer - Defines number of n+1 elements in list.
                      lowerlimit - float - first term of interval.
                      upperlimit - float - last term of interval.
                      function - function type object - evaluates  $f(x)$  at  $x$ .

        Method output: diff_array - list - list of derivated values in (a,b).

        call sequence example: centraldiff.FCDA(4, 0, 2, (lambda x: 4*sin(x)**2));

        Dependencies: None.

        Version: 1.2 for Python 3.4

        Definitions are taken from:
        Jaan Kiusalaastr. "Numerical Methods in Engineering With Python 3".
        3th ed. "Chapter 5 - Numerical Differentiation".
        Cambridge University Press. 2013. PP. 183 - 185.

        Author: J.J. Cadavid - SFTC - EAFIT University.
        Contact: jcadav22@eafit.edu.co

        Date: 28/12/2014.
        """

def FCDA(n,lowerlimit, upperlimit, function, x, flag = False):
    test = lambda: None;
    if isinstance(function,type(test)) == 0:
        raise Exception("function must be lambda object-type");

    h = upperlimit-lowerlimit;

    if flag:
        hc = h/(n-1);
        arrayStart = lowerlimit;
        x_array = [arrayStart + hc*x for x in range(n)];
        diff_array = [0]*n;

        for i in range(n):
            diff_array[i] = (function(x_array[i] + h) - \

```

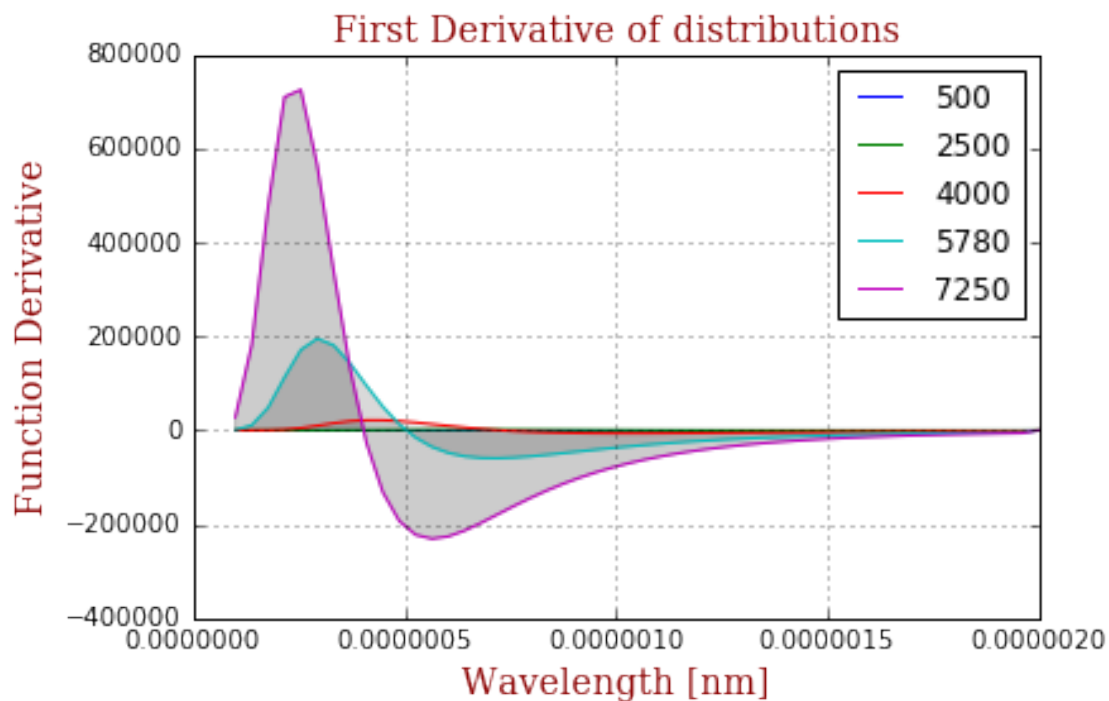
```

        function(x_array[i] - h))/2*h;
    else:
        diff_array = (function(x + h) - \
            function(x - h))/2*h;
    return(diff_array);

In [7]: DerivMat = [[0 for x in range(dataSize)] for x in range(VectSize)];
        # First derivative for each Planck's Law at different temperatures
        for n in range(0,VectSize):
            DB = lambda waveLength: (2*h*c**2)/waveLength**5*(1/(np.exp((h*c)/(waveLength*Kb*Temperatur
            for b in range(0,dataSize-1):
                DerivMat[n][b] = FCDA(n, waveLengthVect[b], waveLengthVect[b+1],DB,waveLengthVect[b]);

