A rational mechanics course where everything is made with Python code

Bettachini, Víctor A.[1] Real, Mariano A.[1] Palazzo, Edgardo[2]

[1] DIIT, Universidad Nacional de La Matanza, [2] FRA, Universidad Tecnológica Nacional, ARGENTINA

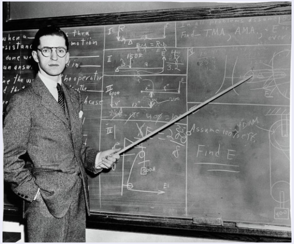
New Media Pedagogy 23: research trends, methodological challenges and succesful implementations
21-24 november 2023

Value student's and professor's time

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

Value student's and professor's time

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

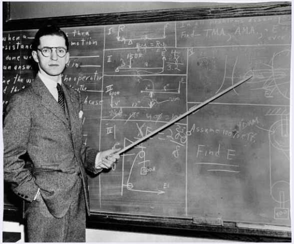


Most current classrooms: an excercise on transcription

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

Value student's and professor's time

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

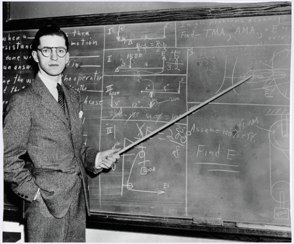


Most current classrooms: an excercise on transcription

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

Value student's and professor's time

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

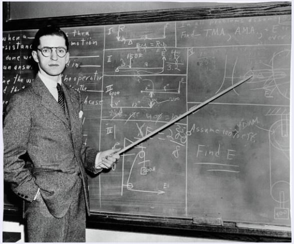


Most current classrooms: an excercise on transcription

- Professor: prepares lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{transcribes}}$ own notebooks
- Assignments: do **again** same calculations, drawings,...

Value student's and professor's time

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

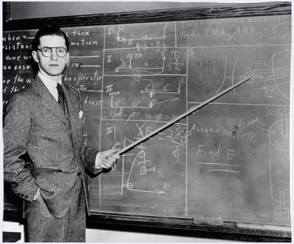


Most current classrooms: an excercise on transcription

- Professor: prepares lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{transcribes}}$ own notebooks
- Assignments: do **again** same calculations, drawings,...
- Boredom $\rightarrow$ loss of concentration on the subject

Value student's and professor's time

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

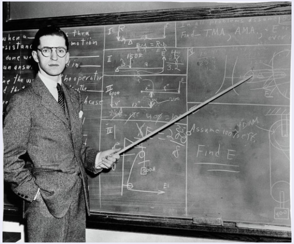


Most current classrooms: an excercise on transcription

- Professor: prepares lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{transcribes}}$ own notebooks
- Assignments: do **again** same calculations, drawings,...
- Boredom → loss of concentration on the subject

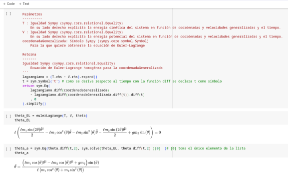
Value student's and professor's time

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



Most current classrooms: an excercise on transcription

- Professor: prepares lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{transcribes}}$ own notebooks
- Assignments: do **again** same calculations, drawings,...
- Boredom $\rightarrow$ loss of concentration on the subject

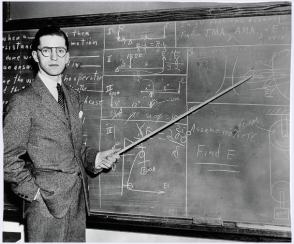


Digital course material with embedded code

- Professor: ideas $\xrightarrow{\textit{updates}}$ code/theory in repository

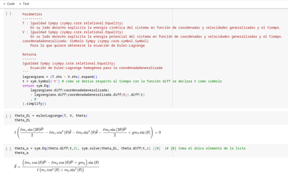
Value student's and professor's time

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



Most current classrooms: an excercise on transcription

- Professor: prepares lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{transcribes}}$ own notebooks
- Assignments: do **again** same calculations, drawings,...
- Boredom $\rightarrow$ loss of concentration on the subject

