Fiuba
Facultad de Ingeniería
Universidad de Buenos Aires

Trabajo Práctico N°1

PROGRAMACIÓN Y ERRORES

Análisis Numérico I (75.12) – Curso 007

Fecha de entrega: 3/10/2011

Grupo N°: 004

Integrantes:

Rossi, Federico Martín - 92086

Montoya, Diego Ramiro - 91939

Bertucci, Juan Pablo - 92816

Di Nella, Camila - 91821

1 Introducción

En el estudio de los materiales es muy común realizar distintas experiencias que implican calentamientos donde debe conocerse la temperatura (por ejemplo los tratamientos térmicos). Existen muchas formas de medir temperatura, con termocuplas, pirómetros ópticos, termómetros; todos con distintos rangos y campos de aplicación.

En particular, se puede medir temperatura por medio de termografías, es decir, a través de fotografías de un material a distintas temperaturas. Como consideración inicial, vamos a suponer a la radiación emitida por los materiales a alta temperatura en el Infrarrojo Cercano (NIR), siendo así susceptible de ser captada por medios ópticos.

1.1 Objetivos

En este trabajo práctico se tienen 8 matrices de 23x23 elementos, las cuales representan 8 fotografías tomadas en un rango que va desde los 900ºC a los 1400ºC. Cada elemento de una matriz es un número natural, entre 0 y 255, el cual representa un color en la escala de grises.

Se buscará un valor representativo para cada matriz, realizando un promedio de todos sus elementos, y se volcarán luego a un gráfico que relacionará la temperatura con la escala de grises, encontrando una relación entre las mismas mediante una aproximación polinómica de segundo orden, utilizando MS Excel.

Luego se determinará una expresión del error en la medición de la temperatura, calculando las derivadas parciales de la expresión obtenida en MS Excel, el cual será además graficado. También se buscará el factor de amplificación de errores.

2 Desarrollo

El desarrollo del trabajo práctico lo dividiremos en varios apartados de acuerdo a lo mencionado en los objetivos, a fin de lograr un mayor entendimiento y organización de lo expuesto.

Todos los archivos y códigos fuente aquí mencionados, así como también el presente informe, pueden ser descargados de la sección *Downloads* del repositorio del grupo alojado en Google Code (*http://code.google.com/p/7512-analisis-numerico-grupo004/*).

2.1 Correlación entre temperatura y escala de grises

Vamos a comenzar buscando una correlación entre la temperatura y la escala de grises de una foto representada por una matriz de dimensión 23x23 donde cada casillero representa un pixel de la imagen.

En este primer caso, disponemos ya de 8 matrices correspondientes a 8 fotos tomadas por el pirómetro óptico. Inicialmente deberemos calcular el promedio de los valores contenidos en cada matriz. Para esto recurriremos a la aplicación *Matlab*, la cual nos permitirá mediante un sencillo algoritmo llegar al resultado. Además de esto, se calcularán los errores debido a los promedios, lo cual se explica en el apartado *2.1.2* de esta sección. Luego de calcular los promedios y sus respectivos errores, se guardaran dichos datos en dos archivos con formato CSV: *promediosMatrices.csv* y *erroresMatrices.csv*. El código fuente de *Matlab* que se encarga de esto se muestra en el *código 2.1*.

Código 2.1 – tp1ItemA.m

```
% Función que calcula los promedios de las matrices del TP1 (Item A).
% POST: se crean dos archivos con formato CSV: uno que contiene los
% promedios (promediosMatrices.csv) y otro que alberga al error de cada
% matriz (erroresMatrices.csv). Ambos archivos se crean en el directorio
% raiz del workspace actual del MATLAB.
function [] = tp1ItemA()
```

```
    % Configuración
    directorio = 'Matrices/';
    extension = '.txt';
    matrices = [917 977 1038 1100 1162 1225 1289 1353];
    dimensionMatrices = size(matrices);
    cantidadMatrices = dimensionMatrices(1,2);

    % Calculamos los promedios y los errores
    [promediosDeMatrices , erroresDeMatrices] =
tp1CalculoDePromediosYErrores(matrices, directorio, extension);

    % Generamos los archivos con formato CSV donde se albergan
    % los promedios calculados y los errores.
    csvwrite('promediosMatrices.csv',
promediosDeMatrices(1,1:cantidadMatrices));
    csvwrite('erroresMatrices.csv', erroresDeMatrices(1,1:cantidadMatrices));

end
```

