Programación en lenguaje MATLAB

Clase 2

Dr. Ing. Rodrigo Gonzalez
rodralez@frm.utn.edu.ar

Universidad Tecnológica Nacional,
Facultad Regional Mendoza.

Resumen

1. Operaciones con matrices
   - Suma y resta
   - Multiplicación
   - Sistema de ecuaciones
   - Operaciones elemento a elemento (element-wise)
2. Operadores relacionales
   - Direccionamiento con matrices lógicas
3. Operadores lógicos
4. Control del flujo
   - Introducción
   - Estructuras condicionales
   - Bucles
5. Vectorización

Suma

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{23} & a_{33} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{23} & b_{33} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \\ a_{31}+b_{31} & a_{32}+b_{23} & a_{33}+b_{33} \end{bmatrix}$$

- Condición: `size(A) == size(B)`
- Se cumplen reglas de suma de matrices: asociativa, conmutativa, elementro neutro, elemento opuesto. Por ej. `A+B = B+A`

Ejercicio 1

1. Genere 3 matrices cuadradas, con sus elementos iguales a 3, 5 y 7, respectivamente.
2. Sume las matrices.
3. Verifique la propiedad asociativa.

```
D1 = A + (B + C), D2 = (A + B) + C, D1 == D2
```

Resta

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{23} & a_{33} \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{23} & b_{33} \end{bmatrix} = \begin{bmatrix} a_{11} - b_{11} & a_{12} - b_{12} & a_{13} - b_{13} \\ a_{21} - b_{21} & a_{22} - b_{22} & a_{23} - b_{23} \\ a_{31} - b_{31} & a_{32} - b_{23} & a_{33} - b_{33} \end{bmatrix}$$

- Condición: `size(A) == size(B)`.
- Se cumplen reglas de suma de matrices.

Suma por escalar

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{23} & a_{33} \end{bmatrix} + b = \begin{bmatrix} a_{11}+b & a_{12}+b & a_{13}+b \\ a_{21}+b & a_{22}+b & a_{23}+b \\ a_{31}+b & a_{32}+b & a_{33}+b \end{bmatrix}$$

Ejercicio 2

1. `v = 1:6`
2. `v1 = v + 4`
3. `A = ones(3)*3 - 1`
4. `A + v`

Suma por escalar

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{23} & a_{33} \end{bmatrix} + b = \begin{bmatrix} a_{11}+b & a_{12}+b & a_{13}+b \\ a_{21}+b & a_{22}+b & a_{23}+b \\ a_{31}+b & a_{32}+b & a_{33}+b \end{bmatrix}$$

Ejercicio 2

1. `v = 1:6`
2. `v1 = v + 4`
3. `A = ones(3)*3 - 1`
4. `A + v`   % ERROR

Multiplicación de matrices

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{23} & a_{33} \end{bmatrix} * \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{23} & b_{33} \end{bmatrix}$$

$$=$$

$$\begin{bmatrix} a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} & a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32} & a_{11} * b_{13} + a_{12} * b_{23} + a_{13} * b_{33} \\ a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31} & a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32} & a_{21} * b_{13} + a_{22} * b_{23} + a_{23} * b_{33} \\ a_{31} * b_{11} + a_{23} * b_{21} + a_{33} * b_{31} & a_{31} * b_{12} + a_{32} * b_{22} + a_{33} * b_{32} & a_{31} * b_{13} + a_{32} * b_{23} + a_{33} * b_{33} \end{bmatrix}$$

- Condición: `[n,m]=size(A), [p,q]=size(B)` $\rightarrow$ `m == p`
- Se cumplen reglas del producto de matrices: asociativa, distributiva respecto a la suma (por derecha y por izquierda), elementro neutro. Por ej. `A * (C+B) = AC + AB`

Multiplicación por matriz por escalar

$$a * \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{23} & b_{33} \end{bmatrix} = \begin{bmatrix} a * b_{11} & a * b_{12} & a * b_{13} \\ a * b_{21} & a * b_{22} & a * b_{23} \\ a * b_{31} & a * b_{23} & a * b_{33} \end{bmatrix}$$

