Libro: Quarteroni, “Scientific Computing with MATLAB and Octave”

PROBLEMA 3.1 – CLIMATOLOGÍA (pag. 77)

La temperatura del aire cerca al suelo depende de la concentración K del ácido carbónico (H2CO3) en el aire. En la Tabla 3.1 (pag. 78 del libro) se reporta el valor δ_K como función de la latitud de nuestro planeta y la concentración K. El valor δ_K se define como:

$$\delta_K = \theta_K - \theta_{\bar{K}}$$

donde θ_K es la temperatura promedio que corresponde a una concentración K y a cierta latitud, y $\theta_{\bar{K}}$ es la temperatura promedio que corresponde a esa latitud pero a una concentración de referencia $\bar{K}$. Este valor de referencia fue medido en 1896 y es normalizado como $\bar{K} = 1$.

Ahora, el objetivo es generar una función que nos dé un valor δ_K aproximado para cualquier latitud (y para un K fijo), a partir de los datos disponibles. Las siguientes etapas nos ayudarán a cumplir el objetivo:

1. Importar la tabla de datos
2. Escoger una función modelo
3. Cálculo de los coeficientes de la función
4. Análisis de la función modelo

1 IMPORTAR LA TABLA DE DATOS

La Tabla 3.1 se encuentra en el archivo “problema_3_1.xlsx”. Vamos a importar este archivo al Matlab y a guardarlo como una matriz de números. El proceso es el siguiente:

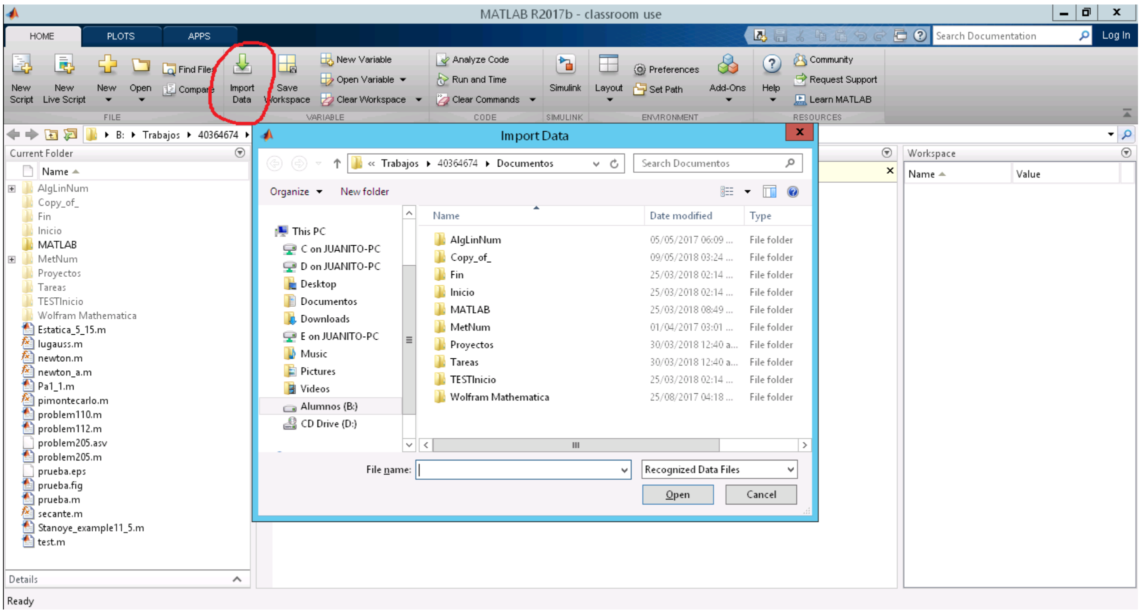

Figura 1.

Hacer click en Import Data y se abre una nueva ventana (Figura 1). En ésta deben buscar el archivo “problema_3_1.xlsx”. Si usan el elabs.pucp.edu.pe deben seleccionar el disco de su computador (en mi caso es “C on JUANITO-PC”).

