Universidad Nacional de La Matanza

DIIT

Departamento de Ingeniería e Investigaciones Tecnológicas

A rational mechanics course where everything is made with Python code

Bettachini, Víctor A.; Real, Mariano A.; Palazzo, Edgardo

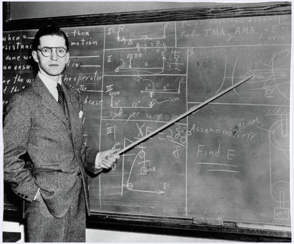
New Media Pedagogy 23

Value students and professors time

Licklider (1957): 85 % of “thinking” are actually mundane task (calculations, drawing, etc.)

Value students and professors time

Licklider (1957): 85 % of "thinking" are actually mundane task (calculations, drawing, etc.)

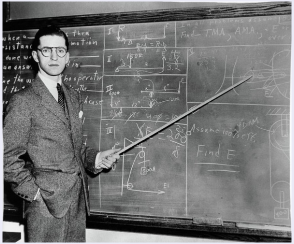


Classroom and practice: an excercise on transcription

- Professor: lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides

Value students and professors time

Licklider (1957): 85 % of "thinking" are actually mundane task (calculations, drawing, etc.)

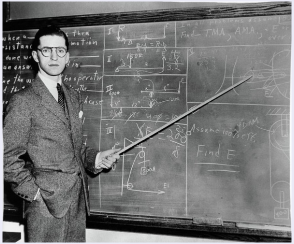


Classroom and practice: an excercise on transcription

- Professor: lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{copies}}$ notebooks

Value students and professors time

Licklider (1957): 85 % of "thinking" are actually mundane task (calculations, drawing, etc.)

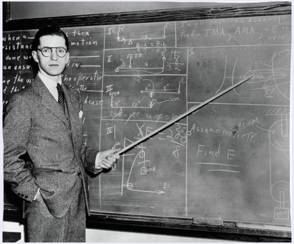


Classroom and practice: an excercise on transcription

- Professor: lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{copies}}$ notebooks
- Práctice **reiterate** diagrames, calculations, etc.

Value students and professors time

Licklider (1957): 85 % of "thinking" are actually mundane task (calculations, drawing, etc.)

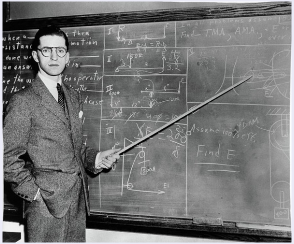


Classroom and practice: an excercise on transcription

- Professor: lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{copies}}$ notebooks
- Práctice **reiterate** diagrames, calculations, etc.
- Boredom $\implies \downarrow$ concentration on the subject

Value students and professors time

Licklider (1957): 85 % of “thinking” are actually mundane task (calculations, drawing, etc.)

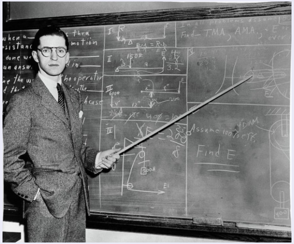


Classroom and practice: an excercise on transcription

- Professor: lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{copies}}$ notebooks
- Práctice **reiterate** diagrames, calculations, etc.
- Boredom $\implies \downarrow$ concentration on the subject

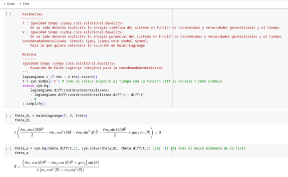
Value students and professors time

Licklider (1957): 85 % of "thinking" are actually mundane task (calculations, drawing, etc.)



Classroom and practice: an excercise on transcription

- Professor: lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{copies}}$ notebooks
- Práctice **reiterate** diagrames, calculations, etc.
- Boredom $\implies \downarrow$ concentration on the subject



+ Code + Text

```
    Parámetros
    ----------
    T : Igualdad Sympy (sympy.core.relational.Equality)
        En su lado derecho explicita la energía cinética del sistema en función de coordenadas y velocidades generalizadas y el tiempo.
    V : Igualdad Sympy (sympy.core.relational.Equality)
        En su lado derecho explicita la energía potencial del sistema en función de coordenadas y velocidades generalizadas y el tiempo.
    coordenadaGeneralizada: Símbolo Sympy (sympy.core.symbol.Symbol)
        Para la que quiere obtenerse la ecuación de Euler-Lagrange

    Retorna
    -------
    Igualdad Sympy (sympy.core.relational.Equality)
        Ecuación de Euler-Lagrange homogénea para la coordenadaGeneralizada
    '''
    lagrangiano = (T.rhs - V.rhs).expand()
    t = sym.Symbol('t') # como se deriva respecto al tiempo con la función diff se declara t como símbolo
    return sym.Eq(
        lagrangiano.diff(coordenadaGeneralizada)
        - lagrangiano.diff(coordenadaGeneralizada.diff(t)).diff(t)
        , 0
    ).simplify()
```

```
theta_EL = eulerLagrange(T, V, theta)
theta_EL
```

$$\ell\left(\frac{\ell m_1 \sin(2\theta)\dot{\theta}^2}{2} - \ell m_1 \cos^2(\theta)\ddot{\theta} - \ell m_2 \sin^2(\theta)\ddot{\theta} - \frac{\ell m_2 \sin(2\theta)\dot{\theta}^2}{2} + g m_2 \sin(\theta)\right) = 0$$