Ejercicio 3

1. Genere 2 matrices de 4x6, con elementos iguales a 4 y a 6, respectivamente.
2. Multiplique la 1era por la 2da transpuesta.
3. Multiplique la 1era por 2. ¿Qué observa?.
4. `v = 1:5, v * v     % ¿Qué observa?`
5. `v * v'     % ¿Qué observa?`
6. `v' * v     % ¿Qué observa?`

División por izquierda

La división entre matrices está ligada a la inversa de una matriz y a la resolución de un sistema de ecuaciones.

$$R * T = T * R = I \iff R = T^{-1},\ T = R^{-1}$$

Sistema de ecuaciones:

$$\begin{aligned} A * \boldsymbol{x} &= \boldsymbol{b} \\ A^{-1} * A \cdot \boldsymbol{x} &= A^{-1} * \boldsymbol{b} \\ \boldsymbol{x} &= A^{-1} * \boldsymbol{b} \end{aligned}$$

División por izquierda

En MATLAB un sistema de ecuaciones se puede resolver de varias maneras:

$$A_{\{N\times N\}} * \boldsymbol{x}_{\{N\times 1\}} = \boldsymbol{b}_{\{N\times 1\}}$$

Se calcula inversa luego se multiplica.

- `x = A^(-1) * b`
- `x = inv(A) * b`.

División por izquierda. Se calcula el resultado por eliminación gaussiana. Recomendado para resolución de sistemas de ecuaciones.

- `x = A\b` .

$$\boldsymbol{x}_{\{1\times N\}} * C_{\{N\times N\}} = \boldsymbol{d}_{\{1\times N\}}$$
$$\boldsymbol{x} * \underbrace{C * C^{-1}}_{I} = \boldsymbol{d} * C^{-1}$$
$$\boldsymbol{x} = \boldsymbol{d} * C^{-1}$$

Se calcula inversa luego se multiplica.

- `x = d * C^(-1)`
- `x = d * inv(C)`

División por derecha.

- `x = d/C`

Sistema de ecuaciones

Ejercicio 4

Resuelva el siguiente sistema de ecuaciones por los 5 métodos indicados. Verifique si los resultados coinciden.

$$4x - 2y + 6z = 8$$
$$2x + 8y + 2z = 4$$
$$6x + 10y + 3z = 0$$

1. `X1 = A^(-1) * b`
2. `X2 = inv(A) * b`
3. `X3 = A\b`
4. `C = A'; d = b';`
5. `X4 = d * inv(C)`
6. `X5 = d / C`

Ejercicio 4

Resuelva el siguiente sistema de ecuaciones por los 5 métodos indicados. Verifique si los resultados coinciden.

$$4x - 2y + 6z = 8$$
$$2x + 8y + 2z = 4$$
$$6x + 10y + 3z = 0$$

1. `X1 = A^(-1) * b`
2. `X2 = inv(A) * b`
3. `X3 = A\b`
4. `C = A'; d = b';`
5. `X4 = d * inv(C)`
6. `X5 = d / C`

Respuesta $X3 = [-1.8049, 0.2927, 2.6341]$

- `det(A)`
- `rank(A)`
- `rcond(A)`
- `eig(A)`
- ...

$$\begin{aligned} 4x - 2y + 6z &= 8 \\ 2x + 8y + 2z &= 4 \\ 6x + 10y + 3z &= 0 \end{aligned}$$

Ejercicio 5

1. `det(A), rank(A), rcond(A)     % ¿Qué observa`
2. `C = [1 2 3; 2 4 6; 2 12 43]`
3. `det(C),rank(C), rcond(C)     % ¿Qué observa?`

Operaciones elemento a elemento (element-wise)

- Multiplicación, `A .* B`
- División por izquierda, `A .\ B`
- División por derecha, `A ./ B`
- `B` puede ser matriz o escalar.
- Si `B` es matriz, `size(A) == size(B)`
- ¡No dejar espacio entre el punto y el operador!