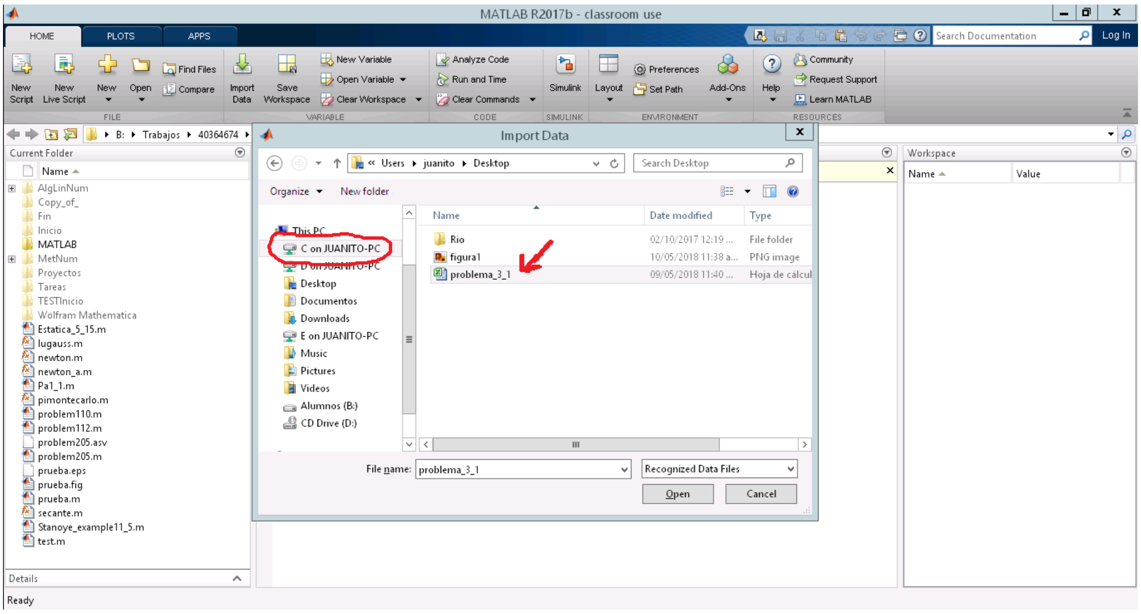

Figura 2

Al encontrar el archivo Excel (Figura 2), hacen click en Open y se abre una nueva ventana (Figura 3). En ésta, verifiquen que el rango de datos corresponde a todos los datos numéricos.

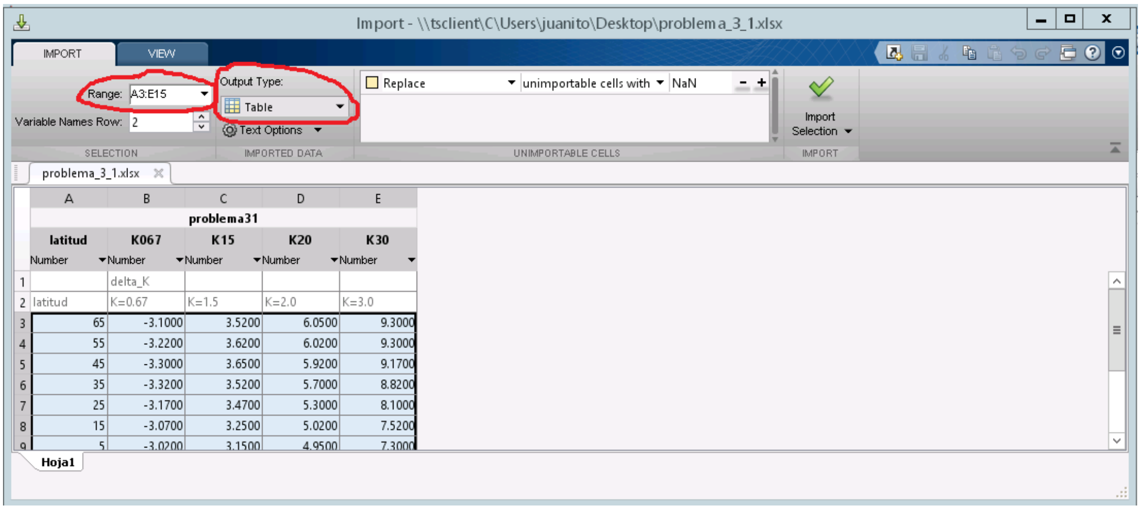

Figura 3

Además, observen el tipo de salida "Output Type" (en mi caso, aparece "Table"). Aquí deben seleccionar tipo "Numeric Matrix" (Figura 4) y hacer click en Import Data.