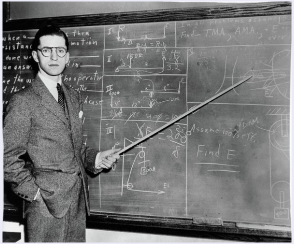


Digital course material with embedded code

- Professor: ideas $\xrightarrow{\textit{updates}}$ code/theory in repository
- Student: course repository $\xrightarrow{\textit{clones}}$ its own modifiable one

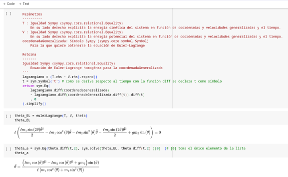
Value student's and professor's time

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



Most current classrooms: an excercise on transcription

- Professor: prepares lessons $\xrightarrow{\textit{by heart}}$ blackboard/slides
- Student: blackboard/slides $\xrightarrow{\textit{transcribes}}$ own notebooks
- Assignments: do **again** same calculations, drawings,...
- Boredom $\rightarrow$ loss of concentration on the subject



Digital course material with embedded code

- Professor: ideas $\xrightarrow{\textit{updates}}$ code/theory in repository
- Student: course repository $\xrightarrow{\textit{clones}}$ its own modifiable one
- **Recycles** professor's code to solve new problems

Engineering students can take advantage of code at every single lecture

Engineering students can take advantage of code at every single lecture

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

Engineering students can take advantage of code at every single lecture

- Currently they use a pocket calculator **after they learnt** arithmetics at school
- Now 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]:

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

Engineering students can take advantage of code at every single lecture

- Currently they use a pocket calculator **after they learnt** arithmetics at school
- Now 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 can take advantage of code at every single lecture

- Currently they use a pocket calculator **after they learnt** arithmetics at school
- Now 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 can take advantage of code at every single lecture

Engineering students can take advantage of code at every single lecture

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

- An example problem is solved by the professor provided code

Engineering students can take advantage of code at every single lecture

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

- An example problem is solved by the professor provided code
- The student **recycles** it to solve other related problems

Engineering students can take advantage of code at every single lecture

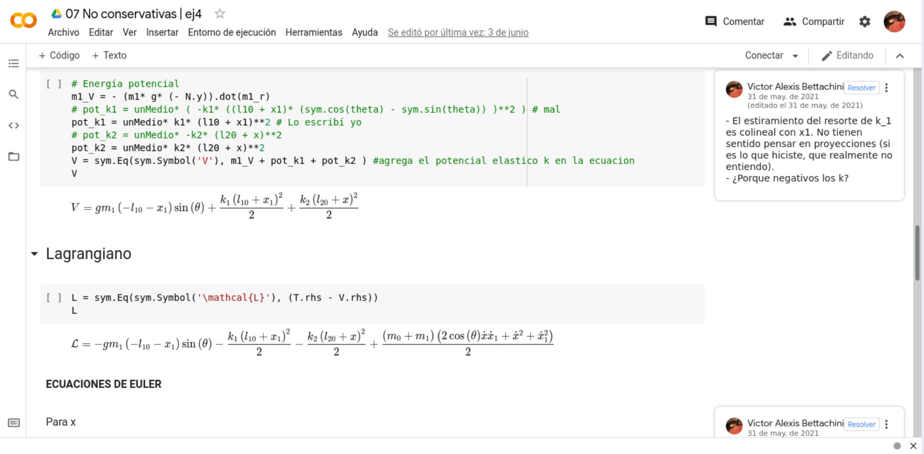
Papert (1980) "...the best learning takes place when the learner takes charge"

- An example problem is solved by the professor provided code
- The student **recycles** it to solve other related problems
- Gradually becoming autonomous by reusing not the provided but his own code

