



## Cálculo Numérico (M107)

[ntripp@fcen.uncu.edu.ar](mailto:ntripp@fcen.uncu.edu.ar)

Aula Virtual

<http://fcen.uncuyo.edu.ar/calculo-numerico>

# Quien les habla...



Ing. Nicolás G. Tripp

<mailto:ntripp@fcen.uncu.edu.ar>

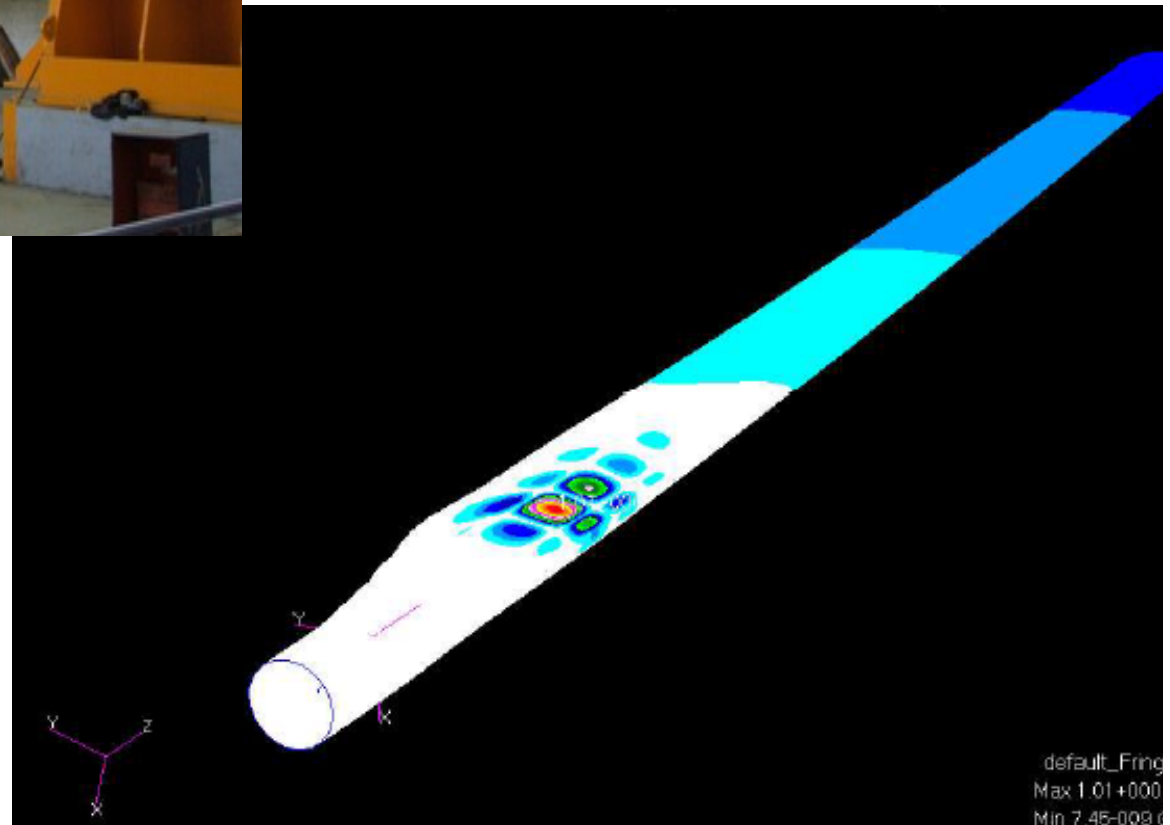
Docente de Métodos Numéricos.

