"Experiencia de un curso de mecánica racional basado en código"

Bettachini, Víctor A.[a]; Palazzo, Edgardo[b]
a Dep. Ing. e Inv. Tec, Univ. Nac. de La Matanza (DIIT-UNLaM), San Justo, Buenos Aires, Argentina
b UDB Física, Fac. Reg. Avellaneda, Univ. Tec. Nacional (UTN), Villa Domínico, Buenos Aires, Argentina
vbettachini@unlam.edu.ar

Resumen

"Fundamentos de programación" y "Cálculo numérico" se mencionan entre los "descriptores de conocimiento" para un Ingeniero Mecánico en la "Propuesta de Estándares de Segunda Generación para la Acreditación de Carreras de Ingeniería en la República Argentina" aprobado por el Consejo Federal de Decanos de Ingeniería (CONFEDI) en 2018, mejor conocido como "Libro Rojo de CONFEDI". Lamentablemente tras que la teoría y uso de estas herramientas son aprendidos por los alumnos no suelen aprovecharse en profundidad en cursos de años posteriores.

En este trabajo se describe la experiencia que se tuvo en la asignatura Mecánica General del 3.er año de la carrera en la UNLaM. Tradicionalmente los sistemas modelados se limitan a los resolubles analíticamente por trabajar en pizarrón o papel. En este curso, los estudiantes resolvieron sus ejercicios utilizando código en lenguaje Python, haciendo uso de herramientas de este siglo, como bibliotecas de funciones para el cálculo simbólico, numérico, graficación, etc.

Todas las clases se dictaron íntegramente usando cuadernos de Jupyter como plataforma. En estos se intercala código con información gráfica y texto incluyendo una clara notación matemática con simbología LaTeX. Este código es re-utilizable por el estudiante para resolver la ejercitación del curso con la misma herramienta, así como para ser aprovechado en asignaturas futuras y en su vida profesional.

Estos cuadernos se ejecutan sobre software libre. Plataformas web de acceso gratuito a través del navegador permitieron a los estudiantes ejecutarlos en su hogar o trabajo, permitiendo comentar y editar en forma conjunta un mismo cuaderno entre alumnos y/o docentes.

La pandemia nos forzó a enseñar a través de una computadora. Tras un periodo inicial de adaptación los estudiantes reconocieron las virtudes de esta metodología. Inclusive la evaluación fue más enriquecedora que en un curso convencional al alcanzar la complejidad de simular sistemas mecánicos similares a los industriales.

Abstract

"Numerical Analysis" and "Programming Fundamentals'' are mentioned as "Knowledge Descriptors'' for a Mechanical Engineer in the "Proposal of standards for the second generation for engineering degrees accreditation in the Argentine Republic" ("Propuesta de Estándares de Segunda Generación para la Acreditación de Carreras de Ingeniería en la República Argentina") approved by the "Federal council of engineering deans" (Consejo Federal de Decanos de Ingeniería, CONFEDI) in 2018, and best known as "Libro Rojo de CONFEDI". Regrettably after theory and use of these tools are learnt by students they are not fully exploited in courses at later years.

In this work the experience had at the subject Mecánica General (General Mechanics) of the UNLaM's mechanical engineering degree third year is described. Traditionally modeled systems are limited to those analytically solvable working on blackboard or paper. In this course the students solved their problem sets using Python language code, applying this century tools, such as library functions for symbolic and numerical analysis, plotting, etc.

All classes were conducted in full using Jupyter notebooks as a platform. In those notebooks code is interbowen with graphical information and text including clear mathematical notation with LaTeX symbols. Students can re-use this same code to solve course's problem sets as well as in future courses and their professional life.

The notebooks mentioned above run on free software. Students operate on them with free to use web platforms that allow concurrent commenting and editing among them and/or their professor.

The pandemic pushed us all to teach through a computer. After an initial adaptation period the students recognized the virtues of this methodology. Even assessments were more enriching than in a conventional course as it reached the complexity of simulating industrial-like mechanical systems.

***Palabras clave**: Mecánica, Código, Jupyter, Python*

INTRODUCCIÓN

Los últimos tres cuatrimestres la pandemia de *SARS-CoV-2* forzó a que los cursos para los estudiantes de Ingeniería Mecánica en la *Universidad Nacional de La Matanza* (*UNLaM*) se dicten en forma remota. El hecho de que los alumnos estén tras una computadora durante la clase se aprovechó para imponer una metodología pedagógica que obvió los tradicionales soportes pizarrón, papel y presentaciones informáticas no interactivas en favor de presentar conceptos teóricos y ejercitar su uso en actividades en una plataforma informática interactiva basada en código escrito en el lenguaje *Python*.

La complejidad del código se incrementa a medida que se contemplan clase a clase nuevos aspectos que afectan a un sistema mecánico. En un curso basado en pizarrón y papel se repiten cálculos similares en cada nueva actividad mientras que en un curso basado en código este presenta la ventaja de su *reutilización*. Realizando modificaciones al código probado en clases anteriores con sistemas mecánicos simples se expande la capacidad de análisis sin la pérdida de tiempo que insumiría una escritura desde cero del largo conjunto de cálculos que insume un análisis en el esquema de *Euler-Lagrange* de un sistema mecánico más complejo.

