



TRABAJO FIN DE GRADO

DESARROLLO DE HERRAMIENTAS DE SOPORTE A UN CURSO OCW- UPM SOBRE RESOLUCIÓN DE ECUACIONES DIFERENCIALES MEDIANTE SIMETRÍAS DE LIE

TRABAJO FIN DE GRADO PARA
LA OBTENCIÓN DEL TÍTULO DE
GRADUADO EN INGENIERÍA EN
TECNOLOGÍAS INDUSTRIALES

OCTUBRE 2020

Javier Díaz Mena

DIRECTOR DEL TRABAJO FIN DE GRADO:
Manuel Mendoza de Haro

Agradecimientos

Con este trabajo finaliza una larga etapa de mi vida. Mi paso por esta universidad ha sido un camino muy difícil en algunos aspectos, pero también maravilloso en muchos otros. Para mí ha supuesto un despertar como amigo, como pareja, como profesional y, sobre todo, como persona. A lo largo de estos 8 años he conocido a muchísima gente, y creo que es justo mencionar a algunos de los que han hecho posible que yo llegue hasta donde estoy hoy.

Quiero agradecer en primer lugar a mis padres, Juana y José, su paciencia. Han confiado en mí poniendo todo su esfuerzo y esperanza en que termine mis estudios, sabiendo que eso me llevará a buen puerto. Os quiero mucho.

A María, mi hermana, por esos momentos de carcajada absoluta que no he vivido con nadie más, y también por los mejores consejos de la vida en los momentos que más los he necesitado.

A Manuel, por todos tus intentos de engancharme al trabajo hasta que lo has conseguido. Tu paciencia y energía constantes son dignas de admiración.

A Alicia, con quien me siento muy identificado en todas las situaciones en la Escuela, y con quien he podido contar siempre a la primera llamada.

A Liza, quien nos ha mantenido unidos a todos los "12miles", y siempre nos tiene en mente. También a Burgos, amigo y referente a seguir, por confiar en mí y escucharme desde que nos conocemos.

A Natalia, amiga y una de las personas más auténticas que conozco, y alguien que siempre se ha preocupado por mí. Vas a llegar muy alto.

A Diego, que te has convertido en cuestión de meses en un pilar de confianza y una persona capaz de sacarme una sonrisa cuando todo parecía perdido. Luego nos damos una vueltecilla.

Y por último a Carlos, quien a pesar de encontrarse aún muy lejos de este mundo, es la única persona que ha sabido darme la motivación necesaria para arrancar todos mis *sprints* y coger siempre el camino más largo.

Para ti, lector y alumno de la Escuela. Pasarás momentos muy duros e injustos en este lugar. Pero debes saber que eres un gran estudiante y lo vas a conseguir. Tu futuro te espera sonriendo.

Muchas gracias a todos.

Javi

Índice

Agradecimientos	2
1. Introducción	6
2. Resumen	8
Enlaces	10
3. Objetivos	11
4. Fundamentos teóricos.....	14
Primeras nociones	14
Qué es una simetría	14
Qué es un grupo	14
Grupos continuos uniparamétricos de Lie	15
Resolución de EDOs mediante simetrías de Lie	16
Transformaciones Infinitesimales.....	16
Problema inverso. Determinación del Grupo Continuo Uniparamétrico asociado a un generador infinitesimal	18
Determinación de las simetrías admitidas por una EDO	21
Resolución mediante el uso de Simetrías de Lie.....	29
5. Librerías Python: Tkinter y Sympy	32
Implementación y uso de las librerías	32
Librería Tkinter	32
Necesidad de una interfaz gráfica	32
Alternativas a <i>Tkinter</i>	33
Descripción de la librería	34
Distribución de los elementos o <i>Layout</i>	37
Librería Sympy	39
Clase <i>Eq(rhs, lhs)</i>	39
Función <i>dsolve</i>	40
Función <i>classify_ode</i>	40
Función <i>checkodesol</i>	41
Función <i>infinitesimals</i>	41
Función <i>checkinfsol</i>	42
Función <i>sympify</i>	42

Desarrollo de herramientas de soporte a un curso OCW-UPM sobre
resolución de ecuaciones diferenciales mediante simetrías de Lie

Hints de <i>dsolve</i>	42
Heuristics de Lie	45
Benchmarking de Sympy. Kamke 88	47
Test 1 - Capacidad de resolución de la librería al completo	47
Test 2 - Importancia de las heurísticas de Lie	51
Test 3 – Comparación del método de Lie con el resto de procedimientos	55
Ecuaciones exclusivas del método de Lie	57
Ecuaciones exclusivas reducibles a cuadraturas	59
Conclusión	59
6. Contenidos docentes	60
Criterios de diseño	60
Estructura modular	61
Sesiones y actividades	62
Contenidos desarrollados	63
Conclusión	65
7. Herramientas de soporte al curso	66
Espacio Moodle	66
Herramienta DELHI	69
Conclusión	75
8. Viabilidad de desarrollo de un MOOC	76
Oportunidad y contexto	76
Formato OCW-UPM	76
Conclusión	79
9. Soporte a un curso MOOC	80
<i>DELHI Web</i> . Herramienta <i>Web</i>	80
ODE Solver	81
Benchmark	81
Conclusión	84
10. Conclusiones y líneas futuras	85
Conclusiones por objetivos	85
Librerías Python	85
Contenidos docentes	85

Herramientas de soporte	86
Viabilidad de un curso MOOC	86
Herramientas web.....	86
Funcionalidades adicionales.....	87
Futuro TFG 1. Benchmark con Maple, Mathematica y Sage.....	88
Futuro TFG 2. Proyecto GNU de mejora de Sympy.....	88
Creación de un proyecto MOOC.....	88
11. Bibliografía	89
Documentación y referencias software.....	89
Otras fuentes.....	89
Bibliografía teórica	89
12. Planificación temporal y presupuesto	91
Planificación temporal.....	91
Presupuesto del proyecto	95
13. Anexo I. Contenidos docentes del Módulo 2	96
14. Anexo II. Guía del usuario DELHI	106
15. Anexo III. Guía de referencia DELHI	120
Estructura del proyecto.....	120
Descripción de los módulos.....	121
Módulo delhiSymbols.py	121
Módulo main.py	121
Módulo Content.py	121
Módulo benchmarkv5.py	123
16. Anexo IV. Tablas completas de resultados	125
17. Anexo V. Glosario	132
Siglas	132
Términos	132
18. Anexo VI. Índice de figuras	133
19. Anexo VII. Índice de tablas	134

1.Introducción

El documento actual describe el desarrollo del Trabajo de Fin de Grado del alumno Javier Díaz Mena.

La evolución constante de los contenidos docentes que se imparten en las universidades genera cierta variabilidad en cuanto a cursos y fuentes de referencia. En concreto, tratando aquí la materia de las ecuaciones diferenciales, históricamente se han ido descubriendo nuevos problemas que, modelados con sistemas de ecuaciones, suponían cada vez un mayor reto en su resolución.

El gran espectro de métodos *ad-hoc* presentes actualmente hace ver las ecuaciones diferenciales como un conjunto de clasificaciones sin relación, en el que cada una posee un procedimiento asociado válido sólo para dicha categoría.

Entra en juego en este contexto el método de simetrías de Lie, que, a priori, parece presentarse como otra “receta” más a la hora de resolver ecuaciones diferenciales. En este trabajo se va a tratar con dicho método para averiguar si es capaz de arrojar algo de luz sobre el establecido canon de procedimientos de resolución en esta materia.

Paralelamente al estudio de las simetrías de Lie, se plantea la meta de desarrollar un curso oficial que ofrezca contenido suficiente tanto a nivel teórico como práctico sobre el mencionado método, y que en un futuro sirva de referencia en su temática, siendo de acceso libre y gratuito, enfocado especialmente al mundo universitario.

Finalmente, se describe el proceso de desarrollo de las herramientas de soporte ideadas para el curso docente, que consisten en una aplicación de escritorio codificada en el lenguaje Python, y un curso activo en Moodle UPM. Se plantean sus características actuales, puntos fuertes y también sus posibles vías de continuación.

Preámbulo: Notas del director del trabajo

Exige experiencia docente y conocimientos de los métodos de resolución de ecuaciones diferenciales que van más allá de los conocimientos del autor de este trabajo y de la de cualquier alumno, salvo muy infrecuentes excepciones, que cursa cuarto curso del Grado en Ingeniería en Tecnologías Industriales en la UPM, por lo que deben ser atribuidos al director de este trabajo que, además, en cualquier caso, debe considerarse responsable último, en su calidad de director, de la calidad final de este trabajo en su conjunto.

Esta responsabilidad adicional del director del trabajo, como especialista docente en la materia, es especialmente reseñable en los apartados que detallan los contenidos docentes diseñados para un curso completo. En cualquier caso, tanto en estos objetivos como en el resto de este trabajo, la participación del autor de este trabajo ha sido activa, aportando sus críticas, comentarios, y visión propia. En todos los alcances de este trabajo se han articulado los medios para aumentar la comprensión del alumno de todos los conceptos tratados, independientemente de su experiencia y conocimientos previos.

Por el contrario, en los objetivos relativos al desarrollo de herramientas informáticas de soporte, el autor de este trabajo ha sido, dados sus conocimientos y experiencia previa, casi totalmente autónomo, limitándose, en ese caso, el director del trabajo a contribuir con sus comentarios, sugerencias y posibles mejoras siguiendo la evolución del proyecto.

Se integran finalmente algunos contenidos que no se han desarrollado completamente por situarse más allá del ámbito de desarrollo de este trabajo de fin de grado, mención especial para la versión web del software, que se mantiene como germen de una línea de expansión del mismo.

2. Resumen

El presente Trabajo de Fin de Grado (en adelante TFG o trabajo) se encarga de introducir en la docencia el **método de simetrías de Lie** para la resolución de ecuaciones diferenciales ordinarias. Al ser un método poco utilizado y relativamente desconocido en el entorno académico actual, se ha decidido dividir el proyecto en diferentes módulos que lo hagan más atractivo y funcional de cara al público docente, tanto a los alumnos como a los profesionales de la enseñanza que deseen incluirlo en sus lecciones.

Este método supone a las ecuaciones diferenciales lo que el método de Galois a las ecuaciones algebraicas, consistente en aprovechar la existencia de simetrías inherentes a estas ecuaciones para simplificar su resolución transformándolas en otras resolubles. Estas simetrías son en ocasiones apreciables y representables gráficamente, dotando al método de una belleza intrínseca.

Además, otra gran ventaja es que proporciona un enfoque abstracto y conceptual que va más allá de la colección de recetas y/o métodos de resolución de las ecuaciones. Estos procedimientos han estado fuertemente influenciados por los problemas planteados y resueltos en la evolución histórica de la materia durante el siglo XVII y la primera mitad del XVIII, que han llegado hasta nuestros días dejando un panorama disperso y no cohesionado de métodos aislados para resolver ciertos tipos de ecuaciones. Este problema se soluciona en parte con el método de Lie, pues muchos de los problemas con procedimiento propio pueden descomponerse a través de las simetrías. Así, se convierte en un método **más general y potente**, presentando una alternativa a las recetas exclusivas tradicionales.

Para demostrar su gran potencial, el primer capítulo introduce los instrumentos principales utilizado durante todo el trabajo. Se trata de dos **librerías del lenguaje de programación Python**, con las cuales se ha realizado una serie de test a una batería de 88 ecuaciones, conocida como conjunto *Kamke*, en referencia al autor del libro del cual se han extraído. El objetivo de estos experimentos es analizar la calidad de la implementación del método de Lie en estas librerías. Los resultados obtenidos son sorprendentemente buenos, consiguiendo resolver **22 de las ecuaciones con éxito (un 25%)**, contando además con la presencia de **5 ecuaciones exclusivas** que no podrían ser resueltas sin utilizar sus simetrías. Si además añadimos aquellas ecuaciones que se han resuelto reduciéndolas a **cuadraturas**, el porcentaje **asciende a un 62% (58 ecuaciones)**. **Para 28 de ellas**, el método de Lie **es el único** que es capaz de hallar la solución explícita o reducida a cuadraturas.

Se proporciona en el segundo capítulo unos **contenidos docentes desarrollados** con el enfoque de convertirse en un **curso OpenCourseWare** (OCW en adelante) según la normativa de la Universidad Politécnica de Madrid. Este tipo de cursos online se caracterizan por cumplir una serie de requisitos comunes en cuanto a forma y contenido, que los hacen fácilmente accesibles por todo aquel que desee cursarlos, siendo además de manera gratuita. Los contenidos cubren desde algunos aspectos teóricos básicos en

materia de ecuaciones diferenciales, hasta la explicación del propio método de Lie para ilustración del lector.

En posteriores capítulos se describen las **herramientas informáticas de soporte** que se han diseñado y desarrollado específicamente para dicho curso. Se pretende dar una primera forma a los contenidos docentes a impartir incluyéndolos en un curso oficial del **espacio Moodle** de la Universidad Politécnica de Madrid para darle mayor visibilidad y facilidad de acceso.

A su lado, porta el estandarte principal la **aplicación DELHI**, programada utilizando las librerías Python mencionadas. Consta de diferentes apartados que permiten al usuario experimentar con el método de las simetrías con aproximaciones diferentes para ayudarle a completar su formación y poner en práctica los conocimientos descritos en los capítulos teóricos.

Por último, este trabajo incluye en su capítulo final una versión preliminar de las **herramientas web** de soporte, que sustituyan en un futuro a la aplicación de escritorio que se ha diseñado. Al tener los contenidos docentes la meta de convertirse en un curso OCW oficial, se concibe como necesaria la adaptación a una versión web de la aplicación DELHI. Esto permite un acceso libre desde cualquier parte, sin necesidad de contar con acceso a Moodle ni descargar una aplicación. Sin embargo, tanto por razones de alcance como por limitaciones técnicas de la funcionalidad web, se ha decidido apostar por un desarrollo más completo en DELHI que muestre todo su potencial, quedando como línea de expansión su adaptación web. **Puede accederse a las versiones beta con el usuario y contraseña indicados en el capítulo 9.**

Por todo lo expuesto, se ha de destacar el deseo de sacar a colación las simetrías de Lie e incluirlas en la docencia de los cursos actuales, pues su aprendizaje ha pasado a considerarse muy recomendable a nivel universitario.

Enlaces

En esta sección puede accederse a los enlaces de dos vídeos sobre el presente trabajo mediante códigos QR, alojados en la página web del Departamento de Matemática Aplicada a la Ingeniería Industrial de la ETSII UPM.

Vídeo Resumen



http://dmaii.etsii.upm.es/mmendoza/TFGs/2020_JDM/video_resumen.html

Vídeo Presentación



http://dmaii.etsii.upm.es/mmendoza/TFGs/2020_JDM/video_presentacion.html

3.Objetivos

El presente trabajo tiene un **carácter exploratorio**. Se han explorado las posibilidades del método de resolución de ecuaciones diferenciales mediante simetrías de Lie en el ámbito docente. La exploración se ha realizado en varias direcciones:

1. **Librerías de Python**. Se ha analizado la **implementación** que de este método hace la librería **Sympy**, codificada en el lenguaje de programación Python. Asimismo, se ha analizado la potencia de esta librería en la resolución de una batería de ecuaciones diferenciales ordinarias —EDO's, en inglés ODEs— que se utilizada habitualmente para comparar distintos paquetes de cálculo simbólico. El conjunto de EDO's se basa en el libro [17] de Erich Kamke.

Adicionalmente, con la meta de poner a disposición del público toda esta potencia de cálculo, se ha utilizado la librería *Tkinter* para elaborar una interfaz gráfica que permite al usuario experimentar con los métodos de *Sympy* mediante tres aproximaciones diferentes.

2. **Diseño de contenidos docentes** que puedan ser susceptibles de incluirse en las asignaturas de Matemáticas que se imparten en los diferentes niveles formativos (grado, máster, doctorado) universitarios. Este diseño se ha orientado, particularmente, a las asignaturas impartidas en la actualidad, o que puedan impartirse en el futuro, en la Universidad Politécnica de Madrid, UPM.

Este objetivo, dada su naturaleza, ha sido liderado y desarrollado, principalmente, por el director de este trabajo. En consecuencia, **la aportación del alumno autor del trabajo a este objetivo ha sido parcial** y ha consistido en analizar y comprender estos contenidos, complementarlos dentro de sus posibilidades, elaborarlos desde un punto de vista administrativo, mejorarlos desde un punto de vista pedagógico aportando su perspectiva como alumno.

Es de reseñar que los contenidos docentes objeto del presente trabajo, es decir la resolución de ecuaciones diferenciales ordinarias mediante simetrías de Lie están presentes en la oferta docente de la Universidades más prestigiosas a nivel internacional dentro del ámbito de la Ingeniería Industrial como el MIT y CALTECH.

3. **Herramientas de soporte** para asignaturas presenciales dedicadas en exclusividad a la resolución de EDO's mediante simetrías de Lie, o a unidades didácticas dedicadas a esta temática, dentro de asignaturas con un contenido más amplio.

Dado que el contexto en el que se desarrolla este trabajo es la UPM, lugar donde podría en un primer lugar utilizarse el material docente desarrollado, se ha orientado su desarrollo a las herramientas de soporte existentes en esta

universidad. En concreto, se decidió integrar el material desarrollado en un **espacio Moodle**, por ser esta la herramienta utilizada en la UPM.

Asimismo, dentro del alcance de este trabajo, se planteó como objetivo el desarrollo de una **aplicación standalone**, es decir, ejecutable sin necesidad de acceso a Internet, que pudiera ser instalada localmente por los alumnos de la UPM y que sirviera de soporte al aprendizaje y/o al desarrollo de prácticas para los contenidos docentes desarrollados.

Este objetivo se enmarca dentro de la dinámica actualmente en marcha en el *Departamento de Matemática Aplicada a la Ingeniería Industrial* de la UPM. Se está ampliando la oferta docente de este departamento con la creación de una **especialidad en Matemáticas** en el grado de GITI (Grado en Ingeniería de Tecnologías Industriales) de la Escuela Técnica Superior de Ingenieros Industriales. En los próximos cursos, y como consecuencia de la creación de esta especialidad en grado, se van a ampliar los contenidos ofertados por el Departamento de Matemática Aplicada en el **Master de Ingeniería Industrial (MII)** de esta Escuela.

Por otro lado, la UPM ha aprobado **la creación de un Grado en Matemáticas (GeM)** en esta universidad. Se prevé que, si este grado es aprobado por los órganos competentes de la Comunidad de Madrid, pudiera pasar a impartirse tan pronto como en el curso 2021-22. En la definición de dicho grado, se han previsto impartir gran cantidad de asignaturas opcionales, entre las que tendría cabida la resolución de EDO's mediante simetrías de Lie con el alcance definido el presente trabajo de fin de carrera.

4. **Analizar la viabilidad, a medio plazo, de la creación de un curso MOOC [8]**

(Curso *En-línea Masivo y Abierto*, en terminología inglesa *Massive Open Online Course*) dedicado a la resolución de ecuaciones diferenciales ordinarias, y eventualmente en derivadas parciales, mediante simetrías de Lie dentro de la plataforma de la UPM.

Estos MOOC tuvieron su precedente en la iniciativa OpenCourseWare (OCW) lanzada por el Instituto de Tecnología de Massachusetts (MIT) en 2001. A la fecha, el MIT oferta unos 2.400 cursos de temática muy variada y tiene 500 millones de visitantes. La iniciativa OCW del MIT dio origen posteriormente, con la participación de otras instituciones, al OpenCourseWare Consortium.

Dentro de los cursos MIT-OCW, su Departamento de Matemáticas oferta 164 MOOC, de ellos se ofertan 11 sobre ecuaciones diferenciales. En varios de ellos se trata la resolución de ecuaciones mediante simetrías de Lie. Adicionalmente, se ofertan dos cursos sobre grupos y álgebras de Lie, pero no específicamente orientados a la resolución de ecuaciones diferenciales.

La UPM lanzó una iniciativa OCW. En la plataforma UPM-OCW, ya se ofertan más de 100 cursos, en concreto 108 a la fecha. De ellos 8 en el ámbito de las

Matemáticas: Matemática Aplicada (3), Álgebra (4) y Análisis Matemático (1). Dentro de los de Matemática Aplicada, uno tiene por objeto las Ecuaciones Diferenciales, aunque **no abarca su resolución mediante simetrías de Lie**.

La plataforma UPM-OCW establece, sin lugar a dudas, un entorno contrastado y muy útil para el desarrollo de MOOC, si bien, existe una clara **oportunidad para ampliar los contenidos ofertados**. Adicionalmente, la transición abrupta a la docencia online motivada por la irrupción de la actual pandemia ha reforzado la tendencia, ya existente, a la oferta de este tipo de servicios por la UPM y el resto de universidades a nivel mundial.

Dentro de este trabajo de fin de grado, se plantea como objetivo **analizar los requerimientos de la plataforma UPM-OCW para el desarrollo de MOOC**. De tal forma, los contenidos docentes planificados y/o desarrollados se ajusten a estos requerimientos y se facilite, en un futuro, su inclusión en un MOOC dedicado a la resolución de ecuaciones diferenciales mediante simetrías de Lie.

5. **Herramientas Web de soporte para MOOC**. Por su naturaleza, el material de soporte para MOOC no puede estar alojado en recursos propios de la UPM, como Moodle.

Las herramientas informáticas que dan soporte a la realización de prácticas o resolución de ejercicios en MOOC son, de forma natural, **herramientas Web**.

Se plantea como objetivo analizar la viabilidad de tales herramientas y desarrollar, como prueba de concepto, alguna de ellas.

Por el carácter exploratorio del presente trabajo, y dado que el desarrollo completo de estos objetivos excede con mucho el alcance de un trabajo fin de grado y, para algunos de ellos, las capacidades docentes del autor de este trabajo; **estos objetivos se han desarrollado parcialmente**. Se ha pretendido que los logros a obtener en el desarrollo de cada objetivo sean acordes a su posición en la enumeración realizada. De esta forma, debe entenderse una **prelación** en los objetivos planteados, de tal forma que los objetivos enumerados en primer lugar son más prioritarios y su consecución puede alcanzarse a más corto plazo, por lo que se ha dedicado más esfuerzo en el presente trabajo a su consecución.

Los objetivos se han definido con cierta amplitud y los avances alcanzados en el presente trabajo deben entenderse como una “prueba de concepto”. Estos desarrollos tendrían que ampliarse una vez se aprobaran, por los órganos competentes, objetivos más concretos, como la inclusión de los contenidos objeto del presente trabajo en asignaturas oficiales de la UPM.

De todo lo anterior, se deduce que el presente trabajo puede dar lugar, sin duda, a **trabajos posteriores** en los que se avance en los ambiciosos objetivos planteados.

4. Fundamentos teóricos

Primeras nociones

Qué es una simetría

El primer pensamiento que puede venir al reflexionar en la idea de simetría se relaciona con la geometría. Tradicionalmente, se ha asociado este concepto con una propiedad de los cuerpos geométricos. Se dice de estos cuerpos que poseen ciertos ejes o puntos sobre los cuales pueden plegarse o reflejarse, quedando iguales al otro lado de los mismos. Este caso es conocido como simetría especular.

Esta primera definición es simplemente la punta del iceberg. Siguiendo con las figuras geométricas, una simetría también puede expresarse como cualquier clase de acción o movimiento que se puede efectuar sobre el cuerpo y cuyo resultado sea dejarlo en el mismo estado, o con el mismo aspecto.

De aquí, se deduce que una simetría es en realidad, citando a [18] *“algo activo, no es una propiedad pasiva y estática de los cuerpos”*. Entrando en el terreno de las matemáticas, las simetrías son un tipo de invarianza. Una simetría consiste en una transformación u operación (o un conjunto de ellas) que deja invariante al objeto matemático en cuestión. Finalmente puede extenderse hasta definirlo como una aplicación del objeto sobre sí mismo tras la cual conserva su estructura.

Este interesante punto de vista sobre las simetrías ayuda a dar una vuelta de tuerca al concepto, y de esta manera comprender las próximas definiciones para finalmente embocar con la descripción completa del método de Lie.

Qué es un grupo

En el campo del álgebra un grupo es una estructura que reúne las siguientes características.

- Está formado por un **conjunto no vacío** de elementos y **una operación interna** entre ellos.
- Esta operación cumple tres propiedades: **asociativa**, la existencia del **elemento neutro** y del **inverso**.

Lo más importante a señalar es que cumplen la llamada “Ley de composición interna” u operación binaria, esto es, que una operación interna al grupo que combine dos o más elementos del mismo, siempre da lugar a otro elemento, que igualmente pertenece al conjunto.

Grupos continuos uniparamétricos de Lie

En tercer lugar, se van a definir los grupos necesarios para la resolución de ecuaciones diferenciales mediante el método de simetrías.

Se trata de los grupos continuos de Lie. Estas estructuras se caracterizan por contener un número infinito de elementos, que en este caso son transformaciones. En el caso de los grupos continuos de Lie, estas transformaciones han de ser continuas. Esto quiere decir que inducen movimientos tan pequeños que se consideran infinitesimales, de ahí que se los denomine continuos. En concreto, para el contexto en el que se enlaza este trabajo, se van a utilizar los grupos continuos de Lie que solo dependen de un parámetro canónico λ . Este aspecto es clave y se desarrolla su utilidad en los próximos apartados.

Resolución de EDOs mediante simetrías de Lie

En el desarrollo de este apartado se han seguido, principalmente, las referencias [15] , [19], [20] y [21].

Transformaciones Infinitesimales

Dado G Grupo Continuo Uniparamétrico de Transformaciones en \mathbb{R}^2 , con parámetro canónico¹ λ , dado por:

$$\begin{cases} \overline{x_1} = f(x, y, \lambda) \\ \overline{y_1} = g(x, y, \lambda) \end{cases} \Leftrightarrow X_1 = F(X, \lambda)$$

Cada transformación puede aproximarse por:

$$\begin{cases} \overline{x_1} \approx x + \left(\frac{\partial f}{\partial \lambda} \right)_{\lambda=0} * \lambda \\ \overline{y_1} \approx y + \left(\frac{\partial g}{\partial \lambda} \right)_{\lambda=0} * \lambda \end{cases}$$

Se denomina **Transformación Infinitesimal** a la definida por estos segundos miembros. Aproxima localmente el grupo de transformaciones G .

A los coeficientes de λ en las ecuaciones de la transformación infinitesimal se les ha denominado históricamente en la bibliografía ξ y η . Son funciones de x e y .

$$\begin{cases} \xi(x, y) = \left. \frac{\partial f}{\partial \lambda} \right|_{\lambda=0} \\ \eta(x, y) = \left. \frac{\partial g}{\partial \lambda} \right|_{\lambda=0} \end{cases}$$

Por lo tanto, en función de ξ y η , la transformación infinitesimal queda definida por:

$$\begin{cases} x_1 = x + \xi(x, y) * \lambda \\ y_1 = y + \eta(x, y) * \lambda \end{cases}$$

¹ El parámetro λ de un Grupo de Transformaciones Uniparamétrico G se denomina canónico cuando la composición de transformaciones tiene como ley de composición la suma de parámetros, es decir, $F[F(X, \lambda_1), \lambda_2] = F(X, \lambda_1 + \lambda_2)$. En ese caso, se verifica que $F(X, 0) = X$, es decir, la transformación identidad corresponde a $\lambda=0$. Véase [19] apartado 7.1.3. Siempre es posible determinar parametrizar un grupo de transformaciones G mediante un parámetro canónico Véase [19] apartado 7.1.6.

Ejemplo. Grupo de **rotaciones** en \mathbb{R}^2 . Transformación Infinitesimal

El grupo continuo uniparamétrico de rotaciones en \mathbb{R}^2 , queda definido por las ecuaciones:

$$\begin{cases} \bar{x}_1 = x \cos(\lambda) - y \operatorname{sen}(\lambda) \\ \bar{y}_1 = x \operatorname{sen}(\lambda) + y \cos(\lambda) \end{cases}$$

ξ y η para estas transformaciones son:

$$\begin{cases} \xi(x, y) = -x \operatorname{sen}(\lambda) - y \cos(\lambda)|_{\lambda=0} = -y \\ \eta(x, y) = x \cos(\lambda) - y \operatorname{sen}(\lambda)|_{\lambda=0} = x \end{cases}$$

Por lo que la transformación infinitesimal asociada es:

$$\begin{cases} x_1 = x - y\lambda \\ y_1 = y + x\lambda \end{cases}$$

Dado G grupo con transformación infinitesimal determinada por (ξ, η) , se define como **generador (u operador) infinitesimal de G** al operador diferencial lineal de primer orden:

$$X = \xi(x, y) * \frac{\partial}{\partial x} + \eta(x, y) * \frac{\partial}{\partial y}$$

Sophus Lie llama a X “the symbol”. Es un operador, actuando por tanto sobre funciones de dos *variables*.

Ejemplo. El **generador infinitesimal** del grupo continuo de rotaciones viene dado por:

$$X = -y \frac{\partial}{\partial x} + x \frac{\partial}{\partial y}$$

Problema inverso. Determinación del Grupo Continuo Uniparamétrico asociado a un generador infinitesimal

Dado un generador infinitesimal, X , es posible determinar el Grupo Continuo Uniparamétrico, G asociado².

TEOREMA

PROBLEMA INVERSO: $X \Rightarrow G$ EXISTENCIA Y UNICIDAD

Dado el Grupo continuo dado por

$$\begin{cases} \overline{x_1} = f(x, y, \lambda) \\ \overline{y_1} = g(x, y, \lambda) \end{cases}$$

con generador infinitesimal dado por (ξ, η) .

Las funciones f y g son solución única de las **Ecuaciones de Lie**, dadas por el sistema de ecuaciones diferenciales ordinarias:

$$\begin{cases} x'(\lambda) = \frac{df}{d\lambda} = \xi(x, y) \\ y'(\lambda) = \frac{dg}{d\lambda} = \eta(x, y) \end{cases}$$

o, equivalentemente,

$$\frac{dx}{\xi(x, y)} = \frac{dy}{\eta(x, y)} = d\lambda$$

con valores iniciales

$$\begin{cases} x(0) = f(x, y, 0) = x \\ y(0) = g(x, y, 0) = y \end{cases}$$

² La determinación de G , grupo, a partir de X , generador infinitesimal, presenta cierta analogía con la determinación de una función de una variable conocida su derivada o, más generalmente, la resolución de una EDO de primer orden con valores iniciales.

COROLARIO

MAPEO EXPONENCIAL

Si ξ y η son analíticas, las ecuaciones de Lie pueden resolverse mediante:

$$\begin{cases} \overline{x_1} = f(x, y, \lambda) = e^{\lambda X}(x) \\ \overline{y_1} = g(x, y, \lambda) = e^{\lambda X}(y) \end{cases}$$

, siendo

$$e^{\lambda X} = \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} X^n$$

El mapeo exponencial presenta una evidente analogía con la representación de una función mediante su serie de Taylor.