Como ha podido observar, en el código se utilizó una función externa *tp1CalculoDePromediosYErrores* la cual es la que se encarga de realizar los cálculos y devolver los resultados. Dicha disposición del código se debe a que se trató de modularizar distintas acciones, de manera tal de evitar tener un solo código fuente extenso y poco legible. El código fuente de la función antes mencionada se muestra en el *código 2.2.*

Código 2.2 – tp1CalculoDePromediosYErrores.m

```
% Función que dado un vector con los nombres de archivo de matrices
% calcula el promedio y el error de cada una y devuelve un vector con los
% mismos.
% PRE: matrices es el vector con los nombre de archivo de las matrices;
% directorio es el directorio en donde se encuentran los archivos;
% extensión es la extensión de los archivos que contienen las matrices.
% POST: se devuelven dos matrices de 1 fila, alojandose en la primera
% los promedios y en la segunda los correspondientes errores.
function [promediosDeMatrices , erroresDeMatrices] =
tp1CalculoDePromediosYErrores(matrices, directorio, extension)

    % Vemos la cantidad de matrices a analizar
    matricesDimension = size(matrices);
    cantMatrices = matricesDimension(1,2);

    % Creamos una matriz de promedios finales y otra de errores
    promediosDeMatrices = zeros(1, cantMatrices);
    erroresDeMatrices = zeros(1, cantMatrices);
    % Calculamos los promedios de los elementos de cada matriz.
    for i = 1:cantMatrices

        matriz = load(strcat(directorio, int2str(matrices(1,i)), extension));
        promediosDeMatrices(1,i) = mean(mean(matriz));
        erroresDeMatrices (1,i) =
```

```
tp1CalculoDeError(matriz,promediosDeMatrices(1,i));

    end

end
```

Hay dos cosas para observar en este último código. Por una parte, nuevamente nos encontramos con el uso de otra función externa, *tp1CalculoDeError*, que es la encargada de calcular los respectivos errores de las matrices. El funcionamiento de dicha función será explicada en el apartado *2.1.2* de esta sección. Otro punto a detallar es la forma en la que se calculó el promedio. Para esto se utilizó la función *mean* de *Matlab*, la cual devuelve un vector con los promedios de las columnas de una matriz pasada por parámetro. En nuestro caso, para conseguir el promedio total de los elementos hemos hecho el promedio de los promedios de las columnas con dicha función. La demostración de porque es posible realizar esto se encuentra en el apartado *2.1.1* de esta sección.

Habiendo ejecutado la función *tp1ItemA()*, se crearán de forma automática los archivos con formato CSV ya mencionados. Estos últimos son importados desde el archivo de Microsoft Excel *GraficoTempVsProm.xlsx*, estableciéndose en este una tabla de temperaturas, promedios y errores (*Tabla 2.1*).

T(°C)	**917**	**977**	**1038**	**1100**	**1162**	**1225**	**1289**	**1353**
Promedio	25	30	39	55	80	102	128	153
Error	5,992	5,712	8,369	12,247	14,143	15,182	16,964	23,742
Cota inf.	18,906	24,547	31,009	42,451	66,090	86,658	110,636	129,038
Cota sup.	30,890	35,971	47,747	66,945	94,376	117,022	144,564	176,522

Tabla 2.1 – Valores de temperatura, promedios y errores.

En el mismo archivo se generó el gráfico correspondiente a la *Temperatura Vs. Promedios* (*T vs. V*), el cual se muestra en el *Gráfico 2.1*. De la curva generada en este se obtiene una función de tendencia la cual se refiere a la correlación entre temperaturas y escala de grises, siendo esta la siguiente:

$$T = 490{,}05.V^{0,2}$$

donde T es la temperatura y V es un número natural comprendido entre 0 y 255 correspondiente al color en la escala de grises.