In [8]: for n in range(0,VectSize):
        plt.plot(waveLengthVect,DerivMat[n][:], label = TemperatureVect[n],);
        plt.fill_between(waveLengthVect,DerivMat[n][:], alpha=0.05*n, facecolor='k');
plt.axis('auto');
plt.title('First Derivative of distributions', fontdict=font);
plt.xlabel('Wavelength [nm]', fontdict=font);
plt.ylabel('Function Derivative', fontdict=font);
plt.grid();
plt.legend();
plt.show();

```



Tras encontrar las derivadas de toda la función, se puede observar una región de interés. Con una lectura de derecha a izquierda de la gráfica se observa una pendiente que se vuelve más negativa. Ésto indica que el cambio negativo y es más abrupto mayor para las longitudes de onda que comienzan a alejarse del espectro visible, indicando que tienen menor densidad asociada.

Al llegar a la zona del espectro visible, la cantidad de energía aumenta rápidamente; pendiente positiva “empinada”. Es en ésta región de aumento aproximadamente constante de energía donde ocurre el cambio

de signo, y gráficamente se ve como el cruce por cero del eje vertical. Es en éste valor donde se encuentra el máximo de emisividad.

El siguiente paso está en encontrar los índices del arreglo que tengan la pendiente negativa más pronunciada y la pendiente positiva más pronunciada. Ésta aproximación tiene en consideración que la función tiene un único máximo que es global y no hay mínimos. De ésta forma se garantiza que el valor entre éstas pendientes es el máximo que se busca.

La comparación de valores busca un primer máximo y lo compara con el valor vecino del arreglo. En el momento de que encuentra el valor más cercano a cero, se considera como el máximo de la función, devolviendo las índices de las posiciones que corresponden a los máximos de las distribuciones. El algoritmo compara todos los valores independientemente de haber encontrado el máximo.

Los resultados son presentados en la tabla al final de la celda.

```
In [9]: # Find Derivative Zeros - all zeros are assumed to be between global maximum and minimum
MinDataMat = [[0 for x in range(2)] for x in range(VectSize)];

# Loop to find individual maximum at each temperature
CurrentMax = 0;
for n in range(0,VectSize):
    CurrentData = DerivMat[n][:];
    MaxIndex = CurrentData.index(max(CurrentData));
    MinIndex = CurrentData.index(min(CurrentData));
    TGTIntval = DerivMat[n][MaxIndex:MinIndex]
    IntvalMin = min(abs(i) for i in TGTIntval)
    try:
        a = CurrentData.index(IntvalMin);

    except:
        a = CurrentData.index(-IntvalMin);
    MinDataMat[n][0] = waveLengthVect[a];
    MinDataMat[n][1] = RadianceMat[n][a]
    CurrentData = RadianceMat[n][a];

    if CurrentMax < CurrentData:
        MaxRadiance = RadianceMat[n][a];
        MaxWlength = waveLengthVect[a];
        Temperature = TemperatureVect[n];
    else:
        CurrentMax = RadianceMat[n][a];

print("The Black Body with Temperature: %s, has the Maximum radiance: %s which is reached at Wa

FrameB = pd.DataFrame(MinDataMat,index=TemperatureVect, columns=['Max Radiance WL [m]','Max Rad
FrameB
```

The Black Body with Temperature: 7250, has the Maximum radiance: 8.17429629438e+13 which is reached at W