Ejercicio 6

Genere 2 matrices cuadras de 5x5, con elementos iguales a 10 y 2, respectivamente.

1. `M1 = A .* B, D1 = A ./ B`
2. `M2 = A * B, D2 = A / B`
3. `M3 = A .* B(1,1), D3 = A ./ B(3,1)`
4. Compare los resultados.

Operaciones elemento a elemento (element-wise)

Ejercicio 6

Genere 2 matrices cuadras de 5x5, con elementos iguales a 10 y 2, respectivamente.

1. `M1 = A .* B, D1 = A ./ B`
2. `M2 = A * B, D2 = A / B % ERROR!`
3. `M3 = A .* B(1,1), D3 = A ./ B(3,1)`
4. Compare los resultados.

Operaciones elemento a elemento (element-wise)

- Potencia, `A .^ B`
- ¡No dejar espacio entre el punto y el operador!

Ejercicio 7

1. Genere 2 matrices 2x3, una con elementos iguales 2 y la otra con elementos de 1 a 6 por columnas.
2. `P1 = A .^ B`
3. `P2 = A .^ B(2)`
4. Compare los resultados.

Operaciones elemento a elemento (element-wise)

Ejercicio 8

Resuelva la siguiente expresión para todos los números enteros entre 0 y 20 inclusive:

$$y = \frac{x^3 + 5x}{4x^2 - 10}$$

Operaciones elemento a elemento (element-wise)

Ejercicio 8

Resuelva la siguiente expresión para todos los números enteros entre 0 y 20 inclusive:

$$y = \frac{x^3 + 5x}{4x^2 - 10}$$

Respuesta

```
x = 0:20;
y = (x.^3 + 5*x)./(4*x.^2 - 10)
```

Operadores relacionales

Operador relacional	Descripción
$<$	Menor que...
$>$	Mayor que...
$<=$	Menor o igual que...
$>=$	Mayor o igual que...
$==$	Igual a...
$\sim=$	Diferente a...

- No confundir $'=='$ con el operador asignación $'='$.
- No dejar espacio entre símbolos.
- Una expresión verdadera es igual a **variable lógica** con valor 1.
- Una expresión falsa es igual a **variable lógica** con valor 0.

Operadores relacionales

- Se pueden comparar matrices, siempre de igual dimensión.
- Las matrices se comparan elemento a elemento.
- El resultado es una matriz de igual dimensión con valores lógicos (0 o 1).

Ejercicio 9

```
1 » A = ones(5) ;
2 » B = ones(5) ;
3 » lo1 = (A == B) %¿Qué tipo de variable es lo1?
4 »                %Ver columna Class en Workspace
5 » B(:,5) = 0 ;
6 » lo2 = (A == B) %¿Qué observa?
```

Operadores relacionales

- Se pueden combinar operadores aritméticos y relacionales en una misma expresión.
- Variables lógicas y números pueden combinarse en operaciones matemáticas. El resultado es un número.
- Los operadores aritméticos se calculan antes que los relacionales.

Ejercicio 10

1. `» 3 + 4 < 16 /2;`
2. `» 3 + (4 < 16) /2;`
3. `» l1 = 7 == 8;`
4. `» sin(l1) %¿Qué observa?`
5. `» l1 * 5 %¿Qué observa?`

Direccionamiento con matrices lógicas

- Se puede usar una matriz con elementos lógicos para direccionar elementos de otra matriz.
- Matrices con 0s y 1s (double) no pueden direccionar otras matrices.

Ejercicio 11

```
1  » r = [8 12 9 6 5 5 13];
2  » idx = (r <= 10);
3  » t = r(idx)
4  » u = [1 0 1 1 1 1 0];
5  » t = r(u) %¿Qué observa?
```

Operadores lógicos

Operador lógico	Nombre
&	AND
|	OR
$\sim$	NOT

- `c = a & b` si `a` $\neq$ `0` **y** `b` $\neq$ `0` $\rightarrow$ `c = 1`; si no `c = 0`.
- `c = a | b` si `a = 0` **o** `b = 0` $\rightarrow$ `c = 0`; si no `c = 1`.
- `b =` $\sim$ `a` si `a` $\neq$ `0` $\rightarrow$ `b = 0`; si `a = 0` $\rightarrow$ `b = 1`.