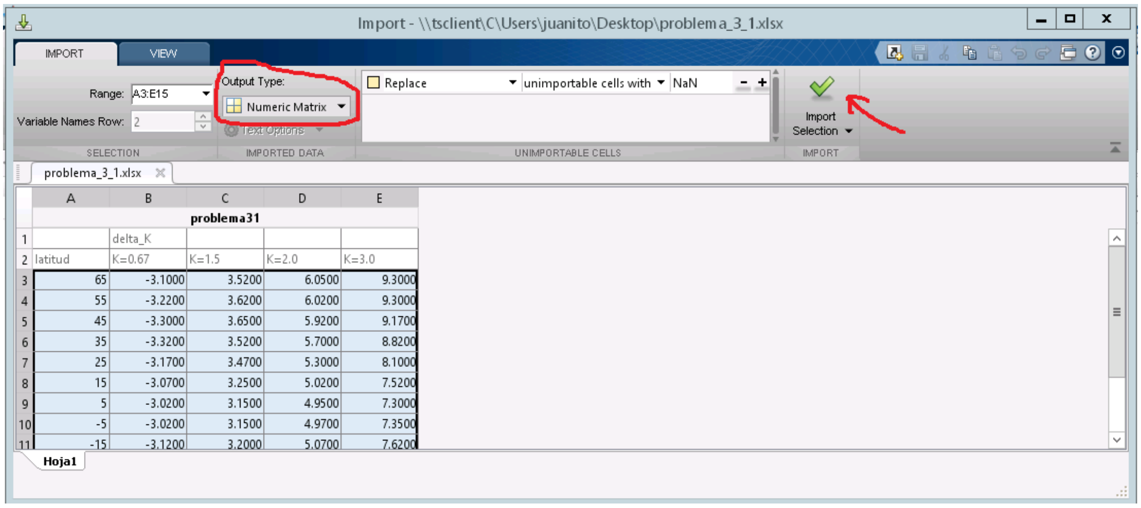

Figura 4

Así se genera una variable "problema31" tipo matriz de tamaño 13x5. Note que las latitudes están en la primera columna y los δ_K en las otras cuatro columnas.

```
>> problema31

problema31 =

   65.0000   -3.1000    3.5200    6.0500    9.3000
   55.0000   -3.2200    3.6200    6.0200    9.3000
   45.0000   -3.3000    3.6500    5.9200    9.1700
   35.0000   -3.3200    3.5200    5.7000    8.8200
   25.0000   -3.1700    3.4700    5.3000    8.1000
   15.0000   -3.0700    3.2500    5.0200    7.5200
    5.0000   -3.0200    3.1500    4.9500    7.3000
   -5.0000   -3.0200    3.1500    4.9700    7.3500
  -15.0000   -3.1200    3.2000    5.0700    7.6200
  -25.0000   -3.2000    3.2700    5.3500    8.2200
  -35.0000   -3.3500    3.5200    5.6200    8.8000
  -45.0000   -3.3700    3.7000    5.9500    9.2500
  -55.0000   -3.2500    3.7000    6.1000    9.5000
```

2 ESCOGER UNA FUNCIÓN MODELO

Recuerde que el objetivo es generar una función que nos dé un valor δ_K aproximado para cualquier latitud (y para un K fijo), a partir de los datos disponibles. Entonces primero elegimos K= 0,67 (el lector podría escoger otro) y plotamos δ_K versus latitud, es decir la columna 2 vs la columna 1 de la matriz de datos "problema31".

```
>> plot(problema31(:,1),problema31(:,2),'*-')
```

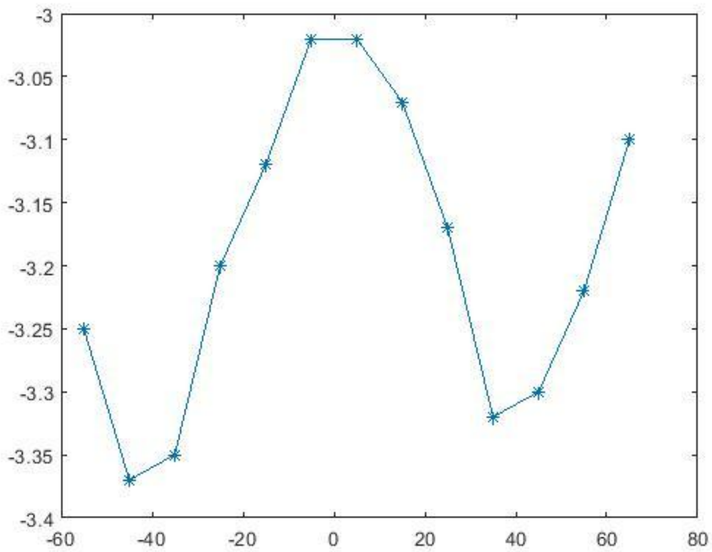

Figura 5

Ahora debemos conjeturar cuál es el comportamiento que "gobierna" estos datos (Figura 5). Una función podría ser polinómica de grado 4 pues estos polinomios presentan oscilaciones (ver la Figura 3.16 del libro). Otra función podría ser periódica, del tipo $\cos\left(\frac{2\pi x}{T}\right)$, pues la Figura 5 así lo sugiere y además el comportamiento parece ser simétrico respecto a la latitud cero (x=0). Así, se sugieren dos opciones:

a) Polinomio de grado 4,
$$\tilde{f}(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4$$
En este caso, el siguiente paso será encontrar los coeficientes a_0, a_1, a_2, a_3, a_4 que multiplican a los elementos de la base de $\tilde{f}(x)$, es decir: $1, x^1, x^2, x^3, x^4$.

b) Suma de Cosenos,
$$\tilde{f}(x) = a_0 + a_1 \cos\left(\frac{2\pi x}{T}\right) + a_2 \cos\left(\frac{4\pi x}{T}\right)$$
donde T=65-(-55)=120, es el tamaño del intervalo de latitudes dado; o sea
$$\tilde{f}(x) = a_0 + a_1 \cos\left(\frac{\pi x}{60}\right) + a_2 \cos\left(\frac{\pi x}{30}\right)$$
En este caso, el siguiente paso será encontrar los coeficientes a_0, a_1, a_2, que multiplican a los elementos de la base de $\tilde{f}(x)$, es decir: $1, \cos\left(\frac{\pi x}{60}\right), \cos\left(\frac{\pi x}{30}\right)$. El lector podría tomar más elementos de este tipo, es decir: $\cos\left(j\frac{2\pi x}{T}\right), j = 3,4,5, \ldots$
Nota: La Figura 3.17 del libro muestra la respuesta para este caso, donde la función $\tilde{f}(x)$ está basada en cosenos.