```
Out[9]:
```

	Max Radiance WL [m]	Max Radiance
500	1.961224e-06	1.727468e+06
2500	1.146939e-06	3.990157e+11
4000	7.204082e-07	4.184867e+12
5780	4.877551e-07	2.631612e+13
7250	4.102041e-07	8.174296e+13

2.4 Regresión lineal en la ley de desplazamiento de Wien

La ley de desplazamiento de Wien permite calcular la posición del espectro máximo en un cuerpo negro. Una ley que refleja de manera operacional la relación inversamente proporcional existente entre la temperatura y la longitud de onda, del cual confirma que ha mayor temperatura menor la longitud de onda que emite. Matemáticamente la ley se expresa de la siguiente manera.

$$\lambda_{max} = \frac{0,0028976}{T} \quad (21)$$

donde T corresponde a la temperatura del cuerpo y λ corresponde a la longitud de onda en el pico máxima del espectro de emisión.

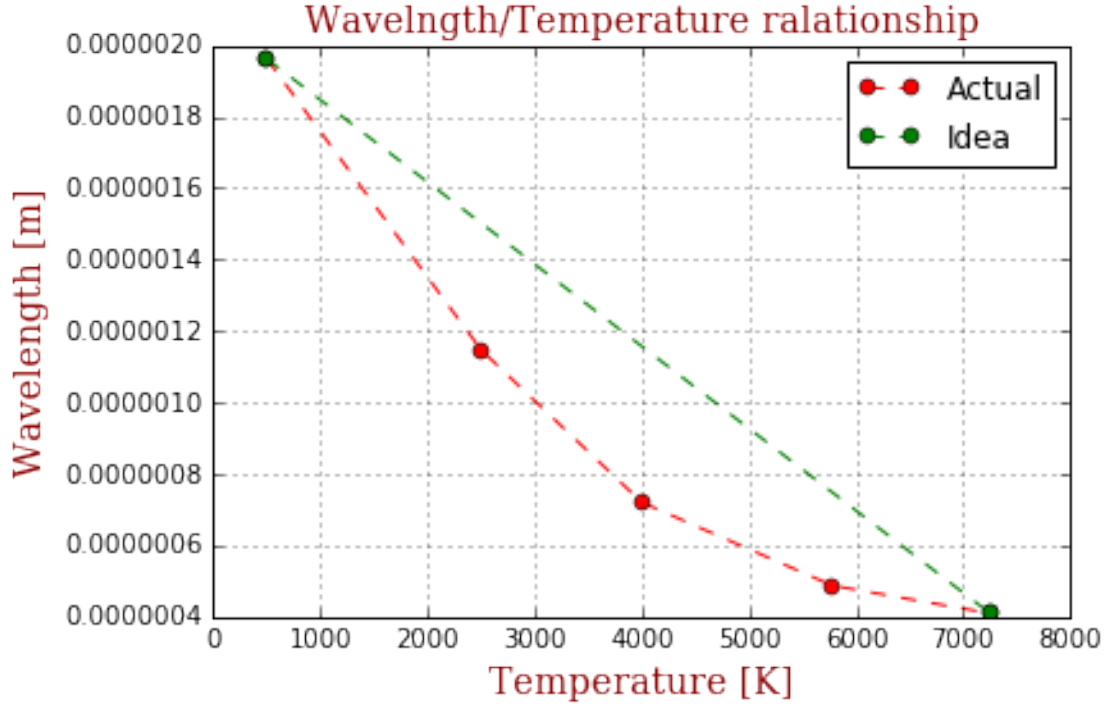
Con los datos computados de los valores Máximos de la función se desea realizar una regresión lineal que permita determinar una función que aproxime el modelo. Sin embargo ésta aproximación no tiene validez cuando la longitud de onda se vuelve pequeña, principalmente por el comportamiento asintótico de la función, cuya indeterminación ocurre en $\lambda = 0$.

En la figura siguiente, se muestra el comportamiento no lineal de los máximos (línea roja), pero se desea aproximar con una función de la forma:

$$f(x) = ax + b \quad (22)$$

Esta recta del gráfico, únicamente se traza desde el primer y último dato del arreglo de máximos. Por consiguiente entre más grande sea la diferencia entre la primera y última temperatura, menor será el comportamiento lineal.

```
In [10]: MinDataArray = np.asarray(MinDataMat);
plt.plot(TemperatureVect,MinDataArray[:,0], marker='o', linestyle='--', color='r',label='Actual')
plt.plot([TemperatureVect[0],TemperatureVect[VectSize-1]], [MinDataArray[0,0],MinDataArray[VectSize-1,0]], color='b',label='Fit')
plt.axis('auto');
plt.title('Wavelength/Temperature relationship', fontdict=font);
plt.xlabel('Temperature [K]', fontdict=font);
plt.ylabel('Wavelength [m]', fontdict=font);
plt.grid();
plt.legend();
plt.show();
```



Para el ajuste de la curva, se reapiza una regresión por mínimos cuadrados de un polinomio de grado m . En nuestro caso, un polinomio de grado 1. También es posible utilizar otro tipo de funciones para el ajuste. De ésta forma se realiza una minimización, cuya representación forma es:

$$S(a_0, a_1, a_2 \dots a_m) = \sum_{i=1}^n [y_i - f(x_i)]^2 \therefore \frac{\partial S}{\partial a_i} = 0 \quad (23)$$

donde $f(x)$ es la función del ajuste y n es la cantidad total de información para el ajuste. De tal forma, se minimiza sobre los m coeficientes del polinomio, tal que produzca el menor residuo (discrepancia) entre la información que se está ajustando. El problema del ajuste lineal se plantea como:

$$S(a, b) = \sum_{i=1}^n [y_i - f(x_i)]^2 = \sum_{i=1}^n (y_i - a - bx_i)^2 \quad (24)$$

La solución a los parámetros que satisfacen las derivadas son:

$$a = \frac{\bar{y} \sum x_i^2 - \bar{x} \sum x_i y_i}{\sum x_i^2 - n \bar{x}^2} \quad b = \frac{\sum x_i y_i - \bar{x} \sum y_i}{\sum x_i^2 - n \bar{x}^2} \quad (25)$$

Sin embargo ésta definición de los parámetros es más sensible a errores de redondeo. Por tanto, se pueden calcular como:

$$b = \frac{\sum y_i (x_i - \bar{x})}{\sum x_i (x_i - \bar{x})} \quad a = \bar{y} - \bar{x}b \quad (26)$$

Éstas definiciones son aplicadas en la siguiente celda.

