

**Proyecto Final**

*Brian de Jesus Herberth Guerrero*

*A00398663*

# 1 Del siguiente tutorial, capturar y documentar en latex los ejercicios

<https://www.gnu.org/software/octave/doc/v4.0.0/Simple-Examples.html>

## 1.1 Elementary calculations

Octave can easily be used for basic numerical calculations. Octave knows about arithmetic operations(+,-,\*,/), exponentiation, natural logarithms / exponents (log, exp), and the trigonometric functions (sin, cos, ...). Moreover, Octave calculations work on real or imaginary numbers (i,j). In addition, some mathematical constants such as the base of the natural logarithm (e) and the ratio of a circle's circumference to its diameter (pi) are pre-defined.

For example, to verify Euler's Identity,

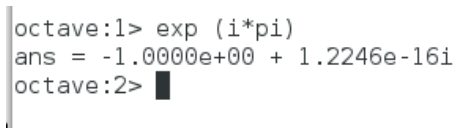
$$i * pi \tag{1}$$

$$e = -1 \tag{2}$$

type the following which will evaluate to -1 within the tolerance of the calculation.

$$octave : 1 > \exp(i * pi) \tag{3}$$

Here is an image which shows the program execution



```
octave:1> exp (i*pi)
ans = -1.0000e+00 + 1.2246e-16i
octave:2> █
```

Figure 1: Elementary calculations

## 1.2 Creating a matrix

Vectors and matrices are the basic building blocks for numerical analysis. To create a new matrix and store it in a variable so that you can refer to it later, type the command

$$octave : 1 > A = [1, 1, 2; 3, 5, 8; 13, 21, 34] \tag{4}$$

Octave will respond by printing the matrix in neatly aligned columns. Octave uses a comma or space to separate entries in a row, and a semicolon

or carriage return to separate one row from the next. Ending a command with a semicolon tells Octave not to print the result of the command. For example,

$$\text{octave : 2} > B = \text{rand}(3,2); \quad (5)$$

will create a 3 row, 2 column matrix with each element set to a random value between zero and one.

To display the value of a variable, simply type the name of the variable at the prompt. For example, to display the value stored in the matrix B, type the command

$$\text{octave : 3} > B \quad (6)$$

Here is an image which shows the program execution

```
octave:1> A = [1,1,2;3,5,8;13,21,34]
A =
     1     1     2
     3     5     8
    13    21    34

octave:2> B = rand (3,2);
octave:3> B
B =
    0.249738    0.118654
    0.850347    0.278361
    0.032423    0.214474

octave:4> █
```

Figure 2: Creating a Matrix

### 1.3 Matrix arithmetic

Octave has a convenient operator notation for performing matrix arithmetic. For example, to multiply the matrix A by a scalar value, type the command

$$\text{octave : 4} > 2 * A \quad (7)$$

To multiply the two matrices A and B, type the command

$$\text{octave : 5} > A * B \quad (8)$$

and to form the matrix product transpose (A) \* A, type the command

$$\text{octave : 6} > A' * A \quad (9)$$

Here is an image which shows the program execution

```
octave:4> 2*A
ans =
      2      2      4
      6     10     16
     26     42     68

octave:5> A*B
ans =
      2.7448      2.3105
     11.3488      9.0185
     48.1398     38.3844

octave:6> A'*A
ans =
     179     289     468
     289     467     756
     468     756    1224
```

Figure 3: Matrix arithmetic

## 1.4 Solving systems of linear equations

Systems of linear equations are ubiquitous in numerical analysis. To solve the set of linear equations  $Ax = b$ , use the left division operator, ' $\backslash$ ':

$$x = A \backslash b \quad (10)$$

This is conceptually equivalent to  $\text{inv}(A) * b$ , but avoids computing the inverse of a matrix directly.

If the coefficient matrix is singular, Octave will print a warning message and compute a minimum norm solution.