LA ASIGNATURA MECÁNICA GENERAL

En el plan de la carrera de grado en *Ingeniería Mecánica* del *Departamento de Ingeniería e Investigaciones Tecnológicas* (*DIIT*) de la *UNLaM* esta asignatura es el nexo entre las primeras propias de la especialidad con las básicas en las que se imparten herramientas de álgebra, análisis matemático, cálculo numérico, y mecánica Newtoniana. El esquema de correlatividades inmediatas a la asignatura *Mecánica General*, que muestra la figura 1, deja en claro que esta debe tener entre sus objetivos el mostrar al alumno cómo dichas herramientas tienen aplicación en su tema de interés.

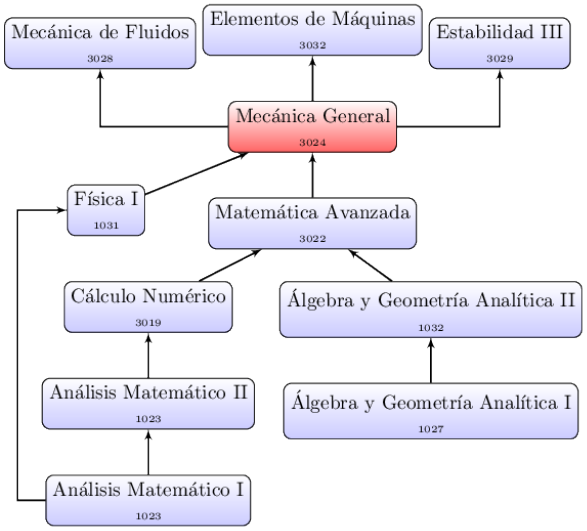

Figura 1: Precedida de asignaturas de álgebra, análisis y física, Mecánica General es la primera en que se aplican tales conocimientos a la ingeniería mecánica.

La asignatura entrena a los alumnos en la habilidad de modelizar la física de sistemas mecánicos simples. Se entiende por modelizar el realizar una serie de procedimientos con los que se construye un esquema simplificado de la física partiendo de una evaluación semi-cuantitativa de las fuerzas y campos que actúan sobre el sistema así como de las ligaduras que limitan sus grados de libertad. Con tal información se priorizan algunas de estas y se descartan otras para arribar al esquema mencionado. Disponer de tal modelo permite:

- elegir coordenadas generalizadas para describir los grados de libertad relevantes,
- escribir relaciones matemáticas entre estas que den cuenta de ligaduras,
- describir las fuerzas generalizadas que no sean efecto de campos (gravitatorio, electromagnéticos, etc.),
- y describir la energía potencial y cinética del sistema en su conjunto.

Tras realizar lo anterior se demuestra y se pone en práctica en el curso el formalismo de *Euler-Lagrange* para obtener un conjunto de ecuaciones diferenciales que describen la dinámica del sistema y/o los esfuerzos mecánicos que cada uno de sus componentes debe soportar en cada instante de tiempo.

De lo expuesto en los párrafos precedentes se evidencia que la temática del curso objeto de este trabajo está circunscrita a la convencional de los cursos de mecánica racional como se detalla en su literatura canónica de referencia [1]. No es en la temática sino en su metodología didáctica donde se hizo una innovación.

El pizarrón, única herramienta didáctica

Tradicionalmente los sistemas que se trabajan en los cursos de mecánica racional son relativamente simples para acotar el tiempo y/o dificultad de los cálculos de análisis matemático y/o de álgebra que requieren los pasos comentados en el párrafo anterior. Pero esta simplificación extrema lleva a un notorio salto en la complejidad de la que requiere modelar de dispositivos mecánicos, la dinámica de fluidos o a estructuras rígidas, las respectivas temáticas de las asignaturas subsiguientes a *Mecánica General* (ver figura 1).

La limitación a la complejidad la impone lo que el docente puede, en la duración de una clase, calcular en el pizarrón. Estos al irse borrando sucesivamente no sólo impide al alumno servirse de referencia de algo que ya no está a la vista sino que además le impone dedicar buena parte de su atención a no cometer errores al transcribir lo allí escrito en su cuaderno. Este soporte en papel a su vez limita la extensión y complejidad de los problemas que pueden proponerse al alumnado para ejercitar lo aprendido.

Lo que sintetiza el párrafo precedente no es otra cosa que el proceder en el dictado de clases de ciencias o ingeniería a nivel universitario que se reproduce casi en forma inalterada desde el siglo XIX hasta nuestros días.

Herramientas didácticas informáticas

Los pasos previos y posteriores al obtener un sistema de ecuaciones diferenciales que describen la dinámica y esfuerzos para un modelo mecánico complejo pueden realizarse con herramientas informáticas, pero que son distintas para cada caso.

Para resolver y analizar el resultado de las ecuaciones se aplican las herramientas aprendidas en la asignatura *Cálculo Numérico*, cursada previamente a *Mecánica General*, y las

ubicuas de graficación para visualizar la evolución temporal de distintas magnitudes. Si bien es cierto que el cálculo numérico se aprovecha ocasionalmente en la ejercitación [2, 3], este rara vez es utilizado por el docente durante la clase. Se pierde así una oportunidad de ejemplificar mejor y profundizar el análisis del comportamiento de los sistemas modelados.