3 CÁLCULO DE LOS COEFICIENTES DE LA FUNCIÓN

El objetivo en esta Sección es encontrar los coeficientes correspondientes a cada función $\tilde{f}(x)$ considerada en la Sección 2. Para ello, recuerde que la función $\tilde{f}(x)$ nos debe dar un valor δ_K aproximado para cualquier latitud (es decir, para cualquier $x \in [-55{,}65]$). En particular, si x es la latitud 65

$$\tilde{f}(65) \approx \delta_K = -3{,}1$$

(ver matriz de datos "problema31" o archivo Excel). Aproximaciones similares debemos conseguir con las otras doce latitudes dadas. Formalicemos estas aproximaciones construyendo el vector "latitud" y el vector "b", a partir de la matriz "problema31"

```
>> latitud=problema31(:,1)

latitud =

    65
    55
    45
    35
    25
    15
     5
    -5
   -15
   -25
   -35
   -45
   -55

>> b=problema31(:,2)

b =

   -3.1000
   -3.2200
   -3.3000
   -3.3200
   -3.1700
   -3.0700
   -3.0200
```

-3.0200

-3.1200

-3.2000

-3.3500

-3.3700

-3.2500

Así, las aproximaciones se pueden escribir así

$$\tilde{f}(latitud_i) \approx b_i \tag{1}$$

para i=1,2,…,13. Estas aproximaciones deben ser satisfechas por la función $\tilde{f}$. Ahora, para encontrar los coeficientes de $\tilde{f}$, se re-escriben estas aproximaciones como:

$$A \cdot xx \approx b$$

donde el vector "xx" está formado por los coeficientes desconocidos de la función $\tilde{f}$. Y la matriz A depende de la función $\tilde{f}$ y del vector "latitud"; la construcción de esta matriz A se detalla a continuación:

a) Cuando $\tilde{f}$ es un polinomio de grado 4, la Ecuación (1) queda:

$$a_0 + a_1 latitud_i{}^1 + a_2 latitud_i{}^2 + a_3 latitud_i{}^3 + a_4 latitud_i{}^4 \approx b_i$$

para i=1,2,…,13. O sea:

$$\begin{bmatrix} 1 & latitud_1 & \cdots & latitud_1{}^4 \\ 1 & latitud_2 & & latitud_2{}^4 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & & \vdots \\ 1 & latitud_{13} & \cdots & latitud_{13}{}^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_4 \end{bmatrix} \approx \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ \vdots \\ b_{13} \end{bmatrix} \tag{2}$$

La matriz 13x5 de esta Ecuación (2) es almacenada en la variable A, en el Matlab:

```
>> A=zeros(13,5);
>> A(:,1)=1;
>> A(:,2)=latitud;
>> A(:,3)=latitud.^2;
>> A(:,4)=latitud.^3;
>> A(:,5)=latitud.^4
A =
     1    65    4225   274625   17850625
     1    55    3025   166375    9150625
     1    45    2025    91125    4100625
     1    35    1225    42875    1500625
     1    25     625    15625     390625
     1    15     225     3375      50625
     1     5      25      125        625
     1    -5      25     -125        625
     1   -15     225    -3375      50625
     1   -25     625   -15625     390625
     1   -35    1225   -42875    1500625
```

```
1      -45      2025     -91125    4100625
1      -55      3025    -166375    9150625
```

El lado derecho de la Ecuación (2) es el vector “b” y el vector de incógnitas $[a_0, a_1, a_2, a_3, a_4]^T$ será denominado como vector “xx”.
Es decir, los coeficientes de la función polinomial $\tilde{f}$ constituyen la solución “xx” de mínimos cuadrados del sistema:

$$A \cdot xx \approx b$$

Este problema será resuelto vía la descomposición QR y luego, la factorización SVD.