Tools — Jupyter notebook: text + equations + executable code

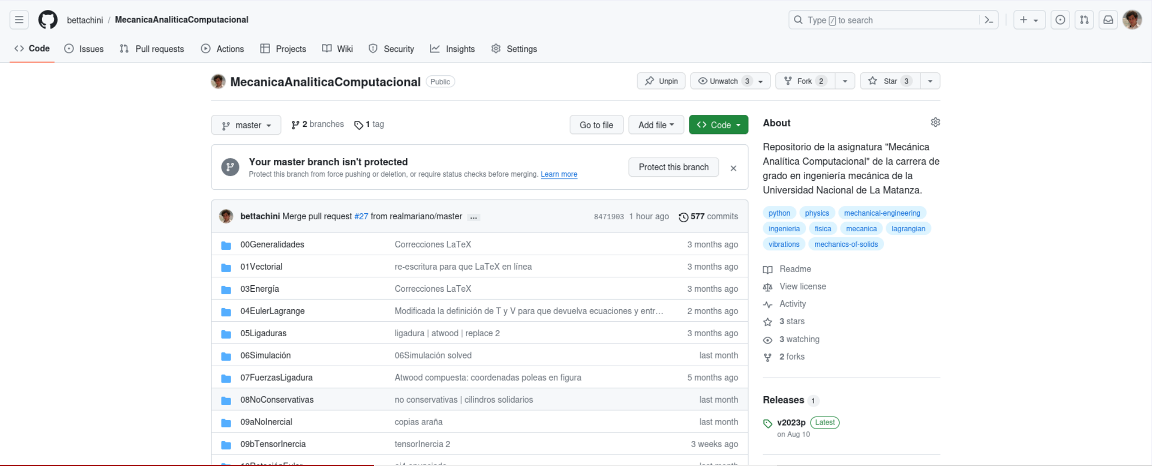
Tools — Google Colaboratory runs Jupyter notebooks on-line

- The platform allows professors to edit and comment on the work of students.
- Students can collaborate remotely, working on the same Jupyter notebook.



Tools — All course material in a GitHub repository

- Clearly organised, freely accessible, and easy to update.
- Google Colaboratory loads Jupyter notebooks directly from GitHub.



- LaTeX typesetting follows strictly the standards of 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

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

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 cílindricas $\ddot{\vec{r}} = (\ddot{r} - r\dot{\varphi}^2)\hat{r} + (\dot{r}\dot{\varphi}^2 + r\ddot{\varphi})\hat{\varphi} + \ddot{z}\hat{z}$, 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

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

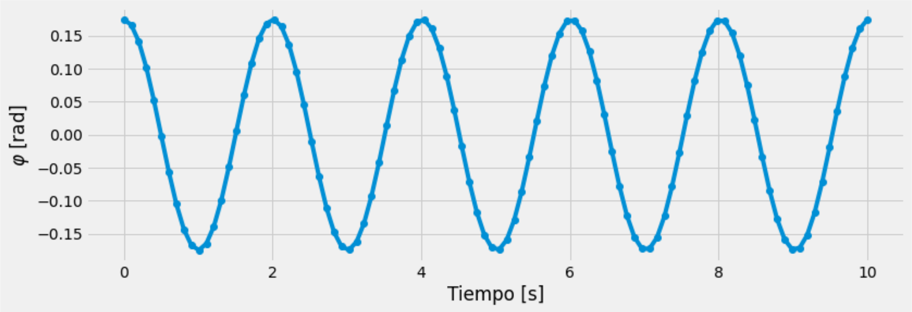
En el lado derecho la derivada del potencial respecto a φ es

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

- Explicit Python code for producing graphics. Students are enticed to 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]')
```



- Student's effort centred on new subjects by freeing them of calculus and algebraic tasks.

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

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
```

- Mathematical complexity doesn't limit the scope of tackled mechanical problems.

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

```
[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
```

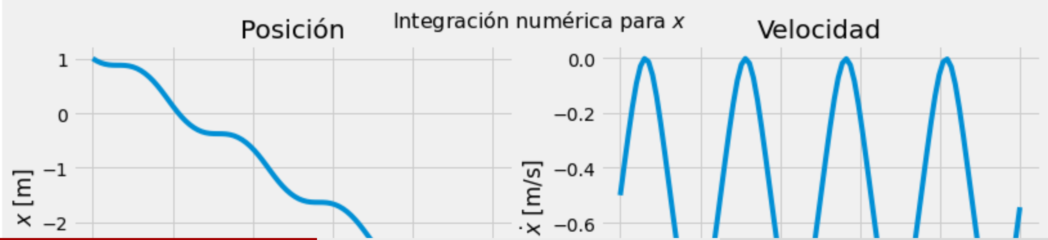
```
[23]: # Integración de a pasos en el tiempo
      y_ode2 = solve_ivp(y_punto, (t_rango[0], t_rango[-1] ), y_inicial, t_eval = t_rango)
```

```
[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,
```