A simple example comes from chemistry and the need to obtain balanced chemical equations. Consider the burning of hydrogen and oxygen to produce water.



The equation above is not accurate. The Law of Conservation of Mass requires that the number of molecules of each type balance on the left- and right-hand sides of the equation. Writing the variable overall reaction with individual equations for hydrogen and oxygen one finds:

$$x_1 * H_2 + x_2 * O_2 \rightarrow H_2O \quad (12)$$

$$H : 2 * x_1 + 0 * x_2 \rightarrow 2 \quad (13)$$

$$O : 0 * x_1 + 2 * x_2 \rightarrow 1 \quad (14)$$

The solution in Octave is found in just three steps.

$$\text{octave : 1} > A = [2, 0; 0, 2]; \quad (15)$$

$$\text{octave : 2} > b = [2; 1]; \quad (16)$$

$$\text{octave : 3} > x = A \backslash b \quad (17)$$

Here is an image which shows the program execution

```
octave:1> A=[2,0;0,2];
octave:2> b=[2;1];
octave:3> x=A\b
x =
    1.00000
    0.50000
```

Figure 4: Solving systems of linear equations

## 1.5 Integrating differential equations

Octave has built-in functions for solving nonlinear differential equations of the form

$$dx/dt = f(x, t) \quad (18)$$

with the initial condition

$$x(t = t_0) = x_0 \quad (19)$$

For Octave to integrate equations of this form, you must first provide a definition of the function  $f(x,t)$ . This is straightforward, and may be accomplished by entering the function body directly on the command line. For example, the following commands define the right-hand side function for an interesting pair of nonlinear differential equations. Note that while you are entering a function, Octave responds with a different prompt, to indicate that it is waiting for you to complete your input.

```
octave:1>function xdot = f (x, t)
>
>r = 0.25;
>k = 1.4;
>a = 1.5;
>b = 0.16;
>c = 0.9;
>d = 0.8;
>
>xdot(1) = r*x(1)*(1 - x(1)/k) - a*x(1)*x(2)/(1 + b*x(1));
>xdot(2) = c*a*x(1)*x(2)/(1 + b*x(1)) - d*x(2);
>
>endfunction
```

Given the initial condition

$$octave : 2 > x_0 = [1; 2]; \quad (20)$$

and the set of output times as a column vector (note that the first output time corresponds to the initial condition given above)

$$octave : 3 > t = linspace(0, 50, 200)'; \quad (21)$$

it is easy to integrate the set of differential equations:

$$octave : 4 > x = lsode('f', x_0, t); \quad (22)$$

The function `lsode` uses the Livermore Solver for Ordinary Differential Equations, described in A. C. Hindmarsh, ODEPACK, a Systematized Collection of ODE Solvers, in: Scientific Computing, R. S. Stepleman et al. (Eds.), North-Holland, Amsterdam, 1983, pages 55–64.

Here is an image which shows the program execution

```
octave:1> function xdot = f(x,t)
>
> r = 0.25;
> k = 1.4;
> a = 1.5;
> b = 0.16;
> c = 0.9;
> d = 0.8;
>
> xdot(1) = r*x(1)*(1-x(1)/k) - a*x(1)*x(2)/(1+b*x(1));
> xdot(2) = c*a*x(1)*x(2)/(1+b*x(1)) - d*x(2);
>
> endfunction
octave:2> x0 = [1;2];
octave:3> t = linspace (0,50,200)';
octave:4> x = lsode("f", x0, t);
octave:5> █
```

Figure 5: Integrating differential equations

## 1.6 Producing graphical output

To display the solution of the previous example graphically, use the command

$$\text{octave : 1} > \text{plot}(t, x) \quad (23)$$

If you are using a graphical user interface, Octave will automatically create a separate window to display the plot.