Vía descomposición QR:

```
>> Q=zeros(13,13);
>> R=zeros(13,5);
>> [Q,R]=qr(A);              % FACTORIZACIÓN COMPLETA
>> A-Q*R                     % COMPROBANDO
ans =
  1.0e-08 *
  -0.0000   -0.0000   -0.0004   -0.0058   -0.7451
        0   -0.0000         0    0.0029    0.1863
        0   -0.0000   -0.0001   -0.0044   -0.8382
        0   -0.0000   -0.0001   -0.0007   -0.3725
        0   -0.0000   -0.0001   -0.0007   -0.4075
        0   -0.0000   -0.0001   -0.0036   -0.4424
        0   -0.0000   -0.0001   -0.0038   -0.4065
        0   -0.0000   -0.0001   -0.0044   -0.6461
        0   -0.0000   -0.0001   -0.0029   -0.4657
        0   -0.0000   -0.0001   -0.0044   -0.4424
        0   -0.0000   -0.0001   -0.0015   -0.3725
        0   -0.0000   -0.0001   -0.0058   -0.2794
        0   -0.0000   -0.0000         0         0
>> Qred=Q(:,1:5); Rred=R(1:5,:);          % FACTORIZACIÓN REDUCIDA
>> A-Qred*Rred                            % COMPROBANDO
ans =
  1.0e-08 *
  -0.0000   -0.0000   -0.0004   -0.0058   -0.7451
        0   -0.0000         0    0.0029    0.1863
        0   -0.0000   -0.0001   -0.0044   -0.7916
        0   -0.0000   -0.0001   -0.0007   -0.4191
        0   -0.0000   -0.0001   -0.0007   -0.4075
        0   -0.0000   -0.0001   -0.0036   -0.4424
        0   -0.0000   -0.0001   -0.0038   -0.3958
        0   -0.0000   -0.0001   -0.0044   -0.6519
        0   -0.0000   -0.0001   -0.0029   -0.4657
        0   -0.0000   -0.0001   -0.0044   -0.4424
        0   -0.0000   -0.0001   -0.0015   -0.3492
        0   -0.0000   -0.0001   -0.0058   -0.2794
        0   -0.0000   -0.0000         0         0
```

Ahora resolvemos Rred * xx = Qred’ * b, donde Rred es una matriz triangular superior:

```
>> y=Qred'*b
```

```
y =
  11.5128
  -0.0993
  -0.1909
   0.1196
  -0.3225
>> format long e
>> xx=Rred\y                          % SOLUCIÓN DE MÍNIMOS CUADRADOS VÍA QR
xx =
   -3.030273866758244e+00
    1.650290885585012e-03
   -2.987753667900723e-04
   -6.088029617441450e-07
    7.210681475387340e-08
```

Nota: el rango de A es 5 (use el comando rank del Matlab) y por lo tanto la matriz Rred es invertible y la solución de mínimos cuadrados xx es única.

Ahora veamos la solución vía la descomposición en valores singulares SVD:

```
>> U=zeros(13,13);S=zeros(13,5);V=zeros(5,5);
>> [U,S,V]=svd(A);        % FACTORIZACIÓN COMPLETA
>> A-U*S*V'               % COMPROBANDO
ans =
  1.0e-08 *
  -0.0000  -0.1167   0.0024  -0.0116  -0.3725
  -0.0000  -0.0097   0.0024        0   0.3725
  -0.0000   0.0456   0.0018  -0.0058  -0.5588
        0   0.0686   0.0014  -0.0029  -0.3725
        0   0.0740   0.0011  -0.0016  -0.3260
        0   0.0727   0.0008  -0.0050  -0.3827
   0.0000   0.0713   0.0006  -0.0069  -0.3622
  -0.0000   0.0720   0.0004  -0.0067  -0.6012
  -0.0000   0.0731   0.0002  -0.0043  -0.3958
  -0.0000   0.0685  -0.0003  -0.0029  -0.3900
  -0.0000   0.0480  -0.0012  -0.0015  -0.2561
  -0.0000  -0.0028  -0.0028   0.0015        0
  -0.0000  -0.1024  -0.0053   0.0087   0.1863
>> Ured=U(:,1:5); Sred=S(1:5,:);          % FACTORIZACIÓN REDUCIDA
>> A-Ured*Sred*V'                         % COMPROBANDO
ans =
  1.0e-08 *
  -0.0000  -0.1167   0.0024  -0.0116  -0.3725
  -0.0000  -0.0097   0.0024        0   0.3725
  -0.0000   0.0456   0.0018  -0.0058  -0.5588
        0   0.0686   0.0014  -0.0029  -0.3725
        0   0.0740   0.0011  -0.0016  -0.3260
        0   0.0727   0.0008  -0.0050  -0.3827
   0.0000   0.0713   0.0006  -0.0069  -0.3622
```

```
  -0.0000    0.0720    0.0004   -0.0067   -0.6012
  -0.0000    0.0731    0.0002   -0.0043   -0.3958
  -0.0000    0.0685   -0.0003   -0.0029   -0.3900
  -0.0000    0.0480   -0.0012   -0.0015   -0.2561
  -0.0000   -0.0028   -0.0028    0.0015         0
  -0.0000   -0.1024   -0.0053    0.0087    0.1863
```

Ahora resolvemos Sred * V’ * xx = Ured’ * b, donde Sred es una matriz diagonal:

```
>> Ured'*b
ans =
    6.7546
   -1.8698
   -7.0085
   -0.8141
    5.8144
>> format long e;
>> y=Sred\ans                          % SISTEMA DIAGONAL Sred * y = Ured’ * b
y =
     2.949056164165404e-07
    -5.749872279014369e-06
    -3.586584257617697e-03
    -1.627741409592406e-02
     3.030228490138737e+00
>> xx=V*y                              % SOLUCIÓN DE MÍNIMOS CUADRADOS VÍA SVD
xx =
    -3.030273866757066e+00
     1.650290885583649e-03
    -2.987753667898960e-04
    -6.088029617391947e-07
     7.210681475302263e-08
```

Una comparación entre la solución xx vía descomposición QR y vía SVD se muestra en la Sección 4.