Investigador con gran tolerancia a la frustración

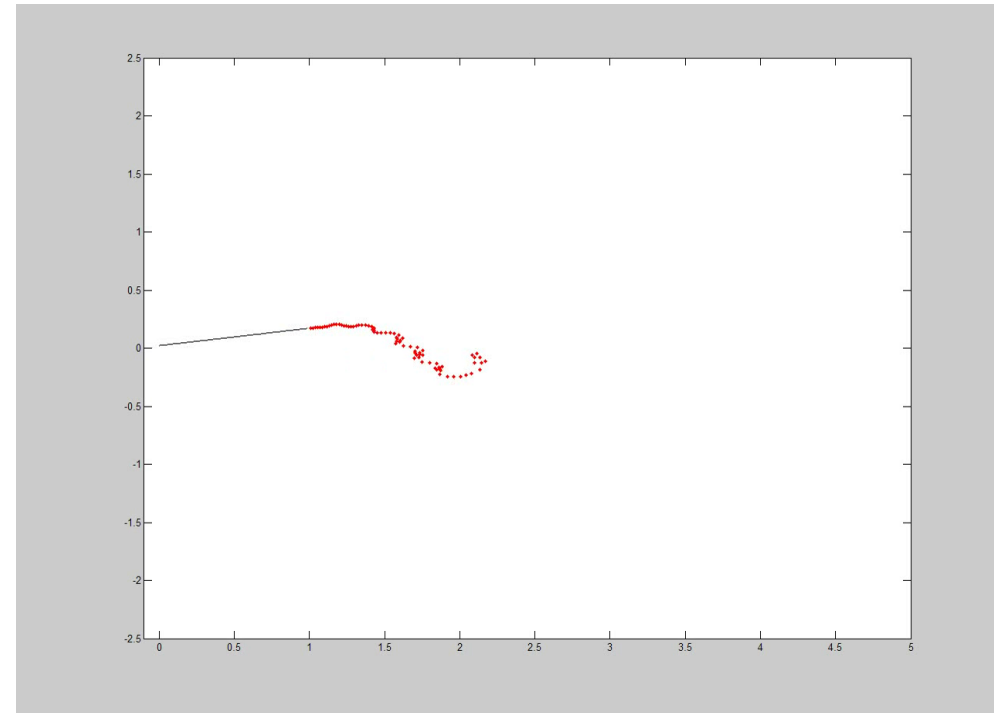
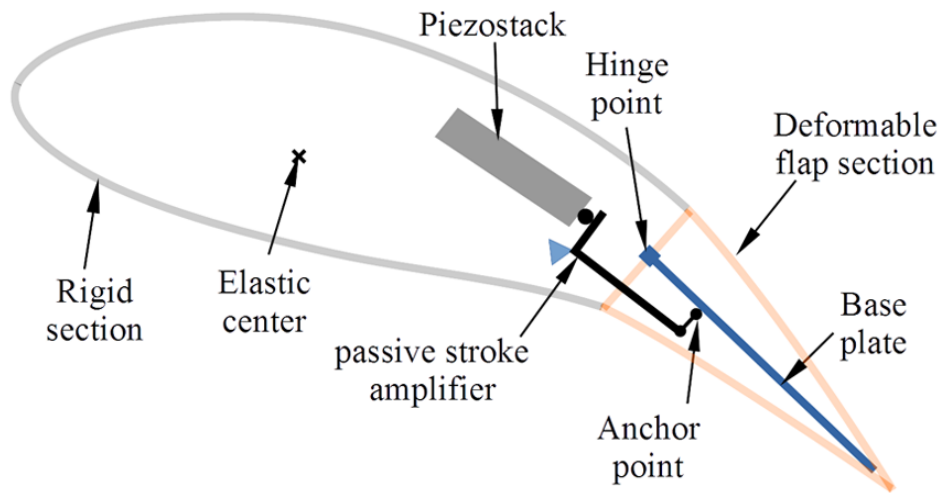
Fanático de los métodos numéricos y software

