Importante: Durante el desarrollo de los problemas se le pide que genere determinados scripts de MATLAB. En la versión final traslade el código de los scripts al LiveScript, ejecute el código para que se visualize los resultados y genere el archivo PDF.

Para los problemas del 1 al 5 escribe un Script llamado problemasCortos.m.

1. Variables escalares.

Defina las siguientes variables

a. $a = 10$

b. $b = 2.5 \times 10^{23}$

c. $c = 2 + 3i$ donde i es la raíz cuadrada de -1

d. $d = e^{j\,2\,\pi\ /3}$ donde j es la raíz cuadrada de -1 y e es el número de Euler

2. Variables vectoriales

Defina las siguientes variables

a. aVec = [3.14 15 9 26]

b. bVec = $\begin{bmatrix} 2.71 \\ 8 \\ 28 \\ 182 \end{bmatrix}$

c. cVec = [5 4.8 ... −4.8 −5] (todos los números de 5 a -5 en incrementos de 0.2)

d. dVec = $\begin{bmatrix} 10^0 & 10^{0.01} & \dots 10^{0.99} & 10^1 \end{bmatrix}$ (números espaciados logarítmicamente entre 1 y 10, use **logspace**, ¡asegurese de que la longitud es la correcta!)

e. eVec = Hello (eVec es una cadena, que es un vector de carateres)

3. Variables de matriz

Defina las siguientes variables

a. aMat = $\begin{bmatrix} 2 & \cdots & 2 \\ \vdots & \ddots & \vdots \\ 2 & \cdots & 2 \end{bmatrix}$ una matriz de 9×9 llena de 2s (usa **ones** o **zeros**)

b. bMat = $\begin{bmatrix} 1 & 0 & \cdots & & 0 \\ 0 & \ddots & 0 & \ddots & \\ \vdots & 0 & 5 & 0 & \vdots \\ & \ddots & 0 & \ddots & 0 \\ 0 & & \cdots & 0 & 1 \end{bmatrix}$ una matrix de 9×9 llena de ceros, pero con los valores [1 2 3 4 5 4 3 2 1] en la diagonal principal.

c. cMat = $\begin{bmatrix} 1 & 11 & \cdots & 91 \\ 2 & 12 & \ddots & 92 \\ \vdots & \vdots & \ddots & \vdots \\ 10 & 20 & \cdots & 100 \end{bmatrix}$ una matriz de 10×10en la que el vector recorre por columnas (utilice **reshape**)

d. dMat = $\begin{bmatrix} \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} \\ \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} \\ \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} \end{bmatrix}$ una matriz NaN de 3×4 (usar **nan**)

e. eMat = $\begin{bmatrix} 13 & -1 & 5 \\ -22 & 10 & -87 \end{bmatrix}$

f. Haz que fMat sea una matriz de 5x3 de enteros aleatorios con valores en el rango de -3 a 3 (Primero usa **rand** y **floor** o **ceil.** Luego sólo usa **randi**)

4. Ecuaciones escalares

Utilizando las variables creadas en 1, calcular x, y y z

a. $x = \dfrac{1}{1 + e^{(-(a-15)\ /\ 6)}}$

b. $y = \left(\sqrt{a} + \sqrt[21]{b}\right)^{\pi}$, recuerda que $\sqrt[g]{h} = h^{1/g}$,y utiliza **sqrt.** También puedes usar **nthroot** (consulta la ayuda de MATLAB para entender la diferencia entre **nthroot** y una potencia fraccionaria)

c. $z = \frac{\log(\Re[(c+d)(c-d)]\sin(a\pi/3))}{c\overline{c}}$donde $\Re$ indica la parte real del número complejo entre paréntesis, $\overline{c}$ es el conjugado del complejo c y log es el logaritmo natural (use **real, conj, log).**

5. Funciones comunes e indexación

a. Haz que cSum sea la suma por columnas de cMat. La respuesta debe ser un vector fila (utilice **sum**)

b. Haz que eMean sea la media de las filas de eMat. La respuesta deber ser una columna (utilice **mean**)

c. Reemplaza la fila superior de eMatpor 1 1 1.

d. Haz que cSub sea la submatriz de cMat que sólo contiene las filas 2 a 9 y las columnas 2 a 9.

e. Generar el vector lin $=$ $[1\ 2 \cdots 20]$ (los enteros de 1 a 20), y luego hacer cada elemento par en el negativo para obtener lin $=$ $[1\ -2\ 3\ -4 \cdots -20]$

f. Haz de r un vector de 1x5 usando **rand.** Encuentra los elementos que tienen valores <0.5 y pon esos valores a 0 (usa **find**).

6. Manipulación de variables.

Escribe un script para leer algunas calificaciones. Para hacer esto, necesitarás emplear el archivo `classGrades.mat`.

a. Abra un script y nómbrelo `calculateGrades.m`. Escriba todos los comandos en este script.

b. Cargue el archivo`classGrades` utilizando **load**.Este archivo contiene una única variable llamada *namesAndGrades*

c. Para ver cómo está estructurado *namesAndGrades*, muestre las primeras 5 filas en su pantalla. La primera columna contiene los "nombres" de los estudiantes, son sólo los números enteros del 1 al 15. Las 7 columnas restantes contienen la puntuación de cada estudiante (en una escala de 0 a 5) en cada una de las 7 tareas. También hay algunos NaN que indican que un estudiante en particular estuvo ausente ese día y no hizo la tarea.