Preguntas:

¿Qué sucede si uso el comando polyfit del Matlab? ¿A cuál de las soluciones xx calculadas arriba se acerca más? Y si escribo xx=A\b en el Matlab, ¿qué observa?

b) Cuando $\tilde{f}$ es una suma de cosenos, la Ecuación (1) queda:

$$a_0 + a_1 \cos\left(\frac{\pi \cdot latitud_i}{60}\right) + a_2 \cos\left(\frac{\pi \cdot latitud_i}{30}\right) \approx b_i$$

para i=1,2,…,13. O sea:

$$\begin{bmatrix} 1 & \cos\left(\frac{\pi \cdot latitud_1}{60}\right) & \cos\left(\frac{\pi \cdot latitud_1}{30}\right) \\ 1 & \cos\left(\frac{\pi \cdot latitud_2}{60}\right) & \cos\left(\frac{\pi \cdot latitud_2}{30}\right) \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1 & \cos\left(\frac{\pi \cdot latitud_{13}}{60}\right) & \cos\left(\frac{\pi \cdot latitud_{13}}{30}\right) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \approx \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ \vdots \\ b_{13} \end{bmatrix} \qquad (3)$$

La matriz 13x3 de esta Ecuación (3) es almacenada en la variable A, en el Matlab:

```
>> A=zeros(13,3);
>> A(:,1)=1;
>> A(:,2)=cos(pi*latitud/60);
>> A(:,3)=cos(pi*latitud/30);
```

El lado derecho de la Ecuación (3) es el vector “b” y el vector de incógnitas $[a_0, a_1, a_2]^T$ será denominado como vector “xx”.

Es decir, los coeficientes de la suma de cosenos $\tilde{f}$ constituyen la solución “xx” de mínimos cuadrados del sistema:

$$A \cdot xx \approx b$$

Como el caso anterior, este problema será resuelto vía la descomposición QR y luego, la factorización SVD.

Vía descomposición QR:

```
>> Q=zeros(13,13);
>> R=zeros(13,3);
>> [Q,R]=qr(A);                        % FACTORIZACIÓN COMPLETA
>> Qred=Q(:,1:3); Rred=R(1:3,:);       % FACTORIZACIÓN REDUCIDA
```

Ahora resolvemos Rred * xx = Qred’ * b, donde Rred es una matriz triangular superior:

```
>> y=Qred'*b
y =
  11.5128
   0.2835
   0.2425
>> format long e
>> xx=Rred\y                  % SOLUCIÓN DE MÍNIMOS CUADRADOS VÍA QR
xx =
   -3.190532099183473e+00
    1.188337227514265e-01
    9.434117159585140e-02
```

Nota: el rango de A es 3 (use el comando rank del Matlab) y por lo tanto la matriz Rred es invertible y la solución de mínimos cuadrados xx es única.

Ahora veamos la solución vía la descomposición en valores singulares SVD:

```
>> U=zeros(13,13);S=zeros(13,3);V=zeros(3,3);
>> [U,S,V]=svd(A);                     % FACTORIZACIÓN COMPLETA
>> Ured=U(:,1:3); Sred=S(1:3,:);       % FACTORIZACIÓN REDUCIDA
```

Ahora resolvemos Sred * V’ * xx = Ured’ * b, donde Sred es una matriz diagonal:

```
>> Ured'*b
ans =
  11.3579
   1.8835
   0.3664
>> format long e;
>> y=Sred\ans                 % SISTEMA DIAGONAL Sred * y = Ured’ * b
y =
    3.114424296177970e+00
    6.931858919729723e-01
```

```
   1.495707484340845e-01
>> xx=V*y                          % SOLUCIÓN DE MÍNIMOS CUADRADOS VÍA SVD
xx =
  -3.190532099183473e+00
   1.188337227514267e-01
   9.434117159585116e-02
```

Preguntas:

¿Qué sucede si escribo xx=A\b en el Matlab? ¿A cuál de las soluciones xx calculadas arriba se acerca más?

4 ANÁLISIS DE LA FUNCIÓN MODELO

a) Cuando $\tilde{f}$ es un polinomio de grado 4,

$$\tilde{f}(x) = a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + a_4 x^4$$

se tiene que los coeficientes son:

coeficientes	QR	SVD
a0	-3.030273866758244e+00	-3.030273866757066e+00
a1	1.650290885585012e-03	1.650290885583649e-03
a2	-2.987753667900723e-04	-2.987753667898960e-04
a3	-6.088029617441450e-07	-6.088029617391947e-07
a4	7.210681475387340e-08	7.210681475302263e-08

Los coeficientes obtenidos con la descomposición QR tienen de 10 a 12 cifras iguales a los correspondientes coeficientes calculados con la factorización SVD. La Figura 6 muestra ambos polinomios y los datos δ_K (ver también Figura 5).