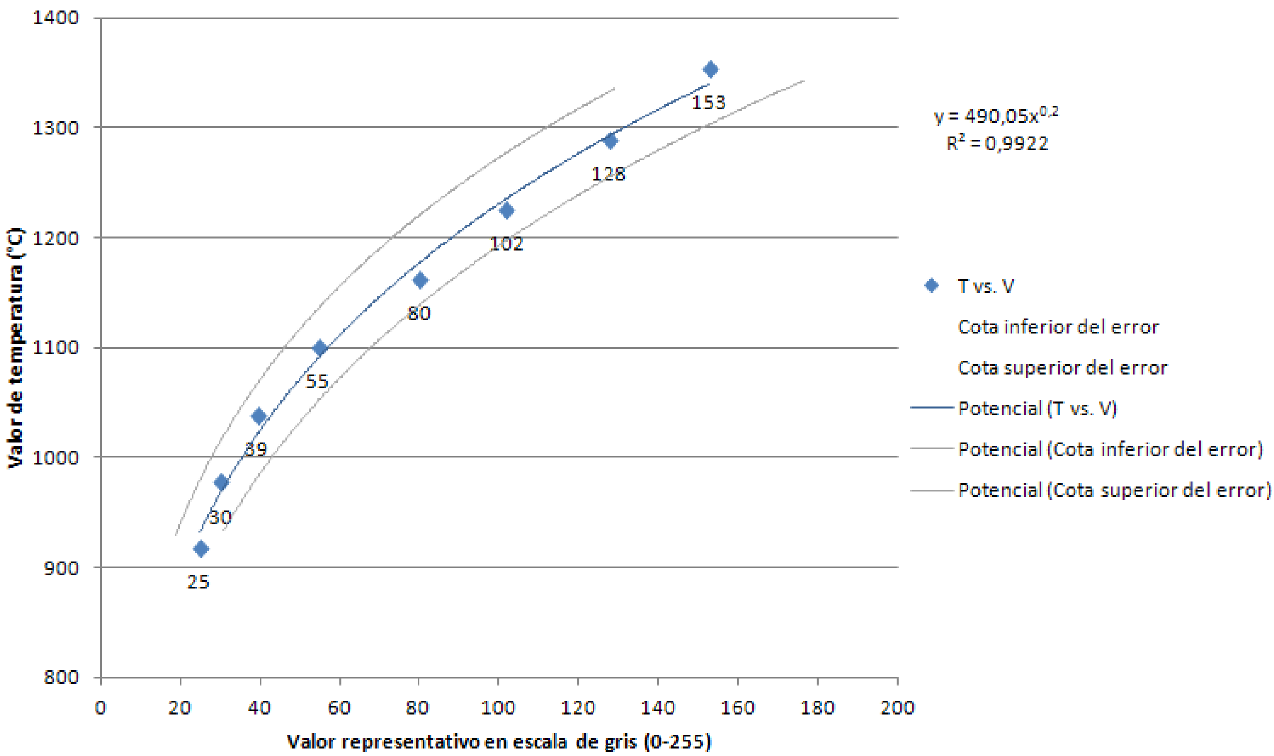

Gráfico 2.1 – T vs. V.

2.1.1 Demostración del cálculo del promedio de valores de una matriz

Pasaremos ahora a demostrar el porque es posible calcular el promedio de los valores de una matriz haciendo el promedio del promedio de las columnas de esta. Para ello, supongamos una matriz genérica de dimensión *mxn:*

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

Vamos entonces a partir de plantear el promedio de los promedios de las columnas de la matriz. Los promedios correspondientes a cada columna resultan:

$$Prom(C_1) = \frac{1}{m}\sum_{i=1}^{m} a_{i1} = \frac{a_{11} + a_{21} + \cdots + a_{m1}}{m}$$

$$Prom(C_2) = \frac{1}{m}\sum_{i=1}^{m} a_{i2} = \frac{a_{12} + a_{22} + \cdots + a_{m2}}{m}$$

$$\vdots$$

$$Prom(C_m) = \frac{1}{m}\sum_{i=1}^{m} a_{in} = \frac{a_{1n} + a_{2n} + \cdots + a_{mn}}{m}$$

siendo C_i la i-ésima columna de la matriz. Ahora, si calculamos el promedio de los promedios anteriores, resulta:

$$Prom(total) = \frac{Prom(C_1) + Prom(C_2) + \cdots + Prom(C_m)}{n} =$$

$$= \frac{\frac{a_{12} + a_{22} + \cdots + a_{m2}}{m} + \frac{a_{12} + a_{22} + \cdots + a_{m2}}{m} + \cdots + \frac{a_{1n} + a_{2n} + \cdots + a_{mn}}{m}}{n} =$$

$$= \frac{\frac{1}{m}\left(a_{12} + a_{22} + \cdots + a_{m2} + a_{12} + a_{22} + \cdots + a_{m2} + \cdots + a_{12} + a_{22} + \cdots + a_{m2}\right)}{n} =$$

$$= \frac{a_{12} + a_{22} + \cdots + a_{m2} + a_{12} + a_{22} + \cdots + a_{m2} + \cdots + a_{12} + a_{22} + \cdots + a_{m2}}{mn}$$