libre desde el 2006

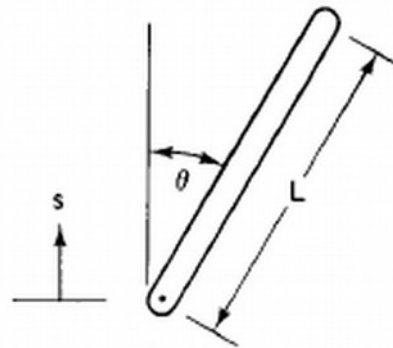
# Análisis estructural



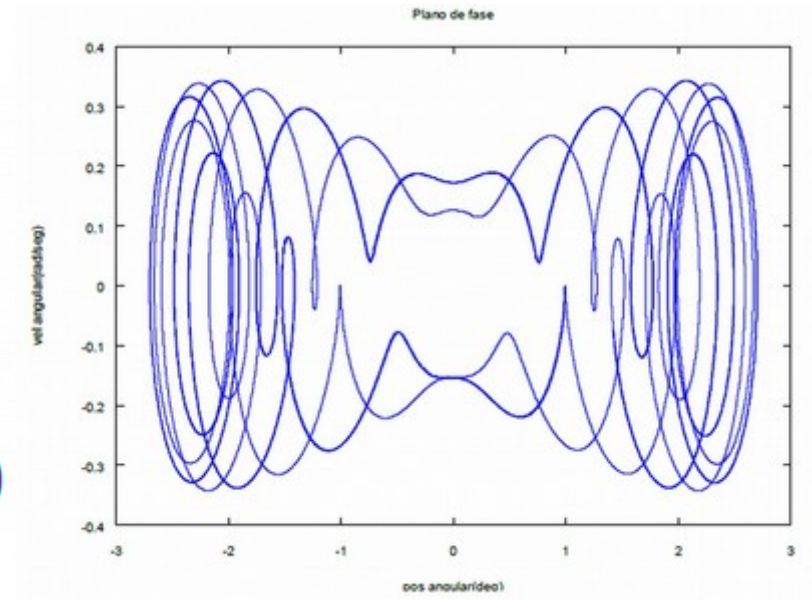
# Interacción fluido-estructura



## *Ecuaciones no lineales (Estabilidad del péndulo