- Se puede operar con matrices, siempre de igual dimensión.
- Las matrices operan elemento a elemento.
- El resultado es una matriz de igual dimensión con valores lógicos (0 o 1).
- Se pueden combinar operadores aritméticos y lógicos en una misma expresión.

Operadores lógicos

Ejercicio 12

1. » `l1 = logical(1) & logical(0);`
2. » `l2 = logical(1) & logical(1);`
3. » `l3 = 4 & 7;`
4. » `v = 1:50; i = (v > 20) & (v <= 30); v2 = v(i)`
5. » `i = (v < 20) | (v > 40); v3 = v(i)`
6. » `i = (v~=20) & (v>=15) & (v<=25); v4 = v(i)`
7. » `l6 = logical(1) | logical(1);`
8. » `l7 = 1.1 | 9.9;`
9. » `l8 = 0 | 0;`
10. » `b1 = ~9;`
11. » `b3 = ~logical(1);`
12. » `b4 = ~logical(0);`

Orden de cálculo completo

1. Paréntesis.
2. Exponente.
3. NOT lógico.
4. Multiplicación y división.
5. Suma y resta.
6. Operadores relacionales.
7. AND lógico.
8. OR lógico.
9. Operadores con misma prioridad, se evalúan de izquierda a derecha.

Funciones relacionadas

- Funciones lógicas: `and(A,B)`, `or(A,B)`, `not(A)`, `xor(A,B)`.
- `all(A)`, verdadero si todos los elementos de un vector son `~=0`.
- `any(A)`, verdadero si algún elemento de un vector es `~=0`.
- `find(A)`, devuelve vector con índices que son `~=0`.
- ...

Ejercicio 13

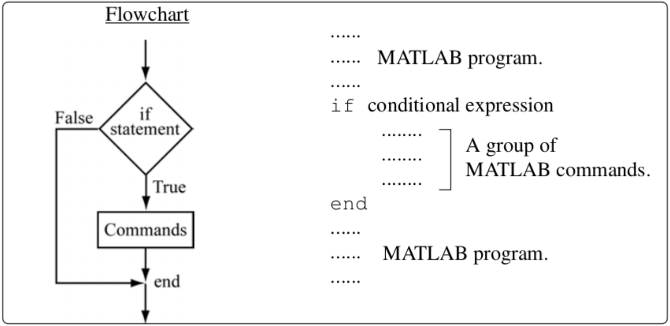
1. `» t = 101:110;`
2. `» idx1 = find( t > 104); v1 = t(idx1)`
3. `» idx2 = find( t > 104, 1, 'first' ); v2 = t(idx2)`
4. `» l1 = t > 104; v3 = t(l1);`
5. `» idx4 = find( t > 104 & t == 101) %¿Qué observa?`
6. `» v5 = t( t >= 104 & t < 108); %¿Qué observa?`

Control del flujo, introducción

- Hasta ahora hemos visto programas del tipo “receta de cocina”.
- Un programa puede tomar decisiones comparando variables.
- Tomar decisiones diferencia a una computadora de una calculadora :-)
- Estructuras de bifurcación:
  - `if`
  - `else`
  - `elseif`
  - `switch case`
- Bucles:
  - `for`
  - `while`
- Las decisiones se llevan a cabo evaluando expresiones con operadores relacionales y/o lógicos.

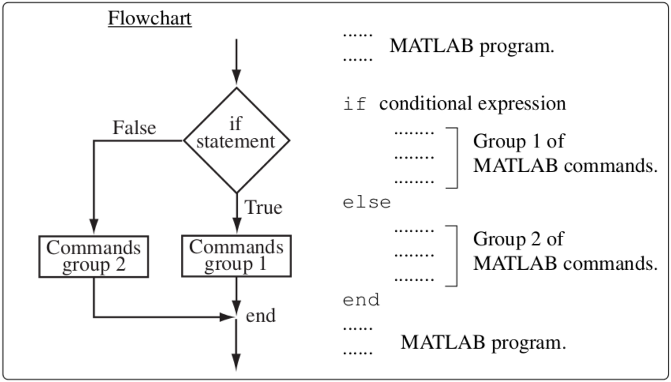
Estructura `if`

Se evalua una condición, si es verdadera se ejecuta un grupo de comandos.