Existen 4 métodos para calcular el grupo continuo uniparamétrico asociado a un generador infinitesimal:

- ❖ **Tabulación.** Consulta a tablas de los Grupos de Lie más habituales (Anexo I)
- ❖ Resolución de las **Ecuaciones de Lie.**
- ❖ **Mapeo Exponencial.**
- ❖ **Variables Canónicas.**

Ejemplo. Vamos a determinar, mediante distintos métodos, el grupo continuo uniparamétrico de Lie asociado al generador infinitesimal

$$X = \frac{\partial}{\partial x} + 3y \frac{\partial}{\partial y}$$

a. Tabulación. Consultando las **tablas** se puede determinar que el grupo infinitesimal asociado a este generador infinitesimal es el de las **dilataciones no uniformes** dadas por

$$\begin{cases} \overline{x_1} = x e^{\lambda} \\ \overline{y_1} = y e^{3\lambda} \end{cases}$$

b. Ecuaciones de Lie. El sistema de ecuaciones diferenciales a resolver es

$$\begin{cases} \overline{x_1}'(\lambda) = \overline{x_1} \\ \overline{y_1}'(\lambda) = 3\overline{y_1} \end{cases}$$

o, equivalentemente,

$$\frac{d\bar{x}_1}{x} = \frac{d\bar{y}_1}{3y} = d\lambda$$

En este caso, el SEDO se reduce a dos ecuaciones diferenciales ordinarias de primer orden desacopladas que pueden, por tanto, resolverse independientemente.

$$\begin{cases} \bar{x}_1 = c_1 e^\lambda \\ \bar{y}_1 = c_2 e^{3\lambda} \end{cases}$$

Imponiendo las condiciones iniciales

$$\begin{cases} \bar{x}_1(0) = x \\ \bar{y}_1(0) = y \end{cases}$$

Obtenemos,

$$\begin{cases} c_1 = x \\ c_2 = y \end{cases}$$

Con lo que el grupo de transformaciones queda definido por:

$$\begin{cases} \bar{x}_1 = x e^\lambda \\ \bar{y}_1 = y e^{3\lambda} \end{cases}$$

c. Mapeo Exponencial. El grupo de transformaciones queda definido por

$$\begin{cases} \bar{x}_1 = e^{\lambda X}(x) \\ \bar{y}_1 = e^{\lambda X}(y) \end{cases}$$

donde

$$e^{\lambda X} = \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} X^n$$

Se cumple $X(x) = x$. Y, por tanto, $X^2(x) = X[X(x)] = X(x) = x$. Se demuestra por inducción que $X^n(x) = x$.

Análogamente, $X(y) = 3y$. Y, por tanto, $X^2(y) = X[X(y)] = X(3y) = 3^2 y$. Se demuestra por inducción que $X^n(y) = 3^n y$.

Por lo tanto,

$$\bar{y}_1 = e^{\lambda X}(y) = \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} 3^n y = y \sum_{n=0}^{\infty} \frac{(3\lambda)^n}{n!} = y e^{3\lambda}$$

Con lo que el grupo de transformaciones queda definido por:

$$\begin{cases} \bar{x}_1 = x e^\lambda \\ \bar{y}_1 = y e^{3\lambda} \end{cases}$$

Determinación de las simetrías admitidas por una EDO

Sea la ecuación diferencial ordinaria:

$$y' = h(x, y)$$

Condición de simetría

La ecuación diferencial **admite una simetría** cuando es invariante ante un grupo de transformaciones de Lie. Es decir que, en cualquiera de los **nuevos sistemas de coordenadas** dados por cualquiera de las transformaciones del grupo, **la ecuación diferencial queda invariante**.

Sea cualquiera de los nuevos sistemas de coordenadas (\bar{x}_1, \bar{y}_1) , dado por cualquiera de las transformaciones del grupo de Lie, parametrizadas por λ .

$$\begin{cases} \bar{x}_1 = f(x, y, \lambda) \\ \bar{y}_1 = g(x, y, \lambda) \end{cases}$$

En este nuevo sistema de coordenadas la ecuación diferencial se transforma en:

$$\frac{d\bar{y}_1}{d\bar{x}_1} = \frac{D_x \bar{y}_1}{D_x \bar{x}_1} = \frac{\left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y} \frac{dy}{dx}\right) \bar{y}_1}{\left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y} \frac{dy}{dx}\right) \bar{x}_1} = \frac{\frac{\partial \bar{y}_1}{\partial x} + \frac{\partial \bar{y}_1}{\partial y} y'}{\frac{\partial \bar{x}_1}{\partial x} + \frac{\partial \bar{x}_1}{\partial y} y'} = \frac{\frac{\partial \bar{y}_1}{\partial x} + \frac{\partial \bar{y}_1}{\partial y} h(x, y)}{\frac{\partial \bar{x}_1}{\partial x} + \frac{\partial \bar{x}_1}{\partial y} h(x, y)}$$

Por lo tanto, la **condición de simetría** es:

$$\frac{\frac{\partial \bar{y}_1}{\partial x} + \frac{\partial \bar{y}_1}{\partial y} h(x, y)}{\frac{\partial \bar{x}_1}{\partial x} + \frac{\partial \bar{x}_1}{\partial y} h(x, y)} = h(\bar{x}_1, \bar{y}_1)$$

Condición de simetría linealizada

Si en lugar de considerar los nuevos sistemas de coordenadas dados por el grupo de transformaciones continuas uniparamétricas de Lie, consideramos los **nuevos sistemas de coordenadas dados por las transformaciones infinitesimales de Lie**, se obtiene una nueva la condición de simetría.

$$\begin{cases} x_1 = x + \xi(x, y) * \lambda \\ y_1 = y + \eta(x, y) * \lambda \end{cases}$$

La condición de simetría en este nuevo sistema viene dada por

$$\frac{\frac{\partial y_1}{\partial x} + \frac{\partial y_1}{\partial y} h(x, y)}{\frac{\partial x_1}{\partial x} + \frac{\partial x_1}{\partial y} h(x, y)} = h(x_1, y_1)$$

Las derivadas parciales, en función de ξ y η tienen las siguientes expresiones

$$\begin{cases} \frac{\partial y_1}{\partial x} = \lambda \eta_x \\ \frac{\partial y_1}{\partial y} = 1 + \lambda \eta_y \end{cases} \quad \begin{cases} \frac{\partial x_1}{\partial x} = 1 + \lambda \xi_x \\ \frac{\partial x_1}{\partial y} = \lambda \xi_y \end{cases}$$

y la función h que define la ecuación diferencial:

$$h(x_1, y_1) = h(x, y) + \lambda(h_x \xi + h_y \eta)$$

Sustituyendo en la condición de simetría se obtiene

$$\frac{\lambda \eta_x + h(1 + \lambda \eta_y)}{1 + \lambda \xi_x + \lambda \xi_y h} = h(x, y) + \lambda(h_x \xi + h_y \eta)$$

Simplificamos la expresión, despreciando términos de orden superior a $O(\lambda)$, y se obtiene

$$\begin{aligned} h + \lambda(\eta_x + h \eta_y) &= [1 + \lambda(\xi_x + h \xi_y)] [h + \lambda(h_x \xi + h_y \eta)] \Leftrightarrow \\ \Leftrightarrow h + \lambda(\eta_x + h \eta_y) &= h + h\lambda(\xi_x + h \xi_y) + \lambda(h_x \xi + h_y \eta) \end{aligned}$$

Que, simplificada, es la **condición de simetría linealizada**

$$\eta_x - \xi_y h^2 + h(\eta_y - \xi_x) - (\xi h_x + \eta h_y) = 0$$

Resolución de la condición de simetría linealizada

La condición de simetría linealizada es una ecuación que relaciona las **derivadas parciales de primer orden de dos funciones**: $\xi(x, y)$ y $\eta(x, y)$. En ella interviene la función $h(x, y)$ que define la ecuación diferencial de la que se están buscando sus simetrías, por lo que, lógicamente, esta condición de simetría es específica para cada ecuación diferencial.

El número de simetrías admitidas por una ecuación diferencial de primer orden es infinito. Sin embargo, para resolver la ecuación diferencial basta con obtener una de esas simetrías. Con objeto de simplificar la condición de simetría linealizada, se hacen hipótesis (*heuristics*) sobre la naturaleza de las funciones $\xi(x, y)$ y $\eta(x, y)$.

En el capítulo 5 se recogen las heurísticas que se consideran en la librería *Sympy*.

Dependiendo de la función $h(x, y)$, con algunos de estos *hints*, la condición de simetría linealizada se puede transformar en una ecuación diferencial ordinaria para alguna de las funciones $\xi(x)$, $\xi(y)$, $\eta(x)$ o $\eta(y)$ que se obtienen al suponer univariantes las funciones: $\xi(x, y)$ o $\eta(x, y)$.

Así, por ejemplo, la heurística “**abaco1_simple**” comprende 4 sub-heurísticas y para cada una de ellas se obtiene una versión simplificada distinta de la condición de simetría linealiza, así:

- a) Sub-heurística 1. $\xi = 0, \eta = f(x)$:

Se verifica, por tanto, que:

$$\begin{cases} \xi_x = 0, & \xi_y = 0 \\ \eta_y = 0, & \eta_x \text{ es función sólo de } x \end{cases}$$

Y la ecuación **condición de simetría linealizada se simplifica a:**

$$\eta_x - \eta h_y = 0$$

Para ecuaciones diferenciales en que la función h_y es sólo función de x , la condición de simetría linealizada simplificada es una ecuación diferencial ordinaria de primer orden para la función univariante $\eta(x)$.

b) Sub-heurística 2. $\xi = 0, \eta = f(y)$:

Se verifica, por tanto, que:

$$\begin{cases} \xi_x = 0, & \xi_y = 0 \\ \eta_x = 0, & \eta_y \text{ es función sólo de } y \end{cases}$$

Y la ecuación **condición de simetría linealizada se simplifica a:**

$$\eta_y h - \eta h_y = 0$$

Para ecuaciones diferenciales en que la función h es sólo función de y , la condición de simetría linealizada simplificada es una ecuación diferencial ordinaria de primer orden para la función univariante $\eta(y)$.

c) Sub-heurística 3. $\xi = f(x), \eta = 0$:

Se verifica, por tanto, que:

$$\begin{cases} \xi_y = 0, & \xi_x \text{ es función sólo de } x \\ \eta_x = 0, & \eta_y = 0 \end{cases}$$

Y la ecuación **condición de simetría linealizada se simplifica a:**

$$\xi_x h + \xi h_x = 0$$

Para ecuaciones diferenciales en que la función h es sólo función de x , la condición de simetría linealizada simplificada es una ecuación diferencial ordinaria de primer orden para la función univariante $\xi(x)$.

d) Sub-heurística 4. $\xi = f(y), \eta = 0$:

Se verifica, por tanto, que:

$$\begin{cases} \xi_x = 0, & \xi_y \text{ es función sólo de } y \\ \eta_x = 0, & \eta_y = 0 \end{cases}$$

Y la ecuación **condición de simetría linealizada se simplifica a:**

$$\xi_y h^2 + \xi h_x = 0$$

Para ecuaciones diferenciales en que la función h es sólo función de y , la condición de simetría linealizada simplificada es $\xi_y = 0$, cuya solución es $\xi =$
cte

Para el resto de heurísticas, la condición de simetría linealizada simplificada toma distintas expresiones.

En [21] se hace un análisis detallado de las estrategias de resolución de las condiciones de simetría linealizadas simplificadas según estas heurísticas.

Ejemplos de cálculo de simetrías de ecuaciones diferenciales

Con objeto de ilustrar el uso de estas heurísticas vamos a calcular las simetrías que admiten dos de las ecuaciones diferenciales de [17].

Ejemplo. KAMKE-76.

$$y' = \operatorname{acos}(y + b)$$

Como la función $h(y) = \operatorname{acos}(y + b)$, es función sólo de y , son de aplicación las sub-heurísticas b) y d) de la heurística **abaco1_simple**.

La condición de simetría linealizada simplificada en el caso b), que es $\eta_y h - \eta h_y = 0$, es una **ecuación separable**, cuya solución es:

$$\eta_y h - \eta h_y = 0 \Leftrightarrow \frac{\eta_y}{\eta} = \frac{h_y}{h} \Leftrightarrow \log|\eta| = \log|h| + k \Leftrightarrow \eta = ch$$

Por tanto, una simetría para esta ecuación diferencial es

$$\begin{cases} \xi = 0 \\ \eta = \cos(y + b) \end{cases}$$

La condición de simetría linealizada simplificada en el caso d), que es $\xi_y h^2 + \xi h_x = 0$, en este caso se reduce a $\xi = \text{cte}$.

Por tanto, una simetría para esta ecuación diferencial es

$$\begin{cases} \xi = 1 \\ \eta = 0 \end{cases}$$

Estas simetrías coinciden con las calculadas por la aplicación DELHI.

DELHI App - Escuela Técnica Superior de Ingenieros Industriales UPM

STEP 1 STEP 2A STEP 2B

STEP 1 - INFINITESIMALS

$dy/dx = a \cdot \cos(y+b)$ Calculate

Heuristic: abaco1_simple Clear ■ Clear ODE

Linearized symmetry condition $\eta_x - \xi_y h^2 + (\eta_y - \xi_x)h - (\xi h_x + \eta h_y) = 0$

Substituted symmetry condition $-a^2 \cdot xiy \cdot \cos(b+y)^2 + a \cdot xi \cdot \sin(b+y)$

Number	Xi	Eta
1	0	$\cos(b + y(x))$
2	1	0

Infinitesimals successfully calculated

Ejemplo. KAMKE-77

$$y' = \cos(ay + bx)$$

Esta ecuación es considerablemente más complicada que la anterior, pues mientras la ecuación KAMKE-76 era separable, pues el segundo miembro dependía sólo de y , en este caso el segundo miembro es una función de las dos variables x e y .

En este caso, ninguna de las cuatro heurísticas abaco1_simple da lugar a una condición de simetría linealizada simplificada de fácil resolución.

La aplicación DELHI identifica que en este caso es útil el uso de la heurística **abaco2_similar**. Esta heurística comprende dos sub-heurísticas:

- a) Sub-heurística 1. $\xi = f(x), \eta = g(x)$:

Se verifica, por tanto, que:

$$\begin{cases} \xi_y = 0, & \xi_x \text{ es función sólo de } x \\ \eta_y = 0, & \eta_x \text{ es función sólo de } x \end{cases}$$

Y la ecuación **condición de simetría linealizada se simplifica a:**

$$\eta_x - \xi_x h - (\xi h_x + \eta h_y) = 0$$

- b) Sub-heurística 2. $\xi = g(y), \eta = f(y)$:

Se verifica, por tanto, que:

$$\begin{cases} \xi_x = 0, & \xi_y \text{ es función sólo de } y \\ \eta_x = 0, & \eta_y \text{ es función sólo de } y \end{cases}$$

Y la ecuación **condición de simetría linealizada se simplifica a:**

$$-\xi_y h^2 + \eta_y h - (\xi h_x + \eta h_y) = 0$$

En este caso, puede verificarse realizando transformaciones trigonométricas elementales que se:

$$\begin{cases} h_x = -b \operatorname{sen}(ay + bx) \\ h_y = -a \operatorname{sen}(ay + bx) \end{cases}$$

Se observa que ninguna de las dos condiciones de simetría linealizadas simplificados con estas dos heurísticas es especialmente sencilla de resolver. En cualquier caso, existe una estrategia bien definida para resolver estas ecuaciones, descritas en [21].

a+b) No obstante, si hacemos una simplificación adicional, suponiendo que deben cumplirse simultáneamente ambas heurísticas, es decir, que tanto ξ como η son constantes:

$$\xi = \alpha, \eta = \beta:$$

Se verifica, por tanto, que:

$$\begin{cases} \xi_x = 0, & \xi_y = 0 \\ \eta_x = 0, & \eta_y = 0 \end{cases}$$

Y la ecuación **condición de simetría linealizada se simplifica a:**

$$(\xi h_x + \eta h_y) = 0$$

$$\text{Es decir, } \sin(ay + bx)(b\alpha + a\beta) = 0 \Leftrightarrow \beta = -\frac{b}{a}\alpha$$

Por tanto, una simetría para esta ecuación diferencial es

$$\begin{cases} \xi = 1 \\ \eta = -\frac{b}{a} \end{cases}$$

Estas simetrías coinciden con las calculadas por la aplicación DELHI.

The screenshot shows the DELHI App interface. On the left is a sidebar with buttons: Home, Solver, Step-by-step (highlighted), Benchmark, and Help. The main area is titled 'STEP 1 - INFINITESIMALS'. It contains a 'dy/dx =' input field with 'cos(a*y+b*x)' and a 'Calculate' button. Below it is a 'Heuristic:' dropdown menu with 'abaco2_similar' selected and a 'Clear' button. To the right is a 'Clear ODE' checkbox. Below these are two text boxes: 'Linearized symmetry condition' with the formula $\eta_x - \xi_y h^2 + (\eta_y - \xi_x)h - (\xi h_x + \eta h_y) = 0$ and 'Substituted symmetry condition' with the formula $a * \xi * \sin(a * y + b * x) + b * \eta * \sin(a * y + b * x)$. Below these is a table with columns 'Number', 'Xi', and 'Eta'. The table has one row with '1', '1', and '-b/a' respectively. At the bottom, a green status bar says 'Infinitesimals successfully calculated'.

Resolución mediante el uso de Simetrías de Lie

Una vez conocida una **simetría infinitesimal admitida** por una ecuación diferencial hay dos estrategias alternativas para resolverla:

1. Encontrar un **factor integrante** que la transforme en una ecuación diferencial **exacta**.
2. Hacer un cambio de coordenadas, a las llamadas **coordenadas canónicas**, para transformar la ecuación en una ecuación diferencial **separable**.

Método del Factor Integrante de Lie

TEOREMA

FACTOR INTEGRANTE DE LIE

(enunciado y demostrado en los trabajos originales de Sophus Lie en 1874, ver [\[23\]](#))

Dada la ecuación diferencial de primer orden (expresada como forma diferencial)

$$M(x, y)dx + N(x, y)dy = 0$$

Admite un grupo uniparamétrico G con generador infinitesimal

$$X = \xi(x, y) * \frac{\partial}{\partial x} + \eta(x, y) * \frac{\partial}{\partial y}$$

Si y solo si, la función

$$\mu(x, y) = \frac{1}{\xi M + \eta N}$$

es un factor integrante de la ecuación diferencial, supuesto que $\xi M + \eta N \neq 0$.

Demostración. Véase [\[19\]](#), página 267 y 268.

Ejemplo. KAMKE-77

$$y' = \cos(ay + bx)$$

O, equivalentemente, como forma diferencial:

$$\overbrace{\cos(ay + bx)}^M dx - \overbrace{1}^N dy = 0$$

La ecuación diferencial no es exacta:

$$\begin{cases} M_y = -a \sin(ay + bx) \\ N_x = 0 \end{cases}$$

Para la simetría $(\xi, \eta) \equiv \left(1, -\frac{b}{a}\right)$, calculada anteriormente, el factor integrante de Lie asociado es

$$\mu(x, y) = \frac{1}{\xi M + \eta N} = \frac{1}{\cos(ay + bx) + \frac{b}{a}}$$

Con este factor integrante la ecuación diferencial se transforma en:

$$\overbrace{\frac{\cos(ay + bx)}{\cos(ay + bx) + \frac{b}{a}}}^{M1} dx - \overbrace{\frac{1}{\cos(ay + bx) + \frac{b}{a}}}^{N1} dy = 0$$

que sí es exacta, ya que

$$M1_y = N1_x = \frac{-b \sin(ay + bx)}{\left[\cos(ay + bx) + \frac{b}{a}\right]^2}$$

y, por tanto, **reducible a cuadraturas**. Las integrales que es necesario resolver para obtener la solución explícita son, en este caso, funciones elementales aunque con expresiones alambicadas dependiendo de los parámetros a , b y de funciones trigonométricas e hiperbólicas (en concreto, de \argth).

Con objeto de ilustrar el método con cálculos más sencillos vamos a resolver explícitamente la ecuación diferencial dando valores a los parámetros a y b .

Ejemplo. KAMKE-77 para $a=1$ y $b=1$

$$y' = \cos(y + x)$$

que sigue siendo una ecuación diferencial complicada de resolver, ya que **no es ni exacta**, ni separable ni, por supuesto, lineal.

En este caso, la simetría anterior se concreta en $(\xi, \eta) \equiv (1, -1)$, y el factor integrante de Lie asociado en

$$\mu(x, y) = \frac{1}{\xi M + \eta N} = \frac{1}{\cos(y + x) + 1}$$

Con este factor integrante la ecuación diferencial se transforma en:

$$\overbrace{\frac{\cos(y + x)}{\cos(y + x) + 1}}^{M1 \equiv U_x} dx - \overbrace{\frac{1}{\cos(y + x) + 1}}^{N1 \equiv U_y} dy = 0$$

que **sí es exacta**, ya que

$$M1_y = N1_x = \frac{-\sin(y + x)}{[\cos(y + x) + 1]^2}$$

En este caso, la resolución explícita de la ecuación diferencial es más sencilla.

$$U = \int -\frac{1}{\cos(y + x) + 1} dx \stackrel{\text{cambio } t = \text{tg}\left(\frac{x}{2}\right)}{\cong} -\text{tg}\left(\frac{y + x}{2}\right) + \Phi(x)$$

Imponiendo $U_x = \frac{\cos(y+x)}{\cos(y+x)+1}$, obtenemos

$$U = -\text{tg}\left(\frac{y + x}{2}\right) + x$$

Con lo que la solución implícita de la ecuación diferencial es

$$\text{tg}\left(\frac{y + x}{2}\right) - x = \text{cte}$$

Lo que puede comprobarse, fácilmente, mediante derivación implícita.

Y la solución explícita de la ecuación diferencial

$$y = 2\text{arctg}(x + \text{cte}) - x$$

Lo que también puede comprobarse fácilmente.

5.Librerías Python: Tkinter y Sympy

Implementación y uso de las librerías

En este apartado se van a describir las dos librerías del lenguaje Python que suponen el núcleo de la aplicación desarrollada. Se tratan de *Sympy*, la librería utilizada para trabajar con el cálculo simbólico en dicho lenguaje de programación, y *Tkinter*, una de las múltiples bibliotecas dedicadas a la creación y manipulación de interfaces gráficas.

Para elaborar el presente capítulo se han utilizado referencias y descripciones de la documentación de Python, disponible en [3] [15] . Asimismo, se incluyen también definiciones no tan estrictas que facilitan la descripción del funcionamiento de las funciones, clases y algoritmos utilizados sin necesidad de profundizar en un lenguaje excesivamente técnico.

Librería Tkinter

Por su definición en [9] Tkinter es “*un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación Python. Se considera un estándar para la interfaz gráfica de usuario (GUI) para Python y es el que viene por defecto con la instalación para Microsoft Windows*”. Esto es, consiste en la librería preinstalada de Python para la creación de aplicaciones gráficas, estandarizada y ampliamente extendida entre los desarrolladores del citado lenguaje de programación.

Necesidad de una interfaz gráfica

Se concibió como necesario incluir una interfaz gráfica en el proyecto para, como se da a entender entre los objetivos del mismo, hacer partícipe al alumno de todas las fases de la resolución de una ecuación por el método que aquí se presenta. Esto permite dar un paso más allá en la enseñanza clásica, y en lugar de impartir la teoría y una batería de ejercicios para resolver, se opta por la interacción directa.

Experimentar por sí mismo con las ecuaciones y los diferentes parámetros sin preocuparse por tener que resolver a mano pesadas operaciones da una gran libertad al usuario, y fomenta su rápido aprendizaje. Además, la aplicación puede tener un propósito mucho más general del inicialmente planteado. Puede utilizarse en clase, como apoyo al docente, en casa para el alumno e incluso para realizar ciertos ejercicios durante una prueba de evaluación, para examinar los conocimientos reales del estudiante y su capacidad de aplicarlos para resolver el problema en cuestión.

Por todos estos motivos, se diseñaron tres espacios diferentes a los que “poner cara” mediante la creación de la interfase de usuario, y mediante los cuales se puede probar el método de simetrías de Lie de forma diferente en cada uno. Más adelante se describe en profundidad la herramienta desarrollada y se ofrecen en detalle los apartados mencionados.

Alternativas a *Tkinter*

Como se ha dicho, el software diseñado pretende ser una herramienta sencilla, pero con gran potencial. Es por ello que se pensó en primera instancia en la librería *Tkinter* para introducir la interfaz. Al ser una librería que viene preinstalada con la distribución de Python, es fácil comenzar a utilizarla. Por otra parte, la gran cantidad de documentación, dudas en foros e información disponible en Internet la hacen aún más atractiva para el desarrollador.

Sin embargo, como en todos los ámbitos de la informática, siempre existen opciones alternativas igualmente viables para un proyecto. En particular, para el trabajo que nos atañe, se barajó la opción de usar la librería *PyQt*. De nuevo recurriendo a la definición oficial en [\[10\]](#), se trata de “...un binding de la biblioteca gráfica Qt para el lenguaje de programación Python. La biblioteca está desarrollada por la firma británica Riverbank Computing y está disponible para Windows, GNU/Linux y Mac OS X bajo diferentes licencias.”

Originalmente, la librería gráfica Qt pertenece al lenguaje C++. Para evitar “reinventar la rueda” como suele decirse en la jerga de programación, se ha creado una adaptación para que los desarrolladores del lenguaje Python puedan utilizarla también, naciendo así el paquete *PyQt*.

El autor de este trabajo presenta cierta experiencia manejando la librería original en C++. A pesar de esto, al contar este proyecto inicialmente con una estructura sencilla (que a lo largo del desarrollo se ha ido madurando y haciendo más compleja) se decidió no utilizar *PyQt* por estar pensada para proyectos más grandes. Trae consigo un conjunto de utilidades que hay que instalar y aprender a utilizar antes de comenzar a desarrollar la aplicación propiamente dicha. Todo esto conllevaría una inversión de tiempo inicial mayor que a la larga se desconocía si traería los resultados deseados.

Por el contrario, con *Tkinter* iniciar la programación resulta mucho más sencillo, prácticamente *plug-and-play*, por lo que la librería gráfica predeterminada terminó siendo la opción escogida al contar con herramientas suficientes para el proyecto que se pretendía realizar. Tras un breve periodo de investigación para conocer su funcionamiento se comenzó el desarrollo de la aplicación de escritorio.

Descripción de la librería

En este apartado se van a comentar los elementos principales que ofrece la biblioteca *Tkinter* para desarrollar interfaces gráficas. Dichos elementos son denominados *widgets*, y actualmente hay una gran variedad de ellos disponible. Cada uno de ellos está diseñado con un propósito relativamente concreto. Una de sus ventajas es que se pueden crear clases heredadas de ellos para modificar su comportamiento y adaptarlo a una necesidad concreta.

Estos elementos también son completamente configurables. Sus opciones de configuración van desde definir su tamaño, grosor de borde o márgenes hasta poder alterar su estética variando los colores y los estilos. Seguidamente, se detallan los *widgets* principales de la librería, los más utilizados en el proyecto y su utilidad.

Widget *Frame*

Compone la base de toda aplicación, se define como un contenedor de elementos. Consiste en un marco o cuadro rectangular, a priori invisible, que sirve para agrupar otros *widgets* y desplegarlos encima. Una aplicación correctamente desarrollada debe manejar bien grupos de *Frames* para lograr tanto la encapsulación como el *layout* deseados.

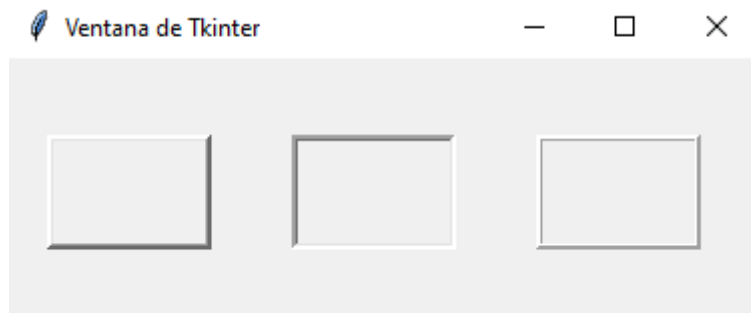


Figura 1. Ejemplo de interfaz con varios *Frame*

Widget *Label*

Otro de los elementos que presenta toda interfaz son las etiquetas o *labels*. Su propósito es claro: contienen texto, anotaciones o imágenes para mostrar al usuario. Están pensadas para acompañar a otros elementos y dar información sobre ellos de manera breve, o guiar al usuario a través de la interfaz, pero no para contener largos párrafos (a pesar de ser posible hacerlo) debido a sus pocas opciones de formatear el texto. Para ello hay otros widgets más apropiados como *Text* o *Message*.

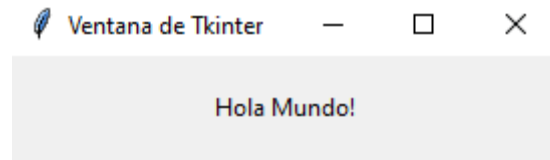


Figura 2. Ejemplo de interfaz con una etiqueta

Widget *Button*

A diferencia de los *Frame* y *Label*, este widget está pensado completamente para la interacción con el usuario. Consiste en un botón, que normalmente al ser presionado ejecuta una tarea en la aplicación. Pueden contener texto e imágenes al igual que las etiquetas para indicar su función, que puede ir desde actualizar algunas variables, enviar un formulario, borrar ciertos elementos o cualquier utilidad que se pueda imaginar. Es uno de los elementos más versátiles, comunes y útiles de todos.

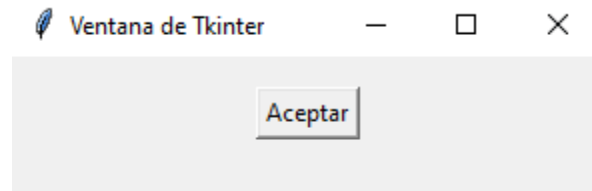


Figura 3. Interfaz con botón de ejemplo

Widget *Entry*

Muy reconocible también, se trata de una caja rectangular donde se puede escribir texto en una línea. Es una parte imprescindible de cualquier formulario o solicitud donde haya que rellenar algunos datos. Existen varias clases que heredan de ella y le aportan funcionalidades extra, como las *Combobox* que generan un desplegable con cierto número de opciones, o las *Spinbox* que agregan unas flechas para aumentar o disminuir el número escrito en la caja de texto habitualmente, marcando un rango de números posibles a introducir.

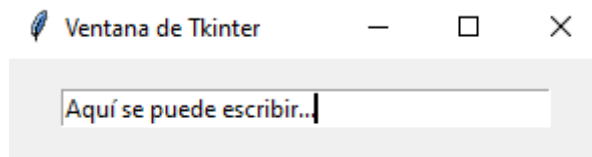


Figura 4. Ejemplo de una caja de texto *Entry*

Widgets de *ttk*

Cabe hacer mención a uno de los submódulos de *Tkinter*, llamado *ttk*. En él se incluyen los “*Tkinter Themed Widgets*”. Gracias a este submódulo se puede trabajar con el estilo de la aplicación desde otro punto de vista.

El submódulo *ttk* redefine las clases de todos los widgets, poniendo a disposición del usuario de nuevo etiquetas, botones y todos los elementos ya incluidos en *Tkinter*, sumando algunos nuevos que aportan utilidades muy interesantes como los *Notebook* (panel de pestañas). La diferencia principal con los anteriores radica en la manera de implementar los estilos.

Para los widgets anteriormente descritos de la librería *Tkinter* estándar, es necesario definir para cada uno de ellos en los argumentos de su clase qué aspecto tendrán finalmente: color de fondo, fuente, color de la letra, relieve del borde o márgenes laterales son ejemplos de algunas características que se pueden editar para dichos elementos.

En cambio, los widgets de *ttk* están gobernados por una nueva clase llamada *Style*, que permite crear temas que afectan a toda la aplicación desde cero, o heredando de otros que vienen incluidos por defecto. También es posible definir estilos aislados para cierto tipo de widget, que afectarán a aquellos que quieran incluirlo en los argumentos del constructor de su clase.