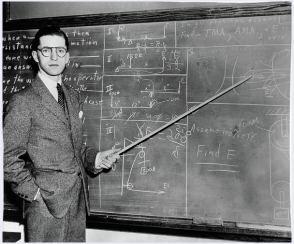
```
theta_a = sym.Eq(theta.diff(t,2), sym.solve(theta_EL, theta.diff(t,2) )[0]  )# [0] toma el único elemento de la lista
theta_a
```

$$\ddot{\theta} = \frac{\left(\ell m_1 \cos(\theta)\dot{\theta}^2 - \ell m_2 \cos(\theta)\dot{\theta}^2 + g m_2\right)\sin(\theta)}{\ell\left(m_1 \cos^2(\theta) + m_2 \sin^2(\theta)\right)}$$

- Professor: ideas $\rightarrow$ new code/notes in repository

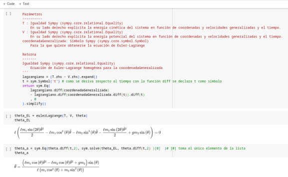
Value students and professors time

Licklider (1957): 85 % of "thinking" are actually mundane task (calculations, drawing, etc.)



Classroom and practice: an excercise on transcription

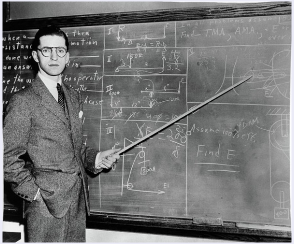
- Professor: lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{copies}}$ notebooks
- Práctice **reiterate** diagrames, calculations, etc.
- Boredom $\implies \downarrow$ concentration on the subject



- Professor: ideas $\rightarrow$ new code/notes in repository
- Student: course repository $\rightarrow$ its own modifiable one

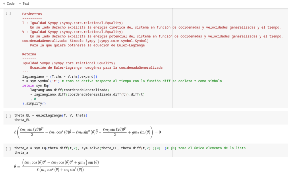
Value students and professors time

Licklider (1957): 85 % of "thinking" are actually mundane task (calculations, drawing, etc.)



Classroom and practice: an excercise on transcription

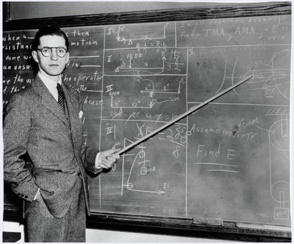
- Professor: lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{copies}}$ notebooks
- Práctice **reiterate** diagrames, calculations, etc.
- Boredom $\implies \downarrow$ concentration on the subject



- Professor: ideas $\rightarrow$ new code/notes in repository
- Student: course repository $\rightarrow$ its own modifiable one
- Use code to solve problems $=$ **recycle** professor's code

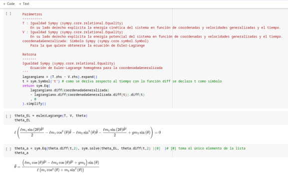
Value students and professors time

Licklider (1957): 85 % of "thinking" are actually mundane task (calculations, drawing, etc.)



Classroom and practice: an excercise on transcription

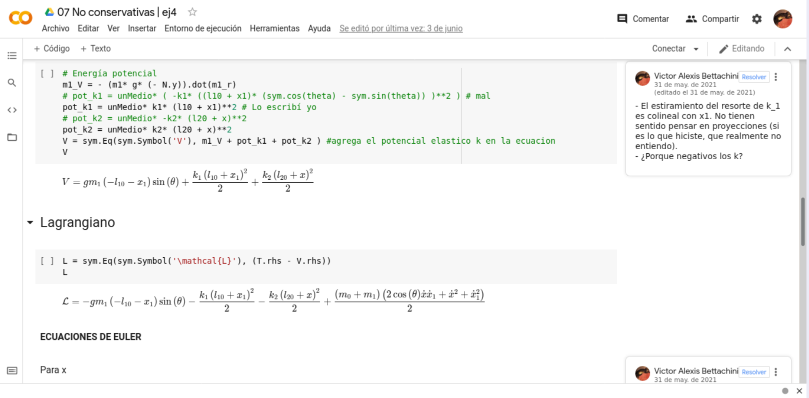
- Professor: lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{copies}}$ notebooks
- Práctice **reiterate** diagrames, calculations, etc.
- Boredom $\implies \downarrow$ concentration on the subject



- Professor: ideas $\rightarrow$ new code/notes in repository
- Student: course repository $\rightarrow$ its own modifiable one
- Use code to solve problems = **recycle** professor's code
- Modifiying it solves different problems

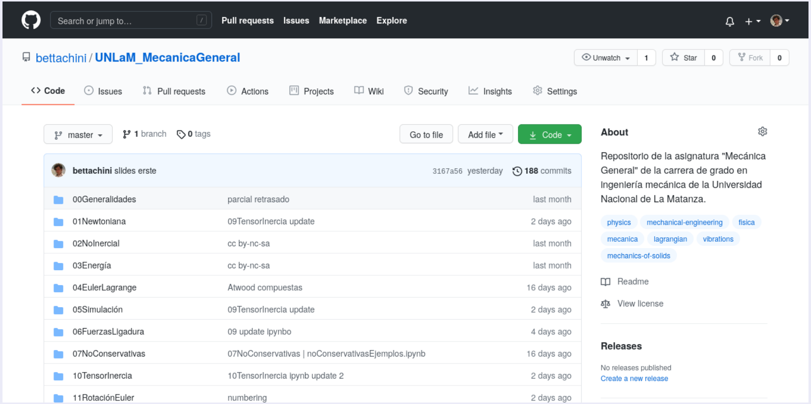
The course setup: Google Colaboratory

- It runs Jupyter notebooks online, and it's free.
- Students can collaborate remotely, working on the same notebook.
- Teachers can edit and comment the work of students.