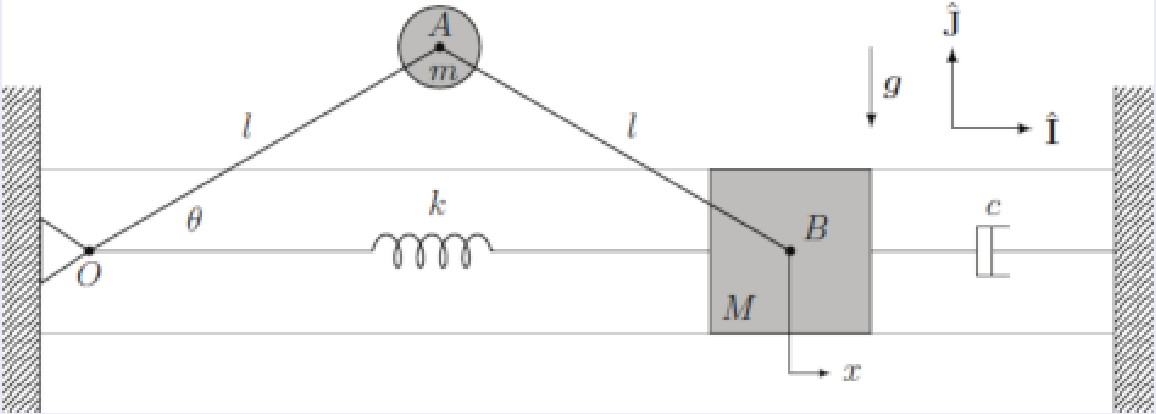
```
[26]: solucion = y_ode2
      nombreCoordenada = 'x'

      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')]
```



Tools — 7th class — Adding complexity

- Code from previous classes is **recycled** to model more realistic devices.



- Now that the students grasp the basics they're invited to take advantage of AI
- After commenting in plain text what they need code is suggested

```
    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()
```

[21]

```
x1_EL = eulerLagrange(T, V, x1)
x1_EL
```

[22]

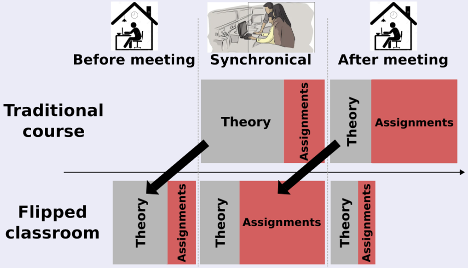
$$\frac{\pi^2 M \ddot{x}_1}{2} - g m_1 + g m_2 + m_1 \ddot{x}_1 + m_2 \ddot{x}_1 = 0$$

Esta es una ecuación diferencial lineal de segundo orden homogena. De aquí se puede despejar $\ddot{x}$

```
#Despejar x1PuntoPunto
x1PuntoPunto = sym.solveset(x1_EL, x1.diff(t, t)).args[0]
```

Methodology — Weekly cycle of our flipped classroom

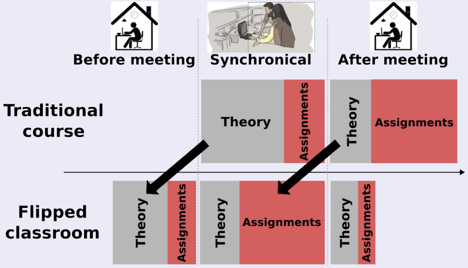
- New theory (Jupyter notebooks and videos) and assignments published on-line



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

Methodology — Weekly cycle of our flipped classroom

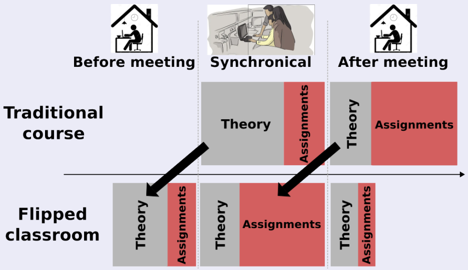
- New theory (Jupyter notebooks and videos) and assignments published on-line
- Previous week assignments **must** be turned-in for scoring