Estableciendo una comparativa, los widgets de *Tkinter* son más fácilmente configurables como se ha indicado anteriormente, y se tiene prácticamente un control total sobre su aspecto. Con los widgets de *ttk*, definir los estilos lleva algo más de trabajo, pero los resultados en general son mejores y una vez creados los estilos deseados, son muy recomendables para interfaces que contengan un gran número de elementos, pues a medio y largo plazo se ahorra tiempo de codificación.

Separadamente, el hecho de poder heredar de los temas predeterminados de *ttk* hace que el usuario pueda imitar el estilo nativo de ciertos sistemas operativos, al tiempo que puede modificar los aspectos que desee. La mayor ventaja de todas es que en una misma aplicación *Tkinter* pueden mezclarse los widgets normales con los del submódulo *ttk* según las necesidades del programador. Esto hace que la biblioteca sea muy flexible y permita resolver diversas situaciones utilizando unos u otros.

Distribución de los elementos o *Layout*

Una vez conocidos los diferentes elementos de los que la librería dispone, es necesario saber cómo distribuirlos en la interfaz para obtener una ventana correctamente colocada y, a ser posible, intuitiva de utilizar.

Para ello, Tkinter dispone de tres métodos de colocación, también conocidos como *Geometry Managers*, que nos ayudan a darle a cada elemento la información sobre su posición en la ventana y respecto a otros elementos.

El primero de ellos es el método ***pack***. Es el más sencillo de utilizar, pero también el que menos opciones de configuración tiene. Cuando se utiliza este método, se ha de indicar la posición del widget en cuestión entre cuatro opciones: *TOP*, *LEFT*, *RIGHT* y *BOTTOM*. Indican a qué borde de su contenedor se adherirán los widgets al aparecer en pantalla.

Este método es especialmente indicado para interfaces con pocos widgets. En aquellas que disponen de gran cantidad de elementos es mejor utilizar una de las otras dos opciones. Por contraparte, el método *pack* es muy útil a la hora de colocar *Frames* que estén en el mismo contenedor, debido a que son rectangulares y son fácilmente distribuibles con estas simples opciones.

En segundo lugar, aparece ***grid***, que, como su nombre indica, se encarga de colocar los widgets en una rejilla o distribución por celdas. Es especialmente indicado para interfaces que se asemejen a una tabla, pudiendo los elementos ocupar varias celdas si es necesario o expandirse cubriendo diversas filas. También es muy útil con gran cantidad de widgets dentro de un mismo contenedor al asignar a cada uno de ellos una fila y una columna, otorgándoles una posición unívoca y fácil de organizar.

Por último, el *manager* ***place*** asigna unas coordenadas según el eje horizontal y vertical, relativas al contenedor del widget. La esquina superior izquierda de dicho contenedor representa el origen de coordenadas. Es útil cuando se conocen bien las dimensiones finales de todos los recuadros, imágenes y ventanas y se quiere asignar con gran precisión la locación de todos los elementos.

En el presente trabajo se han utilizado únicamente los dos primeros métodos de distribución. El primero de ellos (*pack*) se utiliza para relacionar los *Frames* que subyacen a la interfaz para conseguir que se adapten a su lugar, y el segundo (*grid*) se utiliza dentro de los recuadros para colocar todos los elementos de cada sección de la aplicación. No se ha utilizado el método *place* debido a que en ningún momento se han conocido dimensiones finales de ningún elemento (ni se ha pretendido establecerlas hasta el final), y por tanto dar una posición absoluta a un elemento no cobraba sentido debido a la gran cantidad de cambios y mejoras que se han ido introduciendo a lo largo del desarrollo.

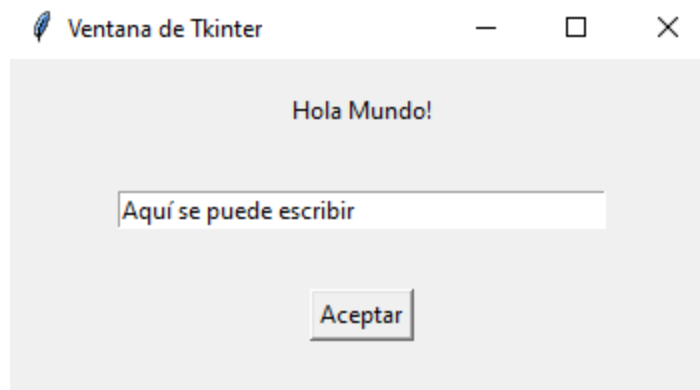


Figura 5. Ejemplo de una distribución utilizando el método pack.

Librería Sympy

Tal y como describe la documentación, *Sympy* (Symbolic Python) es una librería para utilizar matemáticas simbólicas en los programas escritos en Python. A diferencia de otras librerías, como *NumPy* (Numeric Python), que realizan cálculos numéricos, *Sympy* trabaja con unos parámetros definidos previamente, conocidos como símbolos, para operar con ellos. Esto permite evitar costosas operaciones numéricas de gran precisión de dígitos, dejando toda la potencia del procesador dedicarse a resolver los algoritmos propiamente dichos. Por ello, los tiempos de ejecución de las operaciones suelen ser menores al utilizar cálculo simbólico.

En el ámbito del presente trabajo, la resolución de ecuaciones diferenciales mediante cálculo simbólico es una decisión acertada en primera instancia. Al tratar con procesos como la derivación e integración, en general muy caros en cuanto a coste de computación se refiere, es fundamental resolverlos en función de las variables simbólicas.

También es conveniente desde la perspectiva del usuario. Las soluciones expresadas como expresiones simbólicas son suficientemente legibles a simple vista como para que el usuario pueda comprender su significado, y comenzar a extraer conclusiones de los resultados obtenidos.

Clases y métodos

Como se puede deducir, en un proyecto de tamaño medio como es la aplicación que se presenta, se utilizan infinidad de clases y funciones de diversas librerías como las que gestionan la entrada y salida de datos, la conexión con periféricos, tratamiento de ficheros, gestión de errores, etcétera. En este apartado nos centraremos en los principales métodos de la librería *Sympy*.

Para lograr una correcta resolución de una ecuación diferencial mediante cualquiera de los métodos incluidos en la librería, primero ha de saber representarse adecuadamente la ecuación, y a continuación conocer de qué funciones dispone el desarrollador para alcanzar la solución deseada.

Clase **Eq(rhs, lhs)**

Esta clase permite al usuario representar una igualdad de cualquier naturaleza, en nuestro caso una ecuación diferencial ordinaria. Como toda ecuación, dispone de dos miembros, que han de ser pasados como argumento al constructor de la clase. Sus nombres por defecto son *rhs* y *lhs*. La coma hace las veces del símbolo de igualdad. Una vez llamada la clase, se nos devuelve un objeto *Eq*, que podemos utilizar con las siguientes funciones para manipularla y resolverla. Presenta una alternativa mucho más útil que escribir las ecuaciones como expresiones y asignarlas a una variable de texto

String, ya que la mayoría de librerías que trabajan con las matemáticas admiten objetos de igualdad *Eq* en sus algoritmos.

Función *dsolve*

Esta función compone sin duda el núcleo de la aplicación DELHI. Citando a la documentación oficial de Sympy [2], *dsolve* “resuelve cualquier tipo (soportado) de ecuación diferencial, o sistema de ecuaciones diferenciales”.

Este método consta de un gran número de argumentos que nos permiten personalizar la resolución de la ecuación en cuestión. La siguiente tabla ilustra los argumentos que pueden pasarse a *dsolve* y la utilidad de cada uno de ellos.

eq	Una ecuación diferencial soportada por la librería. Puede ser una igualdad o una expresión
func	Función de una variable cuyas derivadas en dicha variable componen la ecuación diferencial a resolver
hint	Método de resolución que el usuario indique. La lista de posibles métodos se obtiene con <code>classify_ode()</code>
simplify	Habilita la simplificación con el método <code>odesimp()</code> . Reduce constantes arbitrarias y cancela términos innecesarios
xi y eta	Son las funciones infinitesimales de una EDO, descritas en el apartado de teoría
ics	Conjunto de condiciones iniciales o de contorno de la ecuación
x0	Punto alrededor del cual se evalúa la solución en series de potencias de la ecuación diferencial.

Tabla 1. Parámetros de *dsolve*

En los diferentes apartados de la herramienta DELHI, se utiliza *dsolve* para obtener la solución en todos ellos. El apartado *Solver* simplemente la llama indicando la ecuación y la función incógnita de la misma, que en general es $y(x)$.

El apartado *Step-by-Step* obliga a resolver la ecuación mediante el *hint* “*lie_group*” en todos los casos. Las opciones de personalización en este punto radican en que el usuario calcula en primer lugar las funciones ξ y η utilizando el método *infinitesimals* para posteriormente indicar cuáles de los obtenidos desea emplear en la ejecución de *dsolve*. Sin ser consciente, el usuario en realidad configura los parámetros de este último método a través de la interfaz.

Por último, en la sección *Benchmark* se ha utilizado *dsolve* cambiando constantemente sus parámetros para realizar los tres test que se detallan más adelante. De esta manera se han podido obtener los resultados que allí se ofrecen.

Función *classify_ode*

Utilizada en el proceso de resolución por *dsolve*, y también disponible para el usuario, esta función devuelve una tupla (una lista) con las posibles clasificaciones para la EDO en

cuestión. Esto nos permite conocer a priori qué métodos de resolución son adecuados para ella. Algunos ejemplos son Ricatti, Bernoulli, separable o de tipo exacta.

Existen un total de 42 clasificaciones posibles para una ecuación diferencial en Sympy. Más adelante se describen con mayor detalle cuáles son las más importantes y las más utilizadas en el contexto de este trabajo.

Si este método se utiliza junto con `dsolve`, se puede especificar en el argumento `hint` de esta última función bajo qué clasificación se quiere tratar la EDO al resolverla. La función `dsolve` intentará obtener la solución con dicho método, arrojando un error en caso de no conseguirlo.

Función `checkodesol`

A pesar de contar con un método tan potente como `dsolve` en esta librería, siempre es conveniente asegurarse de que la solución obtenida es correcta. Para ello se utiliza el método `checkodesol`, cuya misión consiste en sustituir la solución que proporcione el usuario (una función) en la ecuación original y verificar que el resultado es cero. En ese caso, queda comprobada la validez de la solución. En caso contrario debe revisarse la solución propuesta para encontrar el error.

Función infinitesimals

Como se ha descrito en el apartado de teoría, las funciones infinitesimales ξ y η de una ecuación diferencial son los infinitesimales del grupo de transformaciones puntuales de Lie para los cuales la ecuación diferencial queda invariante. Este método permite obtener dichas funciones infinitesimales, a través del proceso de resolución de una ecuación en derivadas parciales llamada condición de simetría linealizada.

La resolución de dicha ecuación en derivadas parciales no es ni mucho menos trivial, y solo puede alcanzarse realizando ciertas asunciones acerca de la forma de las funciones infinitesimales. Estas suposiciones son llamadas heurísticas, y conforman un grupo de nueve condiciones que se pueden asumir para simplificar la EDP y así lograr resolverla, obteniendo ξ y η .

Más adelante en este capítulo se explican cada una de las nueve heurísticas que están programadas en Sympy, junto con un análisis que destaca las más útiles y aquellas que son sustituibles o aportan menos valor a la resolución del problema.

Función *checkinfsol*

Al igual que la función solución de la ecuación diferencial, las funciones infinitesimales también son solución de una ecuación diferencial. Por ello se hace necesaria la existencia de un método que permita comprobar la validez de los *infinitesimals* obtenidos. De esta tarea se encarga *checkinfsol*, sustituyendo ξ y η en la condición de simetría linealizada (la EDP de la que se obtienen) y chequeando que la cumplen.

Función *sympify*

Esta función tiene una participación menor que las anteriormente descritas. Sin embargo, los lugares en los que se utiliza es clave para permitir trabajar correctamente a varias de las herramientas de que se compone DELHI.

El propósito de *sympify* es convertir una expresión o ecuación al lenguaje que Sympy interpreta correctamente. Esto es necesario debido a que la entrada por teclado que pueda introducir un usuario no siempre tendrá la sintaxis necesaria para operar a nivel de código. Así pues, esta función se ha utilizado en el proyecto actual para convertir dichas expresiones en otras manipulables por el resto de clases y métodos.

No debe confundirse con la función *simplify*, que se encarga de simplificar una expresión reduciendo sus constantes arbitrarias, así como cancelando términos de fracciones en numeradores y denominadores para obtener una expresión más sencilla.

Hints de *dsolve*

Existen un total de 42 clasificaciones posibles para una EDO en esta librería. Sus nombres, tal y como se indica en la documentación [2] atienden a una nomenclatura determinada, dependiendo del método de resolución que empleen. Por ejemplo, si el nombre de un cierto hint contiene la palabra *Integral*, esto significa que en alguna fase de la resolución será necesaria una integración. Dependiendo del caso, el programa optará por obtener la primitiva o dejar la solución en forma de cuadraturas, con la integral indicada.

Otro caso puede ser encontrar números de orden al principio de los nombres. Como ejemplo encontramos *1st_exact*, que indica que la ecuación es exacta de primer orden, es decir, que es la derivada total de una función.

Para el trabajo se ha elaborado un módulo de test que se presenta en el siguiente apartado, el 8.2, denominado *Benchmark*. En él se ha probado una batería de 88 ecuaciones diferenciales, entre las cuales podemos destacar como los más importantes métodos de clasificación los siguientes: *1st_power_series*, *lie_group* (correspondiente al método de simetrías de Lie, objetivo de estudio del presente documento), separable,

1st_exact, *1st_linear*, *Bernoulli* y las versiones con integrales de los cuatro últimos citados.

Cabe destacar en esta sección que en *Sympy*, existe una función para cada *hint*. Estas funciones están dedicadas al uso interno de la librería. Sin embargo, al estar descritas y enumeradas en la documentación, el usuario puede forzar su uso en determinadas situaciones. Los nombres de dichas funciones son iguales que los de su correspondiente *hint*.

En la siguiente tabla podemos encontrar una relación de todas las posibles clasificaciones de una ecuación.

factorable
nth_algebraic
separable
1st_exact
1st_linear
Bernoulli
Riccati_special_minus2
1st_homogeneous_coeff_best
1st_homogeneous_coeff_subs_indep_div_dep
1st_homogeneous_coeff_subs_dep_div_indep
almost_linear
linear_coefficients
separable_reduced
1st_power_series
lie_group
nth_linear_constant_coeff_homogeneous
nth_linear_euler_eq_homogeneous
nth_linear_constant_coeff_undetermined_coefficients
nth_linear_euler_eq_nonhomogeneous_undetermined_coefficients
nth_linear_constant_coeff_variation_of_parameters
nth_linear_euler_eq_nonhomogeneous_variation_of_parameters
Liouville
2nd_linear_airy
2nd_linear_bessel
2nd_hypergeometric
2nd_hypergeometric_Integral
nth_order_reducible
2nd_power_series_ordinary
2nd_power_series_regular
nth_algebraic_Integral
separable_Integral
1st_exact_Integral
1st_linear_Integral
Bernoulli_Integral
1st_homogeneous_coeff_subs_indep_div_dep_Integral
1st_homogeneous_coeff_subs_dep_div_indep_Integral
almost_linear_Integral
linear_coefficients_Integral
separable_reduced_Integral
nth_linear_constant_coeff_variation_of_parameters_Integral
nth_linear_euler_eq_nonhomogeneous_variation_of_parameters_Integral
Liouville_Integral

Tabla 2. Hints de dsolve.

Heuristics de Lie

Tal y como se ha descrito en la función *infinitesimals*, existen 9 aproximaciones al problema de la EDP llamada condición de simetría linealizada. Estas parten de la base de asumir ciertas propiedades de las funciones ξ y η para simplificar la forma de la ecuación en derivadas parciales para facilitar su abordaje. Al igual que en el caso de los *hints* de *dsolve*, Sympy cuenta con una función interna para utilizar cada *heuristic* de manera oculta al usuario. Seguidamente puede observarse una imagen con cada heurística y las asunciones que impone.

$$\begin{array}{l} \xi = 0, \eta = f(x) \\ \vdots \\ \xi = 0, \eta = f(y) \\ \xi = f(x), \eta = 0 \\ \xi = f(y), \eta = 0 \end{array}$$

Figura 6. Asunciones de “abaco1_simple”.

$$\begin{array}{l} \eta = 0, \xi = f(x) * g(y) \\ \eta = f(x) * g(y), \xi = 0 \end{array}$$

Figura 7. Asunciones de “abaco1_product”.

$$\begin{array}{l} \eta = g(x), \xi = f(x) \\ \eta = f(y), \xi = g(y) \end{array}$$

Figura 8. Asunciones de “abaco2_similar”.

$$\begin{aligned}\eta &= 0, \xi = f(x) + g(y) \\ \eta &= f(x) + g(y), \xi = 0\end{aligned}$$

Figura 9. Asunciones de “function_sum”.

$$\begin{aligned}\eta &= f(x) \\ \xi &= g(y)\end{aligned}$$

Figura 10. Asunciones de “abaco2_unique_general”.

$$\begin{aligned}\xi &= ax + by + c \\ \eta &= fx + gy + h\end{aligned}$$

Figura 11. Asunciones de “linear”.

Benchmarking de SymPy. Kamke 88

Para analizar la potencia de la implementación vista de la librería *SymPy*, se presentan en este apartado una serie de bancos de pruebas, llamados *benchmark*. Suponen una primera aproximación a la calidad de los métodos incluidos en la librería. Además, sirven de complemento al resto de la aplicación *standalone* a la hora de resolver ecuaciones diferenciales por simetrías de Lie, expuesta más adelante.

Estos test se han realizado sobre un conjunto de ecuaciones típico en este cometido. El conjunto utilizado aquí consiste en ochenta y ocho ecuaciones extraídas del libro de Erich Kamke.

Tradicionalmente dicha batería de prueba consta de 576 ecuaciones de primer orden. Incluyendo las de órdenes superiores el conjunto crece hasta los 1940 ejemplos. Otras plataformas como *Maple* o *Mathematica* han sido probadas por terceros utilizando este extenso conjunto al completo.

Sin embargo, en el presente trabajo se ha limitado este gran número a las ecuaciones de primer orden y que además se presentan en forma normal, esto es, aquellas cuya función incógnita está despejada, a excepción de coeficientes enteros que son fácilmente eliminables. Aplicando dicho criterio se obtienen las 88 ecuaciones mencionadas.

A continuación, se exponen los diferentes test que se han ejecutado sobre la librería *SymPy*. Para conseguir una dirección de trabajo, se plantearon primeramente una serie de cuestiones que se han ido respondiendo a medida que se han obtenido los resultados de las pruebas.

Test 1 - Capacidad de resolución de la librería al completo

El objetivo principal de esta prueba es averiguar la potencia de los métodos incluidos en la librería *SymPy* a la hora de resolver ecuaciones diferenciales mediante cualquiera de los métodos programados en ella.

La siguiente tabla ilustra para las quince primeras ecuaciones del conjunto, a modo de ejemplo, los resultados directos que se han obtenido, y a partir de los cuales se extraerán diferentes conclusiones a lo largo de este apartado. En el anexo IV se puede encontrar una tabla análoga para las 88 ecuaciones que se han testado.

EDO	Resuelta	Resuelta por Lie	Hint por defecto	Tiempo por defecto	Tiempo por Lie
1	Sí	Sí	<i>nth_algebraic</i>	3.14	9.56
2	Sí	Sí	<i>1st_linear</i>	1.02	2.23
3	Sí	Sí	<i>1st_linear</i>	3.31	7.83
4	Sí	Sí	<i>1st_exact</i>	1.68	1.70
5	Sí	Sí	<i>1st_exact</i>	9.96	6.44
6	Sí	Sí	<i>1st_exact</i>	11.39	9.80
7	Sí	Sí	<i>1st_exact</i>	7.03	6.23
8	Sí	Sí	<i>1st_exact</i>	4.98	3.23
9	Sí	Sí	<i>separable</i>	4.41	18.46
10	Sí	No	<i>1st_exact</i>	4.46	2.94
11	Sí	No	<i>1st_exact</i>	3.77	2.96
12	Sí	No	<i>separable</i>	1.77	2.87
13	Sí	No	<i>1st_power_series</i>	11.42	2.26
14	No	No	<i>1st_power_series</i>	0.60	1.32
15	Sí	Sí	<i>1st_power_series</i>	2.77	4.56

Tabla 3. Resultados de las 15 primeras ecuaciones de Kamke.

¿Cuántas ecuaciones resuelve utilizando cualquier método?

Permitiendo a la función *dsolve* utilizar cualquier método, incluyendo las series de potencias, se ha obtenido que son globalmente resolubles un total de 60 ecuaciones de las 88 que componen el conjunto de prueba. Esto supone que un 68,18% de las ecuaciones han sido resueltas y se ha obtenido una expresión de su solución, conformando un porcentaje sorprendentemente elevado a priori.

RESULTADO	Número de ecuaciones	Porcentaje
Globalmente resueltas	60	68,18%
No resolubles por ningún método	28	31,82%
TOTAL	88	100,00%

Tabla 4. Ecuaciones resueltas y no resueltas en el Test 1.

¿Cuántas ecuaciones resuelve utilizando cualquier método a excepción de las series de potencias?

Las soluciones obtenidas con series de potencias se consideran de una calidad muy inferior al resto, debido a que apenas aportan información real sobre la función incógnita al tratarse de un polinomio con una gran cantidad de términos. Asimismo, en la mayoría

de los casos la solución queda en función de una “o de Landau”, que indica una infinidad de términos de orden superior que no se especifican por sencillez de la expresión.

De esta manera, eliminando aquellas ecuaciones que solo podían ser resueltas por series de potencias de cualquier orden, quedan un total de 35 ecuaciones resueltas por métodos que se consideran de calidad. El porcentaje esta vez se queda en un 39,77%, un descenso considerable respecto al anterior pero que refleja mejor la potencia real de los métodos que se prueban.

¿Cuántas ecuaciones pueden resolverse por simetrías de Lie?

Esta vez se tienen en cuenta las ecuaciones de las que se ha podido obtener una solución con el método en el que este trabajo se centra, las simetrías de Lie, independientemente de si también son resolubles mediante otras aproximaciones. El número de ecuaciones que satisfacen esta pregunta se reduce a 22, lo que representa un 25% de las ecuaciones del conjunto. Esto es, en una de cada cuatro ecuaciones es viable aplicar con éxito las simetrías de Lie.

¿Cuántas ecuaciones son resolubles por reducción a cuadraturas?

En varios casos, se obtiene que el método de simetrías de Lie no es capaz de hallar la solución, pero bien por el hecho de que emplean un tiempo considerado como infinito en las tablas adjuntas o bien porque surge un error de ejecución. Es por ello que, probando en estos casos la búsqueda de simetrías, en 58 de ellos la función *infinitesimals* es capaz de hallarlas.

Esto es una ventaja, ya que el hecho de encontrar las simetrías de una ecuación nos garantiza que se puede hallar una solución, al menos en función de cuadraturas. Esto es posible porque, teniendo las simetrías, es posible hallar un factor integrante que transforme la ecuación en una de tipo exacta, que por definición son al menos reducibles a cuadraturas o directamente resolubles.

Por todo ello, considerando estos casos como ecuaciones resueltas por el método de Lie, la cifra total asciende a 58, conformando un 62,5%. Es una importante consideración a tener en cuenta, y ayuda a comprender aún mejor la gran aportación de Lie.

¿Cuántas ecuaciones son únicamente resolubles por simetrías de Lie?

Este criterio es mucho más restrictivo, pero ayuda a comprender la importancia del método de Lie en la resolución de ecuaciones diferenciales. Sólo en 5 de las 88 ecuaciones (un 5,68%) se obtiene su solución únicamente mediante Lie. En este apartado

también se han descartado las series de potencias por las razones anteriormente expuestas.

A priori el porcentaje obtenido en este apartado puede resultar muy bajo, pero su verdadero valor reside en que nos permite deducir que el método de Lie es imprescindible para la librería *Sympy*, al existir casos exclusivamente resueltos con dicha metodología. Una de cada 12 ecuaciones no podría ser resuelta sin la existencia de este método.

¿Cuántas ecuaciones no son resolubles?

Del resto de cuestiones se puede inferir fácilmente esta respuesta. Tras las pruebas, no se ha conseguido obtener por ningún método la solución de 28 de las ecuaciones del conjunto Kamke88, que se corresponde con un 31,82%. Si consideramos igualmente como no resueltas aquellas ecuaciones que sólo se resuelven con series de potencias, el número de ecuaciones sin resolver se eleva a 53 (60,23%) que es un porcentaje más cercano a las estimaciones previas y refleja el margen de mejora que hay por implementar en esta librería.

De la cuestión anterior podemos obtener otra conclusión que pone en valor el método de Lie. Como se ha mencionado, sin la existencia del mismo tendríamos otras 5 ecuaciones sin resolver, llevando el porcentaje hipotético de no resueltas hasta el 65,91% (58 ecuaciones) del conjunto.

RESULTADO	Número de ecuaciones	Porcentaje
Resueltas por simetrías de Lie	22	25,00%
Resueltas pero no por simetrías de Lie	13	14,77%
Resueltas solo con series de potencias	25	28,41%
No resolubles por ningún método	28	31,82%
TOTAL	88	100,00%

Tabla 5. Resumen del Test 1 incluyendo series de potencias.

RESULTADO	Número	Porcentaje
Resueltas por simetrías de Lie	22	25,00%
Resueltas pero no por simetrías de Lie	13	14,77%
No resolubles por ningún método*	53	60,23%
TOTAL	88	100,00%

Tabla 6. Resumen de resultados excluyendo 1st_power_series.

Test 2 - Importancia de las heurísticas de Lie

Esta segunda prueba persigue el objetivo de comparar entre sí las nueve heurísticas que permite utilizar *Sympy* a la hora de obtener el grupo infinitesimal durante la resolución de una ecuación con simetrías de Lie.

De las 88 ecuaciones que componen el test, se obtuvieron 22 resolubles por simetrías de Lie. Es con estas 22 ecuaciones con las que se han probado todas las heurísticas en este test. Las preguntas que se han formulado dirigiendo los ensayos para lograr los resultados son las siguientes.

¿En cuántas ecuaciones halla la simetría cada heurística?

En la tabla siguiente se muestran qué heurísticas son capaces de hallar las simetrías para cada una de las ecuaciones. Se ha indicado con un “1” las heurísticas que funcionan para la ecuación de esa fila, y con un “0” aquellas que no calculan las simetrías para dicha EDO. Asimismo, para facilitar su visualización se han destacado en tonos amarillos las casillas escritas con “1”.

Nótese que existen tres casillas marcadas con un tono más intenso de amarillo. Se tratan de las tres ecuaciones con heurísticas exclusivas para hallar sus simetrías. Más adelante en la tercera cuestión se amplían los detalles sobre estas ecuaciones particulares.

Se observa en la tabla que tres de las nueve heurísticas no logran encontrar las simetrías para ninguna de las ecuaciones, por lo que son completamente prescindibles para esta muestra. Estas son *abaco2_unique_general*, *linear* y *function_sum*. Incluimos asimismo aquí la heurística *abaco2_unique_unknown* que solo funciona con una de las ecuaciones. Sin embargo, los *infinitesimals* de dicha ecuación pueden ser halladas también por otro método, así que resulta ser completamente sustituible para este set.

Las heurísticas *chi* y *abaco1_product*, a pesar de ser válidas con 6 y 9 ecuaciones respectivamente, un porcentaje no desdeñable, son al igual que la mencionada *abaco2_unique_unknown* completamente reemplazables por las demás.

#	abaco1 simple	abaco1 product	abaco2 similar	abaco2 unique unk0wn	abaco2 unique general	linear	function sum	bivariate	chi
1	OK								
2	OK	OK							
3	OK	OK							
4	OK	OK							
5	OK	OK							
6	OK	OK							
7	OK	OK							
8	OK	OK							
9	OK		OK						
15			OK					OK	OK
17	OK		OK					OK	OK
18		OK						OK	
19			OK					OK	OK
26	OK		OK					OK	OK
29	OK		OK					OK	OK
31	OK		OK						
42								OK	
44		OK						OK	OK
60	OK		OK						
75			OK						
76	OK		OK						
84			OK	OK					

Tabla 7. Heurísticas de Lie y ecuaciones para las que pueden hallar las simetrías.

¿Qué heurísticas son las más importantes?

Las tres heurísticas restantes resultan ser las más importantes e imprescindibles.

Abaco1_simple es capaz de encontrar la simetría para un total de 15 ecuaciones, es decir, un 68,2% de las mismas. Es la heurística más importante de todas con gran diferencia.

La segunda heurística más importante es *abaco2_similar*, que sirve para resolver un 50% de las ecuaciones (11 en total).

En tercer lugar, aparece la heurística *bivariate*, cuyas asunciones dan buenos resultados en 8 de las 22 ecuaciones, lo que corresponde a un 36,3% del total. A pesar de ser capaz de resolver una ecuación menos que *abaco1_product*, ésta última es totalmente prescindible para este conjunto como se ha indicado anteriormente.

¿Hay casos de métodos exclusivos para una ecuación?

Sí, tal y como se introducía en la primera cuestión, las tres heurísticas señaladas como más importantes reúnen una característica común y es la presencia de ecuaciones cuyas simetrías calculan de manera exclusiva. Cada una de las tres se corresponde con una ecuación diferente siendo el único método posible para ella. En la tabla se han resaltado dichas ecuaciones exclusivas en un tono más intenso de amarillo.

Como resultado complementario a este análisis, se indican también cuántas de las heurísticas resuelven cada ecuación, así como cuántas de las tres más importantes resuelven cada ecuación.

RESULTADO	abaco1_simple	abaco1_product	abaco2_similar	abaco2_unique_unk0wn
En cuántas halla las simetrías	15	9	11	1
Cuántas halla exclusivamente	1	0	1	0

Tabla 8. Resultados del test 2 para cada heurística

RESULTADO	abaco2_unique_general	linear	function_sum	bivariate	chi
En cuántas halla las simetrías	0	0	0	8	6
Cuántas halla exclusivamente	0	0	0	1	0

Tabla 9. Resultados del test 2 para cada heurística

Número de EDO	Métodos válidos	Métodos principales válidos
1	1	1
2	2	1
3	2	1
4	2	1
5	2	1
6	2	1
7	2	1
8	2	1
9	2	2
15	3	2
17	4	3
18	2	1
19	3	2
26	4	3
29	4	3
31	2	2
42	1	1
44	3	1
60	2	2
75	1	1
76	2	2
84	2	1

Tabla 10. Resultados del segundo test para cada ecuación.

Test 3 – Comparación del método de Lie con el resto de procedimientos

En el tercer test, se espera obtener información en forma de comparativa entre las simetrías de Lie y los demás métodos de resolución de ecuaciones diferenciales existentes.

Para ello, se ha ejecutado el programa de ensayos para resolver todas las ecuaciones con cada uno de los métodos a priori abordables para esa ecuación de entre los programados en esta librería.

Los diferentes puntos en los que se ha decidido enfrentar unos métodos y otros se resumen a continuación.

¿Cuántos métodos inicialmente posibles se proponen?

Para una primera aproximación al problema, se muestran los métodos que la librería señala como factibles para abordar la resolución de cada ecuación. Como puede observarse en las tablas 14 y 15 del Anexo IV, las ecuaciones más sencillas pueden ser resueltas a priori por una gran variedad de metodologías, mientras que según avanzamos a través del conjunto Kamke88 las ecuaciones comienzan a tener menos opciones de resolución. En algunos casos, sólo un par de caminos parecen conducir a la solución.