Pero si es rara la aplicación del cálculo numérico durante las clases lo es aún más el uso de sistemas de álgebra computacional (*Computer Algebra Systems o CAS*, en inglés), que permiten automatizar todos los procedimientos matemáticos que requiere la modelización: desde definir grados de libertad, sistema de coordenadas, campos y fuerzas externas al modelo hasta construir el sistema de ecuaciones diferenciales para la dinámica. El utilizarlos para resolver cálculos de álgebra lineal y análisis matemático permite quitar el énfasis sobre estos y centrar la atención del docente y alumnos en la temática propia de la asignatura.

Pero la realidad cotidiana de nuestras aulas es que durante las clases dichos cálculos se continúan realizando manualmente en el pizarrón o en papel obviando el uso de la informática. Empeñarse en ese proceder en el nivel universitario sería análogo al de privar al alumnado del uso de calculadoras de bolsillo para realizar procedimientos aritméticos aprendidos en el nivel primario. Todo un anacronismo en la tercera década del siglo XXI.

Reciclado del código

Lo precedente puede interpretarse erróneamente como un mero llamado a utilizar la informática como un análogo de la calculadora de bolsillo, supliendo resultados de los cálculos que demanda la resolución de ejercicios en papel. Eso sería una infrautilización de tal recurso en la clase, repitiendo el patrón que sigue la mayor parte de los usuarios de computadoras que obvian en su uso cotidiano un aspecto fundamental de la informática.

La computadora digital se inventó en la *Segunda Guerra Mundial* con el fin de realizar cálculos numéricos que se definían manualmente en cada ejecución operando como una calculadora más rápida y con capacidad de automatizar algunos de los procesos de manipulación numérica. Pero desde la mitad del siglo pasado adquirió la capacidad de operar bajo una serie de instrucciones almacenadas en su memoria sobre cómo procesar información tanto numérica como de otra índole. Tales instrucciones reciben el nombre de programa, y se escriben en un código que respeta la sintaxis de un determinado lenguaje.

Los lenguajes modernos de alto nivel permiten escribir un único código incorporando todos los procedimientos que requiere la resolución y el análisis de un problema de mecánica racional. Esto abarca desde las aproximaciones asumidas para simplificar la física del mismo hasta el análisis con gráficas de su dinámica y esfuerzos mecánicos pasando por todos los cálculos algebraicos y numéricos intermedios.

Partiendo del objetivo de que los alumnos exploten esta herramienta el curso tuvo por metodología el obviar el trabajo en papel y en su lugar desarrollar la habilidad de escribir en un único código el conjunto de operaciones que requiere la resolución de ejercicios. En clase el docente explicó en forma sincrónica ejemplos de códigos que realizan todos los procedimientos requeridos para modelar un sistema mecánico. En cada clase se provee una guía de ejercicios resolubles haciendo pequeñas modificaciones al código provisto por el docente en esa fecha. Sucesivos ejercicios de complejidad creciente requieren pequeñas modificaciones respecto al código con que se resolvió el anterior. Esta *reutilización del código* permite un mejor aprovechamiento del tiempo y esfuerzo del alumno que en resoluciones en papel donde debe repetir procedimientos ya realizados en anteriores oportunidades. Clase a clase el alumnado construye una biblioteca de códigos con capacidades crecientes de análisis [4].

Hay que aclarar que no se forma al alumno en programación para que cree aplicaciones o algoritmos, lo que se denomina *programming* en inglés. Lo que se busca es que puedan *codificar* (por *coding* en inglés) las instrucciones para que la computadora realice tareas específicas, en particular cálculos para la ingeniería mecánica.

Algunos alumnos archivan sus resoluciones de ejercicios resueltas en papel como referencia en

caso de que se presente una problemática similar más adelante en el cursado de la carrera o en la vida profesional. En los hechos esto rara vez sucede y si recurren en el futuro a algún material relacionado a la asignatura es a su bibliografía, que por lo expuesto anteriormente carece de ejemplos adaptables a modelos más complejos que los comúnmente tratados en la asignatura. Por contrapartida un código es fácilmente aplicable al análisis de una problemática profesional análoga a las vistas en el curso. Al figurar en forma explícita las instrucciones para realizar cada paso del procedimiento es sencillo de revisar, expandir y modificar.

HERRAMIENTAS UTILIZADAS

Python, Sympy, Numpy, Scipy y Matplotlib

El lenguaje de programación *Python* está por defecto desprovisto de capacidades de cálculo científico e ingenieril. Esta es una decisión de diseño para hacer que tales funcionalidades deban ser agregadas por bibliotecas especializadas. El efecto de esta decisión es que el desarrollo de las mismas corre por cuenta de usuarios que las aplican en diversos ámbitos del desarrollo científico-tecnológico antes que por profesionales de las informática.

Las funciones del cálculo simbólico las provee la biblioteca *Sympy*. Se aprovecha en particular su módulo *Mechanics* que facilita la generación de ecuaciones para la dinámica de sistemas de cuerpos rígidos con múltiples grados de libertad y en variados sistemas de referencia [5].

Los sistemas de ecuaciones diferenciales se resuelven por métodos numéricos apoyados en las funciones para la manipulación de elementos algebraicos de la biblioteca *Numpy* [6] y de los algoritmos de optimización e integración numérica de *Scipy* [7].

El análisis en ingeniería de resultados numéricos son usualmente interpretados con representaciones gráficas. Esta capacidad la proveen las funciones de la biblioteca *Matplotilb* [8,9].