The course setup: GitHub

- Course material, accesible in a clear way, and easy to keep updated.
- Google Colaboratory loads Jupyter Notebooks directly from GitHub.



Course Overview

Class 1: Concise Mathematical Notation

- Latex typesetting, standard for the American Mathematical Society

Considero que el potencial V es nulo en el origen de coordenadas, es decir que donde se encuentra su mínimo $\varphi = 0, V(\varphi = 0) = -mg\ell$ y por tanto

$$V(\varphi) = mg\left(-\ell\cos\varphi\right) = -mg\ell\cos\varphi,$$

Como vemos la aproximación funciona bastante bien. Conformes con ella calculamos la fuerza

$$\vec{F} = -\vec{\nabla}V = -\left(\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial\varphi}, \frac{\partial}{\partial z}\right)V(\varphi)$$

Pero solo nos interesa expresar la 2.a ley de Newton para lo que pasa en $\hat{\varphi}$

$$m\ddot{\vec{r}} \cdot \hat{\varphi} = -\frac{1}{r}\frac{\partial}{\partial\varphi}V(\varphi)$$

En el lado izquierdo de la expresión de la aceleración en cilíndricas $\ddot{\vec{r}} = (\ddot{r} - r\dot{\varphi}^2)\hat{r} + (\dot{r}\dot{\varphi}^2 + r\ddot{\varphi})\hat{\varphi} + \ddot{z}\hat{z}$, nos quedamos solo con la componente en $\hat{\phi}$,

$$\ddot{\vec{r}} \cdot \hat{\varphi} = \dot{r}\dot{\varphi}^2 + r\ddot{\varphi}$$

y como el hilo del péndulo es rígido e inextensible $r \equiv \ell$ solo queda de esto

$$\ddot{\vec{r}} \cdot \hat{\varphi} = \ell\ddot{\varphi}$$

En el lado derecho la derivada del potencial respecto a φ es

$$\frac{\partial}{\partial\varphi}V(\varphi) = mg\ell\sin(\varphi)$$

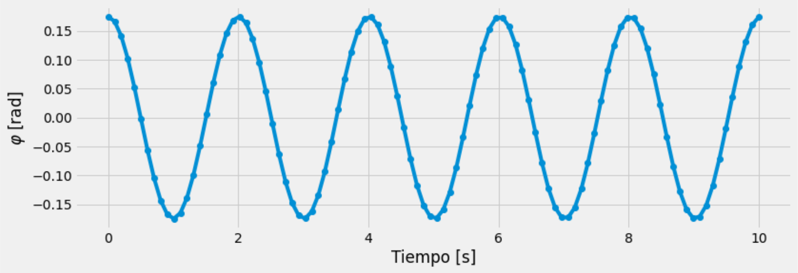
Course Overview

Class 1: precise and reproducible graphics

- The use of Python code for producing graphics is explicit, making the students play with it.

```
# graficación
fig, ax = plt.subplots(figsize=(12, 4))
ax.plot(tiempos, phi(tiempos), 'o-' )
ax.set_xlabel('Tiempo [s]')
ax.set_ylabel(r'$\varphi$ [rad]')
```

```
Text(0, 0.5, '$\\varphi$ [rad]')
```



Class 3: symbolic calculations

- These students have completed courses in Calculus and Algebra, now it's time to focus on applying the tools acquired in those courses.

[8]:
```
m2_v_cuadrado = m2_v.dot(m2_v)
m2_v_cuadrado
```

[8]: $\ell^2 \sin^2(\varphi)\dot{\varphi}^2 + (\ell\cos(\varphi)\dot{\varphi} + \dot{x})^2$

Con esto la energía cinética queda

$$\begin{aligned} T(\dot{x}_1, \varphi, \dot{\varphi}) &= \frac{m_1}{2}\left(\dot{\vec{r}}_1\right)^2 + \frac{m_2}{2}\left(\dot{\vec{r}}_2\right)^2 \\ &= \frac{m_1}{2}\dot{x}^2 + \frac{m_2}{2}\left(\dot{x}^2 + 2\dot{x}\ell\cos\varphi\dot{\varphi} + l^2\dot{\varphi}^2\right) \end{aligned}$$

[9]:
```
# Energía cinética
unMedio = sym.Rational(1,2) # Rational: fracción de enteros, alternativamente podría haberse usado 0.5
m1_T = unMedio* m1* m1_v_cuadrado
m2_T = unMedio* m2* m2_v_cuadrado
T = sym.Eq(sym.Symbol('T'), (m1_T + m2_T) ) # simplify: simplifica usando factor común y otras operaciones
T
```

[9]: $T = \frac{m_1\dot{x}^2}{2} + \frac{m_2\left(\ell^2\sin^2(\varphi)\dot{\varphi}^2 + (\ell\cos(\varphi)\dot{\varphi} + \dot{x})^2\right)}{2}$

Course Overview

Class 4: Dynamics

Ecuaciones de Euler-Lagrange

Para x

[8]:
```
x_EL = sym.Eq(L.rhs.diff(x) - L.rhs.diff(x.diff(t)).diff(t), 0 ).simplify() # ecuación igualando a cero
x_EL
```

[8]:

$$m_1 \ddot{x} + m_2 \left(-\ell \sin(\phi)\dot{\phi}^2 + \ell \cos(\phi)\ddot{\phi} + \ddot{x}\right) = 0$$

Esta es una ecuación diferencial lineal de segundo orden homogena. De aquí podría despejarse $\ddot{x}$

[9]:
```
sym.Eq(x.diff(t,2),
       list( sym.solveset(x_EL, x.diff(t,2) ) )[0]  # solveset devuelve un set, que convertimos a lista
      ) # aceleración = x punto punto [m s-2]
```

[9]:

$$\ddot{x} = \frac{\ell m_2 \left(\sin(\phi)\dot{\phi}^2 - \cos(\phi)\ddot{\phi}\right)}{m_1 + m_2}$$

Pero queda en función de otra aceleración $\ddot{\phi}$.