Las mencionadas tablas han sido obtenidas utilizando la función *classify_ode* en las 88 ecuaciones del conjunto de muestra. Tal y como se indica en el apartado de descripción de la librería Sympy, los resultados de esta función no indican todos los métodos que pueden realmente resolver la ecuación. Para ello, ha de ejecutarse *dsolve* y observar las diferencias, lo cual se ha hecho para responder la pregunta siguiente.

¿Cuántas ecuaciones resuelve cada método?

Fijando la mira ahora en las tablas 16 y 17, presentes en el Anexo IV, corregidas tras utilizar *dsolve* con todo el conjunto de ecuaciones propuesto, pueden visualizarse cuantos métodos son realmente efectivos a la hora de resolver las mismas. Con los resultados de los test anteriores, se sabe que al menos 52 de las ecuaciones no son resolubles (descartando las soluciones en series de potencias), por lo que los métodos propuestos inicialmente no eran válidos en estos casos. En el resto, en ocasiones todos los *hints* obtenidos con *classify_ode* sirven para obtener la solución y en otras solo algunos de ellos. La tabla corregida se expone a continuación.

¿Qué método es el más recurrente o importante?

Observando la tabla corregida vemos que, de nuevo descartando las series de potencias, el método más recurrente es sorprendentemente el de simetrías de Lie, representado por el método *lie_group*, con un total de 22 ecuaciones resueltas como previamente habíamos indicado. Le siguen de cerca las clasificaciones *separable* y *separable_Integral* con 19 casos resueltos adecuadamente, algo entendible teniendo en cuenta que todas las ecuaciones del set están escritas en forma normal, por lo que en los casos en los que el segundo miembro sea lineal en la función incógnita o dependa únicamente de la variable independiente, serán ecuaciones de tipo separable.

¿Qué otros métodos destacan por su efectividad?

Cabe destacar en este rol las clasificaciones de *1st_exact* y *1st_exact_Integral*, con 18 casos resueltos. Junto a ellas, aparecen también *Bernoulli* y *linear*, junto a sus dos versiones *Integral*, resolviendo 14 y 12 ecuaciones respectivamente.

Con estos resultados podemos esbozar una idea de los métodos más importantes en esta librería, siempre relativos a la muestra con la que se está realizando el experimento de *Benchmark*.

¿Qué papel juegan las simetrías de Lie con respecto al resto de métodos?

De la observación de la tabla que recoge las clasificaciones previas de cada ecuación se puede deducir que el método de simetrías de Lie es uno de los más propuestos para resolver las ecuaciones del set, estando presente en todas y cada una de las 88 clasificaciones. Esto lo convierte en una poderosa herramienta a tener en cuenta, pues, a pesar de que no siempre funciona, deja claro que resulta útil buscar las simetrías de una ecuación para intentar simplificarla.

Posteriormente, en la tabla corregida tras ejecutar *dsolve* vemos que el porcentaje de ecuaciones resueltas con *lie_group* se reduce a un 25% (22 ecuaciones). Aun así, tal y como se ha indicado en la primera de las cuestiones de este tercer test, sigue siendo la metodología más importante frente a las demás.

Ecuaciones exclusivas del método de Lie

En los resultados del primer test se puede advertir la presencia de 5 ecuaciones singulares. Destacan por ser ecuaciones del test que únicamente son resueltas por el método de Lie. Por tanto, de no poder aprovechar las simetrías para buscar su solución, ésta no podría ser hallada mediante ningún otro método. Se les dedica pues un pequeño apartado para mostrar cuáles son, qué simetrías ha encontrado cada una de las heurísticas con ellas y finalmente su solución. Todo ello se ilustra en las siguientes dos tablas que recogen esta información.

#	Heurística	Simetrías	
15	abaco2_similar	xi	1
		eta	$2*x$
	bivariate	xi	1
		eta	$2*x$
	chi	xi	$-1/2$
		eta	$-x$
18	abaco1_product	xi	0
		eta	$(y+1)**2*\exp(-x**2/2+2*x)$
	bivariate	xi	1
		eta	$x*y+x+y**2-1$
19	abaco2_similar	xi	1
		eta	-1
	bivariate	xi	1
		eta	-1
	chi	xi	$-1/2$
		eta	$1/2$
42	abaco2_similar	xi	1
		eta	-1
	bivariate	xi	-1
		eta	1
	chi	xi	-1
		eta	1
84	abaco2_similar	xi	1
		eta	$-a/b$

Tabla 11. Heurísticas y simetrías de las EDOS exclusivas de Lie.

Desarrollo de herramientas de soporte a un curso OCW-UPM sobre resolución de ecuaciones diferenciales mediante simetrías de Lie

En el caso de la solución de la ecuación número 42 del Kamke88, por tener una expresión muy grande se ha decidido mostrarla en una figura aparte para no afectar al tamaño de la tabla global.

#	Ecuación	Solución
15	$x^4 - 2x^2y - 2x + y^2 + y' - 1 = 0$	$y = \frac{-x^2 e^{-2C_1 - 2x} + x^2 + e^{-2C_1 - 2x} + 1}{1 - e^{-2C_1 - 2x}}$
18	$-xy - x - y^2 + y' + 1 = 0$	$y = \frac{-C_1 e^{2x} + e^{2x} \int \frac{\sqrt{e^{x^2}}}{e^{2x}} dx + \sqrt{e^{x^2}}}{\left(C_1 - \int \frac{\sqrt{e^{x^2}}}{e^{2x}} dx\right) e^{2x}}$
19	$y' = (x + y)^2$	$y = -x + \tan(C_1 + x)$
42	$axy^3 + by^2 + y' = 0$	FIGURA APARTE
84	$y' = f(ax + by)$	$y = \frac{C_1 a^2 + b^2 \left(C_1 - a \left(\int^{-C_2 - \frac{ax}{b}} \frac{f(-rb)}{a + bf(-rb)} dr\right) + b \left(\int^{-C_2 - \frac{ax}{b}} \frac{1}{a + bf(-rb)} dr\right) + x\right)}{ab}$

Tabla 12. EDOS exclusivas y sus soluciones.

$$\begin{aligned}
 & \left(-\frac{b}{\sqrt{4a+b^2}} + 1 \right) \log \left(xy + \frac{-6a^2 \left(-\frac{b}{\sqrt{4a+b^2}} + 1 \right)^2 - 6a^2 \left(-\frac{b}{\sqrt{4a+b^2}} + 1 \right) + 12a^2 - \frac{7ab^2 \left(-\frac{b}{\sqrt{4a+b^2}} + 1 \right)^2}{2} - \frac{3ab^2 \left(-\frac{b}{\sqrt{4a+b^2}} + 1 \right)}{2} + 11ab^3 - \frac{b^4 \left(-\frac{b}{\sqrt{4a+b^2}} + 1 \right)^2}{2} + 2b^4}{ab(9a+2b^2)} \right) \\
 & C_1 + \frac{2}{2} \\
 & \left(\frac{b}{\sqrt{4a+b^2}} + 1 \right) \log \left(xy + \frac{-6a^2 \left(\frac{b}{\sqrt{4a+b^2}} + 1 \right)^2 - 6a^2 \left(\frac{b}{\sqrt{4a+b^2}} + 1 \right) + 12a^2 - \frac{7ab^2 \left(\frac{b}{\sqrt{4a+b^2}} + 1 \right)^2}{2} - \frac{3ab^2 \left(\frac{b}{\sqrt{4a+b^2}} + 1 \right)}{2} + 11ab^3 - \frac{b^4 \left(\frac{b}{\sqrt{4a+b^2}} + 1 \right)^2}{2} + 2b^4}{ab(9a+2b^2)} \right) \\
 & + \frac{2}{2} - \log(xy) = 0
 \end{aligned}$$

Figura 12. Solución de la ecuación 42 del conjunto Kamke.

Ecuaciones exclusivas reducibles a cuadraturas

Se añade en esta sección el criterio comentado en el Test 1 de este capítulo. Aquellas ecuaciones que, a pesar de no poder hallar la solución en primera instancia a través del método *dsolve*, el método de Lie sí encuentra alguna simetría, por lo que podrían ser reducibles a cuadraturas y por ello, consideradas como resueltas. De los 58 casos indicados anteriormente, hay 23 ecuaciones exclusivas del método de Lie con cuadraturas.

Conclusión

Por todo lo expuesto en los apartados anteriores, se deduce que...

El lenguaje de programación Python a través de la librería **Sympy**, cuenta con un gran número de métodos ya implementados para el tratamiento, clasificación y resolución de ecuaciones diferenciales ordinarias. Se ha aprovechado esta **potencia** para diseñar una interfaz con la librería estándar **Tkinter** que permita a los usuarios entrar en contacto con el método de simetrías de Lie de una forma interactiva.

Por otro lado, se han ejecutado una serie de test, cuyos resultados ilustran la potencia comentada de la librería **Sympy** para un conjunto de ecuaciones determinado. En concreto, se ha buscado comparar la utilidad del método de simetrías de Lie frente al resto de procedimientos tradicionales. Los resultados obtenidos son sorprendentemente buenos, consiguiendo resolver 22 de las ecuaciones con éxito (un 25%), contando además con la presencia de 5 ecuaciones exclusivas que no podrían ser resueltas sin utilizar sus simetrías. Si además añadimos aquellas ecuaciones que se han resuelto reduciéndolas a cuadraturas, el porcentaje asciende a un 62% (58 ecuaciones). Para 28 de ellas, el método de Lie es el único que es capaz de hallar la solución explícita o reducida a cuadraturas.

6. Contenidos docentes

Criterios de diseño

En el diseño de estos contenidos docentes se han seguido los siguientes criterios:

- **Flexibilidad.** Se ha intentado que los materiales elaborados, incluyendo las lecciones magistrales, sean adecuados tanto para un curso oficial, con docencia presencial u online, como directamente, o con una mínima adaptación, a un **MOOC**.
- **Enfoque Práctico.** Aunque la materia objeto de estudio tiene, de forma natural, cierto nivel de abstracción, se ha intentado enfocar desde un punto de vista práctico, reduciendo los contenidos teóricos y teoremas a los más esenciales y potenciando los resultados directamente aplicables a la resolución de ecuaciones diferenciales. Alineado con este enfoque, en cada sesión formativa se incluyen actividades de resolución de problemas y realización de prácticas con o sin soporte de una aplicación informática.
- **Potenciación del autoaprendizaje.** En cada sesión formativa se incluyen actividades (vídeos, resolución de ejercicios, prácticas con o sin soporte de una aplicación informática, cuestionarios) que deben ser resueltos por el alumno de forma autónoma.

Se sigue un enfoque “*learning by doing*”, en la que la labor del profesor es de orientación y apoyo en la resolución de dudas, con un enfoque mayéutico, por ejemplo, con la resolución de cuestionarios en cada sesión formativa.

Esta potenciación del autoaprendizaje, incluso en cursos oficiales presenciales, está en la línea de facilitar el desarrollo futuro de un MOOC.

- **Evaluación y autoevaluación continua.** En cada sesión formativa se propone la realización de un cuestionario a realizar por el alumno. Con esto se pretende, con un enfoque mayéutico, reforzar la asimilación de los conceptos y métodos con los que se ha trabajado en esa sesión.

Asimismo, al finalizar cada módulo formativo, se ha planificado una sesión completa para una prueba de evaluación continua (PEC).

Se pretende que la evaluación del alumno sea literalmente continua.

- **Limitación en el alcance.** La carga de trabajo necesaria para el desarrollo de los contenidos propuestos hace inviable que su desarrollo sea completo y exhaustivo en el ámbito de un trabajo fin de grado. Como consecuencia, se ha decidido desarrollar con la mayor profundidad y detalle posible solo alguno de los contenidos, postergando el desarrollo del resto de los contenidos a futuros trabajos.

Estructura modular

El curso se estructura en 4 módulos formativos. Cada módulo comprende, en el caso de curso oficial, 4 sesiones de 100 minutos de duración con apoyo del profesor más la dedicación correspondiente, de manera autónoma, por el alumno. El curso corresponde, de esta manera, a 3 ECTS.

Se propone en base a las referencias de, entre otros, [15] y [16], el siguiente contenido preliminar para cada uno de los 4 módulos:

1. Módulo 1. Simetría en Matemáticas.

- Concepto de Simetría.
- Teoría Elemental de Grupos.
- Ejemplos elementales de grupos de transformaciones en la línea y en el plano.
- Introducción a la Teoría de Galois.
- Orígenes y evolución de la Teoría de Grupos Continuos de Lie.
- Paralelismos y diferencias entre la Teoría de Galois y la Teoría de Lie.

2. Módulo 2. Generadores Infinitesimales.

- Generador infinitesimal asociado a un Grupo Continuo de transformaciones. Las “greeks” ξ y η . Órbitas.
- Tabulación de transformaciones
- Familia de curvas invariantes. Operador infinitesimal.
- Variables Canónicas.
- Grupo Extendido de transformaciones
- Métodos para la determinación de un Grupo Discreto asociado a un Grupo Infinitesimal de transformaciones: Tablas, ecuaciones determinativas (ecuaciones de Lie), mapeo exponencial, variables canónicas.

3. Módulo 3. Resolución de EDO's de 1^{er} orden mediante simetrías de Lie.

- EDO's de 1^{er} orden invariantes ante un Grupo de Transformaciones. Tabulación
- Cálculo de simetrías de una EDO de 1^{er} orden I: tablas.
- Cálculo de simetrías de una EDO de 1^{er} orden II. EDP asociada. Heurísticas. Sistemas CAS: librería *Sympy*.
- Resolución mediante factor integrante de Lie. Teorema de Lie. Soluciones singulares.
- Inverso del Teorema de Lie: consecuencias.
- Resolución mediante coordenadas canónicas. EDO separable asociada.

4. **Módulo 4. Resolución de EDO's de 2º orden mediante simetrías de Lie**

- El símbolo η_1
- Cálculo de simetrías de una EDO de 2º orden I: tablas
- Cálculo de simetrías de una EDO de 2º orden II: EDP asociada. Heurísticas. Sistemas CAS.
- Integral primera.
- Otras aplicaciones de la teoría de Lie.

En base a la experiencia acumulada, en un futuro los módulos 1 y 2 podrían simplificarse y unificarse dando cabida, de esta manera, a un módulo específico para la resolución de EDP's mediante simetrías de Lie.

Sesiones y actividades

Se propone una estructura, para cada módulo, de 3 sesiones lectivas con soporte del profesor más una sesión dedicada a una PEC. Totalizando, de esta forma el curso, 12 sesiones lectivas y 4 PEC's.

Para las 12 sesiones lectivas se propone la misma estructura de actividades:

1. Actividad 1. Lección magistral. Duración aproximada: 25 minutos.
2. Actividad 2. Resolución de ejercicios por el profesor.
DESCANSO (10 minutos)
3. Actividad 3. Vídeo. Contendrá nuevos contenidos o reforzará contenidos de sesiones previas. Duración aproximada: 10 minutos.
4. Actividad 4. Práctica. Realizada por los alumnos con la supervisión del profesor. Cuando los contenidos lo posibiliten, se utilizarán herramientas informáticas de soporte, preferentemente las desarrolladas a partir del presente trabajo. Duración aproximada: 30 minutos.
5. Actividad 5. Cuestionario. En los cursos oficiales, serán cuestionarios desarrollados en Moodle. Para los MOOC sería recomendable el desarrollo de una plataforma específica en Internet.

Contenidos desarrollados

































































Como se ha indicado en los criterios de diseño, el desarrollo de todos los contenidos propuestos no es posible dentro del alcance de un trabajo fin de grado.

En la siguiente tabla se resume el material ya disponible, en algunos casos como una versión muy preliminar, pendiente de mejora:



ACTIVIDAD	TOTAL (en el curso)	DISPONIBLES	AVANCE
Lección magistral	12	5	17%
Resolución de ejercicios	12	4	33%
Vídeo	8	0	0%
Práctica	12	1	8%
Cuestionario	12	1	8%
Resolución completa colección de ejercicios	4	1	25%




Tabla 13. Resumen del material y su nivel de desarrollo.

En el siguiente gráfico se detalla el avance en el desarrollo de los contenidos a nivel de módulo, sesión y actividad. En el Anexo I, se incluyen los contenidos desarrollados para el módulo 2.

MODULO	S1	S2	S3	S-PEC
1 Simetría en Matemáticas	 L1.1  E1.1 <hr/>  V1.1  P1.1  Q1.1	 L1.2  E1.2 <hr/>  V1.2  P1.2  Q1.2	 L1.3  E1.3 <hr/>  V1.3  P1.3  Q1.3	 PEC 1
2 Generadores Infinitesimales	 L2.1  E2.1 <hr/>  V2.1  P2.1  Q2.1	 L2.2  E2.2 <hr/>  V2.2  P2.2  Q2.2	 L2.3  E2.3 <hr/>  V2.3  P2.3  Q2.3	 PEC 2
3 Resolución de ED0's de 1^{er} orden mediante Simetrías de Lie	 L3.1  E3.1 <hr/>  V3.1  P3.1  Q3.1	 L3.2  E3.2 <hr/>  V3.2  P3.2  Q3.2	 L3.3  E3.3 <hr/>  V3.3  P3.3  Q3.3	 PEC 3
4 Resolución de ED0's de 2^o orden mediante Simetrías de Lie	 L4.1  E4.1 <hr/>  V4.1  P4.1  Q4.1	 L4.2  E4.2 <hr/>  V4.2  P4.2  Q4.2	 L4.3  E4.3 <hr/>  V4.3  P4.3  Q4.3	 PEC 4

Leyenda Símbolos Actividad

 L. Lección Magistral
 E. Ejercicios

 V. Video
 P. Práctica
 Q. Cuestionario

Avance Actividad

Desarrollada
 Parcial
 Pendiente

Figura 13. Avance en los contenidos docentes a nivel de módulo, sesión y actividad.

Conclusión

En base a lo expuesto en los apartados anteriores en relación a los contenidos docentes, concluimos que...

Dentro del presente trabajo se han definido los contenidos de un curso oficial, presencial u online. Se han seguido unos criterios de diseño que, entre otras cosas, simplifican la adaptación de este curso oficial a un MOOC.

Se ha definido una estructura para el curso basado en 4 módulos, que constan conjuntamente de 12 sesiones y 4 PEC's. Las sesiones constan de **distinto tipo de actividades**: clases magistrales (12), resolución de ejercicios (12), vídeos (8), prácticas (12) y cuestionarios de evaluación (12).

Aunque el **desarrollo completo y detallado de todos los contenidos planificados** queda fuera del alcance del presente trabajo, se han desarrollado varios ejemplos de cada uno de los tipos de actividades. Con los contenidos desarrollados se cubre, totalmente, el módulo 2: "Resolución de EDO's de 1er orden mediante simetrías de Lie".

7.Herramientas de soporte al curso

En este capítulo se exponen las herramientas diseñadas para dar soporte a los contenidos docentes anteriormente desglosados. Para ello, se incluye la descripción de una aplicación *standalone*, esto es, ejecutable en el propio equipo sin necesidad de acceso a internet.

Se trata de una primera aproximación al uso de herramientas software como apoyo al aula. En el mejor de los casos, es deseable disponer de una herramienta web, que puede ser utilizada como soporte del curso MOOC y tiene infinidad de usos y aplicaciones en la docencia. Sin embargo, el desarrollo de herramientas web requiere más recursos, mayor formación para su creación y mantenimiento. También presentan algunas limitaciones en cuanto a la funcionalidad que pueden ofrecer. Es por esto que se ha elegido desarrollar una aplicación de escritorio, que dispone de toda la potencia del lenguaje y, para suplir su falta de ubicuidad frente a una herramienta web, será distribuida a los usuarios a través de la plataforma Moodle del curso.

Espacio Moodle

Tal y como se viene indicando en previas citas, se ha creado con el objetivo de albergar los contenidos docentes desarrollados en el capítulo anterior, un curso en el espacio Moodle oficial de la Universidad Politécnica de Madrid.

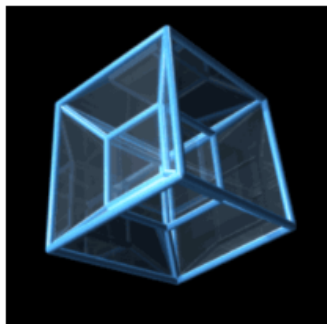
Se trata de un primer paso en la publicación de dicho material docente en la web. Al principio estará disponible únicamente para alumnos UPM, pero más adelante se abrirá este curso al público convirtiéndolo en uno de los descritos *Massive Online Open Course* dentro de la iniciativa OCW de la UPM.

Mediante esta adaptación se pretende hacer llegar estos contenidos a un público mayor y ponerlos a disposición de toda la comunidad universitaria de manera gratuita.

Se ha dividido el espacio Moodle en los cuatro módulos de que dispone el curso diseñado. Dentro de ellos se han habilitado diferentes secciones según el tipo de material que albergarán. Es por ello que encontramos apartados separados para cada lección y cada cuestionario de autoevaluación, así como una carpeta para las Pruebas de Evaluación Continua (PEC).

Las siguientes imágenes ilustran el aspecto actual de este curso en el espacio Moodle de la universidad, así como un ejemplo de las preguntas del primer cuestionario del módulo número 2.

MODULO 1. Importación de la SIMETRÍA en la resolución de EDOs



- LECCION 1.1. ¿Qué es la SIMETRÍA? Papel que juega en la resolución de EDOs
- LECCION 1.2. GRUPOS de SIMETRÍAS
- LECCION 1.3. Grupos INFINITESIMALES Uniparametricos. Generador
- PEC 1

MODULO 2. GENERADORES INFINITESIMALES

- LECCION 2.1. Órbitas. TRANSFORMACIONES INFINITESIMALES. Problema inverso: Ecuaciones de Lie y Mapeo Exponencial.
- ✓ Q 2.1. Generadores Infinitesimales
- LECCION 2.2. PROBLEMA INVERSO. Ejemplos. INVARIANTES. Variables Canónicas.
- ✓ Q 2.2. Problema Inverso
- LECCION 2.3. ALGEBRAS de LIE
- PEC 2
- ✓ Q 2.3. Algebras de Lie

Figura 14. Módulos 1 y 2 del curso en Moodle UPM.

OCW. Lie Symmetries & ODEs

[Área personal](#) / [Mis cursos](#) / [OCW. Lie Symmetries & ODEs](#) / [MODULO 2. GENERADORES INFINITESIMALES](#) /

Pregunta 1

Intentos
restantes: 1

Calificado sobre
2.00

🚩 Marcar
cuestión

⚙ Editar
pregunta

¿Un **grupo continuo** puede tener un número finito de transformaciones?

Seleccione una:

☒ Verdadero

☐ Falso

comprobar

Pregunta 2

Intentos
restantes: 1

Calificado sobre
2.00

🚩 Marcar
cuestión

⚙ Editar
pregunta

¿Si un grupo tiene un número infinito de transformaciones es continuo?

Seleccione una:

☐ Verdadero

☐ Falso

comprobar

Figura 15. Cuestionario 2.1 del curso en Moodle UPM.

Herramienta DELHI

El software *standalone* creado se presenta con el nombre de **DELHI**, proveniente de “ODE Lie”. En ella se han incluido varias utilidades que permiten al usuario entrar en contacto con el método de resolución de **simetrías de Lie** de diferentes maneras. También puede ser utilizada para resolver problemas, o comprobar la solución de los mismos si han sido resueltos a mano.

DELHI ha sido desarrollada en el lenguaje de programación **Python**, tanto la interfaz gráfica, utilizando la librería *Tkinter*, como los algoritmos que la gobiernan y que permiten resolver ecuaciones diferenciales con la librería *SymPy*.

Las funcionalidades principales que DELHI ofrece se describen brevemente en los próximos párrafos. Posteriormente, en el Anexo II se puede encontrar la **guía del usuario**, que hace las veces de manual de instrucciones de uso de la aplicación. Por añadido, en el Anexo III se encuentra la **guía de referencia**, que contiene una explicación de la estructura del proyecto a nivel de software.

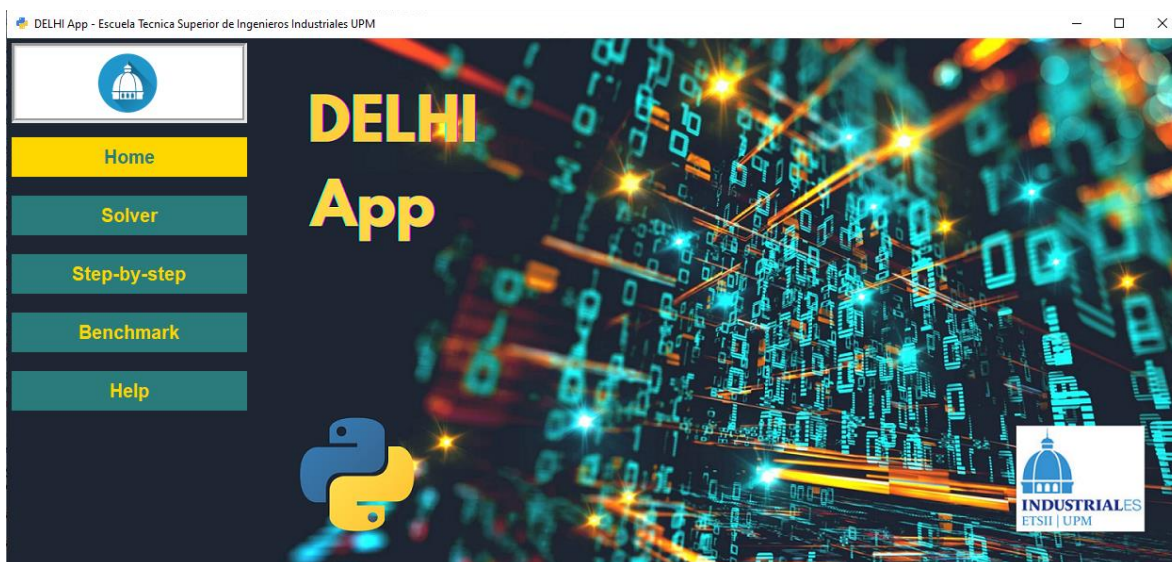


Figura 16. Herramienta DELHI.

Solver

La primera de las utilidades de la aplicación permite al usuario introducir una ecuación diferencial y resolverla utilizando las simetrías de Lie siempre que sea posible. En caso contrario, se utilizará un método diferente, de entre las categorías en las que la ecuación sea clasificada, en aras de obtener la solución.

El valor de esta funcionalidad reside en que está diseñada como una calculadora para conocer rápidamente la solución de una ecuación diferencial sin profundizar en los detalles del proceso.

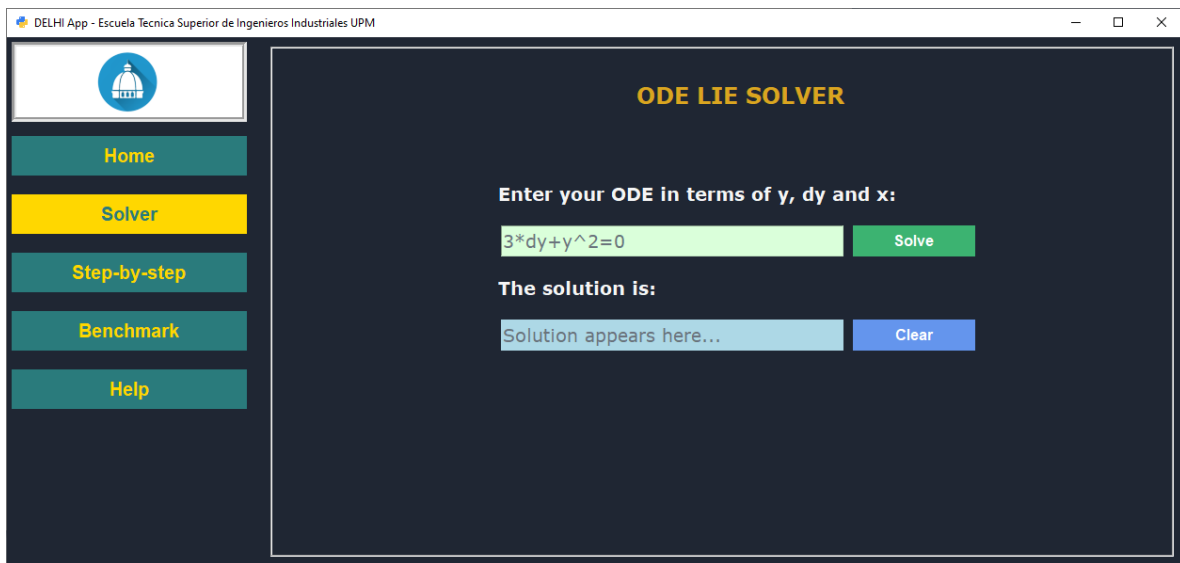


Figura 17. Solver de DELHI.

Step-by-Step

En segundo lugar, la aplicación dispone de una versión más detallada del anterior “Solver”, llamado “Step by Step”. Aquí, el proceso de resolución se ha dividido en varias fases, teniendo cada una de ellas una pantalla propia para interactuar y conocer los detalles.

En la primera de las fases, DELHI permite al usuario hallar el grupo infinitesimal de la ecuación diferencial introducida, siempre que sea resoluble mediante simetrías de Lie. La ventana ofrece una serie de opciones para abordar el problema con diferentes heurísticas y observar las diferencias entre ellas. Este apartado es útil cuando no interesa obtener directamente la solución de la EDO, pero sí se necesita conocer más detalles del método.

Para la segunda fase del proceso de resolución, la aplicación ofrece dos vías alternativas: el cálculo del factor integrante de Lie para transformar la ecuación original en una ecuación exacta, y hallar la solución; o su cambio a coordenadas canónicas, que convierten la ecuación introducida en una ordinaria separable, de fácil resolución. Ambos casos permiten al usuario finalmente conocer la solución de la ecuación habiendo completado previamente el primer paso.

En el caso de optar por calcular el factor integrante, el usuario deberá elegir un grupo infinitesimal de entre los obtenidos en la primera fase del proceso para aplicarlos en la resolución. Se mostrará el factor integrante obtenido, así como la ecuación exacta a la que da lugar. Finalmente se calcula la solución.

En la ventana correspondiente al método de coordenadas canónicas, igualmente el usuario deberá escoger de entre las simetrías obtenidas en el paso 1. Al resolver por este camino, se mostrará el cambio de variable efectuado, la ecuación ordinaria de tipo separable que se ha obtenido y por último, también la solución de la EDO original.



Figura 18. Step-by-Step de DELHI, detalle del paso 1.

Desarrollo de herramientas de soporte a un curso OCW-UPM sobre
resolución de ecuaciones diferenciales mediante simetrías de Lie

The screenshot shows the DELHI App interface. On the left is a sidebar with a logo and buttons: Home, Solver, Step-by-step (highlighted), Benchmark, and Help. The main area has tabs for STEP 1, STEP 2A, and STEP 2B. STEP 2A is active and titled "STEP 2A - LIE INTEGRATING FACTOR". It contains a text input for the differential equation $dy/dx = -(y^2)/3$, a "Solve" button, and a "Symmetry" dropdown set to "None" with a "Clear" button. Below this, it says "Method A: Integrating factor transformation". A list of steps follows: "2 - Integrating factor:" with a formula $\mu(x, y) = \frac{1}{(M\xi + N\eta)}$; "3 - Substituted integ. factor:" with a text input "Integrating factor..."; "4 - Exact ODE:" with a text input "Exact ODE..."; and "5 - The solution is:" with a text input "Solution appears here...". A "Check solution" button is at the bottom.

Figura 19. Step-by-Step de DELHI, detalle del paso 2.