```
In [11]: # Defining the parameters of fitting curve - A linear polynomial
Tmean = sum(TempArray)/VectSize;
WLmean = sum(MinDataArray[:,0])/VectSize;
```

```

SqrTemp = sum(TempArray**2);

SigmaX = np.sqrt(1/VectSize*sum((TempArray - Tmean)**2));
SigmaY = np.sqrt(1/VectSize*sum((MinDataArray[:,0] - WLmean)**2));

# Round-off susceptible fit
ar = (WLmean*SqrTemp - Tmean* sum(TempArray*MinDataArray[:,0]))/(SqrTemp-VectSize*Tmean**2);
br = (sum(TempArray*MinDataArray[:,0]) - Tmean*sum(MinDataArray[:,0]) )/(SqrTemp-VectSize*Tmean**2);
Yr = lambda Temp: ar + br*Temp;

# Less Round-off susceptible fit
b = sum(MinDataArray[:,0]*(TempArray - Tmean))/sum(TempArray*(TempArray - Tmean));
a = WLmean - Tmean*b;
Y = lambda Temp: a + b*Temp;

# Wien's Displacement Law:
WLw = lambda Temp: 2.89776829e-3/Temp; # [m]K/K

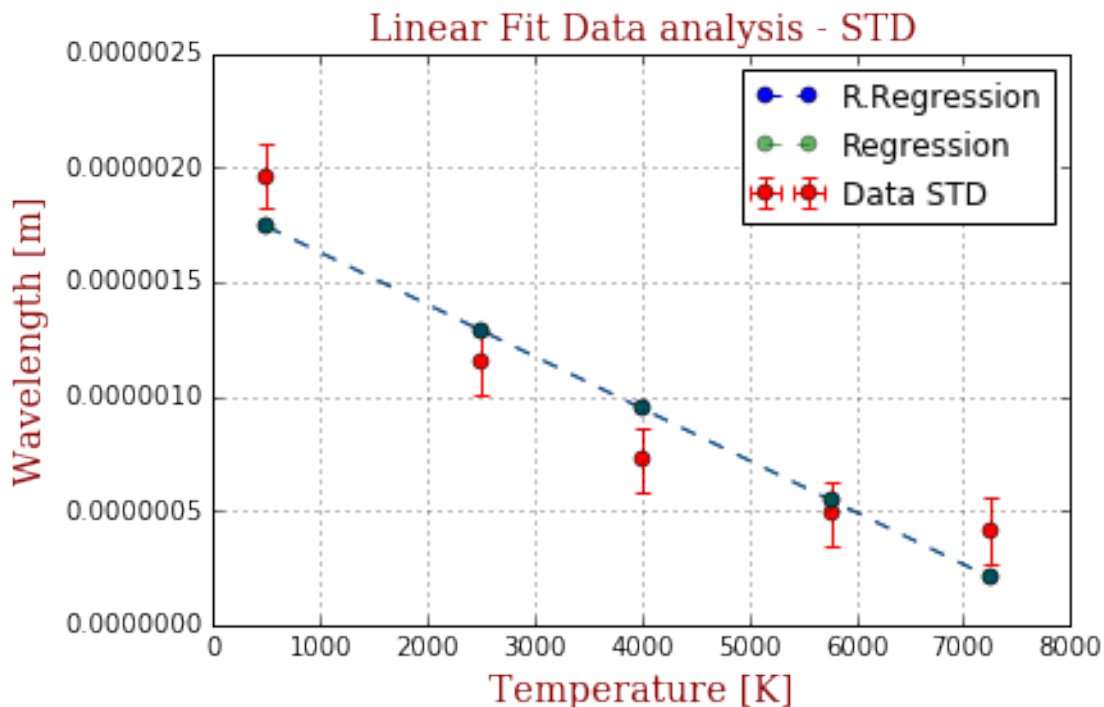
```

```

In [12]: # Plot regression results
DataScale = 10e14;
plt.errorbar(TempArray,MinDataArray[:,0], xerr=SigmaX*DataScale, yerr=SigmaY*DataScale,fmt='o', color='r', label='Data STD');
plt.plot(TempArray,Yr(TempArray), marker='o', linestyle='--', color='b',label='R.Registration');
plt.plot(TempArray,Y(TempArray), marker='o', linestyle='--', color='g',label='Regression',alpha=0.5);