Para ϕ

[10]:
```
phi_EL = sym.Eq(L.rhs.diff(phi) - L.rhs.diff(phi.diff(t)).diff(t), 0 ).simplify() # ecuación igualando a cero
phi_EL
```

Course Overview

Class 4: Automation of resolutions

- No heavy calculations consuming the energy of the students.
- Their complexity does not limit the mechanical problems that can be tackled.

```
[14]: sistemaEcuaciones = [
          x_EL,
          phi_EL,
      ]
      variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas
      variablesDespeje_sol= sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]
```

```
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0] ) # [m s-2]
      phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1] ) # [m s-2]
      x_pp, phi_pp
```

[15]:

$$\left(\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 \left(\ell m_2 \cos(\phi)\dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \ \ddot{\phi} = -\frac{\left(\ell m_2 \cos(\phi)\dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{\ell \left(m_1 + m_2 \sin^2(\phi)\right)} \right)$$

Course Overview

Class 5: Numerical analysis

- Explicit solutions.

```
[22]: # defino una función con el sistema de derivadas
      # t : no se usa en este sistema pero lo dejamos para uso posterior
      # y : lista de estado con [y[0], y[1], y[2], y[3]]
      # y[0]: x
      # y[1]: x punto
      # y[2]: phi
      # y[3]: phi punto
      # dydt : lista de derivadas
      def y_punto(t, y):
          dydt = [y[1],
                  x_pp_numpy(y[0], y[1], y[2], y[3]),
                  y[3],
                  phi_pp_numpy(y[0], y[1], y[2], y[3]),
                 ]
          return dydt

[23]: # Integración de a pasos en el tiempo
      y_ode2 = solve_ivp(y_punto, (t_rango[0], t_rango[-1] ), y_inicial, t_eval = t_rango)

[25]: y_ode2.y[0]

[25]: array([ 1.        ,  0.95510744,  0.92131146,  0.89820932,  0.88468059,
              0.87877042,  0.87745354,  0.87702754,  0.87352768,  0.86357726,
              0.84474673,  0.81565733,  0.77559949,  0.72423163,  0.66166451,
              0.588266  ,  0.50468237,  0.41250381,  0.31433661,  0.21366454,
              0.11444308,  0.02023394, -0.06599563, -0.14244216, -0.20809592,
             -0.26250272, -0.30576388, -0.33796804, -0.35953138, -0.37175469,
```

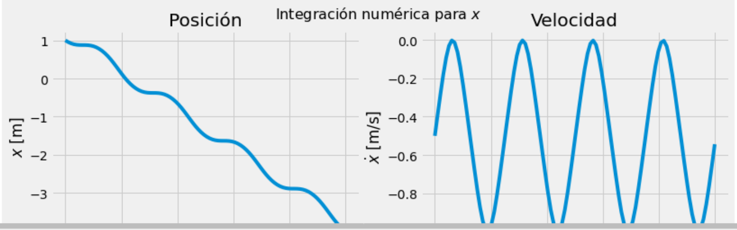
Course Overview

Class 5: analysis of results

```
[26]: solucion = y_ode2
      nombreCoordenada = 'x'

      fig, ax = plt.subplots(nrows= 1, ncols= 2, squeeze=False, figsize=(12, 4)) # dos figuras en la misma fila
      fig.suptitle('Integración numérica para $'+ nombreCoordenada + '$', fontsize=16)
      ax[0,0].plot(solucion.t, solucion.y[0]) # posición x
      ax[0,0].set(xlabel='t [s]', ylabel= '$' + nombreCoordenada+ '$ [m]', title='Posición')
      ax[0,1].plot(solucion.t, solucion.y[1]) # velocidad x
      ax[0,1].set(xlabel='t [s]', ylabel='$\dot{' + nombreCoordenada+ '}$ [m/s]', title='Velocidad')
```

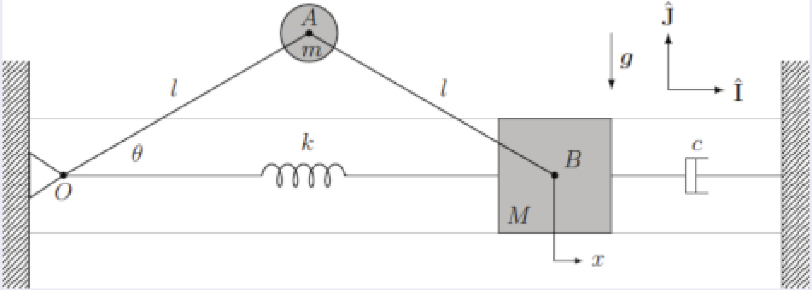
```
[26]: [Text(0.5, 0, 't [s]'),
       Text(0, 0.5, '$\\dot{x}$ [m/s]'),
       Text(0.5, 1.0, 'Velocidad')]
```



0 $_ 1 Python 3 (ipykernel) | Idle Saving completed Mode:

Class 7: Adding complexity

- Reuse code from previous classes to study complex situations.
- Similar to real world problems.