invertido)*

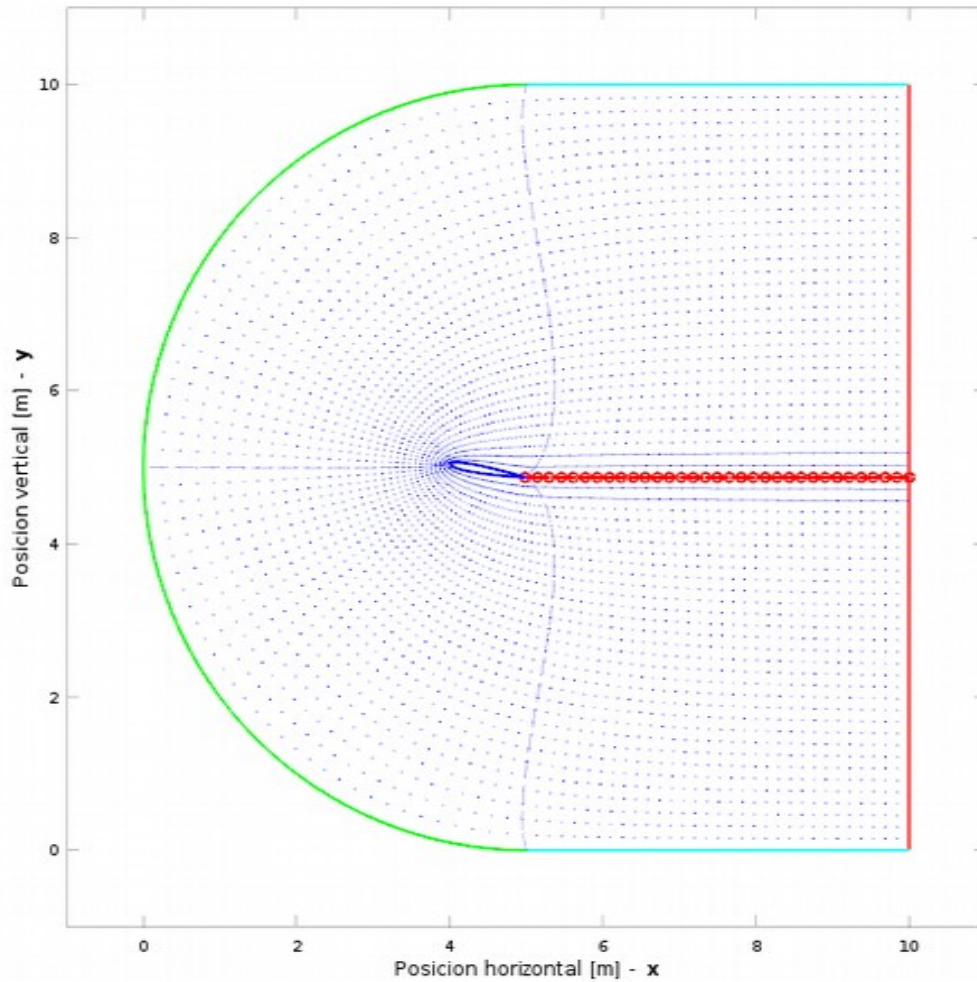


$$\ddot{\theta}(t) = \frac{3}{2L} (g - Aw^2 \sin wt) \sin \theta(t)$$

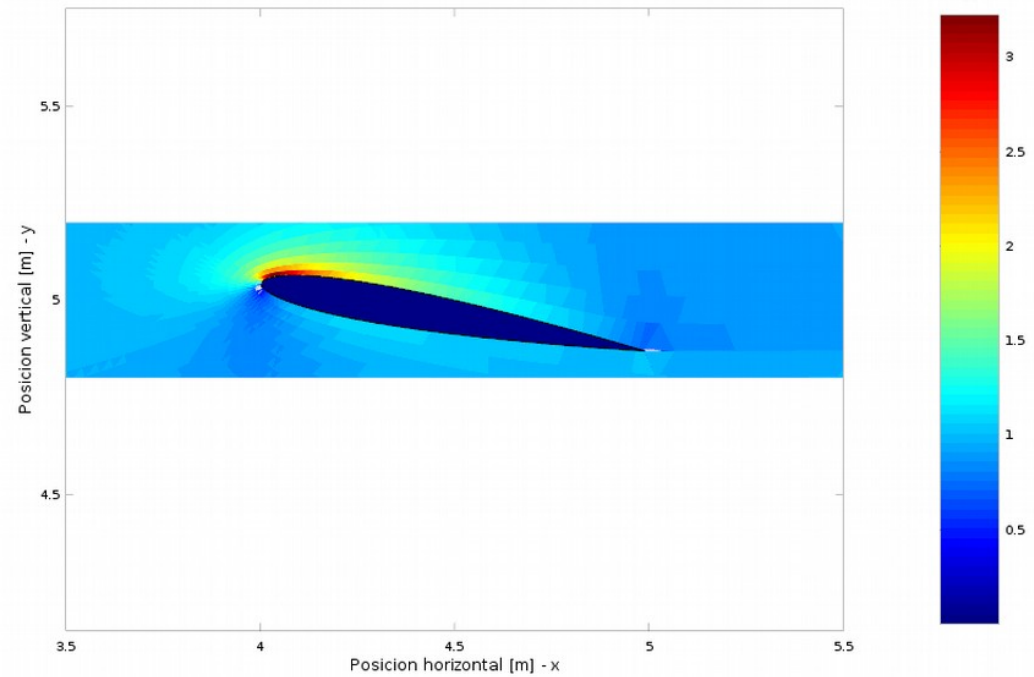


# Mecánica de fluidos

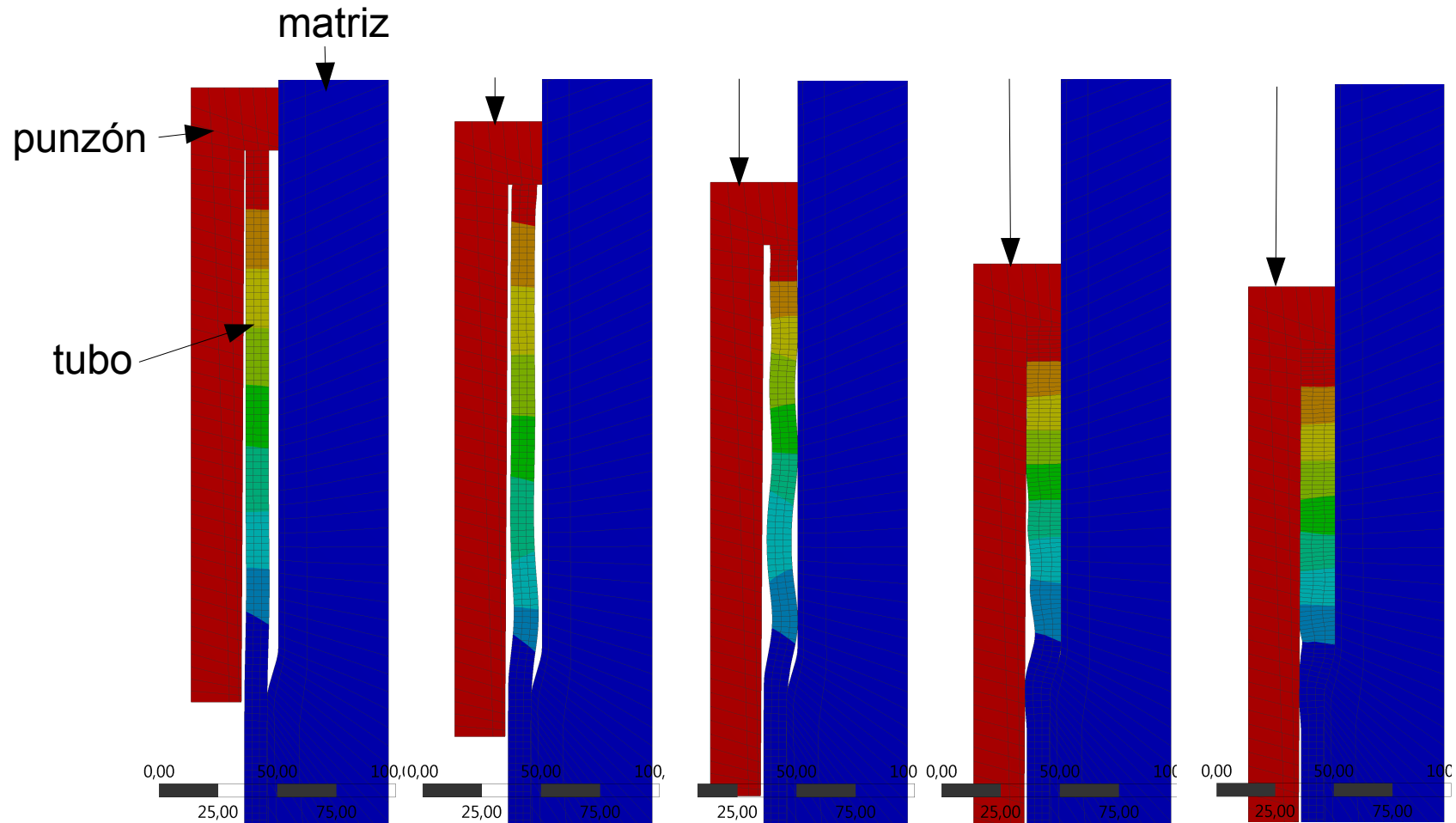