d. Sólo nos interesan las calificaciones, así que extraiga la submatriz que contiene todas las filas pero sólo las columnas 2 a 8 y llame a esta matriz *grades* (para que esto funcione en cualquier tamaño de matriz, no codifique el 8, sino que utilice **end** o **size(namesAndGrades,2)**).

e. Calcule la puntuación media de cada tarea. El resultado debe ser un vector de 1x7 que contiene la nota media de cada tarea.

   i. En primer lugar, hágalo utilizando **mean**. Muestre las calificaciones medias calculadas de esta manera. Observe que los NaNs que había en la matriz de calificaciones hacen que algunas de las calificaciones medias sean también NaN.

   ii. Para solucionar este problema, revise la ayuda de la función **mean** para calcular la media usando sólo los números que nos son NaN. Esto significa que los estudiantes ausentes no se consideran en el cálculo, que es lo que queremos. Nombra este vector de media *meanGrades* y muéstralo en la pantalla para verificar que no tiene NaNs

f. Normalice cada tarea para que la nota media sea 3.5 (esto es un B- en nuestra escala de 5 puntos). Deberá dividir cada columna de *grades* por el elemento correcto de *meanGrades* .

   i. Hacer una matriz llamada *meanMatrix* tal que tenga el mismo tamaño que *grades*, y cada fila tiene los valores *meanGrades*. Hazlo tomando el producto exterior de un vector 15x1 de unos y el vector meanGrades, que es una fila (usa **ones**, *****). Visualice *meanMatrix* para verificar que tiene el aspecto deseado.

   ii. Para calcular las calificaciones curvas,haga lo siguiente: *curvedGrades* = 3.5 (*graded / meanMatrix)*. Ten en cuenta que quieres hacer la división por elementos.

   iii. Calcula y muestra la media de *curvedGrades* para verificar que todos son 3.5 (promedio sin **nan**).

   iv. Como hemos dividido por la media y multiplicado por 3.5, es posible que algunas calificaciones que inicialmente estaban cerca de 5 sean ahora mayores que 5. Para arreglar esto, encuentra todos los elementos en *curvedGrades* que son mayores que 5 y establecerlos en 5. Utilicela indexación lógica y NO utilice **find**.

g. Calcular la nota total de cada alumno y asignarlas calificaciones con letras

   i. Para calcular el vector *totalGrade*, que contendrá la calificación numérica de cada estudiante, se debe tomar la media de *curvedGrades* en todas las columnas (use **mean** sin NaN, vea la ayuda para especificar la dimensión). Además, sólo queremos terminar con números del 1 al 5, así que calcule el techo del vector *totalGrade* (use **ceil**).

   ii. Haz una cadena llamada *letras* que contenga las calificaciones de las letras en orden creciente: FDCBA

   iii. Haga el vector final de calificaciones de letras *letterGrades* utilizando *totalGrade* (que después de la operación ceil sólo debe contener valores entre 1 y 5) para indexar en *letters* .

   iv. Por último, muestre lo siguiente utilizando **disp**: Grades:*letterGrades*

h. Ejecute el script para verificar que funciona. Debería obtener una salida como la siguiente

```
>> calculateGrades
ans =
    1.0000    2.5064    3.6529    2.4617    3.3022    2.5189    0.0963    4.6502
    2.0000    2.1586    3.2324    3.4737    0.2378    2.4480    0.4194    1.9951
    3.0000    4.9878       NaN    4.8637    1.7439    4.3852    4.8740    0.2370
    4.0000    4.0580    1.9914    1.6388    2.2567    1.7657    3.2567    1.7119
    5.0000    2.4283    3.7491    4.1890       NaN    2.2472    1.1562    3.6798
meanGrades =
       NaN       NaN    2.8361       NaN    2.8540    1.6481       NaN
meanGrades =
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
meanMatrix =
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
    2.9690    2.9445    2.8361    2.4879    2.8540    1.6481    2.5677
ans =
    3.5000    3.5000    3.5000    3.5000    3.5000    3.5000    3.5000
Grades: BCBBBACCBCCCCAB
```

7. Gráficas

Abre un nuevo script y guárdalo como leyMoore.m y resuelva el problema siguiente:

De acuerdo con la ley de Moore (una observación hecha en 1965 por Gordon Moore, cofundador de Intel Corporation), el número de transistores que encajaría por pulgada cuadrada en un circuito integrado semiconductor se du- plica aproximadamente cada 18 meses. El año 2005 fue el 40 aniversario de la ley. Durante los últimos 40 años, su proyección se ha satisfecho de manera consistente. En 1965, la entonces tecnología de avanzada permitía 30 transistores por pulgada cuadra- da. La ley de Moore dice que la densidad de transistores se puede predecir mediante $d\left(t\right) = 30\ \left(2^{\frac{t}{1.5}}\right)$, donde t se mide en años.

a. Sea $t = 0$ la representación del año 1965 y $t = 45$ la representación de 2010. Use este modelo para calcular el número predicho de transistores por pulgada cuadrada para los 45 años desde 1965 hasta 2010. Sea t el aumento en incrementos de 1.5 años. Muestre los resultados en una tabla con 2 columnas, una para el año y otra para el número de transistores.

b. Con la característica subplot, grafique los datos en una gráfica lineal

x-y, una gráfica x semilog, una gráfica y semilog y una gráfica log-log. Asegúrese de poner título y etiqueta a los ejes.

c. Guarde la figura como archivo 'ley_moore.pdf'. Pero primero use el comando 'orient tall' para decirle a MATLAB que la figura llene la página impresa. Enumera los comandos que has utilizado a continuación.