plt.title('Linear Fit Data analysis - STD', fontdict=font);
plt.xlabel('Temperature [K]', fontdict=font);
plt.ylabel('Wavelength [m]', fontdict=font);
plt.grid();
plt.legend();
plt.show();

```

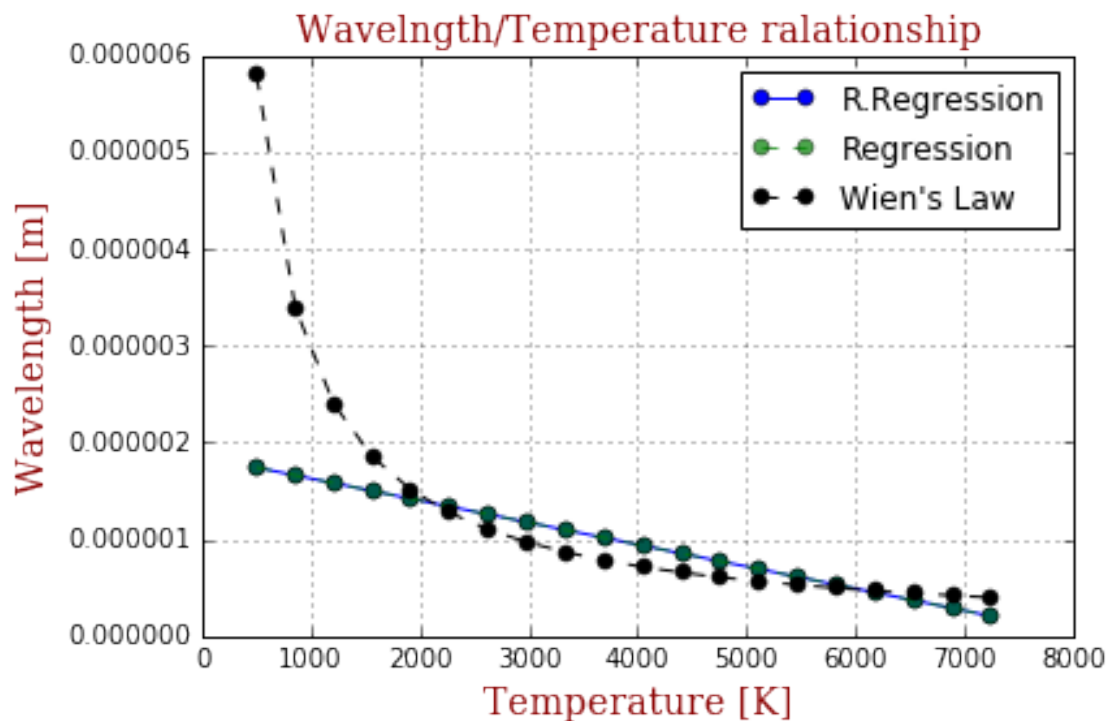



```

In [13]: # Temperature array
TempData = np.linspace(min(TempArray),max(TempArray),20);

# Plot Data
plt.plot(TempData,Yr(TempData), marker='o', linestyle='--', color='b',label='R.Regression');
plt.plot(TempData,Y(TempData), marker='o', linestyle='--', color='g',label='Regression',alpha=0.5);
plt.plot(TempData,WLw(TempData), marker='o', linestyle='--', color='k',label="Wien's Law");
plt.axis('auto');
plt.title('Wavelength/Temperature ralationship', fontdict=font);
plt.xlabel('Temperature [K]', fontdict=font);
plt.ylabel('Wavelength [m]', fontdict=font);
plt.grid();
plt.legend();
plt.show();

```



Como se observa en la figura anterior, el polinomio como la ley de Wien, difieren particularmente para la región en que el comportamiento asintótico es más evidenciado. Para las regiones más distantes de la función se observa que hay cercanía en las formas de la función y la regresión.

En éste caso también se observa que las dos formas de regresión coinciden, implicando que no hay errores por redondeo en los datos. La regresión indica que la ley de Desplazamiento crece lentamente para altas temperaturas por lo que la forma de la función es aproximadamente recta, y la función de ajuste es válida para éstos valores.

2.5 Interpolación y comparación de la ley de Rayleigh-Jeans

La ley de Raleigh-jean consiste en describir la radiación espectral de un cuerpo negro a una temperatura dada y variando las longitudes de ondas. A continuación se expresa matemáticamente la presente ley.

$$B = \frac{2ckT}{\lambda^4} \quad (27)$$

El código en la siguiente sección calcula la interpolación de Newton y compara directamente la relación con la ley de Rayleigh-Jeans.

```
In [14]: ## module newtonPoly - From Numerical Methods in Engineering with Python 3 by Jaan Kiusalass -
"""
p = evalPoly(a,xData,x).
Evaluates Newton's polynomial p at x. The coefficient
vector {a} can be computed by the function 'coeffts'.

a = coeffts(xData,yData).
Computes the coefficients of Newton's polynomial.
"""

def evalPoly(a,xData,x):
    n = len(xData) - 1 # Degree of polynomial
    p = a[n]
    for k in range(1,n+1):
        p = a[n-k] + (x - xData[n-k])*p
    return p

def coeffts(xData,yData):
    m = len(xData) # Number of data points
    a = yData.copy()
    for k in range(1,m):
        a[k:m] = (a[k:m] - a[k-1])/(xData[k:m] - xData[k-1])
    return a

In [15]: # Fixed temperature
FxdTemp = 4000; #[K]
WLSet = [1.146939e-5,4.146939e-06,6.204082e-07,1e-18];
SetSize = len(WLSet);
FxdData = [[0 for x in range(2)] for x in range(SetSize)];