To save a plot once it has been displayed on the screen, use the print command. For example,

$$\text{print} - \text{dpdf} \text{foo.pdf} \quad (24)$$

will create a file called foo.pdf that contains a rendering of the current plot in Portable Document Format. The command

$$\text{helpprint} \quad (25)$$

explains more options for the print command and provides a list of additional output file formats.

Here is an image which shows the program execution

```
octave:3> t = linspace (0, 50, 200)';
octave:4> x = lsode ("f", x0, t);
octave:5> plot (t, x)
octave:6> print -dpdf foo.pdf
```

Figure 6: Creating plot

Here is an image of the Plot

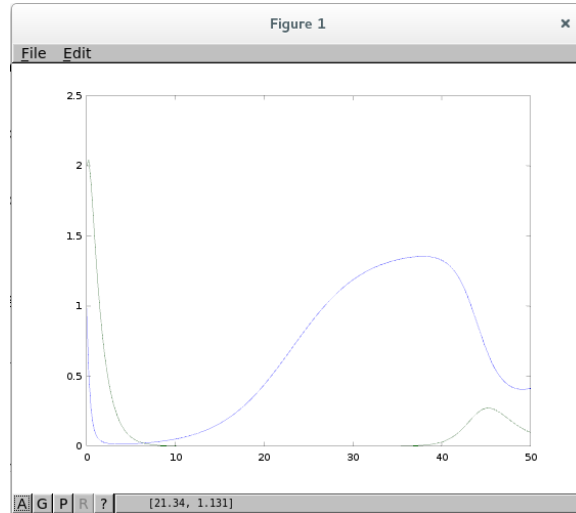


Figure 7: Plot

## 2 Proyecto de clase

Resolver nuevamente el proyecto de clase, pero cambiar el conjunto de datos del archivo .txt, con al menos 200 renglones de algun tema en especial de su interes (salud, sismos, peliculas, musica, videos).

Well, I searched for data about earthquakes, this data is about the earthquakes in the last two weeks

First, we'll want to load the data:

$$data = load('txt/sismos.txt'); \quad (26)$$

Next, let's define x and y. The x vector is for the independent variable (the number of earthquakes), and the y vector is for the dependent variable (magnitude of earthquakes).

$$x = data(:, 1); \quad (27)$$

$$y = data(:, 2); \quad (28)$$

Let's create a function to plot the data

$$function plotData(x, y) \quad (29)$$

$$plot(x, y, 'rx', 'MarkerSize', 2); \quad (30)$$



$$end \quad (31)$$

$$plotData(x, y); \quad (32)$$

$$xlabel('Numeroofearthquakes'); \quad (33)$$

$$ylabel('magnitudeofearthquakes'); \quad (34)$$

$$fprintf('Programpaused.Pressentertocontinue.');$$

$$pause; \quad (36)$$

And the graph of the data looks like this:

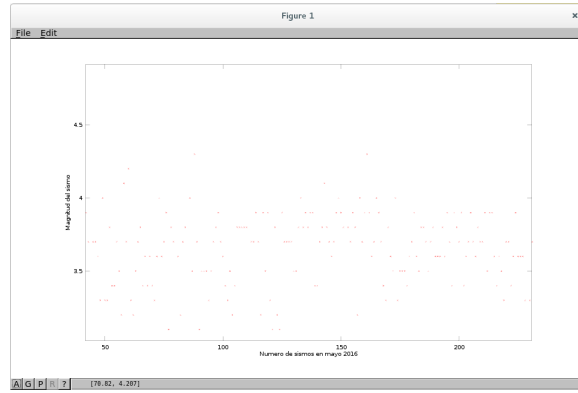


Figure 8: Plot of earthquakes

Add a column of all ones to our x column

$$m = length(x); \quad (37)$$

$$X = [ones(m, 1)x]; \quad (38)$$

Now use the normal equation to calculate theta. Basically, we are minimizing the sum of the squared errors between our predicted equation and the actual y values.