Grilla en dominio físico



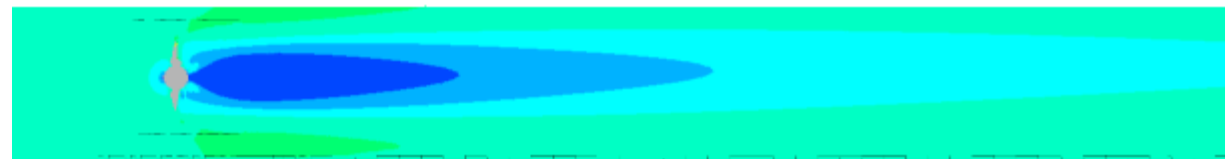
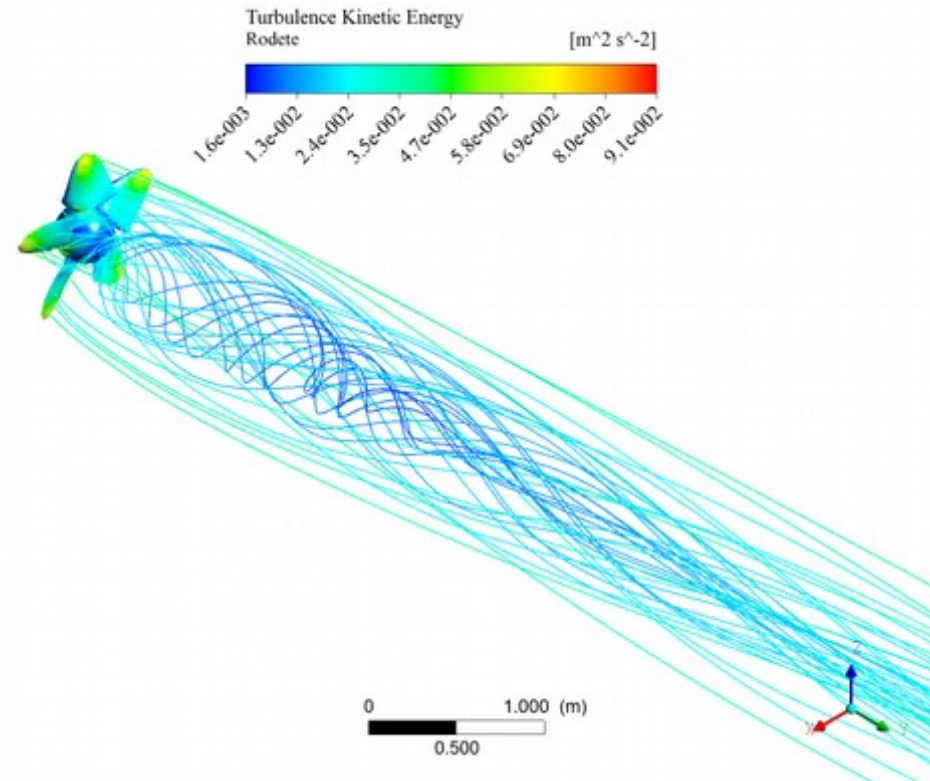
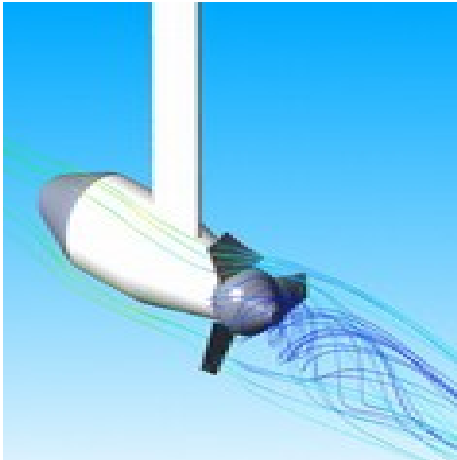
Campo de velocidades total -  $V_t$