The screenshot shows the DELHI App interface. On the left is a sidebar with a logo and buttons: Home, Solver, Step-by-step (highlighted), Benchmark, and Help. The main area has tabs for STEP 1, STEP 2A, and STEP 2B. STEP 2B is active and titled "STEP 2B - CANONICAL COORDINATES". It contains a text input for the differential equation $dy/dx = -(y^2)/3$, a "Solve" button, and a "Symmetry" dropdown set to "None" with a "Clear" button. Below this, it says "Method B: Canonical variable change". A list of steps follows: "2. Coordinate x1:" with a text input "x1..."; "3. Coordinate y1:" with a text input "y1..."; "4. Separable ODE:" with a text input "Separable ODE..."; and "5. The solution is:" with a text input "Solution appears here...". A "Check solution" button is at the bottom.

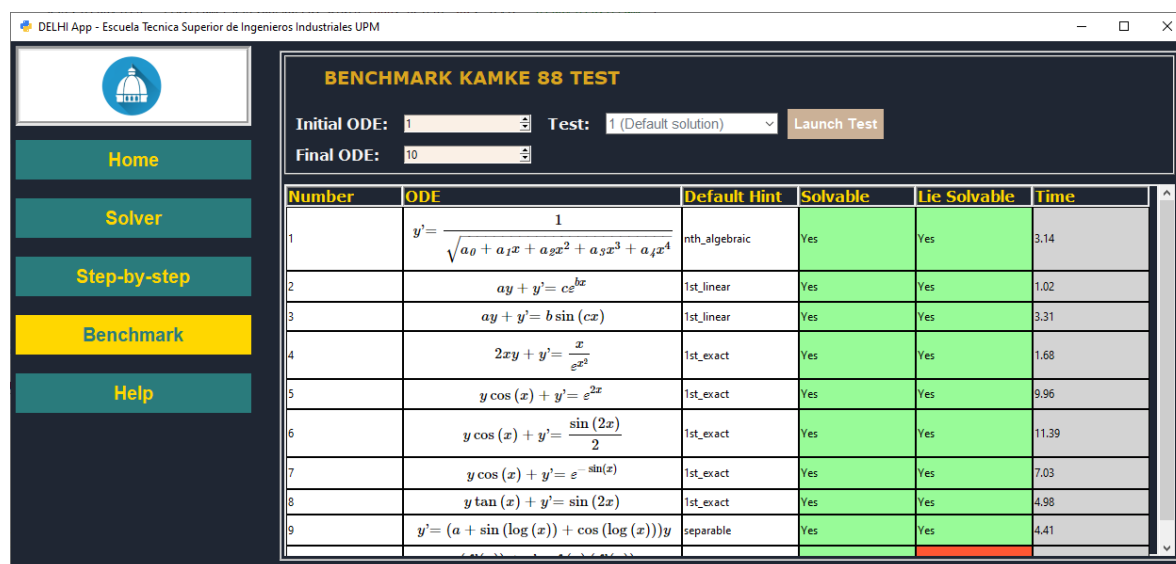
Figura 20. Step-by-Step de DELHI, detalle del paso 3.

Benchmark

En tercer y último lugar, la sección *Benchmark* permite al usuario visualizar los resultados de los dos primeros test del apartado homólogo. Dichos experimentos se detallan en el apartado 5.2 del actual documento. Han sido dotados de una interfaz de usuario para hacerlos más accesibles al usuario en caso de necesitar contrastar los resultados o compararlos con otros conjuntos de ecuaciones.

Para el primer test, el usuario podrá elegir un rango de ecuaciones del conjunto *Kamke 88* utilizado. Se mostrarán en forma de tabla los resultados obtenidos, entre ellos el *hint* utilizado o el tiempo empleado en resolver las ecuaciones.

Con el segundo test el número de ecuaciones, como bien se describe en la sección *Benchmarking*, se reduce a las 22 ecuaciones resolubles por simetrías de Lie. Este segundo test mostrará la información extraída del experimento en relación a las diferentes heurísticas de Lie y su comportamiento con cada ecuación.



Number	ODE	Default Hint	Solvable	Lie Solvable	Time
1	$y' = \frac{1}{\sqrt{a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4}}$	nth_algebraic	Yes	Yes	3.14
2	$ay + y' = ce^{bx}$	1st_linear	Yes	Yes	1.02
3	$ay + y' = b \sin(cx)$	1st_linear	Yes	Yes	3.31
4	$2xy + y' = \frac{x}{e^{x^2}}$	1st_exact	Yes	Yes	1.68
5	$y \cos(x) + y' = e^{2x}$	1st_exact	Yes	Yes	9.96
6	$y \cos(x) + y' = \frac{\sin(2x)}{2}$	1st_exact	Yes	Yes	11.39
7	$y \cos(x) + y' = e^{-\sin(x)}$	1st_exact	Yes	Yes	7.03
8	$y \tan(x) + y' = \sin(2x)$	1st_exact	Yes	Yes	4.98
9	$y' = (a + \sin(\log(x)) + \cos(\log(x)))y$	separable	Yes	Yes	4.41

Figura 21. Benchmark de DELHI.

Help

Se ha añadido una pequeña sección de ayuda que contiene una breve descripción de cada uno de los apartados anteriores. Sirve como primera explicación de la interfaz y su funcionalidad para los usuarios nuevos.

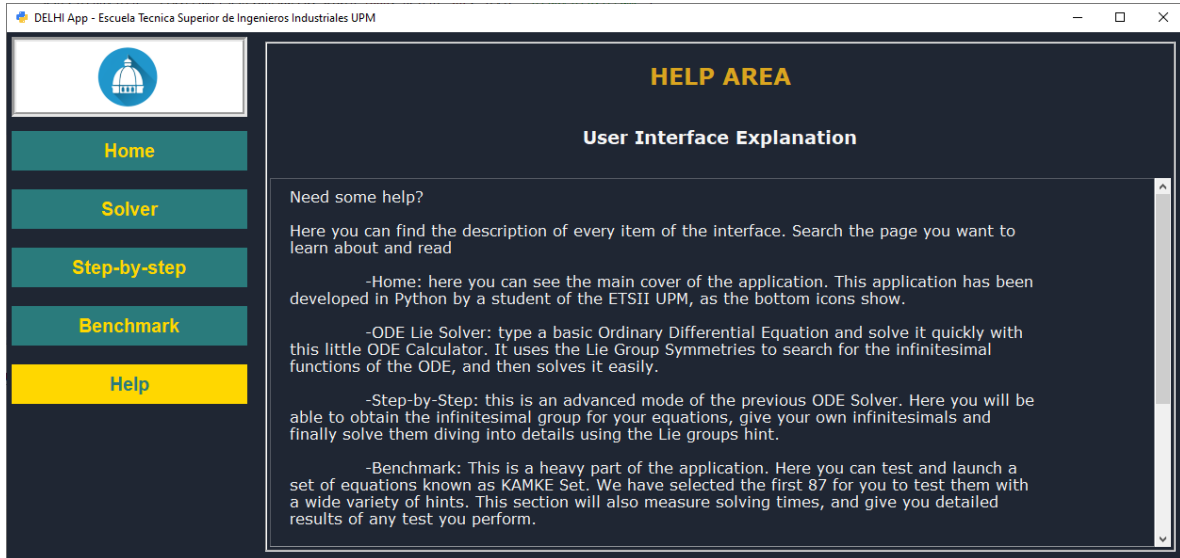


Figura 22. Menú de ayuda de DELHI.

Conclusión

Por lo expuesto a lo largo de este capítulo, se concluye que...

Son necesarias herramientas de soporte informáticas para apoyar a cualquier curso cuyos contenidos se ofrecen de manera online para complementar la formación del alumno. Asimismo, debido a la situación actual se acentúa más esta necesidad de hacer llegar a los hogares mediante la vía telemática la materia, ejercicios y cuestionarios de evaluación.

En adición, para este curso concreto se ofrece un software creado específicamente para experimentar de manera atractiva con los ejercicios y la teoría del curso desde tres perspectivas diferentes para aprovechar toda la potencia que el lenguaje de programación Python pone a disposición del desarrollador.

8. Viabilidad de desarrollo de un MOOC

Oportunidad y contexto

Tal y como se ha descrito anteriormente, un *Curso En línea Masivo y Abierto*, en terminología inglesa Massive Open Online Course [8], es un curso oficial gratuito y disponible para el público general que reúne una serie de características en cuanto a su forma de presentación y estructura de los contenidos.

Los MOOC tienen su origen en la iniciativa *OpenCourseWare* (OCW) iniciada por el *Massachusetts Institute of Technology* (MIT) en el año 2001. A día de hoy, en el MIT se ofertan un total de 2.500 cursos sobre todo tipo de materias. Es una de las plataformas de enseñanza más visitadas a nivel mundial, con 500 millones de visitantes anuales.

Posteriormente, surgió el *OpenCourseWare Consortium* al sumarse otras instituciones oficiales a esta iniciativa de enseñanza.

Particularizando al ámbito que este trabajo se ciñe, en el campo de las matemáticas existen 164 cursos MOOC, de los cuales 11 versan sobre la materia de las ecuaciones con derivadas.

Se hace ver la oportunidad en este contexto de desarrollar un curso con estas características dentro de la iniciativa OCW de la Universidad Politécnica de Madrid sobre el método de simetrías de Lie, ya presente en algunos de los citados cursos del MIT, de manera que pueda disponerse de este material en castellano y así, lograr introducirlo en la docencia actual universitaria.

El entorno OCW de la UPM es sin duda un lugar adecuado para expandir los contenidos que en él se ofertan, de una manera contrastada y ampliamente reconocida a nivel europeo.

The logo for MIT OpenCourseWare is displayed on a dark gray rectangular background. The text "MIT" is in white, "OPEN" is in orange, and "COURSEWARE" is in white. Below this, "MASSACHUSETTS INSTITUTE OF TECHNOLOGY" is written in a smaller, white, sans-serif font.

MIT **OPEN** COURSEWARE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Formato OCW-UPM

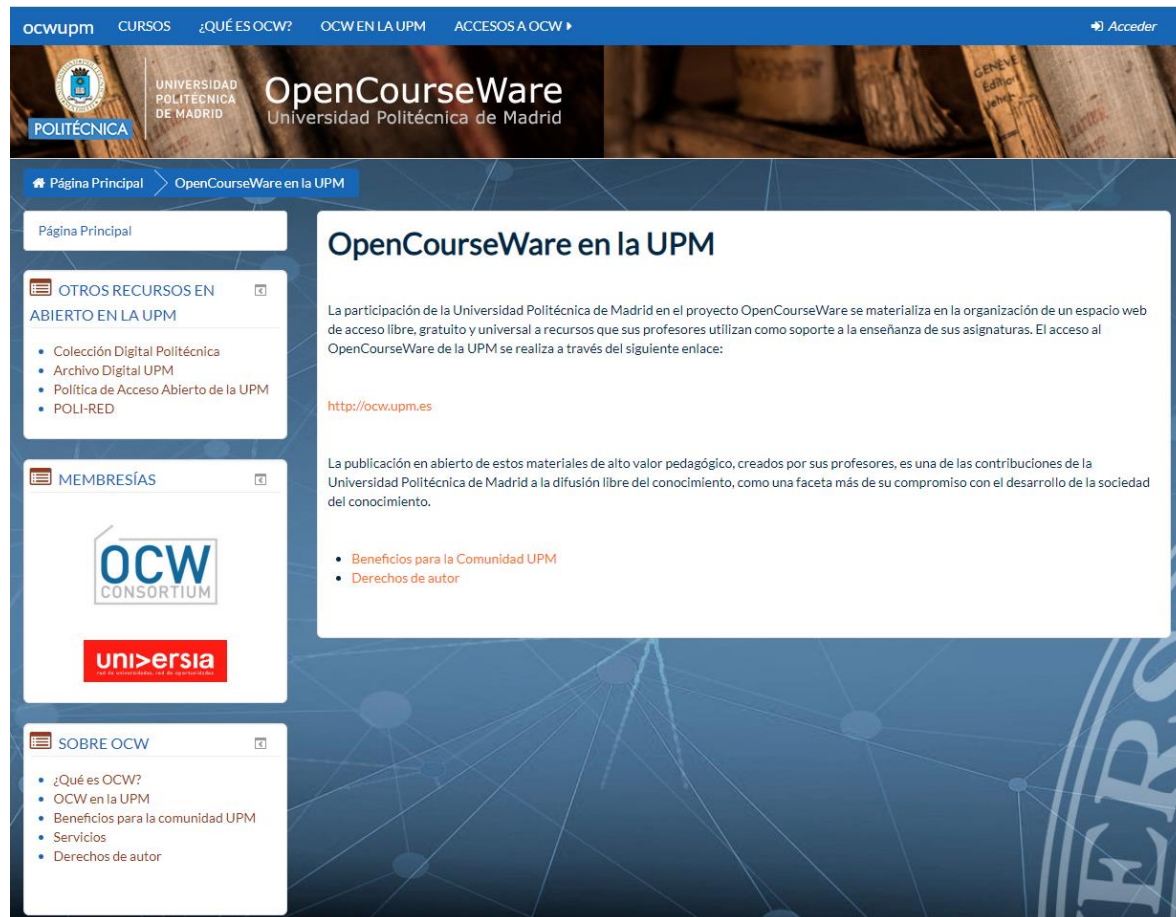


Figura 23. Descripción de la web de OCW UPM.

En la web [11] se indican los requisitos que debe reunir un curso para convertirse en un OCW susceptible de figurar en el repositorio de la UPM. Citando dicha fuente, se referencia lo siguiente:

“

- *Los recursos didácticos publicados en un OCW site se organizan en unidades de “asignaturas” o “cursos”. Con ello se quiere indicar:*
 - *Los accesos se realizan por asignaturas e incluyen un conjunto significativo de todos los materiales asociados a ella.*
 - *Los materiales se ofrecen de forma organizada por categorías: programa de la asignatura, lecturas obligatorias, materiales de clase, ejercicios, guía de aprendizaje.*

- *El profesor o profesores garantizan que el material que publican en el OCW site es original o tiene los derechos, bien directamente por ser propietario o bien a través del tipo de licencia que los soporta, para ser reutilizados en “abierto” sin infringir los “copyright” de otras personas.*
- *Son accesibles universalmente a través de la red:*
 - *Sin limitaciones geográficas.*
 - *Sin exclusión de usuario, ni necesidad de registrarse o utilizar palabras claves de acceso.*
 - *No exigen requisitos técnicos más allá de un navegador Web.*

”

Conclusión

La resolución de ecuaciones diferenciales, tanto ordinarias como en derivadas parciales, mediante simetrías de Lie presenta, citando a Dresner, **dos grandes virtudes**:

- “**Unifica** los muchos métodos *ad hoc* conocidos para resolver ecuaciones diferenciales, y
- Proporciona **nuevos y poderosos caminos** para encontrar soluciones”

De un análisis preliminar de la oferta educativa de las dos grandes universidades americanas de mayor prestigio en el ámbito de la Ingeniería Industrial, a la vista queda la oferta de contenidos relativos a la Teoría de Grupos de Lie para la resolución de ecuaciones diferenciales.

En ambos casos, se ofrecen *MOOC* relativos a la resolución de ecuaciones diferenciales mediante simetrías de Lie.

La UPM ha desarrollado una plataforma que dispone de diferentes *MOOC*, cuya oferta es susceptible de ser ampliada. En concreto, no se incluyen contenidos relativos a la resolución de ecuaciones diferenciales mediante simetrías de Lie.

Por lo tanto, concluimos que...

Existe una oportunidad para ampliar la oferta de MOOC que ya tiene la UPM desarrollando un MOOC específico dedicado a la resolución de ecuaciones diferenciales mediante simetrías de Lie, contenidos que ya se ofertan en universidades del mayor prestigio dentro del ámbito de la Ingeniería Industrial.

Consideramos que esta técnica, de gran belleza y profundidad conceptual, es muy adecuada, dado su nivel de abstracción, a la formación de los futuros ingenieros y complementa los tradicionales métodos de resolución *ad hoc*.

9. Soporte a un curso MOOC

DELHI Web. Herramienta Web

En esta sección se muestran las dos utilidades que se han desarrollado de la herramienta web adaptada de la aplicación de escritorio descrita en el capítulo 7. Se trata de DELHI Web, y a la fecha dispone de dos módulos: *ODE Solver* y *Benchmark*. A continuación, se muestran imágenes del progreso actual, así como una breve descripción de las funciones que son capaces de llevar a cabo.

Para acceder a estos sitios web debe introducirse un usuario y una contraseña. Son los siguientes, temporalmente disponibles para los miembros del tribunal ante los que se defenderá el Trabajo de Fin de Grado:

- Usuario: beta
- Contraseña: demo2020

ODE Solver

Análogo al apartado *Solver* de DELHI, este módulo se encarga de ofrecer al usuario una interfaz sencilla donde introducir una ecuación diferencial en forma normal, y pulsando el botón *Solve*, la aplicación se encarga de calcular la solución utilizando las simetrías de Lie siempre que sea posible. Dispone igualmente de una caja de texto donde se visualiza un mensaje de *feedback* para el usuario sobre los cálculos realizados. Si todo procede adecuadamente se muestra en verde esta afirmación. Si, por el contrario, ha ocurrido algún error, un *warning* aparecerá en la ventana.

The screenshot shows the 'delhi. ODE solver' web interface. At the top, there are logos for 'DELHI' and 'POLITECNICA'. The main heading is 'दिल्ली' (Delhi). Below it, the text 'delhi. ODE solver powered by Python / SymPy' is visible. The input field contains the ODE: $-x^2 y(x) + \frac{d}{dx} y(x) = 0$. A green message box below the input says 'ODE is VALID (this ODE input has been correctly parsed)'. Below this, there is a section for 'Input ODE:' with a text box containing 'yx-x**2*y=0', a dropdown for 'Predefined ODE's (select):' with 'ODE Repository' selected, and a blue 'Solve' button. The solution is displayed as $y(x) = C_1 e^{\frac{x^3}{3}}$. A green message box at the bottom says 'Solved without incidents'.

Figura 24. Utilidad ODE Solver de DELHI Web.

Benchmark

Similar en forma y contenido, se corresponde con su apartado homólogo de la versión de escritorio de DELHI. Como bien se indicaba en el capítulo 7, esta sección está dedicada a mostrar los resultados de los test realizados para el conjunto de 88 ecuaciones diferenciales ordinarias de primer orden en forma normal, llamado Kamke88 por el autor del libro del cual se han extraído.

Aquí se muestran en varias columnas el número de la EDO, su expresión y, a modo de novedad, la expresión interpretada en lenguaje MathML de la solución, en aquellos casos en los que es posible. Las siguientes imágenes representan bien el aspecto a día de hoy de esta aplicación web. Se muestran en ella las 11 primeras entradas de la tabla, pero ésta se extiende más abajo hasta llegar a los 88 mencionados ejemplos.

Aprovechando su naturaleza web, esta aplicación presenta algunas características que no aparecen en la herramienta *standalone*. Se puede visualizar en la imagen el detalle del código MathML, que, una vez interpretado, muestra la expresión de una ecuación diferencial en formato similar al LaTeX, mucho más atractivo a la vista.

Queda pendiente para este módulo web calcular los tiempos de resolución, tanto por el método de simetrías de Lie como por el resto de *hints* mostrados en el capítulo 5, y mostrarlos asimismo en la distribución de la página.

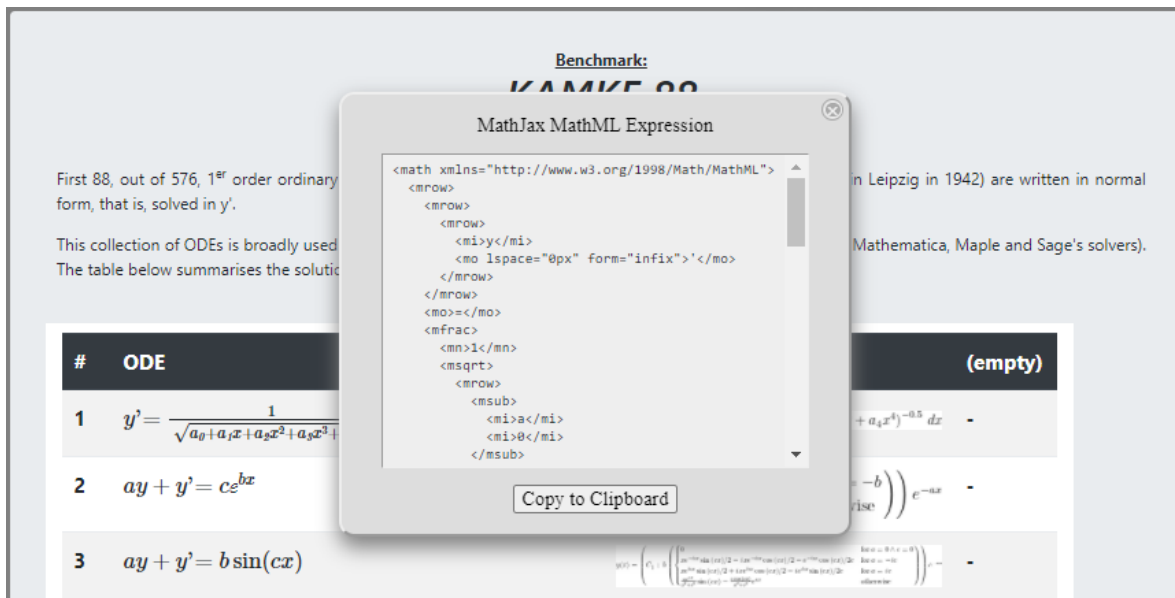


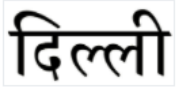


Figura 25. Detalle del código MathML de la EDO número 1.

delhi. Sympy Benchmark powered by Pyhton / Sympy

Benchmark:
KAMKE 88
First order ODEs in solved form

First 88, out of 576, 1st order ordinary differential equations from the [book](#) of [Erich Kamke](#) (first published in Leipzig in 1942) are written in normal form, that is, solved in y' .

This collection of ODEs is broadly used for testing purposes (see, for example, [Nasser M. Abbasi's analysis](#) for Mathematica, Maple and Sage's solvers). The table below summarises the solution provided by **dsolve**, Sympy's symbolic solver, for this data set.

#	ODE	SYMPY.dsolve	(empty)
1	$y' = \frac{1}{\sqrt{a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4}}$	$y(x) = C_1 + \int (a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4)^{-0.5} dx$	-
2	$ay + y' = ce^{bx}$	$y(x) = \left(C_1 + c \left(\frac{x}{a+b} \text{ for } a = -b \right) \right) e^{-bx}$	-
3	$ay + y' = b \sin(cx)$	$y(x) = \left(C_1 + \begin{cases} \frac{b}{c} \sin(cx) & \text{for } a = 0, c \neq 0 \\ \frac{b}{c} \cos(cx) & \text{for } a = -c \\ \frac{b}{c} \sin(cx) & \text{for } a = c \\ \frac{b}{c} \cos(cx) & \text{otherwise} \end{cases} \right) e^{-ax}$	-
4	$2xy + y' = \frac{x}{x^2}$	$y(x) = (C_1 + x^2/2) e^{-x^2}$	-
5	$y \cos(x) + y' = e^{2x}$	$\int (y(x) \cos(x) - e^{2x}) e^{\sin(x)} dx = C_1$	-
6	$y \cos(x) + y' = \frac{\sin(2x)}{2}$	$1.0 \int (0.5e^{i \sin(x)} \sin(2x)) dx + 1.0 \int 1.0y(x)e^{i \sin(x)} \cos(x) dx = C_1$	-
7	$y \cos(x) + y' = e^{-\sin(x)}$	$y(x) = (C_1 + x) e^{-\sin(x)}$	-
8	$y \tan(x) + y' = \sin(2x)$	$y(x) = \frac{\sqrt{-\cos^2(x)} \left(C_1 + \int \frac{y(x) \tan(x) - \sin(2x)}{\sqrt{(\sin(x)-1)(\sin(x)+1)}} dx \right)}{\sqrt{-\cos^2(x)} \int \frac{\tan(x)}{\sqrt{-\cos^2(x)}} dx - 1}$	-
9	$y' = (a + \sin(\log(x)) + \cos(\log(x)))y$	$y(x) = C_1 + \int (a + \sin(\log(x)) + \cos(\log(x))) f(x) dx$	-
10	$y(f'(x)) + y' = f(x)(f'(x))$	$y(x) = C_1 e^{-g(x)} + g(x) - 1$	-
11	$f(x)y + y' = g(x)$	$(e^{\int f(x) dx} - \int g(x) f(x) e^{\int f(x) dx} dx) y(x) + \int (g(x) y(x) - h(x)) e^{\int f(x) dx} dx = C_1$	-

Figura 26. Resultados del Benchmark en DELHI Web.

Conclusión

Por lo expuesto en este capítulo, se extraen las siguientes conclusiones...

A pesar de la potencia alcanzada con la herramienta *standalone*, se presenta como necesaria su adaptación web. De esta forma, el soporte al curso online se realiza de manera más sencilla y accesible desde cualquier parte, pasando por alto la descarga de una aplicación de escritorio gracias a la ubicuidad del sitio web.

Además, se expone que el grado de desarrollo actual de la herramienta web está aún en las primeras fases, pudiendo mostrar únicamente muy limitadas funcionalidades para la resolución de ecuaciones.

10. Conclusiones y líneas futuras

El objetivo de ese apartado consiste en reunir los resultados y conclusiones obtenidas a lo largo del proyecto para ofrecerlas de manera coherente y unificada. A su vez, también se pretender mostrar las posibles líneas de expansión futura del trabajo, así como aquellas partes que han quedado pendientes de ampliar y que por tanto puedan ser continuadas en futuros trabajos de fin de grado que sigan el recorrido del actual.

Conclusiones por objetivos

Librerías Python

El lenguaje de programación Python a través de la librería Sympy, cuenta con un gran número de métodos ya implementados para el tratamiento, clasificación y resolución de ecuaciones diferenciales ordinarias. Se ha aprovechado esta potencia para diseñar una interfaz con la librería estándar *Tkinter* que permita a los usuarios entrar en contacto con el método de simetrías de Lie de una forma interactiva.

Por otro lado, se han ejecutado una serie de test, cuyos resultados ilustran la potencia comentada de la librería Sympy para un conjunto de ecuaciones determinado. En concreto, se ha buscado comparar la utilidad del método de simetrías de Lie frente al resto de procedimientos tradicionales.

Los resultados obtenidos son sorprendentemente buenos, consiguiendo resolver 22 de las ecuaciones con éxito (un 25%), contando además con la presencia de 5 ecuaciones exclusivas que no podrían ser resueltas sin utilizar sus simetrías. Si además añadimos aquellas ecuaciones que se han resuelto reduciéndolas a cuadraturas, el porcentaje asciende a un 62% (58 ecuaciones). Para 28 de ellas, el método de Lie es el único que es capaz de hallar la solución explícita o reducida a cuadraturas.

Contenidos docentes

Dentro del presente trabajo se han definido los contenidos de un curso oficial, presencial u online. Se han seguido unos criterios de diseño que, entre otras cosas, simplifican la adaptación de este curso oficial a un MOOC.

Se ha definido una estructura para el curso basado en 4 módulos, que constan conjuntamente de 12 sesiones y 4 PEC's. Las sesiones constan de distinto tipo de actividades: clases magistrales (12), resolución de ejercicios (12), vídeos (8), prácticas (12) y cuestionarios de evaluación (12).

Aunque el desarrollo completo y detallado de todos los contenidos planificados queda fuera del alcance del presente trabajo, se han desarrollado varios ejemplos de cada uno de los tipos de actividades. Con los contenidos desarrollados se cubre, totalmente, el módulo 2: “Resolución de EDO’s de 1er orden mediante simetrías de Lie”.

Herramientas de soporte

Son necesarias herramientas de soporte informáticas para apoyar a cualquier curso cuyos contenidos se ofrecen de manera online para complementar la formación del alumno. Asimismo, debido a la situación actual se acentúa más esta necesidad de hacer llegar a los hogares mediante la vía telemática la materia, ejercicios y cuestionarios de evaluación.

En adición, para este curso concreto se ofrece un software creado específicamente para experimentar de manera atractiva con los ejercicios y la teoría del curso desde tres perspectivas diferentes para aprovechar toda la potencia que el lenguaje de programación Python pone a disposición del desarrollador.

Viabilidad de un curso MOOC

Existe una oportunidad para ampliar la oferta de MOOC que ya tiene la UPM desarrollando un MOOC específico dedicado a la resolución de ecuaciones diferenciales mediante simetrías de Lie, contenidos que ya se ofertan en universidades del mayor prestigio dentro del ámbito de la Ingeniería Industrial.

Consideramos que esta técnica, de gran belleza y profundidad conceptual, es muy adecuada, dado su nivel de abstracción, a la formación de los futuros ingenieros y complementa los tradicionales métodos de resolución *ad hoc*.

Herramientas web

A pesar de la potencia alcanzada con la herramienta *standalone*, se presenta como necesaria su adaptación web. De esta forma, el soporte al curso online se realiza de manera más sencilla y accesible desde cualquier parte, pasando por alto la descarga de una aplicación de escritorio gracias a la ubicuidad del sitio web.

Además, se expone que el grado de desarrollo actual de la herramienta web está aún en las primeras fases, pudiendo mostrar únicamente muy limitadas funcionalidades para la resolución de ecuaciones.

Funcionalidades adicionales

Desarrollo completo de los contenidos docentes

Queda sin duda como pendiente la tarea de redactar y generar todo el material docente asignado al curso OCW. Como bien se ha indicado en el apartado correspondiente, el presente trabajo de fin de grado se ha limitado a desarrollar únicamente el módulo 2 del curso, en aras de mostrar el formato en el que presentar el material. Sin embargo, los otros tres módulos quedan a la espera de ser completados.

Aplicación en JavaScript para cuestionarios

En el curso de Moodle diseñado para exponer los contenidos docentes del curso, se adjunta el primer cuestionario del segundo módulo del mismo. Al ser una tarea lenta el diseñarlos a mano o con las herramientas de que dispone Moodle, se pensó inicialmente programar un pequeño código en el lenguaje de programación JavaScript para automatizar la tarea de generar los formularios, así como incluir una cuenta atrás de tiempo que entraría en juego durante su realización por parte de un alumno del curso.

Integración de un módulo de cuestionarios en DELHI

Otra de las tareas que sin duda ayudarían a dar más solidez al proyecto sería integrar algunos cuestionarios prácticos en la herramienta DELHI. De esta manera, se complementa su funcionalidad actual de soporte al curso pudiendo realizar ejercicios prácticos tanto con la herramienta como dentro de ella.

Ampliar módulos de la herramienta DELHI

Otra de las metas a conseguir con la herramienta *DELHI* es ampliar sus fronteras de actuación. Como bien se ha expuesto, la aplicación actualmente soporta ecuaciones diferenciales ordinarias de primer orden, siendo además un requisito introducirlas en forma normal en los apartados *Step-by-Step* y *Benchmark*.

Es deseable que la herramienta se expanda para resolver también ecuaciones de segundo orden o incluso de órdenes superiores, dada la potencia de la que se dispone con la librería *Sympy*. Además, aportaría gran valor testear estas nuevas funcionalidades con las ecuaciones siguientes del conjunto *Kamke* y hacer un paralelismo con los resultados obtenidos hasta la fecha en este TFG.

Futuro TFG 1. Benchmark con Maple, Mathematica y Sage

Puede encontrarse en internet los resultados de terceros utilizando el mismo set de ecuaciones de ejemplo que aquí se ha presentado, el *Kamke* 88, en otras plataformas como *Maple*, *Mathematica* o *Sage*. Estos resultados van más allá del rango utilizado en el presente documento, llegando a experimentar con las 1940 ecuaciones del libro de Erich *Kamke*. No obstante, a pesar del enorme esfuerzo, estos informes no están completos.