Engineering students must take advantage of code at every single lecture

Engineering students must take advantage of code at every single lecture

- Currently they use a pocket calculator **after they learnt** learning arithmetics at school

Engineering students must take advantage of code at every single lecture

- Currently they use a pocket calculator **after they learnt** learning arithmetics at school
- They'll employ computational algebra **after they learnt** algebra and calculus

[14]:
```
sistemaEcuaciones = [
    x_EL,
    phi_EL,
]
variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas
variablesDespeje_sol= sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]
```

[15]:
```
x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0] ) # [m s-2]
phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1] ) # [m s-2]
x_pp, phi_pp
```

[15]:

$$\left(\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 \left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{\left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{\ell \left(m_1 + m_2 \sin^2(\phi)\right)} \right)$$

Engineering students must take advantage of code at every single lecture

- Currently they use a pocket calculator **after they learnt** learning arithmetics at school
- They'll employ computational algebra **after they learnt** algebra and calculus
  - Focus on new skills, not in automatable calculations

```
[14]: sistemaEcuaciones = [
          x_EL,
          phi_EL,
      ]
      variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas
      variablesDespeje_sol= sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]
```

```
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0] ) # [m s-2]
      phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1] ) # [m s-2]
      x_pp, phi_pp
```

[15]:

$$\left(\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 \left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \ \ddot{\phi} = -\frac{\left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{\ell \left(m_1 + m_2 \sin^2(\phi)\right)} \right)$$

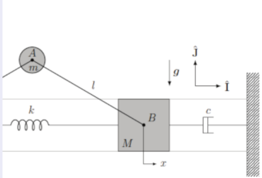
Engineering students must take advantage of code at every single lecture

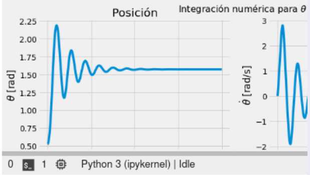
- Currently they use a pocket calculator **after they learnt** learning arithmetics at school
- They'll employ computational algebra **after they learnt** algebra and calculus
  - Focus on new skills, not in automatable calculations
  - Employing numerical calculus they solve what is impossible in a blackboard/paper

[14]:
```
sistemaEcuaciones = [
    x_EL,
    phi_EL,
]
variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas
variablesDespeje_sol= sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]
```

[15]:
```
x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0] ) # [m s-2]
phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1] ) # [m s-2]
x_pp, phi_pp
```

[15]:
$$\left(\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 \left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)},\ \ddot{\phi} = -\frac{\left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{\ell \left(m_1 + m_2 \sin^2(\phi)\right)} \right)$$





Engineering students must take advantage of code at every single lecture

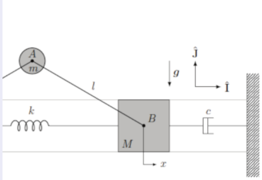
- Currently they use a pocket calculator **after they learnt** learning arithmetics at school
- They'll employ computational algebra **after they learnt** algebra and calculus
  - Focus on new skills, not in automatable calculations
  - Employing numerical calculus they solve what is impossible in a blackboard/paper

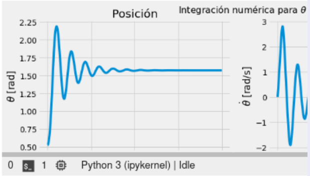
```
[14]: sistemaEcuaciones = [
          x_EL,
          phi_EL,
      ]
      variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas
      variablesDespeje_sol= sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]

[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0] ) # [m s-2]
      phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1] ) # [m s-2]
      x_pp, phi_pp
```

[15]:

$$\left(\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 \left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{\left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{\ell \left(m_1 + m_2 \sin^2(\phi)\right)} \right)$$





Engineering students must take advantage of code at every single lecture

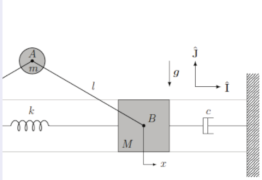
- Currently they use a pocket calculator **after they learnt** learning arithmetics at school
- They'll employ computational algebra **after they learnt** algebra and calculus
  - Focus on new skills, not in automatable calculations
  - Employing numerical calculus they solve what is impossible in a blackboard/paper

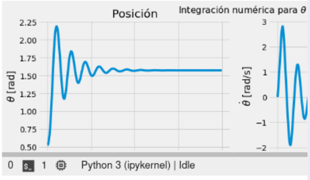
```
[14]: sistemaEcuaciones = [
          x_EL,
          phi_EL,
      ]
      variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas
      variablesDespeje_sol= sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]
```

```
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0] ) # [m s-2]
      phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1] ) # [m s-2]
      x_pp, phi_pp
```

[15]:

$$\left(\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 \left(\ell m_2 \cos(\phi)\dot{\phi}^2 + g m_1 + g m_2\right)\sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{\left(\ell m_2 \cos(\phi)\dot{\phi}^2 + g m_1 + g m_2\right)\sin(\phi)}{\ell \left(m_1 + m_2 \sin^2(\phi)\right)} \right)$$





Papert (1980) "...the best learning takes place when the learner takes charge"

- An expample problem is solved by the professor provided code

Engineering students must take advantage of code at every single lecture

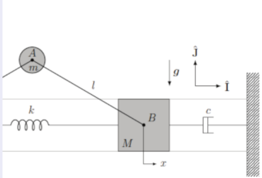
- Currently they use a pocket calculator **after they learnt** learning arithmetics at school
- They'll employ computational algebra **after they learnt** algebra and calculus
  - Focus on new skills, not in automatable calculations
  - Employing numerical calculus they solve what is impossible in a blackboard/paper