Esta última expresión la podemos escribir de la siguiente manera:

$$Prom(total) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}$$

Claramente se puede observar que este último resultado equivale al cociente de la suma de todos los valores y la cantidad de valores albergados en la matriz, por lo que queda demostrado que *es posible calcular el promedio de una matriz genérica haciendo el promedio de los promedios de sus columnas.*

2.1.2 Cálculo del error de desviación de una matriz

El cálculo de errores de las matrices se realiza en la función llamada *tp1CalculoDeErrores*.

Para el cálculo del error de las matrices, se consideró el uso de 3 desviaciones medias estándar. La desviación media (σ) es una medida cuadrática del grado de dispersión de los valores de la matriz respecto del valor promedio de la misma. Es definida también (en estadística) como la raíz cuadrada de la varianza, siendo esta última el promedio de las diferencias con la media elevadas al cuadrado.

Para cada matriz se calculo su desvío medio, utilizando como media el promedio de la misma, calculado anteriormente en la función *tp1CalculoDePromediosYErrores.*

Los valores de error de cada matriz fueron luego guardados en el archivo de formato CSV mencionado con anterioridad.

El código fuente de *tp1CalculoDeErrores* se muestra a continuación en el *código2.3*.

Código 2.3 – tp1CalculoDeErrores.m

```
% Función que calcula el error de una matriz.
% PRE: se debe pasar por parametro una matriz y el promedio de sus elementos.
% POST: se devuelve un escalar representativo del error.
function [error] = tp1CalculoDeError(matriz, promedio)

    % Con los datos de entrada se calcula el error (desviación
    % media estandar)
    sumatoria = 0;

    % Mediante dos ciclos for anidados, recorro la totalidad de la matriz
    % realizando la sumatria de los cuadrados de la diferencia entre el
    % valor (i,j) de la matriz y el promedio de la misma.
    for i = 1: size(matriz,1)
        for j = 1: size(matriz,1)
            sumatoria = sumatoria + ( (matriz(i,j) - promedio) ^2 );
        end
    end

    % Se calcula el desvio medio como la raíz de la sumatoria sobre la
    % cantidad total de elementos
    desvio = sqrt( ( 1/ ((size(matriz,1)^2)) )* sumatoria );


    % El error considerado en este caso, es igual a 3 desviaciones
    error = desvio * 3;

end
```

Una vez calculado el error de cada matriz en el archivo CSV, todos los datos fueron importados por el archivo de MS Excel *GraficoDealtaVvsV.xlsx*, y se generó un gráfico con dicho valores, el cual se muestra en el *Grafico 2.2* a continuación.