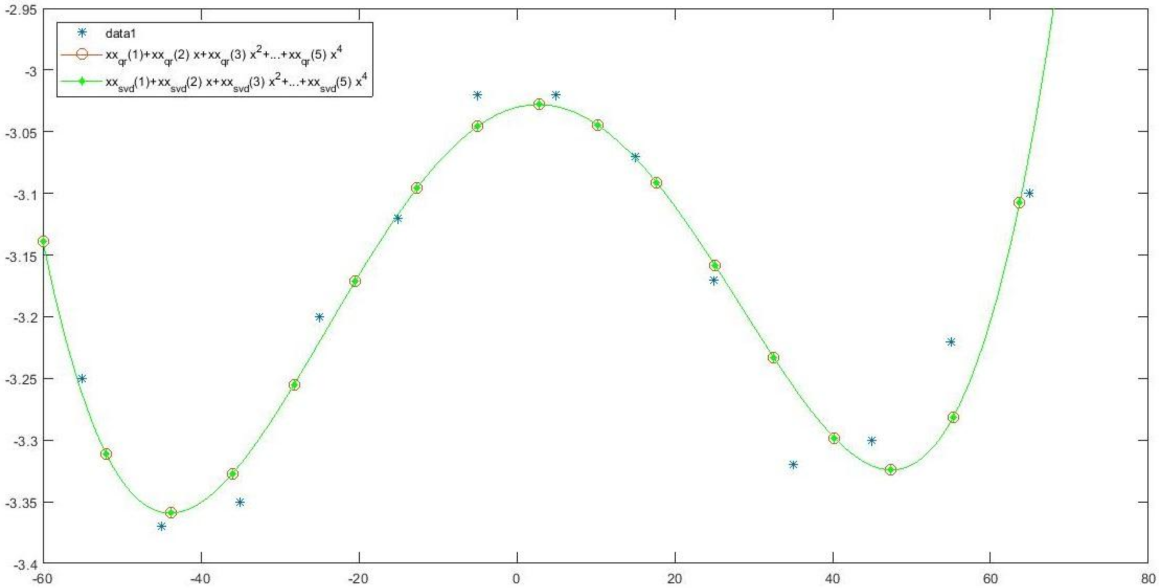

Figura 6

La diferencia entre los valores δ_K y los valores aproximados (dados por un polinomio) se puede medir mediante

$$\Delta = \sqrt{\sum_{i=1}^{13} \left(\tilde{f}(latitud_i) - b_i\right)^2} \qquad (4)$$

donde b_i es el dato δ_K para una $latitud_i$ (ver Ecuación (1)). Este número Δ equivale a la norma-2 del vector A * xx – b (ver Ecuación (2)). Los resultados son los siguientes:

diferencia Δ entre la función polinómica y los datos	
QR	SVD
1.116904149004345e-01	1.116904149004337e-01

La diferencia que se observa en la Figura 6, entre la función polinómica y los datos, está representado por el número $\Delta = 0.11169$. Este número es casi independiente del método usado para construir el polinomio (factorización QR y SVD). La Figura 6 también muestra que los polinomios generados por ambos métodos no se pueden distinguir (por lo menos en la escala de esa figura).

Se sabe que los coeficientes de la función polinomial $\tilde{f}$ constituyen la solución "xx" de mínimos cuadrados del sistema:

$$A \cdot xx \approx b$$

Y se sabe que "xx" es solución de las ecuaciones normales:

$$A^T A \cdot xx = A^T b$$

Como el rango de A es 5 (o sea, coincide con el número de columnas de A), se tiene que la matriz cuadrada $A^T A$ es inversible. Sin embargo, el número de condición de $A^T A$ (correspondiente a la norma-2) es 1.4249e+14 (basta usar el comando "cond" del Matlab), que es relativamente alto. Ahora hagamos un análisis similar para los sistemas asociados a la descomposición QR y SVD.

Cuando se usa la factorización QR, el vector "xx" es solución del sistema:

$$\hat{R} \cdot xx = \hat{Q}^T b$$

donde $\hat{R}$ es invertible (pues el rango de A es 5) y su número de condición es 1.1937e+07, bastante menor al de la matriz de las ecuaciones normales.

Finalmente, cuando se usa la factorización SVD, el vector "xx" es solución del sistema:

$$\hat{\Sigma}\, V^T \cdot xx = \hat{U}^T b$$

donde $\hat{\Sigma}$ es invertible (pues el rango de A es 5) y su número de condición es 1.1937e+07, igual al de $\hat{R}$. Este número de condición también coincide con el de $\hat{\Sigma}\, V^T$.

Es decir, los coeficientes hallados con ambas factorizaciones serían más próximos a los coeficientes exactos (sin error de redondeo) que aquellos que se obtengan al resolver directamente las ecuaciones normales.