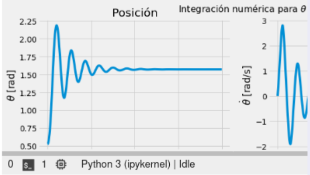
```
[14]: sistemaEcuaciones = [
          x_EL,
          phi_EL,
      ]
      variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas
      variablesDespeje_sol= sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]

[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0] ) # [m s-2]
      phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1] ) # [m s-2]
      x_pp, phi_pp
```

[15]:

$$\left(\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 \left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \ \ddot{\phi} = -\frac{\left(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{\ell \left(m_1 + m_2 \sin^2(\phi)\right)} \right)$$





Papert (1980) "...the best learning takes place when the learner takes charge"

- An expample problem is solved by the professor provided code
- The student modifies it to solve other related problems

Engineering students must take advantage of code at every single lecture

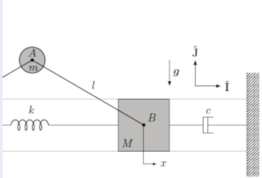
- Currently they use a pocket calculator **after they learnt** learning arithmetics at school
- They'll employ computational algebra **after they learnt** algebra and calculus
  - Focus on new skills, not in automatable calculations
  - Employing numerical calculus they solve what is impossible in a blackboard/paper

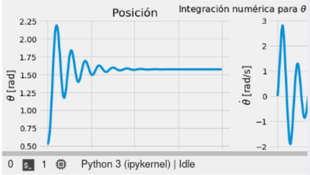
```
[14]: sistemaEcuaciones = [
          x_EL,
          phi_EL,
      ]
      variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas
      variablesDespeje_sol= sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]

[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0] ) # [m s-2]
      phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1] ) # [m s-2]
      x_pp, phi_pp
```

[15]:

$$\left(\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 \left(\ell m_2 \cos(\phi)\dot{\phi}^2 + g m_1 + g m_2\right) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \ \ddot{\phi} = -\frac{\left(\ell m_2 \cos(\phi)\dot{\phi}^2 + g m_1 + g m_2\right)\sin(\phi)}{\ell\left(m_1 + m_2 \sin^2(\phi)\right)} \right)$$





Papert (1980) "...the best learning takes place when the learner takes charge"

- An expapmle problem is solved by the professor provided code
- The student modifies it to solve other related problems
- Gradually he becomes autonomous by reusing not the provided but his own code

All course material can be edited on-line

All course material can be edited on-line

On-line programmable notebook: text + equations + code

File Edit View Run Kernel Tabs Settings Help

PÉNDULOENHEBRADOSOLVED.IPYNB

Launcher | cursoJupyter.ipynb | pénduloEnhebradoSolved.ip

Markdown | Python 3

3. Obtenga una expresión para la tensión que ejerce la barra

$$Q_d = \lambda_1 \frac{\partial f_1}{\partial d} = \lambda_1$$

Por tanto hay que resolver el sistema con las 3 ecuaciones de Euler-Lagrange y la única de ligadura para determinar λ_1. Esta última hay que resolverla para su caso homogéneo y expersar su derivada segunda para que esté en el mismo orden que las de Euler-Lagrange, a fin de cuentas estamos resolviendo sistemas diferenciales de 2.o orden.

[14]:
```
f_1
```

[14]: $f_1 = -l + d$

Determinamos también $\ddot{\theta}_1$ y $\ddot{\theta}_2$ pues serán necesarias para los cálculos numéricos posteriores.

[15]:
```
sistema = [theta1_EL.expand(),
           theta2_EL.expand(),
           d_EL.expand(),
           sym.Eq(f_1.rhs.diff(t,2), 0), # esto es igual a d punto punto = 0
          ]
variables = [theta1.diff(t,2), theta2.diff(t,2), lambda_1]
variables_sol = sym.nonlinsolve(sistema, variables).args[0]
```

[16]:
```
lambda_1_sol = sym.Eq(lambda_1, variables_sol.args[2])
lambda_1_sol.simplify()
```

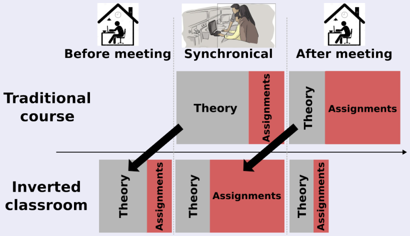
[16]: $\lambda_1 = \quad m \left(2a \cos(\theta_1 - \theta_2)\dot{\theta}_1^2 + g \cos(2\theta_1 - \theta_2) + g \cos(\theta_2) + 2d\dot{\theta}_2^2 - 2\ddot{d}\right)$

Synchronical and asynchronical work on the code

New theory alongisde its worked examples in programmable notebooks

- On-line 24/7 **asynchronical** consultations that are **public** for others to see

Flipped classroom



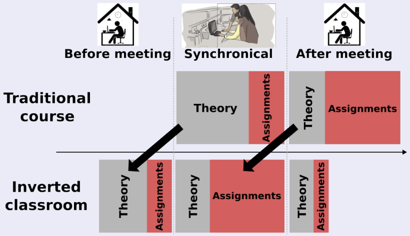
Synchronic	Theory	Assignments
Before	Read and apply	Start them
During	Consultations	Complete them
After	Additional consultations	TA's corrections