BF = lambda waveLength: (2*h*c**2)/waveLength**5*(1/(np.exp((h*c)/(waveLength*Kb*FxdTemp))-1))

for n in range(SetSize):
    FxdData[n][0] = WLSet[n];
    FxdData[n][1] = BF(WLSet[n]);

FxdDataArray = np.asarray(FxdData);

In [16]: # Compute interpolant coefficients and polynomial
Coeff = coeffts(FxdDataArray[:,0],FxdDataArray[:,1]);
PolyData = [0]*dataSize;
print("Polynomial coefficients: %s" %(Coeff));

# Evaluate set with polynomial
WLSet = np.linspace(WLSet[0],WLSet[SetSize-1],dataSize)
for n in range(dataSize):
```

```

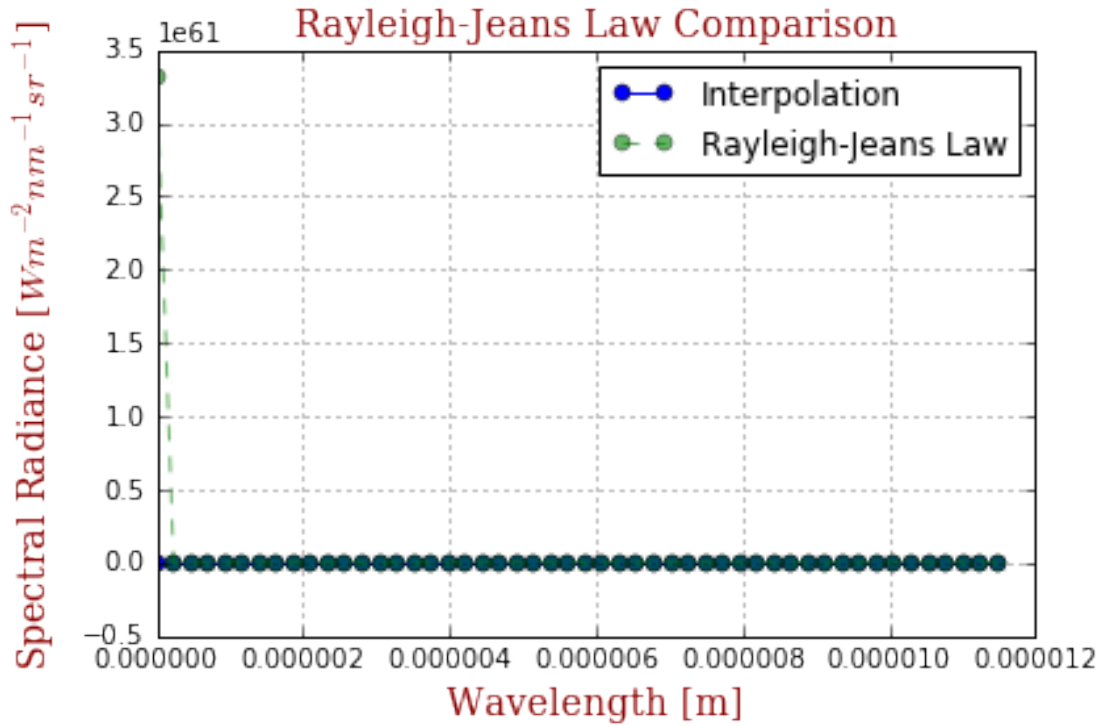
PolyData[n] = evalPoly(Coeff,FxdDataArry[:,0],WLSet[n]);

# Rayleigh-Jeans Law
BRJ = lambda waveLength: (2*c*Kb*FxdTemp)/waveLength**4;

Polynomial coefficients: [ 1.63006572e+09 -9.38689237e+15 1.00112532e+23 1.65069346e+29]