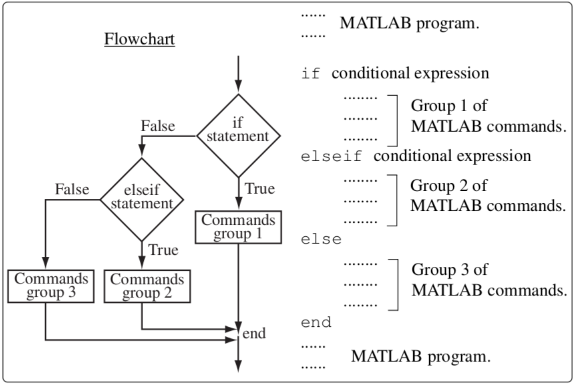
Estructura `if-else`

Se evalua una condición, si es verdadera se ejecuta un grupo de comandos (group 1), si no, se ejecuta otro grupo (group 2).



Estructura `if-elseif-else`

Se evaluan varias condiciones. Si la primera es verdadera, se ejecuta un grupo de comandos (group 1), si no, se evalua la segunda condición. Si la segunda es verdadera, se ejecuta un grupo de comandos (group 2), si no, se evalua la tercera, y así consecutivamente. Finalmente, si no se cumple ninguna condición, se ejecuta el último grupo (group 3).



Ejercicio 14

Un trabajador trabaja normalmente 40 horas por semana. Las horas extras se pagan un 50% más. Cree un programa que calcule el sueldo semanal si por hora se paga $ 50.

Ejercicio 14

Un trabajador trabaja normalmente 40 horas por semana. Las horas extras se pagan un 50% más. Cree un programa que calcule el sueldo semanal si por hora se paga $ 50.

Respuesta

```
horas = 40; valor = 50;
if(horas <= 40)
    sueldo = horas * valor;
else
    sueldo = 40 * valor + (horas - 40) * valor * 1.5;
end
```

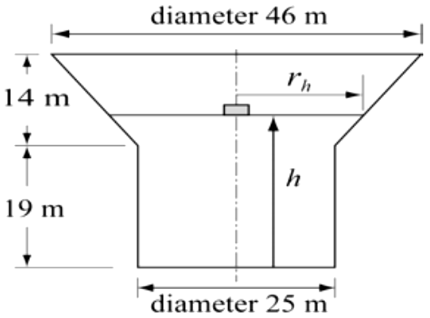
Ejercicio 15

Calcular el vólumen de agua en el siguiente recipiente, según la altura de la boya que se encuentra en el interior. Recuerde que

Volumen del cilindro: $V_{ci} = \pi \cdot 12.5^2 \cdot h$

Volumen del cono: $V_{co} = \pi \cdot 12.5^2 \cdot 19 + \frac{1}{3}\pi(h-19)(12.5^2 + 12.5 \cdot r_h + r_h^2)$

donde $r_h = 12.5 + \frac{10.5}{14}(h-19)$



Estructura `switch-case`

El valor de `switch expression`, se compara con los valores contiguos a cada `case`, estos pueden ser números o caracteres. Si coincide, se ejecuta el grupo de comandos del `case`. Si no coincide ninguno, se ejecuta el grupo de comandos de `otherwise` (opcional).