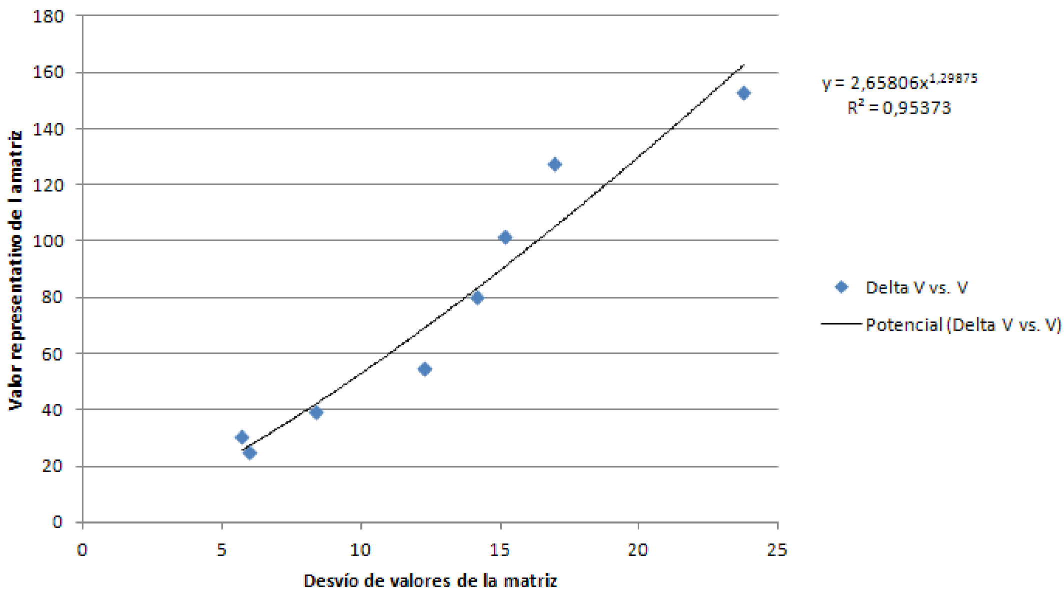

Gráfico 2.2 – ΔV vs. V.

2.2 Expresión para el error de medición de temperatura

Siendo $T = 490.05 * V^{(0.2)}$, aplicando propagación de errores mediante derivadas parciales llegamos a la expresión del error en la medición de la temperatura:

$$\mathbf{\Delta T = |\delta T/\delta V| * \Delta V = |\delta(490.05 * V^{(0.2)})/\delta V| * \Delta V = 490.05 * (0.2) * T^{(-0.8)} * \Delta V}$$

Vemos como se manifiesta los errores en el gráfico T vs. ΔT (*Gráfico 2.3*) obedeciendo a la *Tabla 2.2.*

ΔV	***T***	***ΔT***
59.922	917	0.41803398
57.117	977	0.39641541
830686	1038	0.37764816
12.247	1100	0.35864468
14.143	1162	0.34583516
15.182	1225	0.32965404
16.964	1289	0.31656505
23.742	1353	0.3033393

Tabla 2.2 – Valores de ΔV, T y ΔT.

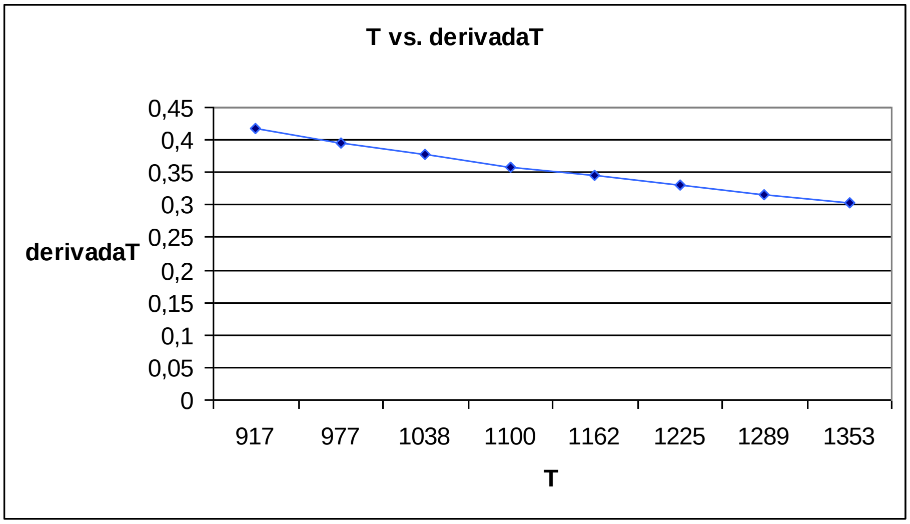

Gráfico 2.3 –T vs. ΔT.

ΔT	**V**
25049,4323	24,898
22642,0589	30,259
31603,8641	39,378
4392,32143	54,698
4891,14673	80,233
5004,80766	101,84
5370,20956	127,6
7201,88161	152,78

Tabla 2.3 – Valores de ΔT y V.