Cuadernos de Jupyter

El entorno usado en el curso para ejecutar código es la aplicación basada en la web del *Proyecto Jupyter* llamada *JupyterLab* cuyo formato de documento es el *cuaderno* (*notebook*) *Jupyter* [10]. Este alterna secciones independientes denominadas celdas. Las de entrada son de código (en variados lenguajes, *Python* es solo uno de los posibles) o de anotaciones, como muestra la figura 2. Esta última variante de celdas se escriben en el lenguaje de marcado *Markdown* [11] que permite incrustar:

- texto y/o expresiones matemáticas en formato *LaTeX* intercaladas,
- contenido multimedia: enlaces web, imágenes, reproductores de video o sonido.

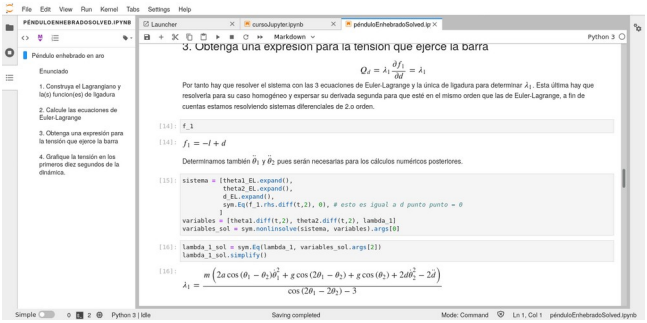

Figura 2: Un cuaderno de Jupyter es un conjunto de celdas. Estas son en formato Markdown o de código ejecutable. Las primeras pueden contener texto, expresiones matemáticas o contenido multimedia. Las segundas líneas de código en variados lenguajes de programación. Intercalando títulos en las celdas Markdown se genera el índice (a la izquierda) que facilita la ubicación dentro del documento.

La utilización de sintaxis *LaTeX* para la simbología matemática provee una notación clara estandarizada bajo los lineamientos de la *American Mathematical Society* [12].

El resultado de la ejecución de una celda de código muestra al usuario el resultado que el mismo instruye a la computadora imprimir. En el curso estas últimas incluyen tanto los comandos para realizar cálculos así como la resolución de un sistema de ecuaciones no lineales que se

imprime en la última celda del cuaderno mostrado en la figura 2.

Ejecución de Jupyter en línea

No se impone a los estudiantes el instalar ningún software para cursar la materia en su dispositivo informático. Solo requieren utilizar un navegador web estándar para utilizar alguno de los servicios que ejecutan cuadernos de *Jupyter* en línea. Este puede tratarse de una instalación de *JupyterHub* del Proyecto Jupyter en servidores propios de la universidad o en nubes comerciales, o en su defecto de alguno de los servicios que ofrecen alternativas incluso gratuitas como, entre otras, *CoCalc, IBM Watson o Google Colaboratory*. De estas se ha utilizado esta última en las últimas ediciones del curso tras cerrar *Microsoft* su servicio gratuito *Azure Notebooks*.

El servicio *Google Colaboratory*, coloquialmente sólo *Colab*, presenta como conveniencia el poder ejecutar cuadernos alojados en un repositorio *Git* gerenciado por el servicio en línea *GitHub*. Basta una modificación en el URL de un cuaderno para que este apunte a un navegador web a ejecutarle en *Colab* [13]. El trabajo con *cuadernos Jupyter* en este servicio puede realizarse en forma concurrente por parte de varios alumnos y/o docentes. También pueden incluirse comentarios cuya actualización es reportada por correo electrónico lo que es útil para la corrección de los ejercicios pues pueden indicarse la ubicación de errores en el código como muestra la figura 3.

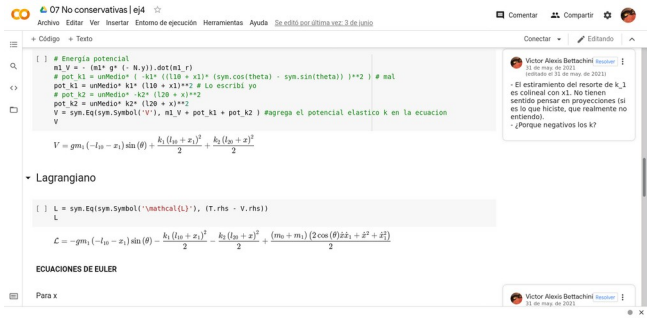

Figura 3: El sitio web Google Colaboratoy permite editar y ejecutar cuadernos Jupyter en forma concurrente entre alumnos y docentes además de incluir comentarios. Esta última característica es útil para las correcciones.

Repositorio Git

El mencionado repositorio en *GitHub* está organizado en sendas carpetas por clase del curso, como muestra la figura 4. Cada una de estas aloja el correspondiente material teórico y ejercicios en el formato de cuadernos *Jupyter* además de guías de ejercicios y algún apunte ocasional en el *formato de documento portátil* conocido por su sigla en inglés *PDF*. Este ordenamiento facilita tanto al docente como a los alumnos una vista de conjunto del material de cada temática así como el verificar las eventuales actualizaciones del mismo. De esta forma el material del curso es de acceso público haciéndolo disponible para ser utilizado a interesados [14] mientras cumplan con citar su origen y no darle uso comercial como indica su licencia *Creative Commons CC-BY-NC-SA* bajo el que está publicado [15].