Synchronical and asynchronical work on the code

New theory alongisde its worked examples in programmable notebooks

- On-line 24/7 **asynchronical** consultations that are **public** for others to see
- **Remote collaboration** on multi-user notebooks

Flipped classroom



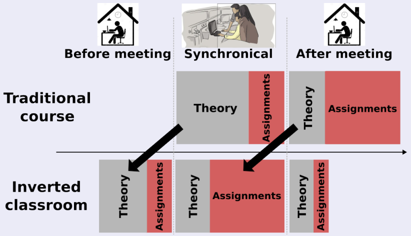
Synchronic	Theory	Assignments
Before	Read and apply	Start them
During	Consultations	Complete them
After	Additional consultations	TA's corrections

Synchronical and asynchronical work on the code

New theory alongisde its worked examples in programmable notebooks

- On-line 24/7 **asynchronical** consultations that are **public** for others to see
- **Remote collaboration** on multi-user notebooks
- Weekly meetings to **synchronically** unfinished assignments with TA's assistance

Flipped classroom



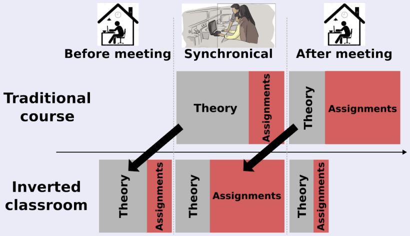
Synchronic	Theory	Assignments
Before	Read and apply	Start them
During	Consultations	Complete them
After	Additional consultations	TA's corrections

Synchronical and asynchronical work on the code

New theory alongisde its worked examples in programmable notebooks

- On-line 24/7 **asynchronical** consultations that are **public** for others to see
- **Remote collaboration** on multi-user notebooks
- Weekly meetings to **synchronically** unfinished assignments with TA's assistance
- On a weekly basis these **must** be turned-in for scoring

Flipped classroom



Synchronic	Theory	Assignments
Before	Read and apply	Start them
During	Consultations	Complete them
After	Additional consultations	TA's corrections

Asynchronic corrections and remote assistance

Student's work can be commented and edited in Google Colaboratory

07 No conservativas | ej4

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se editó por última vez: 3 de junio

Comentar Compartir

+ Código + Texto

Conectar Editando

```
[ ] # Energía potencial
    m1_V = - (m1* g* (- N.y)).dot(m1_r)
    # pot_k1 = unMedio* ( -k1* ((l10 + x1)* (sym.cos(theta) - sym.sin(theta)) )**2 ) # mal
    pot_k1 = unMedio* k1* (l10 + x1)**2 # Lo escribí yo
    # pot_k2 = unMedio* -k2* (l20 + x)**2
    pot_k2 = unMedio* k2* (l20 + x)**2
    V = sym.Eq(sym.Symbol('V'), m1_V + pot_k1 + pot_k2 ) #agrega el potencial elastico k en la ecuacion
    V
```

$$V = g m_1 \left(-l_{10} - x_1\right) \sin\left(\theta\right) + \frac{k_1 \left(l_{10} + x_1\right)^2}{2} + \frac{k_2 \left(l_{20} + x\right)^2}{2}$$

Victor Alexis Bettachini Resolver
31 de may. de 2021
(editado el 31 de may. de 2021)

- El estiramiento del resorte de k_1 es colineal con x1. No tienen sentido pensar en proyecciones (si es lo que hiciste, que realmente no entiendo).
- ¿Porque negativos los k?

Lagrangiano

```
[ ] L = sym.Eq(sym.Symbol('\mathcal{L}'), (T.rhs - V.rhs))
    L
```

$$\mathcal{L} = - g m_1 \left(-l_{10} - x_1\right) \sin\left(\theta\right) - \frac{k_1 \left(l_{10} + x_1\right)^2}{2} - \frac{k_2 \left(l_{20} + x\right)^2}{2} + \frac{\left(m_0 + m_1\right)\left(2 \cos\left(\theta\right) \dot{x} \dot{x}_1 + \dot{x}^2 + \dot{x}_1^2\right)}{2}$$

ECUACIONES DE EULER

Exams

The exam is just another excercise.

- The students send their notebook.
- Feedback is inserted in between the student's work.

```
[ ]: sym.Eq(flig_rho_rep, flig_rho)
```

[119]: $\mathrm{Q}_{\rho}(\lambda) = \lambda$

Esta fuerza efectivamente es igual a λ, pero ¿por qué? No das una justificación.

Entonces la fuerza de ligadura Q_{ρ}, o lo que es lo mismo, la fuerza que hace la barra rigida esta dada por la ecuacion:

```
[ ]: f_lig_rho = sym.Eq(sym.Function('Q_rho')(theta1, theta1.diff(t), theta2, theta2.diff(t)) , lambda_final)
     f_lig_rho
```