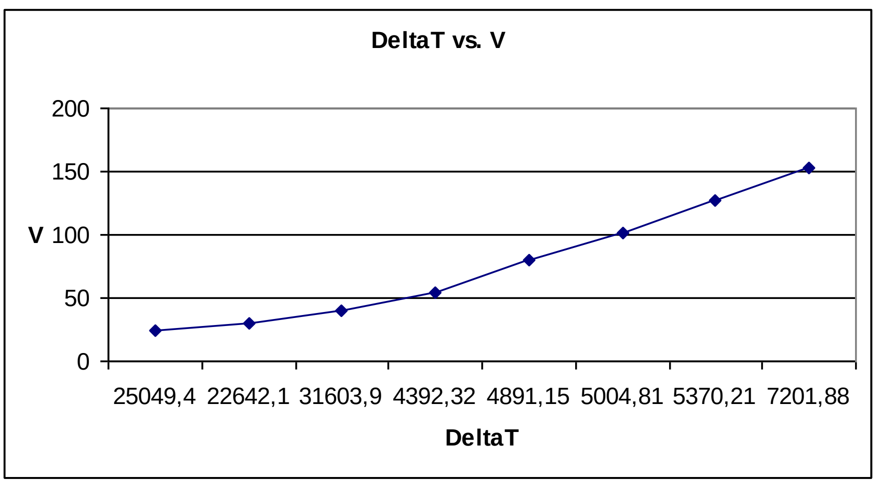

Gráfico 2.4 – ΔT vs. V.

T	ΔT
917	25049,4323
977	22642,0589
1038	31603,8641
1100	4392,32143
1162	4891,14673
1225	5004,80766
1289	5370,20956
1353	7201,88161

Tabla 2.4 – Valores de T y ΔT.

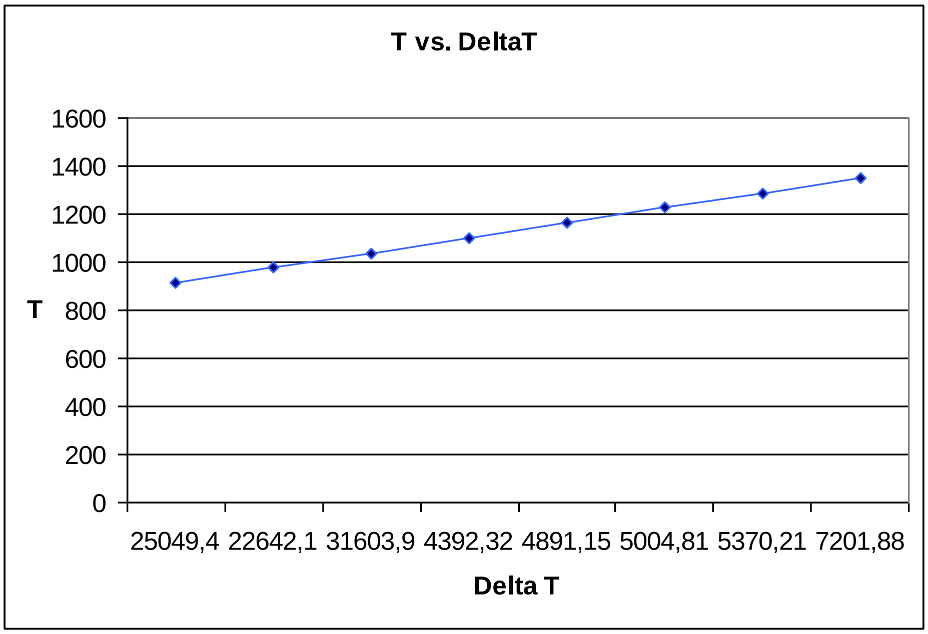

Gráfico 2.5 – T vs. ΔT.

Se ha elegido el *criterio estadístico* por dos razones, en primer lugar ser el de mayor aplicación en la realidad, y en segundo lugar por tener una gran cantidad de datos por foto, la mayoría con valores cercanos y similares. El criterio conservativo proponía errores mayores que no son necesariamente útiles en este caso. Quizás para el caso de la foto con calentamiento no uniforme tendría mas sentido abarcar en su totalidad un mayor espectro de valores, y por ende tendríamos una posibilidad de errores mayores.

Asimismo, calculamos su *Factor de Amplificación*:

$$\mathbf{FA = [\ x/(490.05 * x^{(0.2)})] * 490.05 * 0.2 * x^{(-0.8)} = x/x = 1}$$

2.3 Procesamiento de fotografía

En este caso nos ocuparemos en procesar una fotografía completa de un objeto sometido a un calentamiento no uniforme (*Imagen 2.1*) y a partir de esta generaremos un gráfico del campo de temperaturas del mismo, del cual nos serviremos para ubicar el punto más caliente allí observado.