$$Theta = (Xexp(T)X)exp(negativo(1))Xexp(T)y \quad (39)$$

Putting that into Octave:

$$Theta = (Xexp(T)X)exp(negativo(1))Xexp(T)y \quad (40)$$

$$theta = (pinv(X' * X)) * X' * y \quad (41)$$

$$theta = 3.6026e + 00 \quad (42)$$

$$\theta = 2.5022e - 04 \quad (43)$$

Now, let's plot our fitted equation (prediction) on top of the training data, to see if our fitted equation makes sense.

$$\text{hold on;} \quad (44)$$

$$\text{plot}(X(:, 2), X * \theta, -) \quad (45)$$

$$\text{legend('Training data', 'Linear regression')} \quad (46)$$

$$\text{hold off} \quad (47)$$

And the graph of the data looks like this:

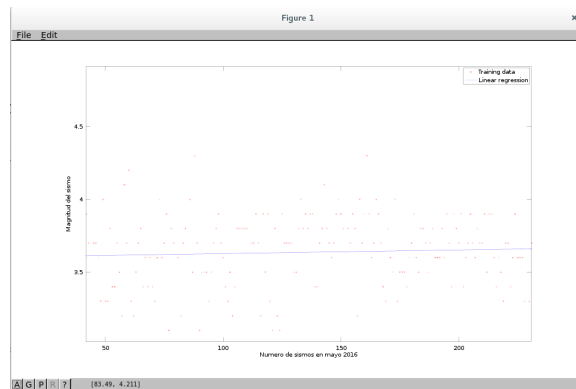


Figure 9: prediction

Here is an image which shows the program execution

```
octave:1> data = load('txt/sismos.txt');
octave:2> x = data(:,1);
octave:3> y = data(:,2);
octave:4> function plotData(x,y)
> plot(x,y,'rx','MarkerSize',2);
> end
octave:5> plotData(x,y);
octave:6> xlabel('Numero de sismos en mayo 2016');
octave:7> ylabel('Magnitud del sismo');
octave:8> fprintf('Programa pausado, presiona enter para continuar.\n');
Programa pausado, presiona enter para continuar.
octave:9> pause;
octave:10> m = length(x);
octave:11> X = [ones(m, 1) x];
octave:12> theta = (pinv(X'*X))*X'*y
theta =

    3.6026e+00
    2.5022e-04

octave:13> hold on;
octave:14> plot(X(:,2), X*theta, '-')
octave:15> legend('Training data', 'Linear regression')
octave:16> hold off
octave:17> █
```

Figure 10: Program execution

### 3 Lectura del artículo: Deep Learning

<http://www.xataka.com/robotica-e-ia/deep-learning-que-es-y-por-que-va-a-ser-una-tecnologia-clave-en-el-futuro-de-la-inteligencia-artificial>

Desde los años 50 del siglo pasado y hasta hace muy pocos años el terreno habitual de la Inteligencia Artificial avanzada era mayoritariamente el laboratorio de investigación y la ciencia ficción.

El gran impulso tecnológico al que solemos referirnos bajo el término Big Data ha revolucionado el entorno empresarial. Las organizaciones sometidas a la necesidad de la transformación digital se han convertido en criaturas sedientas de cantidades ingentes de datos; y por primera vez en la historia de la IA existe una demanda generalizada de sistemas con una inteligencia avanzada, equivalente a la de un humano, que sean capaces de procesar esos datos.

Una de las claves de la IA avanzada está en el aprendizaje. Es cada vez más habitual que les pidamos a las máquinas que aprendan por sí solas, por lo cual necesitamos que las máquinas sean capaces de auto-programarse, en otras palabras, queremos máquinas que aprendan de su propia experiencia. La disciplina del Aprendizaje Automático (Machine Learning)

En el enfoque Deep Learning se usan estructuras lógicas que se asemejan en mayor medida a la organización del sistema nervioso de los mamíferos, teniendo capas de unidades de proceso (neuronas artificiales) que se especializan en detectar determinadas características existentes en los objetos percibidos. La visión artificial es una de las áreas donde el Deep Learning proporciona una mejora considerable en comparación con algoritmos más tradicionales. Existen varios entornos y bibliotecas de código de Deep Learning que se ejecutan en las potentes GPUs modernas tipo CUDA, como por ejemplo NVIDIA cuDNN.