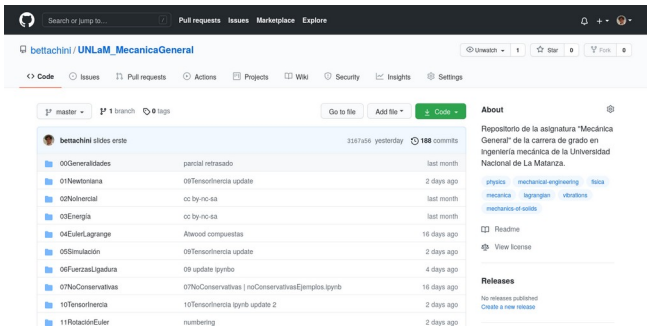

Figura 4: Los alumnos encuentran el material ordenado en sendos directorios por clase.

Sistema de gestión de aprendizaje

En la UNLaM se utiliza la plataforma de comunicaciones de negocios *Microsoft Teams* para suplir la interacción en el aula con los alumnos con videoconferencias. Luego de terminada cada clase el video de las mismas se guarda en el almacenamiento en línea *Microsoft OneDrive*. Enlaces a estos y a los materiales de la clase alojados en el repositorio *Git* se compartimentan en lo que el sistema llama *canales* respetando la misma numeración y denominación que en el repositorio *Git*. La figura

4 muestra los contenidos que encabezan los desplegados para la décima clase.

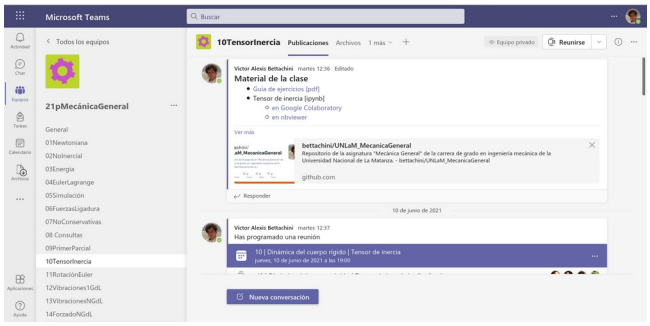

Figura 4: Sendos canales por clase presentan los enlaces a su material.

En cada canal se incluyen enlaces a:

- guía de ejercicios prácticos,
- algún eventual apunte en PDF,
- ambas vías para ver los *cuadernos Jupyter*, la interactiva en *Colab* o estática en *nbviewer, e*
- invitación a la videoconferencia o a su video una vez esta terminó.

Microsoft Teams provee también los rudimentos de un sistema de gestión de aprendizaje (*LMS* por sus siglas en inglés) al permitir asignar tareas a alumnos con fechas límites de aceptación por parte del sistema. Los alumnos pueden cargar al sistema un enlace a su cuaderno en *Colab* o el mismo en formato *.ipynb* en el caso de que no se permita modificación del mismo con posterioridad a una fecha por cuestiones de evaluación.

CRONOLOGÍA DEL CURSO

De los 16 encuentros semanales en línea a lo largo del cuatrimestre en 13 de ellos se presentan nuevos temas. De estos se seleccionaron algunos para ilustrar la progresión del curso.

Clase 1 Repaso de la mecánica Newtoniana, el análisis y el álgebra requeridos para obtener la dinámica del péndulo ideal revisitando las aproximaciones y cálculos vistos en Física 1. Este material de teoría se distribuye tanto en esta como en las subsiguientes clases en un cuaderno *Jupyter*. Se anima a los alumnos a revisar la notación LaTeX con las que el docente escribe fórmulas matemáticas como las que se muestran en la figura 5.

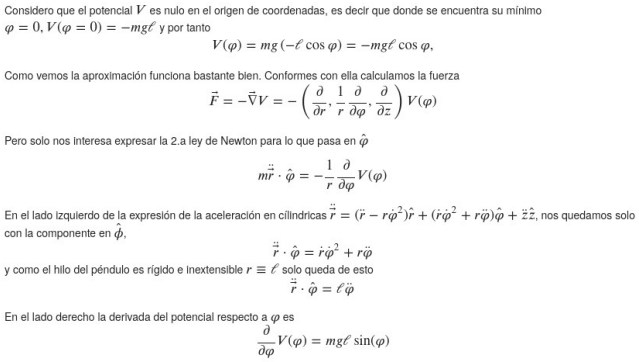

Figura 5: La teoría se presenta en cuadernos Jupyter. Todos las fórmulas matemáticas se expresan en notación estandarizada LaTeX que los alumnos pueden editar o copiar para sus propios fines.

En adición a la reiteración de lo ya visto en cursos anteriores en esta primera clase ya se avanza en el uso de código para analizar resultados. La figura 6 muestra instrucciones para que la biblioteca *Matplotlib* grafique la solución para la dinámica del péndulo ideal.

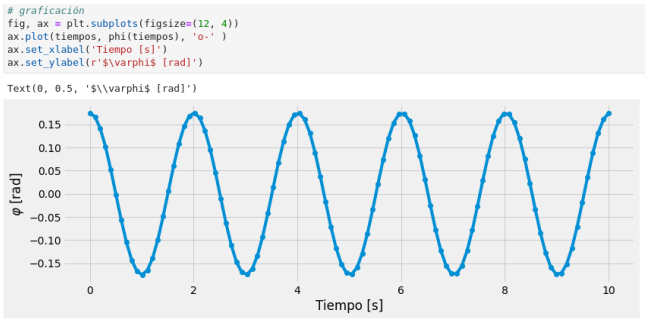

Figura 6: Desde la primera clase se hace explícito a los estudiantes el código utilizado para el análisis de sistemas. Aquí las funciones de Matplotib para graficar la dinámica de un péndulo ideal.