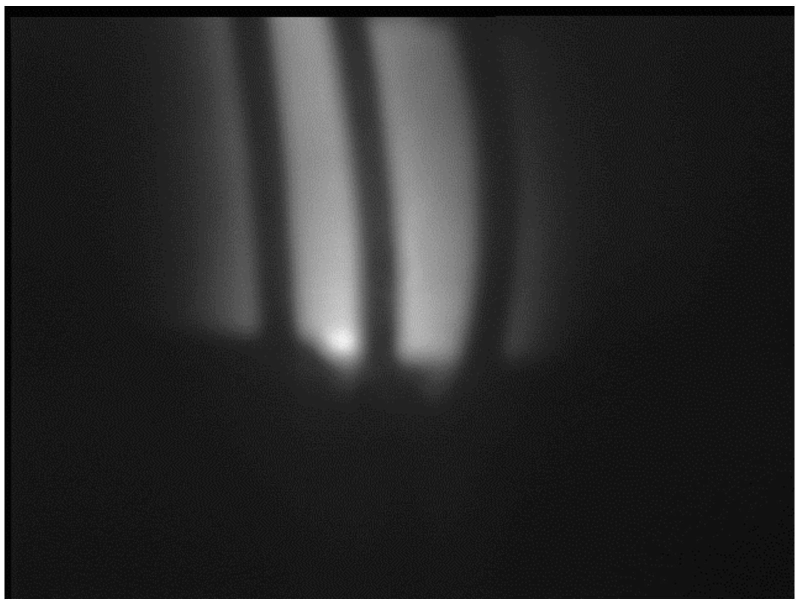

Imagen 2.1 – Fotografía de un objeto sometido a un calentamiento no uniforme.

Nuevamente se hizo uso del software *Matlab* para poder, mediante un corto script, lograr el cometido. Dicho script se encarga de abrir y procesar la imagen, convirtiéndola en una matriz de dimensión 640x480 cuyos índices contienen un número representativo comprendido entre 0 y 255 correspondiente a la escala de grises. Luego se crea otra matriz con las mismas dimensiones que la anterior pero con la diferencia de que ahora, en cada índice se guardará la temperatura respectiva de cada pixel de acuerdo a con la ecuación antes obtenida:

$$T = 490{,}05.V^{0,2}$$

siendo T la temperatura y V el valor representativo del tono de color del punto.

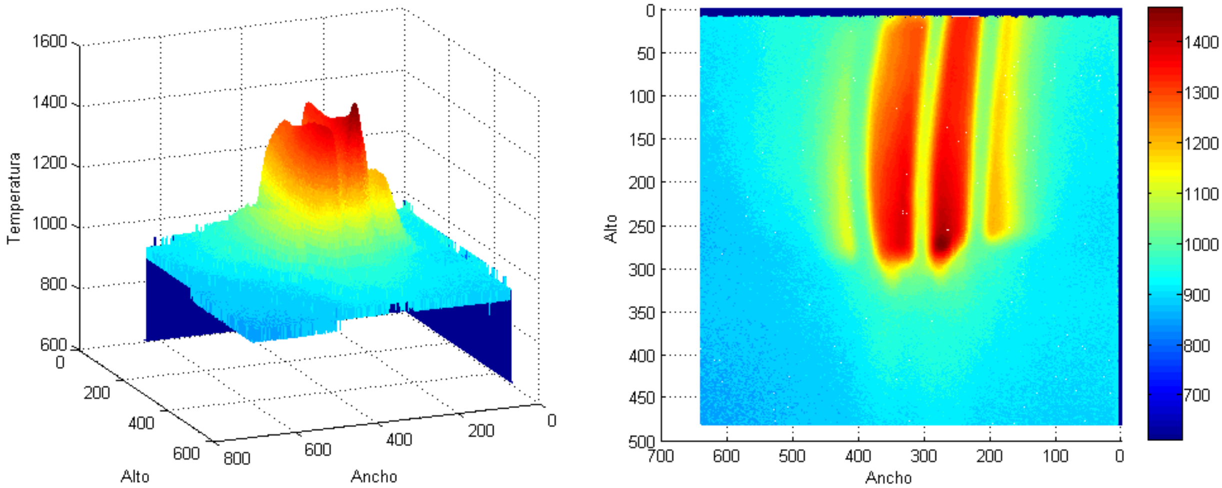

Gráfico 2.6 – Campo de temperaturas de la fotografía de la *Imagen 2.1*.