Es interesante plantear como objetivo de un posterior trabajo de fin de grado comparar las soluciones y número de ecuaciones resueltas por los test *Benchmark* de *DELHI* con los de las aplicaciones mencionadas en el párrafo anterior, que, al ser programas implantados de forma profesional, marcarán un nivel de calidad al que aspirar en dicho trabajo.

Futuro TFG 2. Proyecto GNU de mejora de SymPy

Explorando ahora otras vías, se propone un segundo trabajo de fin de grado que pueda continuar al presente, encargado de desarrollar nuevos métodos en el lenguaje de programación Python para incorporarlos a la librería SymPy. De esta manera se pretende incrementar la tipología de EDOs resolubles por la biblioteca, considerando especialmente la programación de funciones que permitan utilizar las simetrías dinámicas para resolver ecuaciones diferenciales, dando un claro paso más allá del ámbito que alcanza a comprender el presente proyecto.[\[15\]](#)

Creación de un proyecto MOOC

El horizonte final propuesto de este trabajo, como se ha mencionado con anterioridad, sería convertirse en un *Massive Online Open Course* de manera oficial. Queda por tanto pendiente repasar los requisitos necesarios para adaptar los contenidos y realizar los trámites necesarios para solicitar su revisión y aceptación como curso en regla.

Cabe destacar que alcanzar en esta meta un 100% de desarrollo supondría el colofón final de este proyecto, pues supone la idea primera de la que nace y en la cual se ha orientado toda la línea de actuación.

11. Bibliografía

Documentación y referencias software

- [1] Mark Summerfield “*Programming in Python 3. A complete introduction to the Python Language*”.
- [2] Sympy Documentation <https://docs.sympy.org/latest/index.html>
- [3] Python Documentation <https://docs.python.org/3/>
- [4] W3Schools Python Tutorial <https://www.w3schools.com/python/>
- [5] StackOverFlow Questions <https://stackoverflow.com/questions>
- [6] Python Everywhere Cloud <https://www.pythonanywhere.com/>

Otras fuentes

- [7] Carlos Espa Torres, Trabajo de Fin de Grado “*Desarrollo y programación de modelos matemáticos de soporte a la docencia*”. <http://oa.upm.es/53879/>
- [8] Wikipedia: Massive Open Online Course
https://es.wikipedia.org/wiki/Massive_Open_Online_Course
- [9] Wikipedia: Tkinter <https://es.wikipedia.org/wiki/Tkinter>
- [10] Wikipedia: PyQt <https://es.wikipedia.org/wiki/PyQt>
- [11] OCW UPM <http://ocw.upm.es/local/staticpage/view.php?page=que-es-ocw>
- [12] Moodle UPM: <https://moodle.upm.es/formacion/course/view.php?id=5294>
- [13] DELHI web Benchmark:
<https://mmendozaupm.pythonanywhere.com/delhi/benchmark/>
- [14] DELHI web Solver: <https://mmendozaupm.pythonanywhere.com/delhi/ODEsolver/>

Bibliografía teórica

- [15] John Starret “*Solving Differential Equations by Symmetry Groups*”.

[16] E.S. Cheb Terrab, L.C. Duarte, L.A.C.P da Mota “*Computer Algebra Solving of First Order ODEs using Symmetry Methods*”.

[17] Erich Kamke “*Differentialgleichungen reeller Funktionen, Akademische Verlagsgesellschaft. Band 1: Gewöhnliche Differentialgleichungen*” (Ecuaciones Diferenciales de funciones reales. Volumen 1: Ecuaciones Diferenciales Ordinarias).

[18] Marcus du Satoy “*Symmetry*”.

[19] Nail H. Ibragimov “*Elementary Group Analysis and Ordinary Differential Equations*”.

[20] Lawrence Dresner “*Applications of Lie’s Theory of Ordinary and Partial Equations*”.

[21] E.S Cheb Terrab, A.D. Roche “*Symmetries and First Order ODE Patterns*”.

[22] Bruce Char “*Using Lie Transformation Groups to Find Closed Form Solution to First Order Ordinary Differential Equations*”.

[23] S.Lie, G. Scheffers “*Lectures on differential equations with known infinitesimal transformations*”.



Figura 28. Código QR al Moodle UPM.



Figura 27. Código QR a la web OCW UPM.



12. Planificación temporal y presupuesto

Planificación temporal

La realización del presente trabajo de fin de grado se ha dividido en las cuatro fases diferenciadas que se enumeran: investigación previa, aprendizaje, desarrollo de software y redacción de esta memoria.

Fase de investigación previa

Durante este periodo de tiempo se llevó a cabo una búsqueda y comparativa de lenguajes de programación adecuados para el proyecto. Esto tuvo lugar entre el 15 de marzo de 2019 y el 30 de marzo del mismo año, cargando con unas **8 horas** de trabajo.

Se puede incluir también en esta fase, aunque ha tenido lugar de manera continuada a lo largo del desarrollo del trabajo, una investigación en materia de fundamentos teóricos, en la que se han leído y repasado las principales fuentes indicadas en la bibliografía.

Fase de aprendizaje

Claramente diferenciada de las demás, la actividad principal que se llevó a cabo durante este periodo fue un curso del lenguaje de programación Python, para adquirir las competencias básicas y algunas más avanzadas que permitieran realizar las especificaciones del trabajo. El curso tuvo lugar entre el 4 de abril de 2019 y el 27 de junio de ese mismo año, con un total de **48 horas** de trabajo asociadas a él.

Sin embargo, debido a la naturaleza del trabajo, puede decirse al igual que en el caso de la investigación que el aprendizaje ha sido continuo a lo largo del trabajo, pues algunos aspectos incluidos finalmente se conocieron más tarde de las fechas indicadas y estaban fuera de los contenidos que el curso realizado cubría.

Fase de desarrollo del software

Después de la fase de aprendizaje comenzó el grueso principal del TFG que consistió en desarrollar la aplicación *standalone* previamente expuesta, *DELHI*. Se puede subdividir en varios periodos igualmente, que se indican a continuación.

En primer lugar, se realizaron una serie de programas sencillos como toma de contacto con las librerías *Tkinter* y *Sympy*. Más concretamente, con *Tkinter* se elaboró una interfaz

para una calculadora aritmética sencilla, lo cual llevó unas **16 horas** de trabajo. Con *Sympy* los primeros programas han podido ser reaprovechados, siendo el primero de ellos una versión alfa de los test del *Benchmark*. Tomó unas **12 horas** de esfuerzo. Ambos pasos tuvieron lugar en el verano de 2019.

Más tarde, ya dominando los aspectos principales de ambas librerías, comenzó a desarrollarse la primera versión de DELHI. Se comenzó con el apartado *Solver*, al ser el más sencillo de todos, a la par que se continuaba actualizando el *Benchmark* y se utilizaba para obtener los resultados que en el trabajo se han presentado. Estos programas fueron realizados entre octubre y noviembre de 2019, costando **20 horas** de esfuerzo entre ambos.

El trabajo tuvo en este punto un periodo de pausa debido a la cercanía de la evaluación ordinaria de enero. El autor dedicó este periodo a estudiar las asignaturas correspondientes y, por ello, el trabajo no retomó su avance hasta febrero de 2020.

En dicho mes, se continuó con el desarrollo de la interfaz de *DELHI*, que fue paulatinamente incluyendo el resto de apartados que finalmente son visibles en la aplicación.

Esta fase de desarrollo principal se ha extendido desde febrero de 2020 hasta septiembre del mismo año. Al inicio no se pudieron dedicar tantas horas como se hubiera querido debido a que el autor estuvo realizando unas prácticas externas en empresa, contando con menos horas libres para emplear en el presente trabajo.

Entre marzo y junio de 2020 se emplearon **52 horas** en total en el desarrollo del trabajo, en las que se alcanzó la tercera versión del *Benchmark* y un primer esbozo de la sección *Step-by-Step*, así como del apartado de ayuda. El procedimiento aproximado que se ha seguido para todas las funcionalidades ha sido la de pensar en primer lugar qué tareas se querían realizar con ella, después crear la interfaz y por último programar los métodos necesarios para lograr que funcionase correctamente.

El resto del crecimiento de la aplicación ha tenido lugar entre junio y septiembre de 2020. En este periodo la mayor parte del tiempo se ha dedicado a refinar la interfaz, arreglar errores en los algoritmos, probar ejemplos y mejorar la estética general del programa. Esto ha llevado **116 horas** de trabajo hasta dejar la aplicación en el estado en el que se presenta actualmente.

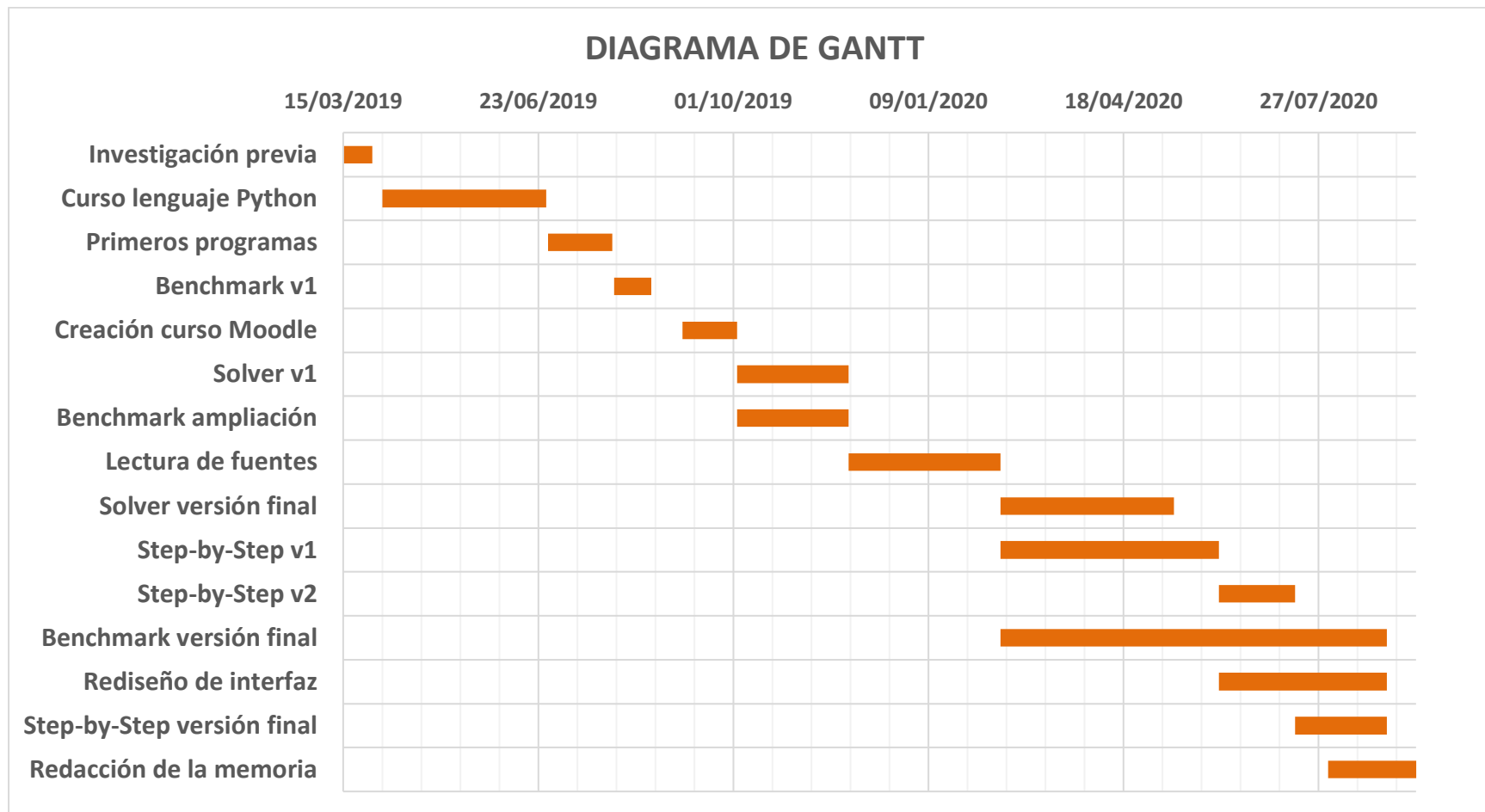
Fase de redacción de la memoria

La memoria ha tenido un avance en dos tramos separados en el tiempo. En el primer momento de escritura, se comenzó creando la estructura general de proyecto, los objetivos y los contenidos teóricos docentes. Esto tuvo lugar en febrero de 2020, tomando **21 horas** de esfuerzo en total.

La segunda parte de la redacción ha sido la que ha llevado más tiempo, pues se han incluido el resto de capítulos incluyendo la descripción de la aplicación, los experimentos realizados y los anexos, imágenes y formato del documento. Se ha llevado a cabo en el periodo comprendido entre mayo y septiembre de 2020, cobrándose en total **34 horas** de trabajo.

Haciendo el compendio de **horas totales** invertidas en el trabajo de fin de grado, el resultado son **327 horas** empleadas a lo largo de un año y medio de desarrollo debido a las circunstancias indicadas anteriormente.

Por último, se expone a continuación el diagrama de Gantt que resume la descomposición de las horas de vida del proyecto.



Presupuesto del proyecto

Este trabajo de fin de grado ha tomado un gran número de horas y esfuerzo para su elaboración. Sin embargo, en cuanto a costes de material y equipos apenas ha consumido recursos, debido a que todo el proyecto se ha llevado a cabo en el ordenador personal del alumno. No hay gastos derivados de préstamo de otros equipos ni material de oficina reseñables.

El total de horas asciende a 327 horas dedicadas por el autor, como se ha indicado en el apartado anterior. Estimando su coste en 35 €/hora, el gasto asciende a 11445€.

En cuanto al director del trabajo, su participación y esfuerzo se estima en unas 90 horas. A un coste de 120€ por hora, supone 10800€ de gasto.

Se incluye un gasto puntual de reserva para un espacio de trabajo municipal para estudiantes, que costó 5€/h por alquiler del local durante 6 horas.

Por último, se va a tener en cuenta el único gasto en equipos comentado, el ordenador portátil del alumno. El valor inicial del equipo se tasa en 1000€. La amortización anual establecida es de un 26%. Por tanto, el coste de uso del ordenador se sitúa en 260€ anuales. Al haberse utilizado durante 1,5 años, el gasto empleado en el equipo queda en 390€ en total.

La siguiente tabla presenta el desglose del presupuesto.

CONCEPTO	CÁLCULO	COSTE FINAL
Horas autor del trabajo	327 h x 35 €/h	11.445 €
Horas director de trabajo	90 h x 120€/h	10.800 €
Alquiler de local	6 h x 5€/h	30 €
Ordenador portátil	260 €/año x 1,5 año	390 €
TOTAL		22.665 €

Tabla 14. Presupuesto desglosado del proyecto.

13. Anexo I. Contenidos docentes del Módulo 2

Este anexo se encarga de reunir las transparencias elaboradas del módulo 2 sobre “**Generadores Infinitesimales**” del curso diseñado, comenzando con el índice del mismo. Posteriormente se muestran todos los contenidos para ponerlos a disposición del lector para su estudio o consulta.

INDUSTRIALES
RSIS | UPM

Simetrías Lie y Ecuaciones Diferenciales

Índice
I

Tr. Infin.
Transformaciones Infinitesimales

Expon.
Mapeo Exponencial

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal en Grupo Continuo

Recursos
R

2

Generadores Infinitesimales


Generadores Infinitesimales

1. *Transformaciones Infinitesimales asociada a un Grupo Continuo de Transformaciones*
2. *Mapeo Exponencial*
3. *Invariantes*
4. *Variables Canónicas*
5. *Métodos para la construcción de un Grupo Continuo Uniparamétrico, conocido su Generador Infinitesimal*
 - a. *Uso de Tablas*
 - b. *Ecuaciones determinantes (o de Lie)*
 - c. *Mapeo Exponencial*
 - d. *Variables Canónicas*

Manuel Mendoza de Haro

Universidad Politécnica de Madrid

Figura 31. Índice de contenidos del Módulo 2



Simetrías Lie y Ecuaciones Diferenciales

I

Indice

Tr. Infin.

Transformaciones Infinitesimales

Expon.

Mapeo Exponencial

Invar.

Invariantes

Canón.

Variables Canónicas

Gen->Gr

Generador Infinitesimal -> Grupo Continuo

R

Recursos

Generadores Infinitesimales

2

L 2.1


Órbitas.

Transformaciones infinitesimales.

Problema inverso: Ecuaciones de Lie y Mapeo Exponencial

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



Simetrías Lie y Ecuaciones Diferenciales

I

Indice

Tr. Infin.

Transformaciones Infinitesimales

Expon.

Mapeo Exponencial

Invar.

Invariantes

Canón.

Variables Canónicas

Gen->Gr

Generador Infinitesimal -> Grupo Continuo

R

Recursos

Generadores Infinitesimales

2

Trayectoria o G-órbita

Dado un Grupo Continuo Uniparamétrico de Transformaciones G , la **trayectoria** (o **G-órbita**) del punto (x, y) es una curva continua, también parametrizada por el parámetro de G , que resulta de la aplicación a (x,y) de cada transformación de G .


LEMA

El transformado de cualquier punto de la trayectoria está en la trayectoria.

Demostr. Por la propiedad de grupo: la composición de transformaciones es una operación interna.

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



INDUSTRIALES
ETSII | UPM

Simetrías Lie y Ecuaciones Diferenciales

Indice

I

Tr. Infin.
Transformaciones Infinitesimales

Expon.
Mapas Exponenciales

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal -> Grupo Continuo

Recursos
R

Generadores Infinitesimales

2

G ⇒ G-órbitas

e.g. Grupo de rotaciones en \mathbb{R}^2 . G-órbitas

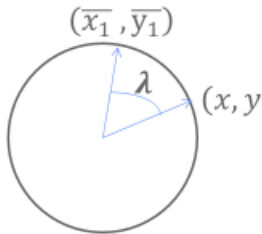
Sea el grupo de rotaciones dado por:

$$\begin{cases} \bar{x}_1 = x \cos(\lambda) - y \sin(\lambda) \\ \bar{y}_1 = x \sin(\lambda) + y \cos(\lambda) \end{cases}$$

Si (\bar{x}_1, \bar{y}_1) es el transformado de $(x, y) \Rightarrow \bar{x}_1^2 + \bar{y}_1^2 = (x \cos(\lambda) - y \sin(\lambda))^2 + (x \sin(\lambda) + y \cos(\lambda))^2 =$
 $= x^2 \cos^2(\lambda) + y^2 \sin^2(\lambda) - 2xy \cos(\lambda) \sin(\lambda) + x^2 \sin^2(\lambda) + y^2 \cos^2(\lambda) + 2xy \cos(\lambda) \sin(\lambda) =$
 $= x^2 [\cos^2(\lambda) + \sin^2(\lambda)] + y^2 [\sin^2(\lambda) + \cos^2(\lambda)] = x^2 + y^2$

Es decir, (\bar{x}_1, \bar{y}_1) , transformados de (x, y) , están en la misma circunferencia centrada en el origen que (x, y) .
 Adicionalmente, cualquier punto (\bar{x}_1, \bar{y}_1) en la circunferencia centrada en el origen es transformado de (x, y) para la transformación con parámetro $\lambda = \arctg(\frac{\bar{y}_1}{\bar{x}_1}) - \arctg(\frac{y}{x})$


Por lo tanto, las G-órbitas son circunferencias centradas en el origen



L 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



INDUSTRIALES
ETSII | UPM

Simetrías Lie y Ecuaciones Diferenciales

Indice

I

Tr. Infin.
Transformaciones Infinitesimales

Expon.
Mapas Exponenciales

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal -> Grupo Continuo

Recursos
R

Generadores Infinitesimales

2

Transformación Infinitesimal

Dado G Grupo Continuo Uniparamétrico de Transformaciones en \mathbb{R}^2 , con parámetro canónico*
 A, dado por:

$$\begin{cases} \bar{x}_1 = f(x, y, \lambda) \\ \bar{y}_1 = g(x, y, \lambda) \end{cases} \Leftrightarrow X_1 = F(X, \lambda)$$

Cada transformación puede aproximarse por:

$$\begin{cases} \bar{x}_1 \approx x + \left(\frac{\partial f}{\partial \lambda} \right)_{\lambda=0} * \lambda \\ \bar{y}_1 \approx y + \left(\frac{\partial g}{\partial \lambda} \right)_{\lambda=0} * \lambda \end{cases}$$

Se denomina **Transformación Infinitesimal** a la definida por estos segundos miembros.
 Aproxima localmente el grupo de transformaciones G.

* El parámetro λ de un Grupo de Transformaciones Uniparamétrico G se denomina canónico cuando la composición de transformaciones tiene como ley de composición la suma de parámetros; es decir, $F[F(X, \lambda_1), \lambda_2] = F(X, \lambda_1 + \lambda_2)$. En ese caso, se verifica que $F(X, 0) = X$, es decir, la transformación identidad corresponde a $\lambda=0$. Véase [IBR-7.1.3]. Siempre es posible determinar parametrizar un grupo de transformaciones G mediante un parámetro canónico Véase [IBR-7.1.6].


Respecto a un parámetro no canónico la aproximación de cada transformación quedaría determinada, respecto al parámetro λ_0 correspondiente a la transformación identidad, por:

$$\begin{cases} \bar{x}_1 \approx x + \left(\frac{\partial f}{\partial \lambda} \right)_{\lambda=\lambda_0} * (\lambda - \lambda_0) \\ \bar{y}_1 \approx y + \left(\frac{\partial g}{\partial \lambda} \right)_{\lambda=\lambda_0} * (\lambda - \lambda_0) \end{cases}$$

L 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



INDUSTRIALES
UPM

Simetrías Lie y Ecuaciones Diferenciales

Índice

1

Tr. Infin.
Transformaciones Infinitesimales

Expon.
Mapas Exponenciales

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal -> Grupo Continuo

Recursos

Generadores Infinitesimales

2

ξ, η (the "greeks")

A los coeficientes de λ en las ecuaciones de la transformación infinitesimal se les ha denominado históricamente ξ y η. Son funciones de x e y.

$$\begin{cases} \xi(x, y) = \frac{\partial f}{\partial \lambda} \Big|_{\lambda=0} \\ \eta(x, y) = \frac{\partial g}{\partial \lambda} \Big|_{\lambda=0} \end{cases}$$

Por lo tanto, en función de ξ y η, la transformación infinitesimal queda definida por :

$$\begin{cases} x_1 = x + \xi(x, y) * \lambda \\ y_1 = y + \eta(x, y) * \lambda \end{cases}$$

e.g. Grupo de rotaciones en en \mathbb{R}^2 . Transformación Infinitesimal

El grupo continuo uniparamétrico de rotaciones en \mathbb{R}^2 , queda definido por las ecuaciones:

$$\begin{cases} \bar{x}_1 = x \cos(\lambda) - y \sin(\lambda) \\ \bar{y}_1 = x \sin(\lambda) + y \cos(\lambda) \end{cases}$$

ξ y η para estas transformaciones son:


$$\begin{cases} \xi(x, y) = -x \sin(\lambda) - y \cos(\lambda) \Big|_{\lambda=0} = -y \\ \eta(x, y) = x \cos(\lambda) - y \sin(\lambda) \Big|_{\lambda=0} = x \end{cases}$$

Por lo que la transformación infinitesimal asociada es:

$$\begin{cases} x_1 = x - y\lambda \\ y_1 = y + x\lambda \end{cases}$$

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



INDUSTRIALES
UPM

Simetrías Lie y Ecuaciones Diferenciales

Índice

1

Tr. Infin.
Transformaciones Infinitesimales

Expon.
Mapas Exponenciales

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal -> Grupo Continuo

Recursos

Generadores Infinitesimales

2

Campo de Vectores Tangentes

(ξ, η) define una campo de vectores tangentes a las órbitas de G.

Generador Infinitesimal

Dado G grupo con transformación infinitesimal determinada por (ξ, η), se define como generador (u operador) infinitesimal de G al operador diferencial lineal de primer orden:

$$X = \xi(x, y) * \frac{\partial}{\partial x} + \eta(x, y) * \frac{\partial}{\partial y}$$

Lie llamo a X "the symbol".
Es un operador, actuando por tanto sobre funciones de dos variables


e.g. El generador infinitesimal del grupo continuo de rotaciones viene dado por:

$$X = -y \frac{\partial}{\partial x} + x \frac{\partial}{\partial y}$$

Manuel Mendoza de Haro

Universidad Politécnica de Madrid

DUDA: Por que X se transforma como un escalar (Ibragimov 138, refiere a clausula 4.2.2 lema 2)



INDUSTRIALES
ETSII - UPM

Simetrías Lie y Ecuaciones Diferenciales

Indice
I

Tr. Infin.
Transformaciones Infinitesimales

Expon.
Mapa Exponencial

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal -> Grupo Continuo

Recursos
R

Generadores Infinitesimales

2

Problema Inverso. Calcular G conocidos (ξ, η)

Dado un generador infinitesimal, X , es posible determinar el Grupo Continuo Uniparamétrico, G asociado⁽¹⁾.

TEOREMA PROBLEMA INVERSO: $X \Rightarrow G^{(1)}$ **EXISTENCIA Y UNICIDAD**

Dado el Grupo Continuo dado por: $\begin{cases} \bar{x}_1 = f(x, y, \lambda) \\ \bar{y}_1 = g(x, y, \lambda) \end{cases}$ con generador infinitesimal dado por (ξ, η)

Las funciones f y g son solución única del SEDO, con valores iniciales:

ECUACIONES de LIE:

$$\begin{cases} x'(\lambda) = \frac{df}{d\lambda} = \xi(x, y) \\ y'(\lambda) = \frac{dg}{d\lambda} = \eta(x, y) \end{cases}$$

o, equivalentemente,

$$\frac{dx}{\xi(x, y)} = \frac{dy}{\eta(x, y)} = d\lambda$$

verificando: $\begin{cases} x(0) = f(x, y, 0) = x \\ y(0) = g(x, y, 0) = y \end{cases}$

⁽¹⁾ La determinación de G , grupo, a partir de X , generador infinitesimal, presenta cierta analogía con la determinación de una función de una variable conocida su derivada o, más generalmente, la resolución de una EDO de primer orden con valores iniciales.

L 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



INDUSTRIALES
ETSII - UPM

Simetrías Lie y Ecuaciones Diferenciales

Indice
I

Tr. Infin.
Transformaciones Infinitesimales

Expon.
Mapa Exponencial

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal -> Grupo Continuo

Recursos
R

Generadores Infinitesimales

2

Problema Inverso. Calcular G conocidos (ξ, η) (cont.)

COROLARIO Las G-órbitas que pasan por el punto (x, y) son las curvas integrales de las ecuaciones de Lie

COROLARIO Si ξ y η son analíticas, las ecuaciones de Lie pueden resolverse mediante:

MAPEO EXPONENCIAL⁽¹⁾:

$$\begin{cases} \bar{x}_1 = f(x, y, \lambda) = e^{\lambda X}(x) \\ \bar{y}_1 = g(x, y, \lambda) = e^{\lambda X}(y) \end{cases}$$

siendo


$$e^{\lambda X} = \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} X^n$$

⁽¹⁾ El mapeo exponencial presenta una clara analogía con la representación de una función mediante su serie de Taylor.

L 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



Simetrías Lie y Ecuaciones Diferenciales

Indice

Tr. Infin.

Expon.

Invar.

Canón.

Gen->Gr

Recursos

Transformaciones Infinitesimales

Mapas Exponencial

Invariantes

Variables Canónicas

Generador Infinitesimal -> Grupo Continuo

2

Generadores Infinitesimales

Cálculo de G-órbitas a partir de (ξ, η)

E. g. Sea el generador infinitesimal del grupo continuo de rotaciones, dado por:

$$X = -y \frac{\partial}{\partial x} + x \frac{\partial}{\partial y} \quad \text{o, equivalentemente,} \quad \begin{cases} \xi(x, y) = -y \\ \eta(x, y) = x \end{cases}$$

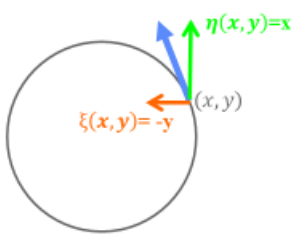
Las órbitas son las curvas integrales de las ecuaciones de Lie:

$$\begin{cases} x'(\lambda) = -y \\ y'(\lambda) = x \end{cases}$$

Las curvas integrales, aplicando la regla de la cadena, vienen dadas por la EDO:

$$y'(x) = \frac{y'(\lambda)}{x'(\lambda)} = -\frac{x}{y} \Rightarrow yy' = -x \Rightarrow \frac{y^2}{2} = -\frac{x^2}{2} + k \Rightarrow x^2 + y^2 = a^2$$


Son *circunferencias* centradas en el origen.



L 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



Simetrías Lie y Ecuaciones Diferenciales

Indice

Tr. Infin.

Expon.

Invar.

Canón.

Gen->Gr

Recursos

Transformaciones Infinitesimales

Mapas Exponencial

Invariantes

Variables Canónicas

Generador Infinitesimal -> Grupo Continuo

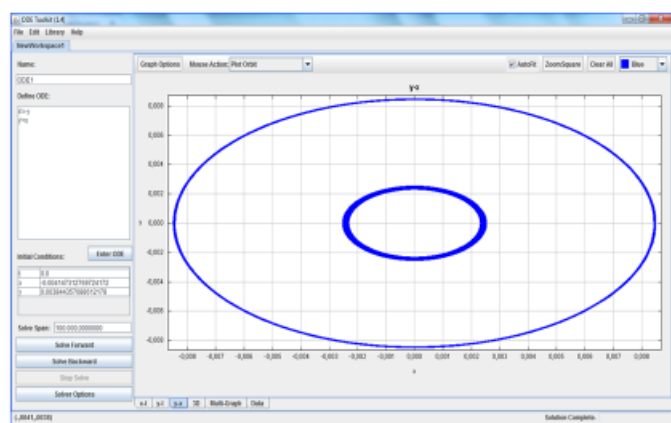
2

Generadores Infinitesimales

Cálculo de G-órbitas a partir de (ξ, η) (cont.)

E. g. (Cont.)

Las curvas integrales de un SEDO pueden estimarse numéricamente y dibujarse mediante paquetes específicos que analizan el diagrama de fases de un SEO, como ODEToolkit^(*)




L 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid

(*) ODEToolkit: [URL](http://www.ode-toolkit.com/)



INDUSTRIALES
ETSII | UPM

Simetrías Lie y Ecuaciones Diferenciales

Índice

1

Tr. Infin.

Expon.

Invar.

Canón.