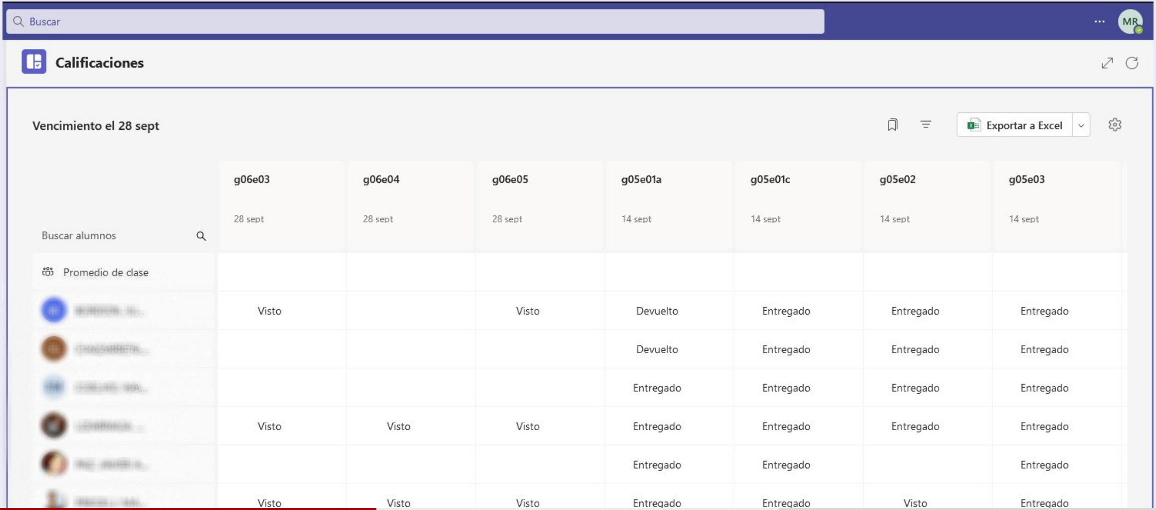
[120]: $\mathrm{Q}_{\rho}\left(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2\right) = -m\left(\ell\dot{\theta}_2^2 + a\sin\left(\theta_1 - \theta_2\right)\ddot{\theta}_1 + a\cos\left(\theta_1 - \theta_2\right)\dot{\theta}_1^2 + g\cos\left(\theta_2\right)\right)$

No es la expresión buscada.
Entre las variables hay una que no está marcada en la izquierda de la igualdad, donde decís que la expresas en función de los θ_i y $\dot{\theta}_i$.

Podés ver que te quedó expresado en función de $\ddot{\theta}_1$. Esta puede obtenerse de resolver el sistema con las otras dos ecuaciones de Euler-Lagrange.

Individualized student follow-up at Microsoft Teams

A record of the weekly turn-up of assignments



Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.

Inverted classroom

Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.
- Reinforced by: suggested bibliography and short professor's videos.

Inverted classroom

Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.
- Reinforced by: suggested bibliography and short professor's videos.
- assignments: professor's code recycling.

Inverted classroom

Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.
- Reinforced by: suggested bibliography and short professor's videos.
- assignments: professor's code recycling.
- On-line:

Inverted classroom

Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.
- Reinforced by: suggested bibliography and short professor's videos.
- assignments: professor's code recycling.
- On-line:
  - Remote collaboration and correction

Inverted classroom

Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.
- Reinforced by: suggested bibliography and short professor's videos.
- assignments: professor's code recycling.
- On-line:
  - Remote collaboration and correction
  - Doesn't require powerful nor at campus computers

Inverted classroom

Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.
- Reinforced by: suggested bibliography and short professor's videos.
- assignments: professor's code recycling.
- On-line:
  - Remote collaboration and correction
  - Doesn't require powerful nor at campus computers
  - A dated record of each student's work

Inverted classroom

Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.
- Reinforced by: suggested bibliography and short professor's videos.
- assignments: professor's code recycling.
- On-line:
  - Remote collaboration and correction
  - Doesn't require powerful nor at campus computers
  - A dated record of each student's work

Inverted classroom

- Theory: emphasis on student's autonomus reading

Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.
- Reinforced by: suggested bibliography and short professor's videos.
- assignments: professor's code recycling.
- On-line:
  - Remote collaboration and correction
  - Doesn't require powerful nor at campus computers
  - A dated record of each student's work

Inverted classroom

- Theory: emphasis on student's autonomus reading
- Consultations: mostly on-line asynchronical and of public access

Summary

A course centred on code

- Theory: text + equations + executable cod in digital notebooks.
- Reinforced by: suggested bibliography and short professor's videos.
- assignments: professor's code recycling.
- On-line:
  - Remote collaboration and correction
  - Doesn't require powerful nor at campus computers
  - A dated record of each student's work

Inverted classroom

- Theory: emphasis on student's autonomus reading
- Consultations: mostly on-line asynchronical and of public access
- TA personal assisantace when completing assignments in synchronical meetings

Current developments

Current developments

2023 Students feedback improved:

Current developments

2023 Students feedback improved:

- Theory notes and code at repository

Current developments

2023 Students feedback improved:

- Theory notes and code at repository
- Grading of assignments methodology

  Evaluating each one of them $\rightarrow$ higher student's performance

Current developments

2023 Students feedback improved:

- Theory notes and code at repository
- Grading of assignments methodology

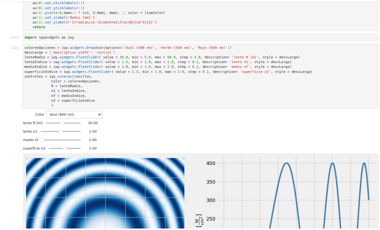
  Evaluating each one of them $\rightarrow$ higher student's performance

Current developments

2023 Students feedback improved:

- Theory notes and code at repository
- Grading of assignments methodology
  Evaluating each one of them → higher student's performance

2024
- A course on optics and waves will incorporate part of the methodology



Current developments

2023 Students feedback improved:

- Theory notes and code at repository
- Grading of assignments methodology
  Evaluating each one of them → higher student's performance

2024

- A course on optics and waves will incorporate part of the methodology
- AI assistance in code generation employing *GitHub Copilot*