Clase 3 A partir de esta clase se aplica en clase la biblioteca *Sympy* para el cálculo simbólico automático. La figura 7 muestra cómo para calcular la energía cinética de un sistema

con dos coordenadas generalizadas se diferencia en su sistema de referencia.

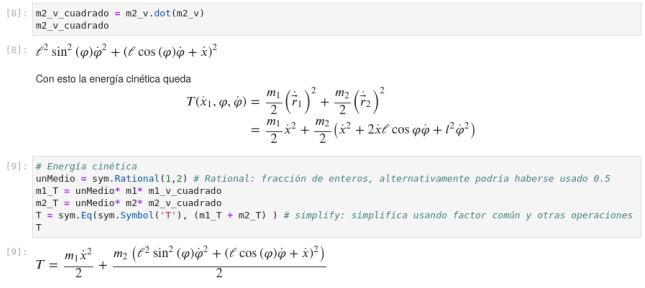

Figura 7: Primeros cálculos simbólicos utilizando la biblioteca SymPy.

Clase 4 Las ecuaciones de *Euler-Lagrange* permiten a los alumnos obtener las ecuaciones que describen la dinámica de un sistema. La figura 8 muestra como funciones de la biblioteca *Sympy* facilitan el obtener tales ecuaciones para un sistema de dos grados de libertad.

Figura 8: Aplicación de funciones de SymPy para general las ecuaciones diferenciales de Euler-Lagrange que describen la dinámica de un sistema.

Hasta aquí se ha utilizado el código para realizar los mismos pasos que en un curso de mecánica racional convencional se resuelven en pizarrón o papel para arribar a ecuaciones diferenciales que solo se resuelven para casos triviales. En contrapartida utilizando *Sympy* se resuelven rápidamente sistemas complejos para aceleraciones en función de coordenadas y velocidades generalizadas como se muestra en la figura 9. Realizar tal tarea manualmente insumiría un tiempo y esfuerzo no despreciable inclusive para este sistema con meros dos grados de libertad.

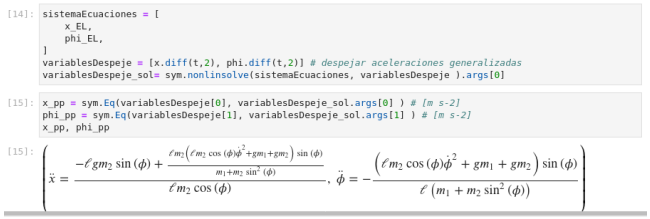

Figura 9: La resolución de sistemas de ecuaciones diferenciales de cierta complejidad se evita en cursos convencionales. En este curso solo insume un par de líneas de código con funciones de la biblioteca Sympy.

Clase 5 Los estudiantes aprobaron un curso de cálculo numérico para poder inscribirse a este curso en el que se hará uso de tales conocimientos. En clase se repasan los fundamentos de los métodos de resolución numérica de ecuaciones diferenciales y cómo se implementarían en una notación de vectores de estado adecuada para un eficiente procesamiento. Tal repaso se presenta a los estudiantes con la misma metodología que para los otros temas, en cuadernos *Jupyter* que los alumnos pueden editar, como se muestra en la figura 10.

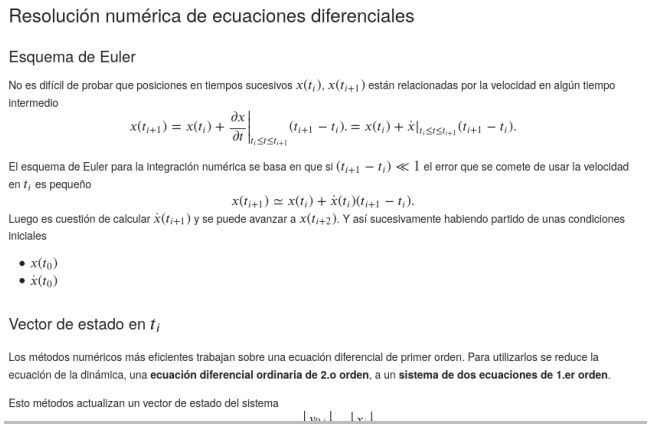

Figura 10: Previo a proceder a la resolución numérica de ecuaciones diferenciales se presenta, en cuadernos de Jupyter, un repaso de sus fundamentos.

Inmediatamente tras el repaso de fundamentos se muestran en acción las funciones de la

biblioteca de cálculo científico *Scipy* para obtener eficientemente las soluciones para la dinámica de un sistema de dos grados de libertad como se ilustra en la figura 11

```
[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,  0.88460859,
              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,
```

Figura 11: Se resuelve numéricamente el sistema de ecuaciones para la dinámica de un sistema de dos grados de libertad con funciones de la biblioteca SciPy.