In [17]: # Plot Data
plt.plot(WLSet,PolyData, marker='o', linestyle='-', color='b',label='Interpolation');
plt.plot(WLSet,BRJ(WLSet), marker='o', linestyle='--', color='g',label='Rayleigh-Jeans Law',al
plt.axis('auto');
plt.title('Rayleigh-Jeans Law Comparison', fontdict=font);
plt.xlabel('Wavelength [m]', fontdict=font);
plt.ylabel('Spectral Radiance [$W m^{-2}nm^{-1}sr^{-1}$]', fontdict=font);
plt.grid();
plt.legend();
plt.show();

```



2.6 Ley de Stefan-Boltzmann

Como se observó en la la ley de Planck una cavidad de cuerpo negro puede emitir en todo el espectro. Un de las propieades que se puede analizar es la Potencia de Emisividad de la cavidad, definiendo la cantidad de energía por unidad de área se emite. En el caso de tener una cavidad con un ángulo sólido de emisión hemisférico, se puede establecer la relación:

$$E_{total} = 2\pi hc^2 \int_0^\infty \frac{d\lambda}{\lambda^5 \left[\exp\left(\frac{hc}{\lambda k_b T}\right) - 1 \right]} d\nu = \sigma T^4 \quad (28)$$

2.6.1 Solución numérica de la constante de Stefan-Boltzmann

Para resolver numéricamente ésta expresión es necesario considerar que se trata de una integral impropia, puesto que sus límite superior de integración no está definido y posee una singularidad en el límite inferior. Por éste motivo las técnicas como el trapecio o las reglas de Newton-Cotes no pueden ser utilizadas, principalmente porque dependen de los límites para definir el dominio de integración.

Otra estrategia está en la integración de punto medio de Gauss-Legendre, sin embargo éste método no arroja resultados acertados sino se utiliza altos costos computacionales. Para plantear una solución a éste planteamiento, primero se realizará un cambio de variable en los argumentos de la función, para obtener un problema análogo:

$$u = \frac{hc}{\lambda k_B T} \longrightarrow d\lambda = \frac{-hc}{\lambda^2 k_B T} \therefore \quad (29)$$

$$E = \frac{2\pi (k_B T)^4}{c^2 h^3} \int_0^\infty \frac{u^3}{\exp(u) - 1} du = \sigma T^4 \quad (30)$$

El segundo paso sugiere cambiar el dominio de integración infinito a un dominio finito, aplicando un segundo cambio de variable, tal que:

$$t = u^{-1}, dt = -u^{-2} \therefore du = -u^{-2}, dt = -t^{-t} \quad (31)$$

de ésta manera se puede aplicar:

$$\int_a^\infty f(u) du = \int_{1/a}^0 -t^{-2} f(t^{-1}) dt = \int_0^{1/a} t^{-2} f(t^{-1}) dt \quad (32)$$

Sin embargo, nótese que en éste caso, el parámetro a equivale a cero, por lo que definiría nuevamente un límite indeterminado. Por éste motivo, otra aproximación que se puede realizar sería evadir la singularidad, desplazando el dominio de integración tal que $a = 0.1$. De ésta forma se puede establecer:

$$E = \frac{2\pi (k_B T)^4}{c^2 h^3} \int_{0.1}^{10} \frac{t^{-5}}{\exp(t^{-1}) - 1} du \quad (33)$$

El integrando planteado, puede ser evaluado numéricamente por los convencionales métodos de intervalo cerrado o el método de Gauss-Legendre. Para éste caso se utilizará éste último y el trapecio compuesto. Para el caso del punto medio, se puede establecer que:

$$\int_a^b f(x) dx = \sum_{i=1}^n c_i f(r_i) \quad (34)$$

donde c_i y r_i son el i -ésimo coeficiente de interpolación y raíz del i -ésimo polinomio de Legendre. En el código siguiente, se realiza la aproximación numérica de estos valores utilizando el método de Newton-Rapson. En éste caso se utilizará $n = 2$ para garantizar que la función del integrando sea interpolable por un polinomio menor al grado $2n$.

In [18]: # Original code: <http://w3mentor.com/learn/python/scientific-computation/gauss-legendre-m-point>

```
''' x,A = gaussNodes(m,tol=10e-9)
    Returns nodal abscissas {x} and weights {A} of
    Gauss-Legendre m-point quadrature.
'''
from math import cos,pi
from numpy import zeros

def gaussNodes(m,tol=10e-9):

    if m<2:
```

```

        raise Exception("m parameter must be greater than 1.");

def legendre(t,m):
    p0 = 1.0; p1 = t;
    for k in range(1,m):
        p = ((2.0*k + 1.0)*t*p1 - k*p0)/(1.0 + k )
        p0 = p1; p1 = p
    dp = m*(p0 - t*p1)/(1.0 - t**2)
    return p,dp;

A = zeros(m);
x = zeros(m);
nRoots = (m + 1)//2          # Number of non-neg. roots

for i in range(nRoots):
    t = cos(pi*(i + 0.75)/(m + 0.5)) # Approx. root
    for j in range(30):
        p,dp = legendre(t,m);          # Newton-Raphson
        dt = -p/dp; t = t + dt          # method
        if abs(dt) < tol:
            x[i] = t; x[m-i-1] = -t
            A[i] = 2.0/(1.0 - t**2)/(dp**2) # Eq. (6.25)
            A[m-i-1] = A[i];
            break;
    return x,A
nRoots = 2;
[roots,Coeff] = gaussNodes(nRoots);
print('Number of roots: ', nRoots);
print('Roots =',roots);
print('Coefficients = ',Coeff);

Number of roots:  2
Roots = [ 0.57735027 -0.57735027]
Coefficients = [ 0.99999997  0.99999997]