Gen->Gr

Recursos

R

Transformaciones Infinitesimales

Mapas Exponencial

Invariantes

Variables Canónicas

Generador Infinitesimal
→ Grupo Continuo

2

Generadores Infinitesimales


Tabulación de Grupos de Transformaciones

	Transformación	ξ, η	X, Y	ψ	
1	Translación				
	a lo largo del eje x	$\begin{cases} \bar{x}_1 = x + \lambda \\ \bar{y}_1 = y \end{cases}$	$\begin{cases} \xi = 1 \\ \eta = 0 \end{cases}$	$X = \frac{\partial}{\partial x}$	$\psi = y$
	a lo largo del eje y	$\begin{cases} \bar{x}_1 = x \\ \bar{y}_1 = y + \lambda \end{cases}$	$\begin{cases} \xi = 0 \\ \eta = 1 \end{cases}$	$X = \frac{\partial}{\partial y}$	$\psi = x$
	a lo largo de recta ax+by=0	$\begin{cases} \bar{x}_1 = x + b\lambda \\ \bar{y}_1 = y + a\lambda \end{cases}$	$\begin{cases} \xi = 0 \\ \eta = 1 \end{cases}$	$X = b \frac{\partial}{\partial x} - a \frac{\partial}{\partial y}$	$\psi = x$
2	Rotación	$\begin{cases} \bar{x}_1 = x \cos \lambda + y \sin \lambda \\ \bar{y}_1 = y \cos \lambda - x \sin \lambda \end{cases}$	$\begin{cases} \xi = y \\ \eta = -x \end{cases}$	$X = y \frac{\partial}{\partial x} - x \frac{\partial}{\partial y}$	$\psi = x^2 + y^2$
3	Dilatación uniforme	$\begin{cases} \bar{x}_1 = x e^\lambda \\ \bar{y}_1 = y e^\lambda \end{cases}$	$\begin{cases} \xi = x \\ \eta = y \end{cases}$	$X = x \frac{\partial}{\partial x} + y \frac{\partial}{\partial y}$	$\psi = \frac{x}{y}$
4	Dilatación no uniforme	$\begin{cases} \bar{x}_1 = x e^\lambda \\ \bar{y}_1 = y e^{a\lambda} \end{cases}$	$\begin{cases} \xi = x \\ \eta = ay \end{cases}$	$X = x \frac{\partial}{\partial x} + ay \frac{\partial}{\partial y}$	$\psi = \frac{x^a}{y}$
5	Transform. de Lorentz	$\begin{cases} \bar{x}_1 = x \cosh \lambda + y \sinh \lambda \\ \bar{y}_1 = y \cosh \lambda + x \sinh \lambda \end{cases}$	$\begin{cases} \xi = y \\ \eta = x \end{cases}$	$X = y \frac{\partial}{\partial x} + x \frac{\partial}{\partial y}$	$\psi = y^2 - x^2$

L 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



INDUSTRIALES
ETSII | UPM

Simetrías Lie y Ecuaciones Diferenciales

Índice

1

Tr. Infin.

Expon.

Invar.

Canón.

Gen->Gr

Recursos

R

Transformaciones Infinitesimales

Mapas Exponencial

Invariantes

Variables Canónicas

Generador Infinitesimal
→ Grupo Continuo

2

Generadores Infinitesimales

Tabulación de Grupos de Transformaciones (cont.)

6	Transform. de Galileo	$\begin{cases} \bar{x}_1 = x + \lambda y \\ \bar{y}_1 = y \end{cases}$	$\begin{cases} \xi = y \\ \eta = 0 \end{cases}$	$X = y \frac{\partial}{\partial x}$	$\psi = y$
7	Transform. proyectivas				
	Tipo 1	$\begin{cases} \bar{x}_1 = \frac{x}{1-\lambda x} \\ \bar{y}_1 = \frac{y}{1-\lambda x} \end{cases}$	$\begin{cases} \xi = x^2 \\ \eta = xy \end{cases}$	$X = x^2 \frac{\partial}{\partial x} + xy \frac{\partial}{\partial y}$	$\psi = \frac{x}{y}$
	Tipo 2	$\begin{cases} \bar{x}_1 = \frac{x}{1-\lambda y} \\ \bar{y}_1 = \frac{y}{1-\lambda y} \end{cases}$	$\begin{cases} \xi = xy \\ \eta = y^2 \end{cases}$	$X = xy \frac{\partial}{\partial x} + y^2 \frac{\partial}{\partial y}$	$\psi = \frac{x}{y}$
8	Fracional lineal	$\begin{cases} \bar{x}_1 = \frac{x}{1-\lambda x} \\ \bar{y}_1 = \frac{y}{1-\lambda y} \end{cases}$	$\begin{cases} \xi = x^2 \\ \eta = y^2 \end{cases}$	$X = x^2 \frac{\partial}{\partial x} + y^2 \frac{\partial}{\partial y}$	$\psi = \frac{1}{y} - \frac{1}{x}$
9	Transform. conformes	$\begin{cases} \bar{x}_1 = \frac{x+\lambda y^2}{1+2\lambda x+\lambda^2 y^2} \\ \bar{y}_1 = \frac{y}{1+2\lambda x+\lambda^2 y^2} \end{cases}$	$\begin{cases} \xi = y^2 - x^2 \\ \eta = -2xy \end{cases}$	$X = (y^2 - x^2) \frac{\partial}{\partial x} - 2xy \frac{\partial}{\partial y}$	$\psi = \frac{y}{x^2}$

L 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid

Simetrías Lie y Ecuaciones Diferenciales

Indice
Tr. Infin.
Expon.
Invar.
Canón.
Gen->Gr
Recursos

Generadores Infinitesimales

2

$G \Rightarrow (\xi, \eta) \Leftrightarrow X$

Ej. 1

Determinar el generador infinitesimal asociado a los grupos de:

- a) Dilataciones no uniformes
- b) Transformaciones Projectivas

R

E 2.1

R

Manuel Mendoza de Haro

Universidad Politécnica de Madrid

Simetrías Lie y Ecuaciones Diferenciales

Indice
Tr. Infin.
Expon.
Invar.
Canón.
Gen->Gr
Recursos

Generadores Infinitesimales

2

$G \Rightarrow G - \text{órbitas}$

Ej. 2

Determinar las G-órbitas asociadas a los grupos de:

- a) Transformaciones de Lorentz
- b) Transformaciones Conformes

R


E 2.1

R

Manuel Mendoza de Haro

Universidad Politécnica de Madrid

Desarrollo de herramientas de soporte a un curso OCW-UPM sobre
resolución de ecuaciones diferenciales mediante simetrías de Lie

 INDUSTRIALES
ETSI I UPM

**Simetrías Lie y
Ecuaciones
Diferenciales**

Indice
1

Tr. Infin.
Transformaciones
Infinitesimales

Expon.
Mapas
Exponencial

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal
-> Grupo Continuo

Recursos
R

2

Generadores
Infinitesimales


$X \Rightarrow G - \text{órbitas}$

Ej. 3 Determinar las G-órbitas asociadas a los generadores infinitesimales de:
a) Dilataciones uniformes
b) Transformaciones de Galileo

E 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid

 INDUSTRIALES
ETSI I UPM

**Simetrías Lie y
Ecuaciones
Diferenciales**

Indice
1

Tr. Infin.
Transformaciones
Infinitesimales

Expon.
Mapas
Exponencial

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal
-> Grupo Continuo

Recursos
R

2

Generadores
Infinitesimales


Práctica: Calcular órbitas analíticamente (ec. Transformación y ec. Lie) y comprobarlas con ODE-Toolkit

Pendiente. EDP asociada a SEDO

P 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid



INDUSTRIALES
ETSI / UPM

Simetrías Lie y Ecuaciones Diferenciales

Índice

I

Tr. Infin.
Transformaciones Infinitesimales

Expon.
Mapas Exponencial

Invar.
Invariantes

Canón.
Variables Canónicas

Gen->Gr
Generador Infinitesimal -> Grupo Continuo

Recursos

R

Generadores Infinitesimales

2

Q 2.1					
Dado G , grupo continuo uniparamétrico de transformaciones, y ξ , generador infinitesimal asociado					
	Q1	Un grupo continuo puede tener un número finito de transformaciones.	NO		
	Q2	Si un grupo tiene un número infinito de transformaciones, es continuo.	SI		
	Q3	Dado G grupo de transformaciones, su generador infinitesimal es único.	NO		
	Q4	Dado G , grupo de transformaciones, su generador infinitesimal es único.	SI		
	Q5	Si los coeficientes del generador infinitesimal son funciones analíticas, el grupo generador	Existe y es único	Existe pero no es único	No está garantizada la existencia
check no radio button	Q6	Dado, G Grupo Uniparamétrico Global de Transformaciones, los G -órbitas de dos puntos distintos P_1 y P_2	No pueden cortarse	Si se cortan son coincidentes	Si son coincidentes, P_2 es transformado de P_1 y P_1 es transformado de P_2
check no radio button	Q7	Los coeficientes del generador infinitesimal ξ y η	no pueden anularse simultáneamente	Pueden ser iguales	Alguno de ellos puede anularse
	Q8	En la tabla de las transformaciones proyectivas, los pares de las variables x e y se intercambian entre las de tipo 1 y 2, sin embargo tienen el mismo invariante	Es un error	y/x no es invariante en el caso 2	y/x es invariante en el caso 2
	Q9	Las ecuaciones de Lie	siempre tienen solución	de haber solución, es única, aunque puede no haberla	Permite determinar el grupo continuo a partir de su generador infinitesimal
	Q10	El trapeo exponencial	Bajo ciertas hipótesis, permite determinar el grupo continuo uniparamétrico de transformaciones conocido su generador infinitesimal	Permite determinar el generador infinitesimal conocido un grupo continuo	Es un método para calcular las órbitas de un grupo continuo o

Q 2.1

Manuel Mendoza de Haro

Universidad Politécnica de Madrid

14. Anexo II. Guía del usuario DELHI

Como se indica en el Capítulo 7, este anexo contiene la **Guía del Usuario**, un manual de instrucciones para los usuarios sobre cómo utilizar la plataforma DELHI desarrollada para este proyecto. En ella se detalla cada uno de los apartados de la interfaz gráfica y cómo comenzar a trabajar con ellos.

Abrir la aplicación

Se debe acceder al directorio donde se haya descargado la aplicación DELHI, tal y como se detalla en el apartado 1 del Capítulo 7. Una vez allí, se deberá pulsar sobre el icono de DELHI y esperar a que la aplicación se abra.

Ventana principal

Una vez abierta la aplicación, se muestra la ventana principal de la misma, que contiene una imagen de portada. Aquí se puede leer el nombre de DELHI y se visualizan los logos de la Universidad Politécnica de Madrid, de la Escuela Técnica Superior de Ingenieros Industriales y del lenguaje Python, en el cual está desarrollada la herramienta.

A la izquierda, aparece un menú lateral bajo el icono de la Escuela de Industriales. Se trata del menú principal de la aplicación. Pulsando en los botones que contiene se accede a los diferentes apartados de la aplicación. Este menú consta de cinco botones, cuya función se describe a continuación.

HOME

Permite acceder a la ventana principal que muestra la portada de DELHI con los logos descritos anteriormente. No tiene ninguna funcionalidad activa programada más allá de servir de ventana de inicio de la aplicación. Sin embargo, pulsando el botón se puede volver a ella en cualquier momento sin perder el progreso en el resto de secciones.

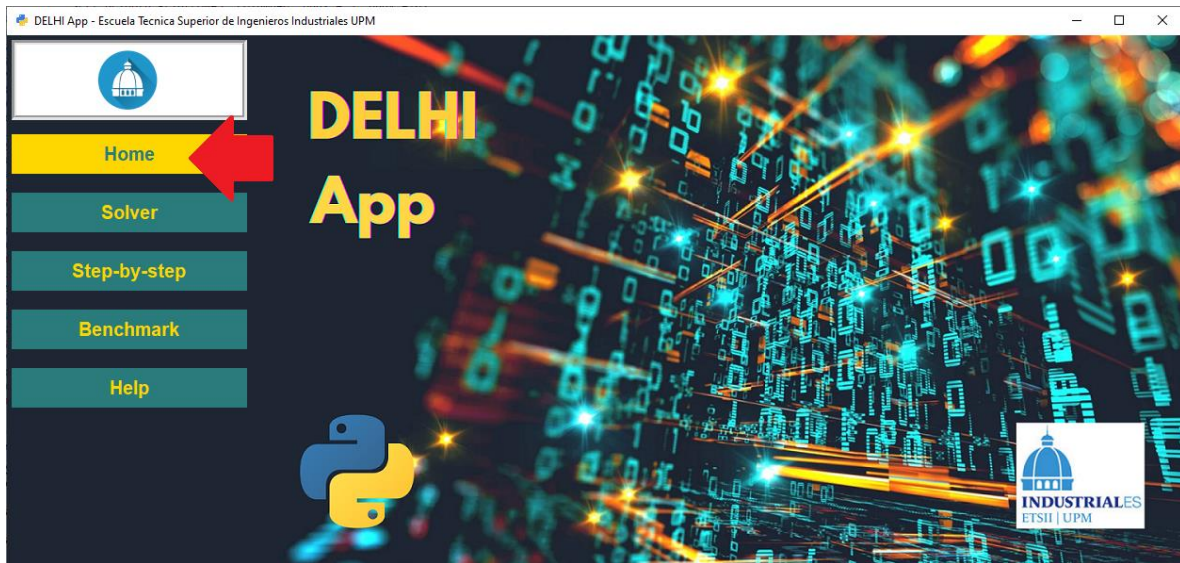


Figura 32. Botón Home y portada de DELHI

SOLVER

¿Para qué sirve?

Se trata de un solucionador sencillo de ecuaciones diferenciales mediante simetrías de Lie. El usuario debe introducir una ecuación en el lugar indicado, y después pulsar el botón de “Solve”.

Se intentará resolver la ecuación mediante las simetrías de Lie en primer lugar y, si la ecuación no es resoluble mediante este método, se pasará entonces a resolverla mediante cualquier otro procedimiento de los contemplados que sea adecuado para dicha ecuación.



Figura 33. Detalle del botón Solver.

Cómo introducir una ecuación

La ventana consta de una caja de texto, indicada con una oración. En ella, se debe escribir la ecuación diferencial a resolver. Ha de tenerse en cuenta la siguiente notación para que el programa interprete correctamente los términos de la ecuación.

Han de utilizarse los símbolos “+”, “-”, “*”, “/” y “^” para representar la suma, resta, multiplicación, división y potencia, respectivamente. Para esta última también puede utilizarse los caracteres “**”, más cercanos a cómo se representan las potencias en lenguaje Python. Pueden utilizarse paréntesis para encerrar numeradores, denominadores o cualquier otra expresión que deba interpretarse previamente al resto de operadores.

Para representar la variable independiente ha de usarse la letra “x” en todos los casos. La función incógnita ha de ser la variable “y”.

Al tratarse de una ecuación diferencial, es importante disponer de una notación para las derivadas de la función incógnita. De esta manera, representamos la primera derivada de “y” respecto de “x” como “dy”.

Si la ecuación diferencial incluye funciones matemáticas especiales, como las trigonométricas, hiperbólicas, logarítmicas o exponenciales, su notación ha de ser la habitual para estas funciones en inglés. Algunos ejemplos pueden ser la función coseno que se representa por $\cos()$, o el seno hiperbólico, $\sinh()$.

Se adjuntan algunos ejemplos de ecuaciones diferenciales y los equivalentes de cómo deberían introducirse en DELHI para ser correctamente interpretadas.

$$x^4 - 2x^2y - 2x + y^2 + y' - 1 = 0$$

$$\text{Equivalente: } x^4 - 2 * x^2 * y - 2 * x + y^2 + dy - 1 = 0$$

Figura 34. Ejemplo 1 de introducción de ecuación.

$$-xy - x - y^2 + y' + 1 = 0$$

$$\text{Equivalente: } -x * y - x - y^2 + dy + 1 = 0$$

Figura 35. Ejemplo 2 de introducción de ecuación.

$$y \cos(x) + y' = e^{2x}$$

$$\text{Equivalente: } y * \cos(x) + dy = \exp(2 * x)$$

Figura 36. Ejemplo 3 de introducción de ecuación.

Resolver la ecuación introducida

Una vez introducida la ecuación en la caja de texto correspondiente con la notación indicada, es necesario pulsar el botón “*Solve*” para que el programa resuelva la ODE.

Primero se comprueba que la sintaxis es correcta. De no ser así, un mensaje en la ventana indicará que hay un error y deberá corregirse la ecuación introducida.

Si todo es correcto, el programa indicará el método que ha utilizado para resolver la ecuación. La solución de la ecuación se muestra en la parte inferior de la pantalla.

Opciones de resolución

Este apartado no permite configurar la resolución de la ecuación diferencial. Para personalizar en detalle la resolución debe utilizarse el apartado *Step-by-Step*. La única funcionalidad adicional a la de resolver la ecuación es el botón *Clear*, que permite borrar la ecuación introducida y el resultado para regresar al estado por defecto de esta pantalla.

STEP-BY-STEP

¿Para qué sirve?

Se trata de una versión más avanzada del anterior apartado “Solver”. Como su nombre indica, descompone la resolución de la ecuación en varios pasos, cada uno dedicado a una fase diferente del procedimiento de simetrías de Lie. El interés de este apartado reside en que permite al usuario conocer los detalles de los pasos intermedios del método de resolución, u obtener información de un paso concreto sin necesidad de tener que solucionar el problema al completo.

Paso 1. Obtener los *infinitesimals*

En este primer paso, el usuario debe introducir una ecuación diferencial y elegir el método que desea para obtener las simetrías ξ y η de la ecuación en cuestión.

En la parte superior de la ventana se observa un panel de pestañas. Al entrar por primera vez, se accederá automáticamente a la primera de ellas, que abre la ventana del primer paso.

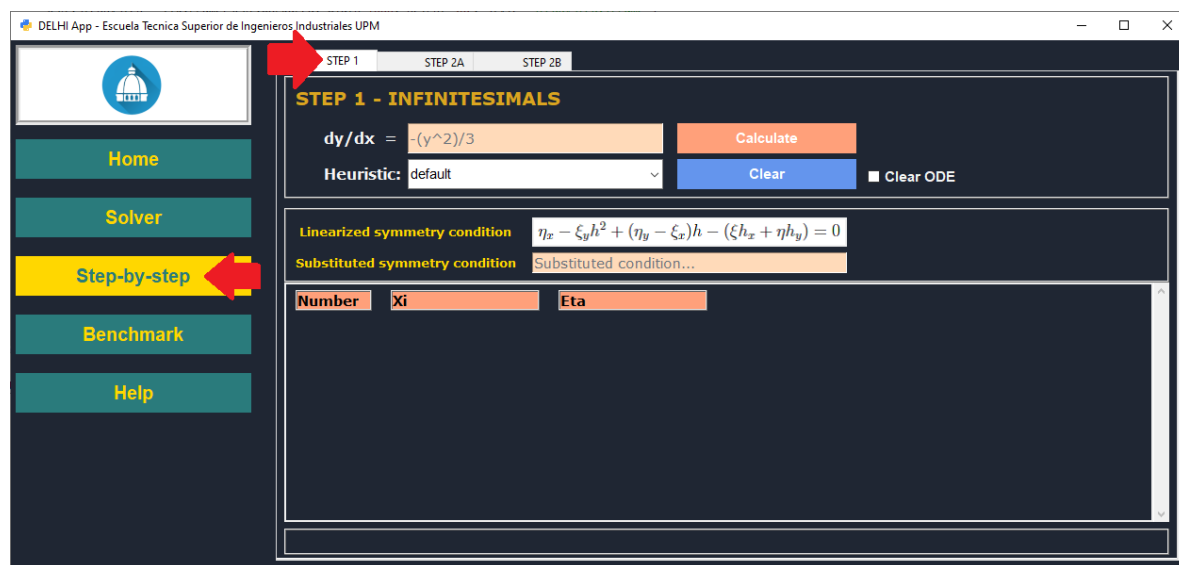


Figura 37. Detalle del acceso al primer paso.

Introducir una ecuación

En la caja de texto indicada se debe introducir la ecuación que se desea estudiar, siguiendo la notación descrita para el apartado *Solver*, a excepción de que, en esta ocasión, y como regla en todo el apartado *Step-by-step* se debe escribir en forma normal tal y como se indica en la ventana. Esto es, la derivada de la función incógnita ha de estar despejada en el primer miembro de la ecuación.

Elegir una heurística

A continuación, se debe elegir en el desplegable “*Heuristic*” qué heurística de Lie, de entre las 9 disponibles, se desea utilizar para obtener las simetrías de la ecuación. Si no se conocen los detalles de estas heurísticas, o no se está seguro de cual utilizar, se debe marcar la opción por defecto “*default*”, que comenzará a utilizarlas en orden hasta que una de ellas de un resultado válido.

Calcular las simetrías

Lo siguiente es pulsar el botón “*Calculate*” y esperar a que la aplicación realice las operaciones necesarias.

En este momento, aparecerán los resultados de este primer paso. Puede observarse un rótulo con la condición de simetría linealizada en la parte central de la ventana. Debajo de él, el programa se encarga de sustituir la ecuación introducida en la condición para mostrarnos la ecuación en derivadas parciales que se ha de resolver para obtener las simetrías. En función de la heurística seleccionada, esta EDP aparecerá simplificada en mayor o menor medida, lo cual facilitará si es el caso la resolución completa.

Si la heurística seleccionada es válida, se mostrará en la parte inferior un mensaje en color verde que informa al usuario de este hecho. Por consiguiente, se mostrarán en el recuadro central las simetrías encontradas. En caso de hallar más de una pareja xi-eta de simetrías, se mostrarán todas en forma de tabla. Si esta tabla es lo suficientemente larga se activa una barra lateral para hacer *scroll* hacia abajo y poder visualizar todas y cada una de las entradas de la tabla.

Si la heurística no es válida, no se mostrarán parejas de simetrías debido al error en los cálculos, y un mensaje rojo en el cuadro inferior así lo comunicará.

Más opciones

Una vez en este punto, el usuario dispone de varias opciones:

- Introducir una nueva ecuación y repetir todo el proceso
- Probar una heurística diferente para la misma ecuación
- Borrar los resultados y/o la ecuación introducida si así lo desea

Si desea analizar otra ecuación diferente puede introducirla borrando la previa y pulsando de nuevo el botón *“Calculate”*.

En caso de querer probar otra heurística, abriendo el desplegable podrá seleccionarla, a la par que se borra el mensaje de error o de confirmación de la parte inferior para evitar confundirlo con el que se generará a continuación. Después debe pulsarse el botón *“Calculate”* de nuevo para iniciar el proceso.

Por último, si desea borrar todos los resultados anteriores y regresar la ventana a su estado inicial, debe pulsarse el botón *“Clear”* para dicha tarea. Esto borrará todos los datos en pantalla salvo la ecuación introducida.

Existe una opción adicional para borrar también la ecuación junto al resto de la información, y es marcando la caja con un check, situada a la derecha del botón de borrar, antes de pulsar el propio botón *“Clear”*. De esta manera, toda la pantalla incluida la ecuación regresarán a su estado por defecto tal y como se encontraba al abrir la aplicación.

Paso 2. Resolver la ecuación introducida

Una vez calculadas las simetrías, el usuario puede utilizar esas funciones ξ y η para continuar los siguientes pasos de la resolución. El método de Lie tiene dos variantes, que se han representado aquí mediante las versiones A y B del paso 2. En ellas, se utiliza bien el factor integrante de Lie o el cambio de coordenadas canónicas para convertir la ecuación original en otra más sencilla, y por último se resuelve dicha ecuación y se muestra el resultado. Seguidamente puede encontrarse la descripción del uso de ambas metodologías en la herramienta DELHI.

Paso 2-A. Factor integrante de Lie

En este segundo paso, el usuario no deberá introducir de nuevo la ecuación, pues la caja de texto dedicada a ello se encuentra ya cumplimentada con la ecuación escrita en el paso 1. Esto ahorra tiempo de uso y repetición de tareas ya realizadas por el usuario que agilizan su interacción con la herramienta.

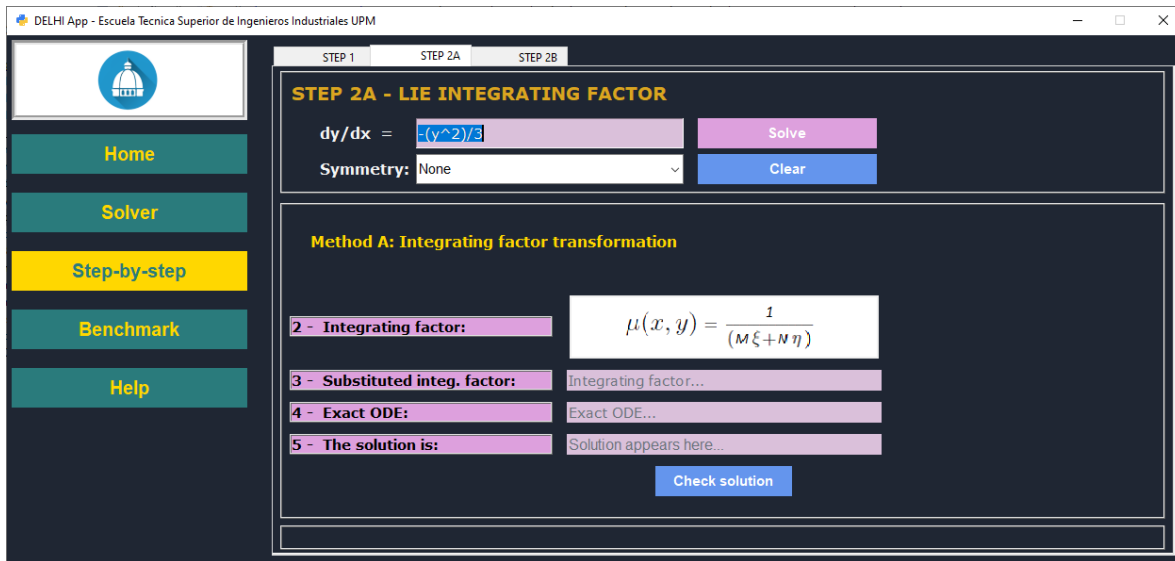


Figura 38. Detalle del acceso al paso 2A.

Elegir una simetría previa

Para acceder a este paso 2-A se debe pulsar en la pestaña correspondiente del panel de la parte superior de la ventana. La interfaz, similar a la del paso 1, difiere de ella tanto en estilo como en la distribución de la parte inferior donde se mostrarán los resultados.

Como la ecuación ya aparece escrita, la única tarea del usuario es abrir el desplegable “Symmetry” y elegir de la lista que aparece una de las simetrías previamente calculadas en el paso primero. Estas funciones se utilizarán para calcular la solución de la ecuación.

Calcular el factor integrante y la solución

Por último, una vez elegida la pareja de funciones ξ y η deseada, se debe pulsar el botón “Solve”.

Si todas las entradas son correctas, la aplicación calculará los diferentes resultados a mostrar. En primer lugar, se visualiza la fórmula del factor integrante generalizada. Tras pulsar el botón, en la caja de texto bajo ella aparecerá la expresión del factor integrante hallado para la ecuación y las simetrías elegidas.

Más abajo se podrá encontrar la ecuación diferencial exacta en la que se ha transformado la EDO original al multiplicarla por su factor integrante. De esta forma, su resolución es ahora mucho más sencilla al contar con un procedimiento conocido para las ecuaciones ordinarias de este tipo.

Finalmente, en la última entrada se podrá visualizar la solución de la ecuación original, hallada por el método de simetrías de Lie a través del cálculo del mencionado factor integrante.

Si todo es correcto, de nuevo en el recuadro de la parte inferior de la interfaz aparecerá un mensaje de color verde que nos proporciona esa información. En caso de encontrar algún error en la ecuación introducida o de no poder hallar la solución, el cuadro contendrá un mensaje de error de color rojo.

Chequear la validez de la solución

Siempre existe la posibilidad de que, a pesar de no haber sufrido ningún error al calcular la solución, la expresión mostrada por la aplicación sea incorrecta por alguna razón. La forma de que dispone el usuario de comprobar la validez de esta función encontrada es mediante el botón "*Check solution*". Una vez se disponga de una expresión en la caja de texto de la solución, se puede pulsar este botón para sustituirla automáticamente en la ecuación original, revisando si la verifica o no.

En caso afirmativo, la función solución encontrada es correcta y el cuadro de mensajes mostrará un rótulo de color azul comunicando que la función verifica la ecuación introducida en el paso 1. Por el contrario, si la solución no es válida de nuevo el color rojo destacará el mensaje de error que aparecerá.

Paso 2-B. Coordenadas canónicas

Como opción alternativa, se puede utilizar el cambio a coordenadas canónicas en esta ventana para calcular la solución de la ecuación.

Al igual que en el paso 2-A, en esta fase no será necesario introducir de nuevo la ecuación diferencial a resolver.

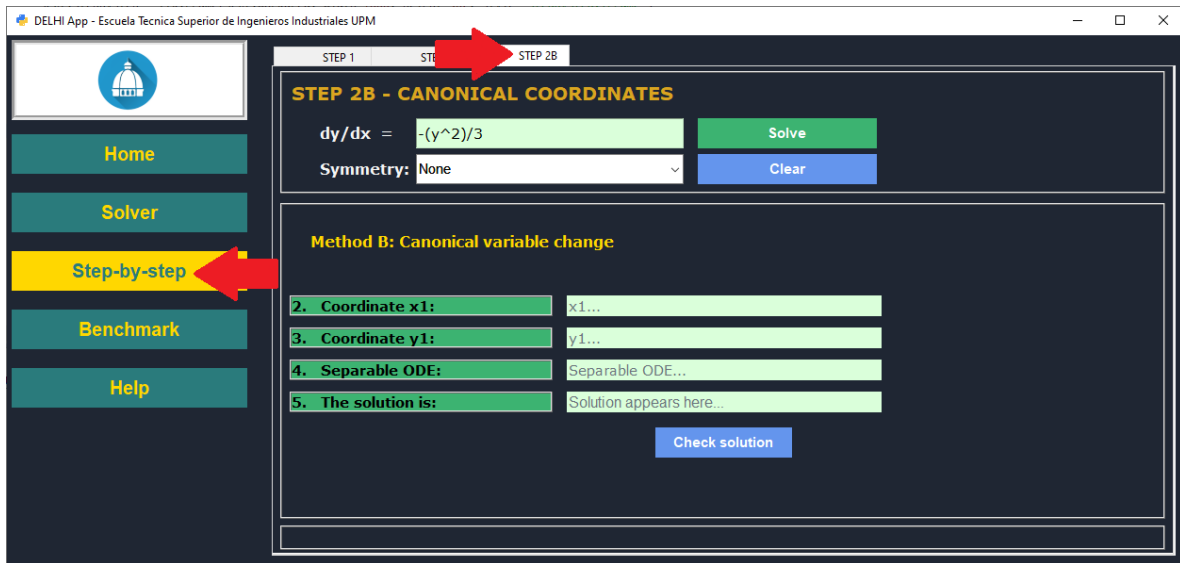


Figura 39. Detalle del acceso al paso 2B.

Elegir una simetría previa

De manera análoga a la descrita para el factor integrante de Lie, en este paso el usuario debe elegir en el desplegable “Symmetry” una de las parejas de funciones *infinitesimals* calculadas en el *Step 1*.

Aplicar el cambio de variable y calcular la solución

De nuevo, una vez elegida la simetría deseada, se debe pulsar el botón “Solve” y comenzará la ejecución.

En la parte central aparece un recuadro similar al del paso 2-A que contiene toda la información a mostrar tras los cálculos.

En primer lugar, aparecen dos cajas de texto con el cambio de variable que se ha aplicado, representado por las variables x_1 e y_1 . Al aplicar este cambio de variable a la ecuación original, ésta se transforma en una ecuación ordinaria de tipo separable, de fácil resolución a través de la integración en la mayoría de los casos. Queda a merced de dicha integración la posibilidad de hallar una expresión para la función incógnita, o por el contrario obtenerla en forma de cuadratura.

La ecuación separable obtenida también se muestra en su recuadro correspondiente. Finalmente, y de forma paralela al método del factor integrante, en la parte inferior aparece la función solución de la ecuación original.

La funcionalidad de este apartado es, como se puede observar, muy parecida a la de la otra versión del paso 2, salvo en los detalles propios.