Por último, se utiliza la matriz de temperaturas calculada para realizar un gráfico 3D del campo de temperaturas de la fotografía (*Gráfico2.6)*. En este se puede observar el punto más caliente a unos 1465,28°C que es donde se genera el pico más alto de la superficie.

El código fuente al que nos referimos en este apartado puede verse en el *Código 2.4*.

Código 2.4 – tp1ItemC.m

```
% Función que a partir de una fotografía completa de un objeto sometido a
% un calentamiento no uniforme, genera un gráfico de temperaturas del mismo
% y encuentra el punto más caliente.
function [] = tp1ItemC()

    % Configuración
    nombreImagen = '1160.tif';
    directorio = 'Matrices/';
    dimM = 480;
    dimN = 640;

    % Procesamos el archivo de la imagen.
    matrizDeImagen =imread(strcat(directorio, nombreImagen));
    %Generamos una matriz nueva de iguales dimensiones a la imagen.
    matrizDeTemperaturasDeImagen = zeros(dimM, dimN);

    % Calculamos la temperatura de cada pixel de la imagen y la almacenamos
    % en la respectiva matriz.
    for i = 1:dimM
        for j = 1:dimN

            matrizDeTemperaturasDeImagen(i,j) = 490.05 * ( (
```

```
double(matrizDeImagen(i,j)) ) ^ 0.2);

        end
    end

    % Imprimimos la temperatura máxima de la imagen
    strcat('Temperatura máxima: ',
num2str(max(max(matrizDeTemperaturasDeImagen))), '°C')

    % Generamos el gráfico 3D de temperaturas.
    x = 1:1:640;
    y = 1:1:480;
    [X,Y] = meshgrid(x,y);

    mesh(X,Y,matrizDeTemperaturasDeImagen)
    title('GRÁFICO DE TEMPERATURAS DE UNA IMAGEN')
    colorbar
    xlabel('Ancho')
    ylabel('Alto')
    zlabel('Temperatura')

end
```

3 Conclusiones

Como cierre del trabajo práctico, podemos observar lo factible que resulta el poder, a partir de datos experimentales, crear un modelo que se ajuste a dichos datos de manera tal de poder ser aplicado en otras situaciones desconocidas.

En nuestro caso, pudimos notar cómo, a partir de 8 fotografías a temperaturas conocidas pudimos elaborar una escala aplicable a otros casos desconocidos. Sin embargo, llegamos no a valores exactos sino a estimaciones que, como tales, conllevan un error de estimación. Son imprescindibles los métodos numéricos a la hora de cuantificar estos errores y por ende conocer el valor que tiene la estimación como modelo de la realidad.

En primer lugar, la propagación de errores calculada a partir del desarrollo de Taylor provee una referencia para saber cuánto error se comete al estimar distintos niveles de temperatura, y por otro lado la estimación del error cometido al hacer los promedios, llevan a poder determinar un intervalo confiable dentro del cual futuras mediciones ópticas de la temperatura pueden ser consideradas como factibles. Es este uso solo un comienzo de una enorme cantidad de aplicaciones que tienen los métodos numéricos dentro de la ingeniería.

Notas finales

Dependemos en este trabajo, de las mediciones iniciales efectuadas con el pirómetro. No tenemos datos en cuanto a su precisión y tomamos las temperaturas dadas como las reales del objeto, sin indagar sobre un error inherente en estas mediciones iniciales. Desde ya podemos añadir, que si los objetos realmente están a la temperatura establecida, deberían presentar un calentamiento uniforme y univoco a través de toda su superficie, y no existir una variedad de temperaturas sobre las cuales hacer un promedio. Es esta también otra razón por la cual optamos por un método estadístico a la hora de estimar el error en los promedios de cada matriz, ya que es más verosímil a una distribución real de temperaturas a través de un sólido.

Estamos también atados desde un comienzo a la escala de grises de 256 posibilidades. Podríamos, quizás en una medida apreciable, mejorar las mediciones en un mayor número de decimales si dispusiésemos de mayores cantidades de valores de gris posibles. Tal estudio sin embargo, escapa a este trabajo práctico.