In [19]: Integrand = lambda t: (1/(t**5))/(np.exp((1/t)-1));
Sm = 0;
for index in range(0,nRoots-1):
    Sm = Coeff[index]*Integrand(roots[index]) + Sm;
sigmaGL = (2*pi*Kb**4)/(h**3*c**2)*(Sm);
sr = comptraprule(0.1, 10, 10, Integrand);
sigmaT = (2*pi*Kb**4)/(h**3*c**2)*(sr);

```

2.6.2 Solución analítica de la constante de Stefan-Boltzmann

Para resolver la forma analítica y comparar los resultados, la clave de la integración está en tener la serie:

$$\frac{e^{-x}}{1 - e^{-x}} = \sum_{n=1}^{\infty} e^{-nx} \quad (35)$$

Para llevar la expresión a esa forma, el integrando se divide por $\exp u$ obteniendo:

$$E = \frac{2\pi (k_B T)^4}{c^2 h^3} \int_0^{\infty} \frac{u^3}{e^u - 1} du = \frac{2\pi (k_B T)^4}{c^2 h^3} \int_0^{\infty} \frac{u^3 e^{-u}}{1 - e^{-u}} du = \frac{2\pi (k_B T)^4}{c^2 h^3} \sum_{n=1}^{\infty} \int_0^{\infty} u^3 e^{-nu} du \quad (36)$$

Esta expresión puede ser evaluada utilizando la función gamma, si se considera:

$$t = nu \longrightarrow u^3 = \frac{t^3}{n^3} \therefore dt = ndu \longrightarrow du = \frac{dt}{n} \quad (37)$$

Planteando:

$$\Gamma(Z) = (Z-1)! = \int_0^\infty t^{Z-1} e^{-t} dt = \int_0^\infty \frac{1}{n^4} t^3 e^{-t} dt \therefore 3 = Z-1 \longrightarrow Z = 4 \therefore \Gamma(4) = 3! = 6 \longrightarrow \int_0^\infty \frac{1}{n^4} t^3 e^{-t} dt = \frac{6}{n^4} \quad (38)$$

Por consiguiente:

$$E = \frac{2\pi (k_B T)^4}{c^2 h^3} \sum_{n=1}^{\infty} \int_0^\infty u^3 e^{-nu} = \frac{2\pi (k_B T)^4}{c^2 h^3} \sum_{n=1}^{\infty} \frac{6}{n^4} \quad (39)$$

La parte de la sumatoria, puede verse como un caso particular de la función zeta $\zeta(s)$ de Riemann, tal que:

$$E = \frac{2\pi (k_B T)^4}{c^2 h^3} \sum_{n=1}^{\infty} \frac{6}{n^4} = \frac{2\pi (k_B T)^4}{c^2 h^3} 6\zeta(4) = \frac{2\pi (k_B T)^4}{c^2 h^3} 6 \frac{\pi^4}{90} = \frac{2\pi^5 k_B^4 T^4}{15 c^2 h^3} \therefore \sigma = \frac{2\pi^5 k_B^4}{15 c^2 h^3} \longrightarrow E(T) = \sigma T^4 \quad (40)$$

Esta expresión se conoce como la ley de Stefan-Boltzmann.

2.6.3 Comparación

Sustituyendo el conjunto de constantes, el valor numérico es $\sigma = 5.670e-8 [W K^4 m^{-2}]$. Comparando éste valor con el numérico se obtiene:

```
In [20]: ErR = lambda VR,V0: abs(VR-V0)/VR*100;
print('Valor analítico: %1.3e'% SIGMA);
print('Método de punto medio: %1.3e. Error relativo: %1.2f%%.' % (sigmaGL, ErR(SIGMA,sigmaGL)));
print('Método del trapecio %1.3e. Error relativo: %1.2f%%.' % (sigmaT, ErR(SIGMA,sigmaT)));
```

Valor analítico: 5.670e-08

Método de punto medio: 6.537e-08. Error relativo: 15.29%.

Método del trapecio 5.982e-08. Error relativo: 5.51%.

2.7 Referencias y otros recursos

1. Serway, Raymond A. Física Moderna. Capítulo 3: “La teoría cuántica de la Luz”, (Ch3), Apéndice Web Segundo. Thomson Learning,inc. 2005.
2. Michael Fowler. Modern Physics. “Black Body Radiation”. University of Virginia 2008. [URL](#)
3. Rafael Zamora Ramos. [Estudio de la radiación del cuerpo negro. Determinación de la constante de Wien](#)
4. Universidad del País Vasco - Campus de Gipuzkoa. [La radiación de cuerpo negro](#)
5. Maria de Fátima Oliveira Saraiva y Kepler de Souza Oliveira Filho - Kepler de Souza Oliveira Filho. [Radiación de cuerpo negro](#)
6. Jaan Kiusalaas. [Numerical Methods in Engineering With Python 3](#). 3th ed. “Chapter 5 - Numerical Differentiation”. Cambridge University Press. 2013. PP. 183 - 185.
7. Richard L. Burden, J. Douglas Faires. Numerical Analysis 9th ed. “Chapter 4 - Numerical Differentiation and Integration”. Cengage Learning. 2010. pp: 153 - 156.
8. Ronald E. Walpole, Roanoke College, Raymond H. Myers, Sharon L. Myers. Probability and statistics for engineers and scientists. Chapter 11. Simple Linear Regression and Correlation. Pearson Education. 2012.