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

Methodology — Weekly cycle of our flipped classroom

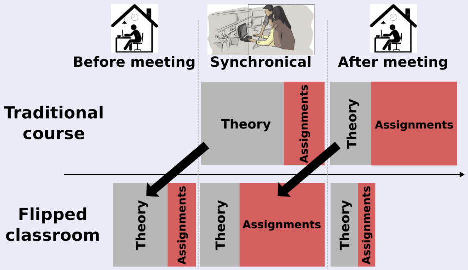
- New theory (Jupyter notebooks and videos) and assignments published on-line
- Previous week assignments **must** be turned-in for scoring
- On-line 24/7 **asynchronic** consultations that are **public** to other students



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

Methodology — Weekly cycle of our flipped classroom

- New theory (Jupyter notebooks and videos) and assignments published on-line
- Previous week assignments **must** be turned-in for scoring
- On-line 24/7 **asynchronic** consultations that are **public** to other students
- **synchronic** meetings with TA's to finish assignments



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

Methodology — Google Colaboratory: asynchronic remote assistance

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
```

$$V = gm_1 \left(-l_{10} - x_1\right) \sin\left(\theta\right) + \frac{k_1 \left(l_{10} + x_1\right)^2}{2} + \frac{k_2 \left(l_{20} + x\right)^2}{2}$$

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
```

$$\mathcal{L} = -gm_1 \left(-l_{10} - x_1\right) \sin\left(\theta\right) - \frac{k_1 \left(l_{10} + x_1\right)^2}{2} - \frac{k_2 \left(l_{20} + x\right)^2}{2} + \frac{\left(m_0 + m_1\right) \left(2 \cos\left(\theta\right) \dot{x} \dot{x}_1 + \dot{x}^2 + \dot{x}_1^2\right)}{2}$$

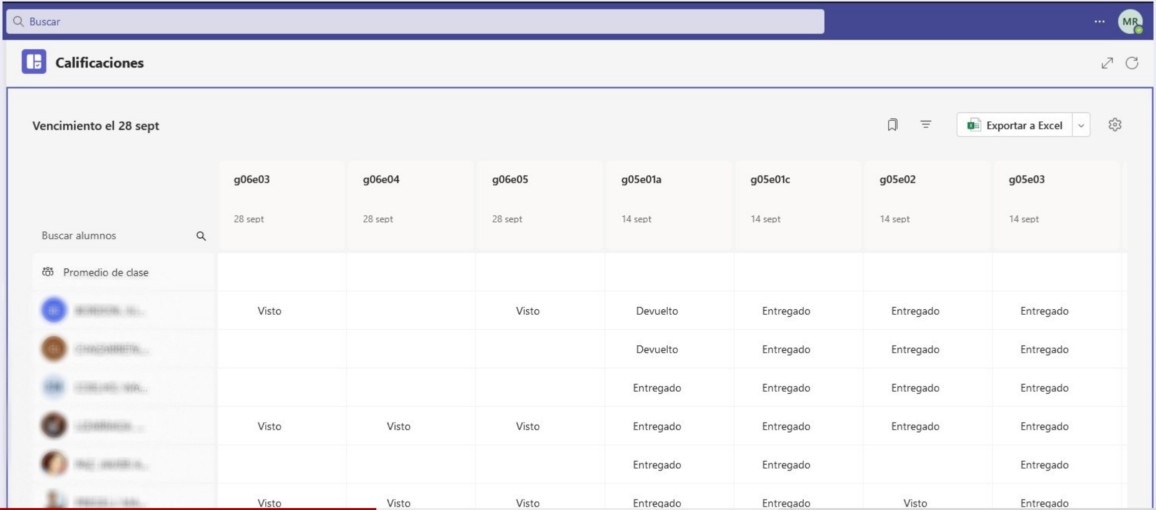
ECUACIONES DE EULER

Para x

Victor Alexis Bettachini Resolver
31 de may. de 2021

Methodology — Microsoft Teams: individualized student follow-up

A record of the weekly turn-up of assignments



Thank You!

Summary

A course centred on code

- Material: text + equations + executable code in digital notebooks in a repository
- Assignments: professor's code recycling
- On-line:
  - Remote collaboration and correction
  - Doesn't require powerful nor at campus computers
  - AI assistance for coding
  - Dated record of each student's work

Flipped classroom

- Theory: emphasis on student's autonomus reading
- Reinforced by: suggested bibliography and short professor's videos
- Consultations: mostly on-line asynchronic and publicly accessible
- Synchronic mettings: TA's personal assistance for completing assignments