Las posiciones y velocidades generalizadas en el rango de tiempos de interés obtenidas numéricamente se representan gráficamente. La figura 12 muestra tal representación que sirve para discutir con los alumnos si el comportamiento del sistema se condice con el que puede predecirse de un análisis cualitativo de este sistema simple. El comprobar que las herramientas utilizadas de cálculo simbólico y numérico obtienen resultados correctos confiere confianza en los mismos en vistas de aplicarles a sistemas más complejos.

```
[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')]
```

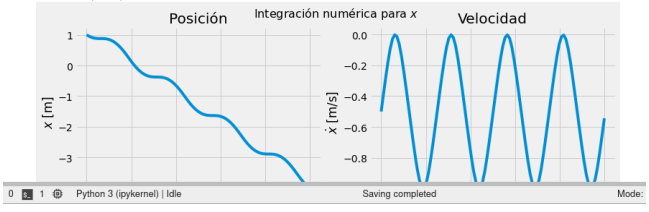

Figura 12: Visualización de resultados obtenidos por cálculo numérico. Corroborando que lo representado corresponde con un análisis cualitativo de la dinámica del sistema se crea confianza en los alumnos en esta herramienta.

Clase 7 Se incorpora a los códigos el análisis de fuerzas no conservativas que a fin de cuentas son la mayoría de las que pueden afectar a un dispositivo mecánico industrial. Como primer ejemplo se extiende la analogía de las oscilaciones del péndulo a un sistema amortiguado que se muestra en la figura 13.

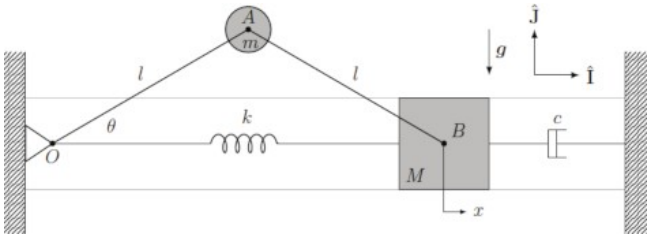

Figura 13: Paulatinamente se extiende el rango de factores analizables. En este sistema actúa un amortiguador lineal con la velocidad. Esta fuerza no conservativa no podía analizarse con el código de clases precedentes.

La figura 14 muestra la gráfica que permite analizar la dinámica calculada con el mismo procedimiento y código que se viene utilizando en las clases precedentes.

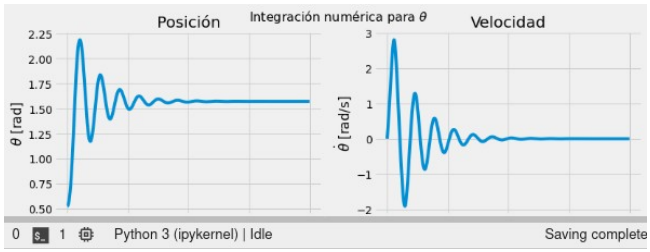

Figura 14: Paulatinamente se extiende el rango de factores analizables. En este sistema actúa un amortiguador lineal con la velocidad. Esta fuerza no conservativa no podía analizarse con el código de clases precedentes.

Siguientes clases El temario habitual de un curso de mecánica racional se va completando centrándose en sistemas extensos analizados en el marco del cuerpo rígido y el análisis de oscilaciones forzadas en sistemas de múltiples grados de libertad. Hacia finales del curso los

alumnos ya han desarrollado la habilidad de analizar en forma autónoma sistemas “realistas” en términos de ser más semejantes a dispositivos mecánicos existentes en la industria. Para plasmar esto último se les propone que calculen los torques que deben realizar los motores de un muy simplificado brazo robótico industrial para que este realice una secuencia de movimientos. Ejemplos del resultado del trabajo de alumnos en respuesta a esta propuesta se muestran en la figura 15

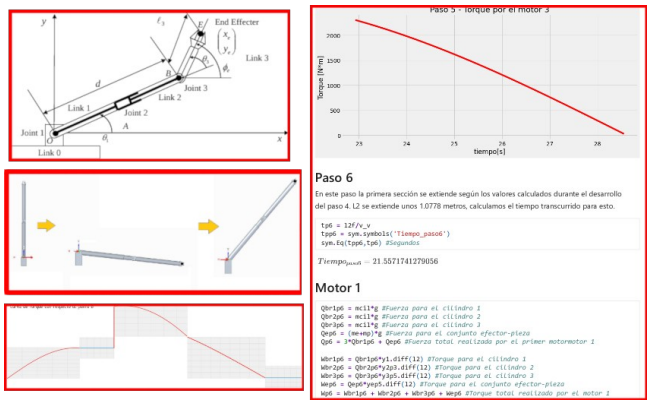

Figura 15: Para que un brazo mecánica industrial realice aún un simple movimiento se requiere que sus motores apliquen una secuencia de torques. Los alumnos realizan el cálculo de los mismos en un trabajo que plasma su dominio de las herramientas analíticas e informáticas cuyo uso fue aprendido en el curso.

EN RESUMEN

La exposición de teoría y la ejercitación práctica tomaron distintas ventajas de la metodología de este curso.

Clases de teoría:

- La exposición en pizarrón o en una presentación en que los alumnos están concentrados en transcribir lo allí escrito se reemplazó por código que pueden reutilizar para resolver sus ejercicios.
- En las clases en línea cada palabra del docente durante la clase queda registrada en video liberando al alumno de la toma de notas.
- El docente puede cambiar el código durante la clase para corregir un error o graficar otro aspecto de la temática.