# Grandes deformaciones en materiales

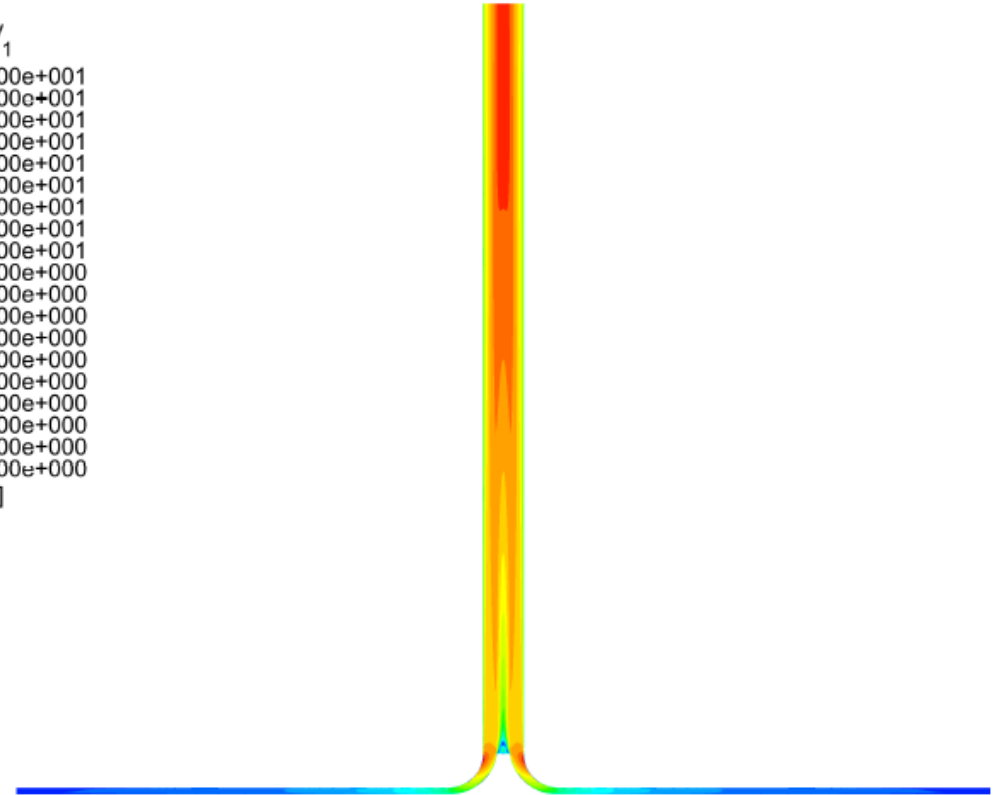
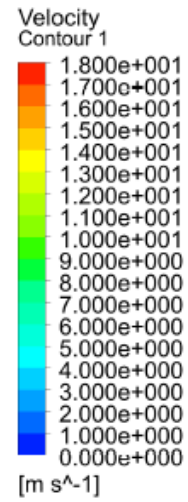
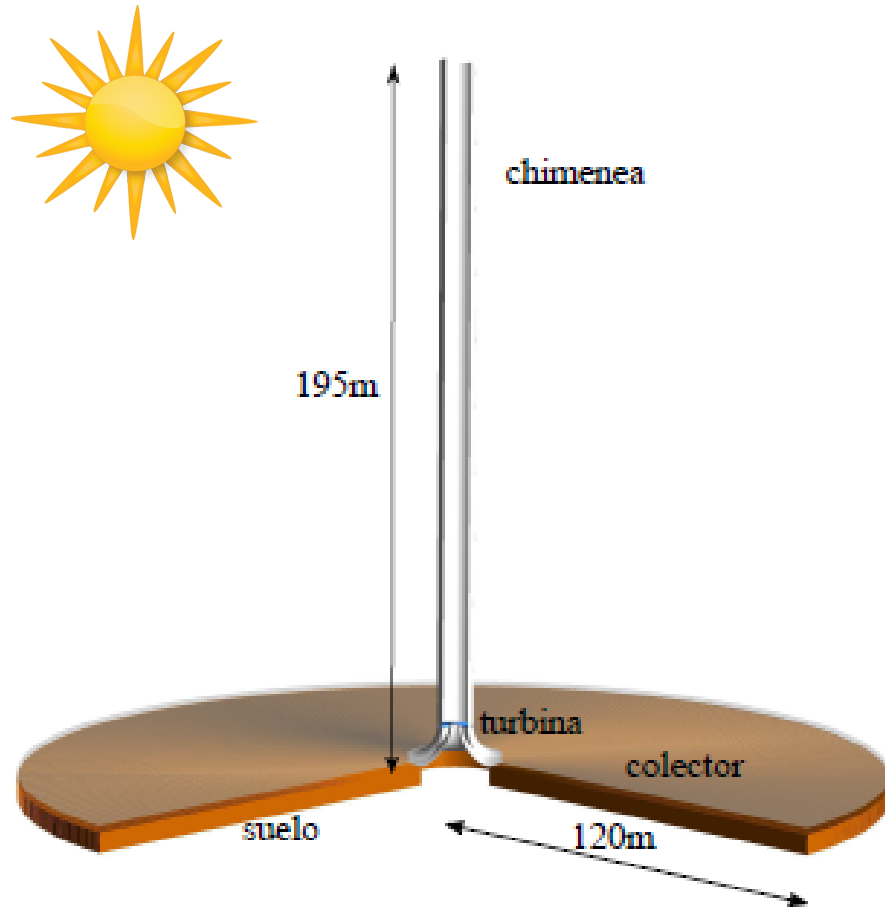


# Hidráulica





# Termodinámica



# Generalidades curso 2020

## Planificación de las clases

15 encuentros semanales, 8 unidades, y 3 exámenes parciales

Programa en <http://fcen.uncuyo.edu.ar/programas-de-las-materias>

Planificación en el aula virtual

## Forma de evaluación

- Cuestionarios teórico-prácticos en aula virtual (individual).
- Tres evaluaciones parciales (grupales).
- Examen final (individual).