Los modelos computacionales de Deep Learning imitan estas características arquitecturales del sistema nervioso, permitiendo que dentro del sistema global haya redes de unidades de proceso que se especialicen en la detección de determinadas características ocultas en los datos. Este enfoque ha permitido mejores resultados en tareas de percepción computacional, si las comparamos con las redes monolíticas de neuronas artificiales.

El internet de las cosas (IoT) supone un gran avance en el reto de la adquisición de los datos, mientras que la computación cognitiva aporta la inteligencia necesaria para la extracción del conocimiento.

La Internet Cognitiva y Ubicua cuenta con un sistema sensorial que no para de extenderse gracias a los miles de millones de sensores conectados y desplegados por todas partes.

El auge de los sistemas M2M (Machine to Machine) en el marco del Inter-

net de las Cosas ha promovido un crecimiento exponencial del intercambio de datos entre las propias máquinas.

Con la llegada de tecnologías como el Deep Learning y la Computación Cognitiva nuestra forma de aprender, de relacionarnos y de entender el mundo va a cambiar también de forma radical.

**Dr. Raúl Arrabales Moreno**

## 4 Resumen

Realizar un resumen de 1 cuartilla del tema que consideren de mayor interés para su carrera profesional

### **La Ingeniería Mecatrónica, una carrera del futuro que se vive desde hoy**

Muchas personas piensan que el ingeniero mecatrónico pasa su vida en la creación de robots sin ninguna utilidad práctica inmediata. Pero la presencia de un profesional de este campo es mucho más importante de lo que crees.

Las labores de un ingeniero mecatronico incluyen:

- Implementar sistemas inteligentes, usando la electrónica moderna y la tecnología de los materiales
- Realizar la automatización y el control de procesos, integrando sistemas con sensores, actuadores e interfases hombre-máquina
- Armar y coordinar equipos de trabajo para realizar proyectos mecatrónicos
- Diseñar e implementar sistemas roboticos

Al igual que la industria aeroespacial, la mayoría de las cosas que necesitamos hoy en día dependen de equipos, máquinas y sistemas mecatrónicos. El ingeniero mecatrónico es responsable de reunir los conocimientos especializados en Mecánica, Electrónica e Informática con el fin de desarrollar la mejor solución a las actividades humanas.

Al igual que en cualquier otra ingeniería, es importante tener una buena capacidad de abstracción y la facilidad para trabajar en equipo. Debido a que es una profesión que interconecta otras disciplinas, la Ingeniería Mecatrónica requiere paciencia y comunicabilidad.

El papel de los profesionales se centra principalmente en la industria manufacturera. Esto incluye las plantas de energía, fábricas de acero, plantas de automóviles y medicamentos, entre otros. El mundo actual obliga a ser interdisciplinario, conocer la Ingeniería de control y automatización para que gracias a ellos los procesos sean óptimos.

La Ingeniería Mecatrónica integra los conocimientos, procedimientos y tecnologías provenientes de la ingeniería mecánica, electrónica, informática y eléctrica. Esta sinergia permite el análisis, diseño de productos y procesos de manufacturas automatizadas.

La UTP forma profesionales altamente calificados en la automatización de sistemas de producción industrial y de manufacturas, utilizando técnicas electrónicas e informáticas de última generación.

Es una carrera con gran futuro, puesto que la tendencia mundial es que las máquinas reemplacen el trabajo manual con lo cual los profesionales de esta carrera serán los más requeridos.