Ejercicios de práctica:

- En papel una variación sobre un ejercicio resuelto anteriormente obliga a reiterar tediosos cálculos similares. Con código basta con modificar ligeramente el mismo para atender al nuevo caso.
- En forma remota varios alumnos pueden trabajar concurrentemente en la resolución en un mismo ejercicio.
- Los alumnos pueden alertar al docente a toda hora vía el *LMS* de un inconveniente que enfrenten en la resolución de un ejercicio. El docente puede dedicarle tiempo y detenimiento en el momento que encuentre propicio a diferencia del acotado tiempo de consultas del que se dispone en el aula.
- Los docentes pueden comentar y corregir el mismo código sobre el que está trabajando el alumno inclusive en tiempo real.

CONCLUSIONES

El uso de herramientas informáticas durante la clase para la resolución de ejercicios de mecánica racional aliviaría, en especial al alumnado, de tediosos cálculos en pizarrón o papel que distraen del temario propio de la asignatura. El no hacerlo se justificaba por cuestiones de disponibilidad de equipamiento, en particular en las universidades del tercer mundo. La pandemia de SARS-CoV-2 forzó a realizar cursos a través de internet que demostraron donde tales recursos estaban disponibles: en los hogares y lugares de trabajo de alumnos y docentes.

El curso objeto de este trabajo instrumentó una metodología para presentar los conceptos de teoría y su puesta en práctica en ejercicios a través de la escritura de código que capitaliza conocimientos de los alumnos adquiridos en materias previamente cursadas: “Fundamentos de programación” y “Cálculo numérico”. Los términos entre paréntesis son “descriptores de conocimiento” para un Ingeniero Mecánico en el “Libro Rojo de CONFEDI” [16]. No hacer uso de los mismos en materias posteriores a aquellas en las que se aprendieron es un desperdicio del esfuerzo de sus docentes y alumnos. Docentes en universidades reconocidas a nivel mundial también lo han entendido así al iniciar en esta

última década cursos de ingeniería con idénticas metodología y herramientas que las presentadas en este trabajo [4,17,18].

Es el convencimiento de los autores que la metodología de este curso presenta una mayor utilidad al estudiante en vistas a próximas asignaturas y su vida profesional respecto a la que pueden brindar clases presenciales. Las ventajas citadas en este trabajo de un curso basado en código por sobre una cursada presencial les lleva a recomendar la continuidad de esta metodología independientemente de si las condiciones sanitarias vuelven a permitir cursos con la metodología convencional basados en tecnología del siglo XIX.

AGRADECIMIENTOS

Los autores quieren agradecer a la coordinación de la carrera de Ingeniería Mecánica del DIIT-UNLaM qué ha acompañado el desarrollo e implementación de este curso.

REFERENCIAS

[1] Landau, L.D.; Lifshitz, E.M. (1994) *Mecánica*. Reverté. Ciudad de México, México.

[2] Mirass, A; Raichman, S.; Totter, E. (2014) *Articulación de estratégias y recursos para el aprendizaje de métodos numéricos en ingeniería.* XXI Congreso sobre Métodos Numéricos y sus Aplicaciones, Bariloche, Argentina. Mecánica Computacional Vol XXXIII, págs. 2099-2109 http://venus.ceride.gov.ar/ojs/index.php/mc/article/viewFile/4809/4740

[3] Caligaris, M.G.; Rodríguez, G.; Favieri, A.; Laugero, L. (2019). *Desarrollo de habilidades matemáticas durante la resolución numérica de problemas de valor inicial usando recursos tecnológicos*. Educación en ingeniería, 14 (27), 30-40. https://doi.org/10.26507/rei.v14n27.928

[4] Barba, L.A.;Forsyth, G.F. (2018) *CFD Python: the 12 steps to Navier–Stokes equations.* Journal of Open Source Education, 1(9), 21, *https://doi.org/10.21105/jose.00021*

[5] Sympy Development Team (2021) *Classical Mechanics*, https://docs.sympy.org/latest/modules/physics/mechanics/index.html

[6] NumPy Steering Council (2019) *Some information about the NumPy project and community,* https://numpy.org/about/

[7] SciPy Developers (2021) *Scientific computing tools for Python,* https://www.scipy.org/about.html

[8] Hunter, J. D.; *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[9] The Matplotlib development team (2021) *Matplotlib: Visualization with Python*, https://matplotlib.org/

[10] Project Jupyter (2018) *JupyterLab Documentation*, https://jupyterlab.readthedocs.io/en/latest/

[11] Gruber, J. (2021) *Daring Fireball, https://daringfireball.net/projects/markdown/*

[12] Letourneau, M; Wright Sharp, J. (2017) *AMS Style Guide*. American Mathematical Society. Providence, Rhode Island, E.U.A. https://www.ams.org/publications/authors/AMS-StyleGuide-online.pdf

[13] Google LLC (2021) *Using Google Colab with GitHub*, https://colab.research.google.com/github/googlecolab/colabtools/blob/master/notebooks/colab-github-demo.ipynb

[14] Bettachini, V.A. (2021) *Repositorio de la asignatura Mecánica General, https://github.com/bettachini/UNLaM_MecanicaGeneral/*

[15] Creative Commons (2021) *Atribución-NoComercial-CompartirIgual 4.0 Internacional,* https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es

[16] Asamblea del Consejo Federal de Decanos de Ingeniería de la República Argentina (2018) *Propuesta de estándares de segunda generación para la acreditación de carreras de ingeniería en la República Argentina, https://www.ing.unlp.edu.ar/sitio/institucional/difusion/archivos/*

LIBRO_ROJO_DE_CONFEDI_estandares_de_segunda_generacion.pdf