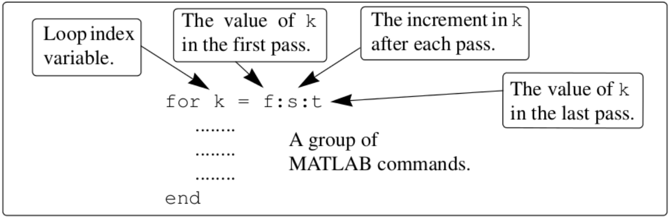
```
......
          MATLAB program.
......

switch  switch expression
     case  value1
     ........
                       Group 1 of commands.
     ........
     case  value2
     ........
                       Group 2 of commands.
     ........
     case  value3
     ........
                       Group 3 of commands.
     ........
     otherwise
     ........
                       Group 4 of commands.
     ........
end
......
          MATLAB program.
......
```

Ejercicio 16

Cree un programa que con datos de posición en metros y tiempo en segundos, y calcule una aceleración. Las unidades en la que entrega el resultado pueden ser m/s^2, cm/s^2 o g.

Bucle `for`

Los comandos se repiten desde `f`, hasta `t`, en pasos de `s`. Bucle con número fijo de iteraciones.



Ejercicio 17

La función $f(x) = e^x$ se expresa en series de Taylor como,

$$e^x = \sum_{n=0}^{N} \frac{x^n}{n!}$$

Cree un programa que haga este cálculo hasta el término número 10. Los argumentos de entrada son x y n. Compare los resultados obtenidos con la función estandar `exp(x)`.

Bucle `while`

Los comandos se repiten hasta que se cumplan la expresión condicional. Generalmente la variable que se evalua en dicha expresión es modificada dentro del bucle. Bucle con número variable de iteraciones.

```
while conditional expression
    ........
    ........        A group of
    ........        MATLAB commands.
end
```

Ejercicio 18

La función $f(x) = e^x$ se expresa en series de Taylor como,

$$e^x = \sum_{n=0}^{N} \frac{x^n}{n!}$$

Cree un programa que haga este cálculo hasta que el elemento *n* de la suma sea menor a un error configurable. Los argumentos de entrada son *x* y el error máximo permitido. Los argumentos de salida son e^x y la cantidad de iteraciones. Compare los resultados obtenidos con la función estandar `exp(x)`.

Vectorización

- MATLAB está optimizado para operar con matrices.
- Al vectorizar evitamos usar bucles (loops) para operar con matrices.

Ejemplo

Calcule el seno de 0 a 1 radian, en pasos de 0.01

Sin vectorización

```
t = 0:0.01:10;
y = zeros(1,100);
for i=1:100
 y(i) = sin(t(i));
end
```

Con vectorización

```
t = 0:.01:10;
y = sin(t);
```

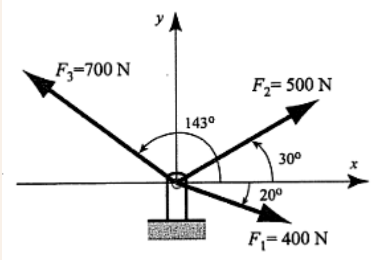
Vectorización

- Con vectorización, nuestro código ejecuta más rápido.
- Todas las funciones estándares de MATLAB soportan vectorización.
- `sin(V)`, `cos(V)`, `tan(V)`.
- `min(V)`, `max(V)`, `sort(V)`.
- `sum(V)`, `diff(V)`
- `repmat(A,M,N)`, crea una matriz a partir de replicar A.
- `reshape(V)`.

Recordar: MATLAB opera por columnas.

Ejercicio 19

Tres fuerzas son aplicadas a un émbolo. Determine la fuerza total aplicada.



En un sistema cartesiano,

$$\boldsymbol{F} = iF_x + jF_y = F(i\cos(\theta) + j\sin(\theta))$$

$$F = \sqrt{{F_x}^2 + {F_y}^2}, \quad \tan\theta = \frac{F_y}{F_x}$$

Ejercicio 19, respuesta

```
F = [400 500 700];
Th = [-20 30 143];
Fcos = F .* cosd(Th);
Fsin = F .* sind(Th);
Fcost = sum(Fcos);
Fsint = sum(Fsin);
Ft = sqrt(Fcost^2 + Fsint^2)
Tht = atand(Fsint/Fcost)
```

$F = 504.6192$ N y $\theta = 64.9453$ grados.