## Bibliografía recomendada

- Mathews y Fink, “Métodos Numéricos con Matlab”, 1Ed, Prentice Hall, 2000.

# Introducción a la computación científica y al lenguaje Octave

## **Temario:**

- Fenómenos reales, modelos matemáticos y modelos numéricos.
- Aritmética de las computadoras.
- Fuentes de error en una solución numérica.
- Introducción a Octave

# Fenómenos reales, modelos y las soluciones numéricas

*Naturaleza*

*Modelos físicos*

- *Leyes de Newton*
- *Electromagnetismo de Maxwell*
- *Transferencia de calor*
- *Principios de Termodinámica*
- *Hipótesis del Continuo*

*Ecuaciones matemáticas*

- *Ecuación de Laplace*
- *Ecuación de la onda*
- *Ecuación del calor*

*Soluciones numéricas*

- *Diferencias Finitas*
- *Elementos Finitos*
- *Volúmenes Finitos*
- *Solver explícitos, implícitos*

Los **modelos físicos pueden albergar errores que lleven a conclusiones erradas** (ver el siguiente artículo [Newton, Einstein y Mercurio](#))

Los principios y leyes de la física se expresan en el lenguaje de la **Matemática**.

La complejidad de las ecuaciones y/o del dominio **pueden hacer imposible la obtención de una solución exacta**.

En estos casos se recurre a la **solución numérica aproximada**.

# El mundo digital

## Sistemas de numeración posicional

Se realiza una combinación lineal de los dígitos posibles con potencias de la base elegida.

Decimal o base-10

$$N = \sum_{i=-k}^{n-1} d_i 10^i, d_i \in [0, 1, 2, \dots, 9]$$

N es un número a representar,

n es la cantidad de dígitos a la izquierda de la coma

k es la cantidad de decimales

Ejemplo:  $193 \rightarrow n=3, k=0, d[193] \rightarrow 193 = \sum_{i=0}^2 d_i 10^i = 1 * 10^2 + 9 * 10^1 + 3 * 10^0$

Binario o base-2

Ejemplo:  $193 = \sum_{i=0}^7 d_i 2^i = 1 * 128 + 1 * 64 + 0 * 32 + 0 * 16 + 0 * 8 + 0 * 4 + 0 * 2 + 1 * 1$

$$193_{10} = 11000001_2$$

[Convertor binario-decimal](#)

# El mundo digital

## Representación en coma flotante normalizada

La computadora almacena una aproximación binaria de los números

$$(-1)^s \times C \times b^q$$

s: bit de signo, puede ser 0 o 1

C: mantisa, es menor a uno y se normaliza para que no tenga cero después de la coma.

b: base, en la computación actual se usa 2.

q: exponente, número natural

Ejemplo:

$$156,78 = (0) \times (00111001100011110101110) \times 2^{(10000110)}$$

[Conversor a coma flotante normalizada](#)

# Hardware

## Representación en la computadora

La computadora trabaja con **sistema binario**. Cada registro de la memoria almacena “bits” que pueden tomar el valor 0 o el 1 (apagado - encendido).

Los números se almacenan según la cantidad de bits disponibles por el hardware de la computadora.

El **estándar IEEE-754** determina dos distribuciones de bits:

- **Precisión simple** para registros de 32 bits (s:1bit q:8bits c:23bits) puede representar números reales entre  $2,938736E-39$  y  $1,701412E+38$
- **Precisión doble** para registros de 64 bits (s:1bit q:11bits c:52bits) puede representar números reales entre  $5,562684646268003E-309$  y  $8,988465674311580E+307$



# Aritmética de las computadoras digitales

## 1+1 = 2 ...¿cierto?

### Operaciones de coma flotante

Para poder sumar, restar, multiplicar dividir números representados por coma flotante se deben ordenar los números previo a la operación.

**Suma y resta:** se aumenta la mantisa del número de menor exponente.

Números que queremos sumar	$123456,7 + 101,7654 = 123558,4654$
Expresados con 6 decimales	$(1,234567 \times 10^5) + (1,017654 \times 10^2)$
Se igualan los exponentes	$(1,234567 \times 10^5) + (0,001018 \times 10^5)$
Se suman las mantisas	$(1,234567 + 0,001018) \times 10^5 = 1,235585 \times 10^5$

Si comparamos la suma inicial con la obtenida, el error por el redondeo es:  $123558,4654 - 123558,5 = -0,0346$

**Multiplicación y división:** se multiplican o dividen las mantisas y se suman o restan los exponentes.



# Fuentes de errores de una solución numérica

Las ecuaciones matemáticas se pueden **aproximar mediante representaciones más simples**. Estas representaciones **introducen errores**.