Preguntas:

Si usted considera uno de los métodos para factorizar A ¿podría estimar el error relativo en el cálculo de xx? En esta Sección, se ha considerado un polinomio de grado 4; ¿qué sucedería si hace el cálculo con un polinomio de mayor grado, por ejemplo grado 12?

b) Cuando $\tilde{f}$ es una suma de cosenos

$$\tilde{f}(x) = a_0 + a_1 \cos\left(\frac{\pi x}{60}\right) + a_2 \cos\left(\frac{\pi x}{30}\right)$$

se tiene que los coeficientes son:

coeficientes	QR	SVD
a0	-3.190532099183473e+00	-3.190532099183473e+00
a1	1.188337227514265e-01	1.188337227514267e-01
a2	9.434117159585140e-02	9.434117159585116e-02

Los coeficientes obtenidos con la descomposición QR tienen de 14 a 16 cifras iguales a los correspondientes coeficientes calculados con la factorización SVD. Es decir, ambas funciones $\tilde{f}$ (con base de cosenos) son casi idénticas. Esta característica se observa en la Figura 7.

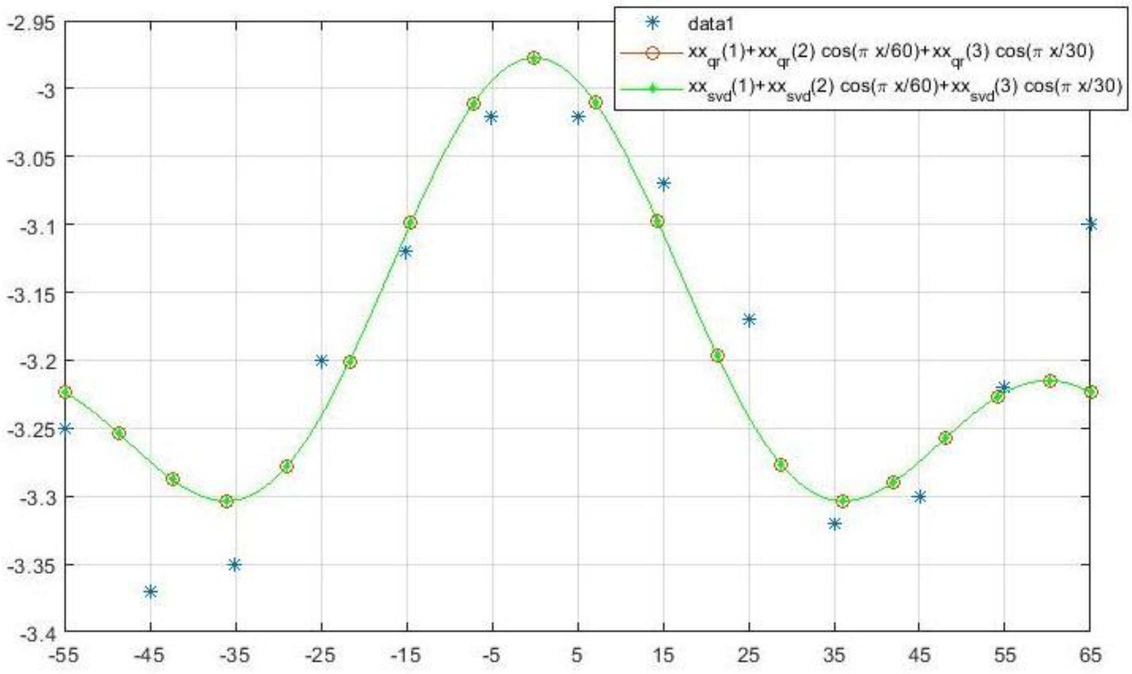

Figura 7

El número Δ de la Ecuación (4), es decir, la diferencia entre los valores δ_K y los valores aproximados (dados por $\tilde{f}$) es igual a 0.19477 (como se ve en la siguiente tabla). Este número es independiente del método usado para construir la función con base de cosenos (factorización QR y SVD).

diferencia Δ entre la función con base de cosenos y los datos	
QR	SVD
1.947726497265043e-01	1.947726497265043e-01

Al comparar con los resultados obtenidos con el modelo polinomial, se diría que el polinomio de grado 4 se ajusta mejor a los datos. Sin embargo, también hay que mencionar que fuera del intervalo de latitudes [-55,65], el polinomio (por su naturaleza) llega a dar valores δ_K lejanos a los reportados dentro del intervalo; esto no sucede con el modelo basado en cosenos pues él es periódico. Otra característica del modelo periódico es el número de condición de la matriz A^TA de las ecuaciones normales:

$$A^TA \cdot xx = A^Tb$$

Dicho número de condición es 2.2166. Este valor sugiere que los coeficientes hallados serían más próximos a los coeficientes exactos (sin error de redondeo) que los respectivos coeficientes del modelo polinomial. El lector podría considerar una función con más cosenos: $\cos\left(j\frac{2\pi x}{T}\right), j = 0,1,\ldots,4$, de modo que calcule tantos coeficientes como con el polinomio de grado 4.

La factorización QR y SVD mejoraría aún la precisión, pues los números de condición de $\hat{R}$ y de $\hat{\Sigma}$ son iguales a 1.4888.