En la parte inferior de la interfaz vuelve a aparecer el cuadro de mensajes. En caso de proceder todo correctamente, se ilustrará en verde un mensaje que así lo indique. De otro modo, en caso de errores diversos el letrero brillará de color rojo avisando de que un fallo ha tenido lugar.

Chequear la validez de la solución

Haciendo referencia de nuevo al paralelismo entre las fases 2-B Y 2-B, se dispone asimismo del botón de *“Check solution”* para verificar la solución obtenida, descrito previamente. Su funcionamiento es igual que en el anterior apartado, es por ello que no se proporcionan más detalles.

BENCHMARK

¿Para qué sirve?

Esta sección de la aplicación permite al usuario visualizar los resultados de los test realizados con el conjunto de ecuaciones Kamke 88, descrito en detalle en el capítulo 5 del presente documento. La idea tras este apartado es mostrar de manera fácil las conclusiones obtenidas al realizar dichos experimentos sin necesidad de que el usuario los replique, con el consiguiente coste de tiempo y computación que conllevan. De esta forma, todo el trabajo realizado se ofrece desglosado para ser aprovechado rápidamente.

Number	ODE	Default Hint	Solvable	Lie Solvable	Time
1	$y' = \frac{1}{\sqrt{a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4}}$	nth_algebraic	Yes	Yes	3.14
2	$ay + y' = ce^{bx}$	1st_linear	Yes	Yes	1.02
3	$ay + y' = b \sin(cx)$	1st_linear	Yes	Yes	3.31
4	$2xy + y' = \frac{x}{e^{x^2}}$	1st_exact	Yes	Yes	1.68
5	$y \cos(x) + y' = e^{2x}$	1st_exact	Yes	Yes	9.96
6	$y \cos(x) + y' = \frac{\sin(2x)}{2}$	1st_exact	Yes	Yes	11.39
7	$y \cos(x) + y' = e^{-\sin(x)}$	1st_exact	Yes	Yes	7.03
8	$y \tan(x) + y' = \sin(2x)$	1st_exact	Yes	Yes	4.98
9	$y' = (a + \sin(\log(x)) + \cos(\log(x)))y$	separable	Yes	Yes	4.41

Tabla 15. Detalle del proceso para el Test 1

Visualizar los resultados del test 1

En la ventana que aparece, el usuario dispone de dos cajas de texto en las que deberá indicar el rango de ecuaciones cuyos resultados desea visualizar. El rango máximo, como es natural, es el que comienza en la primera ecuación y termina en la número 88. Pueden elegirse otros rangos cualesquiera que estén dentro del máximo, por ejemplo, desde la ecuación 22 a la 57.

Una vez elegido el rango deseado, a la derecha figura un desplegable que ofrece al usuario dos opciones. En él debe elegirse Test 1 para mostrar la información del mismo.

Por último, debe pulsarse el botón “Launch Test” visible a la derecha del desplegable, y la aplicación mostrará en el recuadro inferior una tabla con los siguientes campos:

- Number: Indica el número de la ecuación dentro del conjunto Kamke 88

- ODE: Muestra la expresión de la ecuación diferencial tras ser procesada aparte como lenguaje MathML de presentación, y extraída la imagen que se observa.
- Default hint: Se destaca aquí la clasificación que, por defecto, la función `classify_ode` asigna a dicha EDO para comenzar su resolución.
- Solvable: Al haberse lanzado el test de resolución con la clasificación por defecto, esta casilla indica “Yes” en caso de que la librería haya podido calcular la expresión de la solución, y “No” en caso contrario.
- Solvable Lie: Equivalente al campo anterior pero aplicado al método de Lie. Si se ha encontrado la solución mediante las simetrías, las casillas indicarán “Yes”. Si no ha ocurrido así, se indicará “No”.
- Time: Durante la realización de los test se han medido los tiempos de ejecución como variable extra a considerar al cuantificar la calidad de los métodos implementados en *Sympy*. Esta medición no debe tomarse como definitiva, pues es claramente dependiente del ordenador en el que se ejecuten los programas de test y de la calidad del código, no solo de la dificultad intrínseca que presente la ecuación diferencial.

15. Anexo III. Guía de referencia DELHI

Tal y como se indica en el capítulo 7, este Anexo contiene la Guía de Referencia del desarrollador del proyecto software llamado DELHI. Aquí se detallan los pormenores a nivel de programación de la aplicación *standalone* desarrollada, más allá de las funcionalidades de cara al usuario descritas en el Anexo anterior, la Guía del Usuario.

Estructura del proyecto

Los proyectos en Python suelen contar con varios ficheros de código fuente. Cada uno de estos ficheros se denomina módulo. A lo largo de esta sección (y específicamente en la siguiente) se van a presentar y describir cada uno de los módulos que componen el proyecto DELHI a nivel de software, así como las clases y métodos principales que, en conjunto, permiten ejecutar las funcionalidades de alto nivel de la aplicación.

Los módulos que componen el proyecto son los siguientes:

- *delhiSymbols.py*
- *main.py*
- *Content.py*
- *benchmark.py*

Adicionalmente, el proyecto dispone también de una serie de ficheros de texto, que almacenan toda una variedad de datos de entrada y salida de la aplicación y que son indispensables para su correcto funcionamiento. Se resumen en la consiguiente lista:

- *Description*: Contiene el texto a mostrar en el apartado *Help*, que ofrece una breve descripción interna de la aplicación DELHI al usuario. Editando este fichero puede cambiarse el contenido a desplegar en dicha ventana.
- *Kamke*: Indispensable para realizar los test del *Benchmark*, este fichero contiene las 88 ecuaciones del set *Kamke*, acondicionadas a un formato adecuado para procesarlas mediante los métodos del módulo homólogo.
- *Heuristics*: Fichero de salida en el que se graba una tabla con las heurísticas válidas para cada ecuación diferencial probada.
- *Odehints*: Similar al anterior, esta vez genera una tabla indicando qué clasificaciones obtiene cada ecuación con el método *classify_ode*.
- *Times*: Contiene dos columnas con los tiempos medidos durante la ejecución del solucionador de ecuaciones. La primera de ellas se corresponde con el tiempo consumido por el procedimiento por defecto de la función *dsolve*, y la segunda recoge los tiempos empleados por el método de Lie, representado por el *hint lie_group*.

Descripción de los módulos

A continuación, se describe brevemente cada módulo de los mencionados anteriormente. Los detalles menores pueden encontrarse en el código fuente de cada uno de ellos. Se ha intentado que el nombre de todos los ficheros, métodos y variables sea lo más explicativo posible, pues es una buena práctica contribuir a la lectura y mantenimiento del código por parte de cualquier desarrollador que trate con el proyecto.

Módulo **delhiSymbols.py**

Este módulo contiene todas las definiciones de variables, listas y símbolos que utilizan el resto de ficheros Python del trabajo. Esto lo convierte en un módulo imprescindible para lograr una correcta ejecución. Está diseñado de manera similar a un archivo de cabeceras del lenguaje C++, reuniendo aquí las declaraciones de los objetos y reservando la implementación de los métodos para el resto de ficheros.

Entre las variables definidas pueden destacarse los símbolos y funciones creadas con *Sympy* para representar las de las ecuaciones diferenciales que se pretenden resolver. Cabe mencionar también las listas de tiempos, elementos de la interfaz como etiquetas o cajas de entrada que se generan a través de bucles.

Módulo **main.py**

Se trata del módulo de inicialización. En él se ejecuta el bucle principal de la aplicación que permite a la interfaz de DELHI mantenerse actualizada ante los cambios. También se encarga de importar el resto de módulos e instanciar la clase *Content* (se describe seguidamente). Si se desea ejecutar el programa en modo depuración, se debe lanzar este fichero en primer lugar, y desde él se utilizarán el resto de manera transparente.

Módulo **Content.py**

Sin duda alguna el módulo más importante, así como el mayor en tamaño de todo el proyecto. Es el lugar donde se define la clase *Content* homóloga al módulo. Esta clase contiene dos partes claramente diferenciadas: primero aparece el constructor o método “__init__”, y después el resto de métodos de la clase.

En el constructor puede observarse toda la definición de la parte gráfica de DELHI. Todos los widgets y elementos relativos a la interfaz se albergan aquí. Además, se definen también algunos atributos de la clase que son igualmente necesarios para la ejecución de ciertos métodos. Al constar de una cantidad considerable de widgets, no es práctico

listarlos en la memoria, pues en el código fuente pueden encontrarse en orden de creación y con comentarios aclaratorios sobre la utilidad y posición de cada uno de ellos.

Funciones show

Se trata de un conjunto de funciones complementarias, esto es, que actúan sólo una al tiempo de entre todas ellas. Su cometido es sencillo: muestran el *Frame* adecuado al botón del menú lateral que el usuario haya pulsado, para visualizar el apartado correspondiente. La lista siguiente indica los nombres reales de estos métodos.

- *show_welcome*
- *show_solver*
- *show_stepbystep*
- *show_benchmark*
- *show_help*

Solvestep 2A y 2B

Como sus nombres indican, su tarea es recoger la ecuación diferencial introducida y resolverla. Cada una utiliza el procedimiento que corresponde al paso en el que ocurren: la primera resuelve utilizando el factor integrante de Lie y la segunda mediante el cambio a coordenadas canónicas.

Simplesolver

En este caso se trata de una función más sencilla que las dos anteriores. Dedicada a la sección *Solver*, se limita a utilizar *dsolve* con la heurística *lie_group*, y, en caso de encontrarse con un error, intentarlo con la *hint* por defecto que aparezca en la primera posición de la lista generada por *classify_ode*.

Funciones Checksol

Presentes en los pasos “step2A” y “step2B”, realizan la función asociada a los botones de *Check Solution*, esto es, sustituir la expresión obtenida de la solución en la ecuación original, y verificar si la solución es válida mediante esta sencilla comprobación.

Substitute

Necesaria para manipular las expresiones que el usuario introduce. Se encarga de sustituir, como su nombre en inglés indica, las variables simbólicas en tres lugares: el primero de ellos es la condición de simetría linealizada, para obtener dicha condición simplificada y obtener de ella las simetrías; también se utiliza en el cálculo del factor integrante de Lie; y por último a la hora de obtener las EDOs separables y exacta en cada versión del segundo *step* en la resolución paso a paso.

Módulo benchmarkv5.py

Este módulo es el programa que ha permitido realizar los experimentos descritos en el apartado de *Benchmark*. Seguidamente se muestran algunos de los métodos que se han utilizado para obtener las tablas de resultados expuestas en los capítulos principales del presente documento.

analyzeLine

Como bien se ha indicado al principio de este anexo, las ecuaciones diferenciales que componen el set *Kamke88* están almacenadas en un fichero de texto. Esta función se encarga de leer línea por línea el fichero indicado, y analizar su contenido para distinguir dos casos:

1. Si se trata de una línea comentada, es decir, que no se desea procesar, en su inicio cuenta con los caracteres `"/"`. Este método reconoce esa sintaxis y salta las líneas que comiencen de esa manera. Esto es útil cuando no se quiere procesar cierta ecuación o se introduce un comentario escrito en el fichero, pero no se desea que interrumpa la ejecución del programa ni que de origen a errores.
2. Si se trata de una línea que contiene una ecuación, el algoritmo la separa en sus dos miembros para crear un objeto *Eq*, definido en el capítulo 5. De esta manera se crea la ecuación como un objeto en el lenguaje Python, lo que más adelante permitirá resolverla.

obtainHints

Con este método el objetivo es encapsular la ejecución de la función de la librería *Sympy* *classify_ode* en bucle para cada ecuación leída con el método anterior y, a continuación, guardar los resultados en el fichero de texto `"odehints.txt"`. Así, la salida en Python nos ayuda a generar las tablas que, una vez formateadas adecuadamente, se presentan en esta memoria.

obtainInfinitesimals

Muy similar al método anterior, esta vez el objetivo es obtener los resultados de la función *infinitesimals* de *Sympy* y guardarlos en el fichero de texto "*heuristics.txt*". De esta forma, se crea la tabla que indica qué heurísticas son válidas para cada ecuación de las 22 resolubles por el método de Lie. Nuevamente, a esta tabla se le da el formato adecuado para que su presentación sea clara y puedan leerse bien los resultados.

solveODE

Uno de los métodos más utilizados en este Benchmark es *solveODE*. Su cometido es envolver el método nativo *dsolve* configurado para los siguientes parámetros:

- *ODE*: la correspondiente a la iteración en la que se encuentre el bucle que recorre el rango introducido por el usuario
- *Hint*: En este caso se utiliza el argumento "*default*", para que se intente hallar la solución por cualesquiera de los métodos posibles atendiendo a la clasificación de la ecuación.
- *Func*: siempre se busca la función incógnita "*y*" como solución.

Adicionalmente, en esta función se almacenan en la primera columna del fichero "*times.txt*" los tiempos de ejecución medidos para dicha configuración.

solveODELie

Muy similar al método anterior, los parámetros que se indican para *dsolve* son los mismos que en *solveODE* salvo el argumento *hint*, que ahora pasa a ser "*lie_group*" puesto que el objetivo es verificar si la ODE en cuestión es resoluble mediante las simetrías o no. Asimismo, se graban en la segunda columna del fichero de texto "*times.txt*" los tiempos de ejecución para este procedimiento de Lie. De esta manera se puede establecer una comparativa de tiempos entre este caso y el anterior.

16. Anexo IV. Tablas completas de resultados

En el presente anexo se incluyen las tablas más grandes, que no han sido incluidas en la memoria debido a esta característica.

Las tablas que se exponen en las próximas páginas ilustran la información sobre los test realizados en el apartado Benchmarking del capítulo 5.

Desarrollo de herramientas de soporte a un curso OCW-UPM sobre resolución de ecuaciones diferenciales mediante simetrías de Lie

EDO	Resuelta	Resuelta por Lie	Hint por defecto	Tiempo por defecto	Tiempo por Lie
1	Sí	Sí	<i>nth_algebraic</i>	3.14	9.56
2	Sí	Sí	<i>1st_linear</i>	1.02	2.23
3	Sí	Sí	<i>1st_linear</i>	3.31	7.83
4	Sí	Sí	<i>1st_exact</i>	1.68	1.70
5	Sí	Sí	<i>1st_exact</i>	9.96	6.44
6	Sí	Sí	<i>1st_exact</i>	11.39	9.80
7	Sí	Sí	<i>1st_exact</i>	7.03	6.23
8	Sí	Sí	<i>1st_exact</i>	4.98	3.23
9	Sí	Sí	<i>separable</i>	4.41	18.46
10	Sí	No	<i>1st_exact</i>	4.46	2.94
11	Sí	No	<i>1st_exact</i>	3.77	2.96
12	Sí	No	<i>separable</i>	1.77	2.87
13	Sí	No	<i>1st_power_series</i>	11.42	2.26
14	No	No	<i>1st_power_series</i>	0.60	1.32
15	Sí	Sí	<i>1st_power_series</i>	2.77	4.56
16	Yes	No	<i>1st_power_series</i>	10.05	6.11
17	Yes	Yes	<i>separable</i>	1.34	2.38
18	Yes	Yes	<i>1st_power_series</i>	4.03	80.42
19	Yes	Yes	<i>1st_power_series</i>	2.06	2.34
20	Yes	No	<i>1st_power_series</i>	4.20	3.65
21	Yes	No	<i>1st_power_series</i>	12.89	7.81
22	Yes	No	<i>1st_power_series</i>	5.53	9.57
23	Yes	No	<i>separable</i>	16.20	25.10
24	No	No	<i>1st_power_series</i>	0.62	1.72
25	No	No	<i>lie_group</i>	6.18	5.58
26	Yes	Yes	<i>separable</i>	5.78	9.54
27	Yes	No	<i>1st_power_series</i>	5.04	2.26
28	Yes	No	<i>1st_power_series</i>	8.72	16.38
29	Yes	Yes	<i>separable</i>	46.05	47.23
30	No	No	<i>lie_group</i>	9.56	8.85
31	Yes	Yes	<i>separable</i>	3.02	4.00
32	Yes	No	<i>1st_power_series</i>	12.32	14.47
33	Yes	No	<i>1st_power_series</i>	164.23	126.83
34	Yes	No	<i>Bernoulli</i>	2.26	9.53
35	Yes	No	<i>separable</i>	28.47	33.38
36	Yes	No	<i>1st_power_series</i>	5.16	2.76
37	Yes	No	<i>1st_power_series</i>	33.93	2.91
38	No	No	<i>lie_group</i>	6.01	5.80
39	Yes	No	<i>separable</i>	156.10	-
40	Yes	No	<i>1st_power_series</i>	15.63	3.44
41	Yes	No	<i>1st_power_series</i>	20.73	5.60
42	Yes	Yes	<i>1st_power_series</i>	10.63	52.29
43	Yes	No	<i>1st_power_series</i>	51.05	95.15
44	Yes	Yes	<i>Bernoulli</i>	0.87	4.88
45	Yes	No	<i>1st_power_series</i>	10.61	36.48
46	No	No	<i>lie_group</i>	41.58	41.82
47	No	No	<i>1st_power_series</i>	1.71	13.72
48	No	No	<i>1st_power_series</i>	2.47	17.32
49	No	No	<i>1st_power_series</i>	-	-
50	Yes	No	<i>1st_power_series</i>	65.96	110.87

Tabla 16. Test 1, primera parte

51	No	No	<i>1st_power_series</i>	-	-
52	No	No	<i>lie_group</i>	-	-
53	No	No	<i>1st_linear</i>	-	-
54	No	No	<i>1st_power_series</i>	-	-
55	No	No	<i>1st_power_series</i>	-	-
56	No	No	<i>1st_power_series</i>	-	-
57	Yes	No	<i>separable</i>	2.09	3.68
58	Yes	No	<i>1st_power_series</i>	9.35	3.11
59	Yes	No	<i>separable</i>	3.05	10.21
60	Yes	Yes	<i>separable</i>	1.55	2.01
61	Yes	No	<i>separable</i>	25.53	-
62	No	No	<i>lie_group</i>	-	-
63	Yes	No	<i>separable</i>	3.61	-
64	No	No	<i>1st_power_series</i>	-	-
65	No	No	<i>1st_exact</i>	-	-
66	Yes	No	<i>separable</i>	4.49	-
67	Yes	No	<i>separable</i>	4.60	-
68	Yes	No	<i>1st_power_series</i>	42.51	-
69	No	No	<i>1st_power_series</i>	-	-
70	No	No	<i>1st_power_series</i>	-	-
71	No	No	<i>1st_power_series</i>	-	-
72	No	No	<i>1st_power_series</i>	-	-
73	No	No	<i>1st_power_series</i>	-	-
74	No	No	<i>1st_power_series</i>	-	-
75	Yes	Yes	<i>separable</i>	1.34	0.98
76	Yes	Yes	<i>separable</i>	4.41	5.74
77	Yes	No	<i>1st_power_series</i>	24.21	-
78	Yes	No	<i>1st_power_series</i>	34.54	-
79	No	No	<i>1st_power_series</i>	-	-
80	Yes	No	<i>1st_power_series</i>	69.88	198.35
81	Yes	No	<i>1st_power_series</i>	13.74	131.67
82	Yes	No	<i>1st_power_series</i>	53.76	-
83	Yes	No	<i>1st_power_series</i>	10.55	-
84	Yes	Yes	<i>1st_power_series</i>	49.97	4.41
85	No	No	<i>lie_group</i>	7.91	8.01
86	No	No	<i>1st_power_series</i>	-	-
87	No	No	<i>lie_group</i>	-	-
88	Yes	No	<i>1st_power_series</i>	19.54	13.48

Tabla 17. Test 1, segunda parte.

Desarrollo de herramientas de soporte a un curso OCW-UPM sobre resolución de ecuaciones diferenciales mediante simetrías de Lie

#	nth_algebraic	separable	1st_exact	1st_linear	Bernoulli	almost_linear	1st_power_series	lie_group	nth_linear_constant_coeff_undetermined_coefficients	nth_linear_constant_coeff_variation_of_parameters	nth_linear_euler_eq_nonhomogeneous_variation_of_parameters	nth_algebraic_Integral	separable_Integral	1st_exact_Integral	1st_linear_Integral	Bernoulli_Integral	almost_linear_Integral	nth_linear_constant_coeff_variation_of_parameters_Integral	nth_linear_euler_eq_nonhomogeneous_variation_of_parameters_Integral	Métodos válidos
1	1	1	1	1	1		1	1		1	1	1	1	1	1	1	No	1	1	16
2				1	1	1	1	1	1	1					1	1	1	1		11
3				1	1	1	1	1	1	1					1	1	1	1		11
4			1	1	1	1	1	1		1				1	1	1	1			10
5			1	1	1	1	1	1						1	1	1	1			10
6			1	1	1	1	1	1						1	1	1	1			10
7			1	1	1	1	1	1						1	1	1	1			10
8			1	1	1	1	1	1					1	1	1	1	1			10
9	1	1	1		1			1				1	1	1	1	No	1			9
10			1	1	1	1	1	1						1	1	1	1			10
11			1	1	1	1	1	1						1	1	1	1			10
12		1	1				1	1					1	1						6
13							1	1												2
14							1	1												2
15							1	1												2
16							1	1												2
17		1	1				1	1					1	1						6
18							1	1												2
19							1	1												2
20							1	1												2
21							1	1												2
22							1	1					1							2
23		1					1	1					1							4
24							1	1												2
25								1					1							1
26		1					1	1					1							4
27							1	1												2
28							1	1												2
29		1	1		1		1	1					1	1		1				8
30								1												1
31		1	1				1	1					1	1						6
32							1	1												2
33							1	1												2
34					1		1	1								1				4
35		1			No		1	1					1							4
36							1	1												2
37							1	1												2
38								1												1
39		1					1	1					1							4
40							1	1												2
41							1	1												2
42							1	1												2
43							1	1												2
44					1		1	1								1				4

Tabla 18. Resultados de classify_ode para las hints representativas (1) del test 3.

[illegible]

Tabla 19. Resultados de `classify_ode` para las hints representativas (2) del test 3,

Desarrollo de herramientas de soporte a un curso OCW-UPM sobre
resolución de ecuaciones diferenciales mediante simetrías de Lie

#	nth_algebraic	separable	1st_exact	1st_linear	Bernoulli	almost_linear	1st_power_series	lie_group	nth_linear_constant_coeff_undetermined_coefficients	nth_linear_constant_coeff_variation_of_parameters	nth_linear_euler_eq_nonhomogeneous_variation_of_parameters	nth_algebraic_Integral	separable_Integral	1st_exact_Integral	1st_linear_Integral	Bernoulli_Integral	almost_linear_Integral	nth_linear_constant_coeff_variation_of_parameters_Integral	nth_linear_euler_eq_nonhomogeneous_variation_of_parameters_Integral	Métodos válidos
1	1	1	1	1	1		1	1		1	1	1	1	1	1	1		1	1	16
2				1	1	1	1	1	1	1					1	1	1	1		11
3				1	1	1	1	1	1	1					1	1	1	1		11
4			1	1	1	1	1	1						1	1	1	1			10
5			1	1	1	1	1	1						1	1	1	1			10
6			1	1	1	1	1	1						1	1	1	1			10
7			1	1	1	1	1	1						1	1	1	1			10
8			1	1	1	1	1	1						1	1	1	1			10
9		1	1	1		1		1				1		1	1		1			9
10			1	1	1	1	1							1	1	1	1			9
11			1	1	1	1	1							1	1	1	1			9
12		1	1				1					1		1						5
13							1													1
14								1												0
15							1	1												2
16							1													1
17		1	1				1	1				1		1						6
18							1	1												2
19							1	1												2
20							1													1
21							1													1
22							1													1
23		1					1					1								3
24																				0
25																				0
26		1					1	1				1								4
27							1													1
28							1													1
29		1	1		1		1	1				1	1			1				8
30																				0
31		1	1				1	1				1	1							6
32							1													1
33							1													1
34					1		1									1				3
35		1					1					1								3
36							1													1
37							1													1
38																				0
39		1					1					1								3
40							1													1
41							1													1
42							1	1												2
43							1													1
44					1		1	1								1				4

Tabla 20. Resultados de dsolve para las hints válidas (2)

45							1											1
46																		0
47																		0
48																		0
49							1											0
50																		1
51																		0
52																		0
53				1	1	1							1	1	1			6
54																		0
55																		0
56																		0
57		1	1				1					1	1					5
58							1											1
59		1					1					1						3
60		1	1				1	1				1	1					6
61		1	1									1	1					4
62																		0
63		1										1						2
64																		0
65																		0
66		1										1						2
67		1	1									1	1					4
68							1											1
69																		0
70																		0
71																		0
72																		0
73																		0
74																		0
75		1	1				1	1				1	1					6
76		1					1	1				1						4
77							1											1
78							1											1
79																		0
80							1											1
81							1											1
82							1											1
83							1											1
84							1	1										2
85																		0
86																		0
87																		0
88							1											1

Tabla 21. Resultados de dsolve para las hints válidas (2)

17. Anexo V. Glosario

Siglas

- TFG – Trabajo de Fin de Grado.
- PEC – Prueba de Evaluación Continua.
- EDO – Ecuación diferencial ordinaria. En ocasiones citada como ODE, siglas inglesas.
- EDP – Ecuación en derivadas parciales.
- MOOC – *Massive Online Open Course* (Curso Masivo Abierto En Línea).
- OCW – *OpenCourseWare*.
- ETSII – Escuela Técnica Superior de Ingenieros Industriales.
- UPM – Universidad Politécnica de Madrid.

Términos

- Clase – Encapsulación de atributos y métodos.
- Método – En programación, función que puede realizar un objeto de una clase.
- Objeto – Instancia de una clase, un espécimen de la misma.
- Argumento – Palabra clave enviada a una función para su utilización en un algoritmo.
- Constructor – Método que se ejecuta al instanciar un objeto de una clase.
- Módulo – Cada uno de los ficheros Python que componen un proyecto.
- *Hint* – Procedimiento o clasificación que la librería *Sympy* asigna a una ecuación diferencial para resolverla.
- *Heuristic* – Conjunto de asunciones sobre la forma de las funciones infinitesimales, que simplifican la resolución de la ecuación en derivadas parciales que permite obtenerlas.
- *Infinitesimals* – En nomenclatura inglesa se refiere a ξ y η , las funciones infinitesimales del método de resolución de simetrías de Lie.
- *Delhi App* – aplicación de escritorio desarrollada
- *Delhi Web* – adaptación web de la aplicación de escritorio

18. Anexo VI. Índice de figuras

<i>Figura 1. Ejemplo de interfaz con varios Frame.....</i>	<i>34</i>
<i>Figura 2. Ejemplo de interfaz con una etiqueta.....</i>	<i>35</i>
<i>Figura 3. Interfaz con botón de ejemplo.....</i>	<i>35</i>
<i>Figura 4. Ejemplo de una caja de texto Entry.....</i>	<i>35</i>
<i>Figura 5. Ejemplo de una distribución utilizando el método pack.....</i>	<i>38</i>
<i>Figura 6. Asunciones de “abaco1_simple”.....</i>	<i>45</i>
<i>Figura 7. Asunciones de “abaco1_product”.....</i>	<i>45</i>
<i>Figura 8. Asunciones de “abaco2_similar”.....</i>	<i>45</i>
<i>Figura 9. Asunciones de “function_sum”.....</i>	<i>46</i>
<i>Figura 10. Asunciones de “abaco2_unique_general”.....</i>	<i>46</i>
<i>Figura 11. Asunciones de “linear”.....</i>	<i>46</i>
<i>Figura 12. Solución de la ecuación 42 del conjunto Kamke.....</i>	<i>58</i>
<i>Figura 13. Avance en los contenidos docentes a nivel de módulo, sesión y actividad.</i>	<i>64</i>
<i>Figura 14. Módulos 1 y 2 del curso en Moodle UPM.....</i>	<i>67</i>
<i>Figura 15. Cuestionario 2.1 del curso en Moodle UPM.</i>	<i>68</i>
<i>Figura 16. Herramienta DELHI.....</i>	<i>69</i>
<i>Figura 17. Solver de DELHI.....</i>	<i>70</i>
<i>Figura 18. Step-by-Step de DELHI, detalle del paso 1.</i>	<i>71</i>
<i>Figura 19. Step-by-Step de DELHI, detalle del paso 2.</i>	<i>72</i>
<i>Figura 20. Step-by-Step de DELHI, detalle del paso 3.</i>	<i>72</i>
<i>Figura 21. Benchmark de DELHI.</i>	<i>73</i>
<i>Figura 22. Menú de ayuda de DELHI.....</i>	<i>74</i>
<i>Figura 23. Descripción de la web de OCW UPM.....</i>	<i>77</i>
<i>Figura 24. Utilidad ODE Solver de DELHI Web.....</i>	<i>81</i>
<i>Figura 25. Detalle del código MathML de la EDO número 1.....</i>	<i>82</i>
<i>Figura 26. Resultados del Benchmark en DELHI Web.....</i>	<i>83</i>
<i>Figura 27. Código QR a la web OCW UPM.....</i>	<i>90</i>
<i>Figura 28. Código QR al Moodle UPM.....</i>	<i>90</i>
<i>Figura 29. Código QR a DELHI web Benchmark.....</i>	<i>90</i>
<i>Figura 30. Código QR a DELHI web Solver.....</i>	<i>90</i>
<i>Figura 31. Índice de contenidos del Módulo 2.....</i>	<i>96</i>
<i>Figura 32. Botón Home y portada de DELHI.....</i>	<i>107</i>
<i>Figura 33. Detalle del botón Solver.....</i>	<i>108</i>
<i>Figura 34. Ejemplo 1 de introducción de ecuación.....</i>	<i>109</i>
<i>Figura 35. Ejemplo 2 de introducción de ecuación.....</i>	<i>109</i>
<i>Figura 36. Ejemplo 3 de introducción de ecuación.....</i>	<i>109</i>
<i>Figura 37. Detalle del acceso al primer paso.....</i>	<i>111</i>
<i>Figura 38. Detalle del acceso al paso 2A.....</i>	<i>114</i>
<i>Figura 39. Detalle del acceso al paso 2B.....</i>	<i>116</i>

19. Anexo VII. Índice de tablas

Tabla 1. Parámetros de dsolve.....	40
Tabla 2. Hints de dsolve.....	44
Tabla 3. Resultados de las 15 primeras ecuaciones de Kamke.....	48
Tabla 4. Ecuaciones resueltas y no resueltas en el Test 1.....	48
Tabla 5. Resumen del Test 1 incluyendo series de potencias.....	50
Tabla 6. Resumen de resultados excluyendo 1st_power_series.....	50
Tabla 7. Heurísticas de Lie y ecuaciones para las que pueden hallar las simetrías.....	52
Tabla 8. Resultados del test 2 para cada heurística.....	53
Tabla 9. Resultados del test 2 para cada heurística.....	53
Tabla 10. Resultados del segundo test para cada ecuación.....	54
Tabla 11. Heurísticas y simetrías de las EDOS exclusivas de Lie.....	57
Tabla 12. EDOS exclusivas y sus soluciones.....	58
Tabla 13. Resumen del material y su nivel de desarrollo.....	63
Tabla 14. Presupuesto desglosado del proyecto.....	95
Tabla 15. Detalle del proceso para el Test 1.....	118
Tabla 16. Test 1, primera parte.....	126
Tabla 17. Test 1, segunda parte.....	127
Tabla 18. Resultados de classify_ode para las hints representativas (1) del test 3.....	128
Tabla 19. Resultados de classify_ode para las hints representativas (2) del test 3.....	129
Tabla 20. Resultados de dsolve para las hints válidas (2).....	130
Tabla 21. Resultados de dsolve para las hints válidas (2).....	131