1) Cuando se aproximan funciones con polinomios de Taylor se produce un **error de truncamiento**.

2) Cuando se utiliza un **subdominio más simple** se produce un **error de discretización**.

3) Cuando se utiliza aritmética de punto flotante con precisión finita se introduce un **error de redondeo**. Si el error de redondeo es grande, se pueden perder cifras significativas en las operaciones matemáticas (**cancelación**). A su vez, si las operaciones matemáticas dan como resultado un número que no se puede representar (por exceso o defecto), se produce un **desbordamiento (overflow)**.

# Introducción a Octave



## Scientific Programming Language

- Powerful mathematics-oriented syntax with built-in plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Windows
- Drop-in compatible with many Matlab scripts

Página de GNU Octave

Métodos Numéricos Básicos con Octave

The screenshot shows the GNU Octave GUI interface with several windows and annotations:

- File Browser:** Shows the current directory as C:/Users/Chevie. The file list includes folders like .cfx, .config, Contacts, Desktop, Documents, Downloads, Favorites, Intel, Links, Music, Pictures, Saved Games, Searches, Videos, and files like .octave\_hist.
- Command Window:** Displays the GNU Octave version 4.2.1, copyright information, and configuration details. It shows the prompt >> |.
- Workspace:** A table with columns for Name, Class, Dimension, Value, and Attribute. It is currently empty.
- Command History:** Shows a list of commands, including multiple instances of 'figura\_velocidad' and 'exit'.
- Annotations:**
  - 'Directorio de trabajo' is overlaid on the File Browser window.
  - 'Variables en memoria' is overlaid on the Workspace window.
  - 'Historial de comandos' is overlaid on the Command History window.
  - 'Ventana de comandos' is overlaid on the Command Window.
  - 'Pestañas (comandos, editor código fuente, manual de ayuda)' is overlaid on the bottom tabs, with arrows pointing to the Command Window, Editor, and Documentation tabs.

# Nuestro primer programa

```
function [TF]=tempconvert (TC)
%conversión de temperatura expresada en
%Celsius a Fahrenheit
%uso [TF]=tempconvert(TC)
%Argumentos de entrada: TC:Temperatura en
%grados Celsius
%Argumentos de salida: TF:Temperatura en
%grados Fahrenheit
clc; %borra pantalla
TF=9/5*TC+32;
endfunction
```

## Receta para crear una función

Declaración de función  
[salidas]=nombre(entradas)

### Bloque de comentarios para ayuda

Instrucción 1  
Instrucción 2  
Instrucción 3

...

**Fin de la función**

Para probar la función escribir en la ventana de comandos...

```
tempconvert (25)
```

```
tempconvert (25);
```

```
[TF]=tempconvert (25);
```

# Nuestro segundo programa

Vector definido por dos puntos

$$\vec{V} = \vec{P}_2 - \vec{P}_1$$

$$(v_1, v_2) = (x_2, y_2) - (x_1, y_1)$$

$$v_1 = x_2 - x_1$$

$$v_2 = y_2 - y_1$$

Calcular el vector entre los puntos (0,0)  
y (1,1)

Investigar la función “quiver” con el  
comando “help” o “doc”

Instrucciones para crear  
la función

[salidas]=nombre(entradas)

**Descripción**

**Forma de uso**

**Variables de entrada  
y salida**

Leer  $x_1, y_1, x_2, y_2$

Calcular  $v_1$

Calcular  $v_2$

Graficar en pantalla el  
vector

**Fin de la función**

# Nuestro tercer programa

Soluciones de una ecuación cuadrática  
 (fórmula de Baskara)

$$a x^2 + b x + c = 0$$

$$D = b^2 - 4 a c$$

$$x_1 = \frac{-b - \sqrt{D}}{2 a}$$

$$x_2 = \frac{-b + \sqrt{D}}{2 a}$$

Calcular las soluciones a mano y con el programa, para los siguientes casos:

$$a=4, b=0, c=-1$$

$$a=4, b=0, c=1$$

Instrucciones para crear  
 la función

[salidas]=nombre(entradas)

**Descripción**

**Forma de uso**

**Variables de entrada  
 y salida**

Leer a,b,c

Calcular D

Si  $D > 0$  calcular  $x_1$  y  $x_2$

Si  $D = 0$  calcular  $x_1 = x_2$

Si  $D < 0$  no calcular y  
 avisar al usuario

**Fin de la función**