Ecuaciones Diferenciales

Fernando Mazzone

25 de febrero de 2017

Índice general

Capítulo 1

Breve introducción a Python y SymPy

Se trata de una masa puntual m suspendida de un punto por medio de una barra de longitud l a la que suponemos sin masa. Equivale al movimiento sobre una guía circular. Usaremos el ángulo α marcado en la figura, como variable dependiente.

1.1 Descripción

Python es un lenguaje de programación interpretado, abierto, facil de aprender, potente y portátil. Es utilizado en proyectos de todo tipo, no sólo aplicaciones científicas.

SciPy, Python científico, es un conjunto de módulos de python para distintos tipos de cálculos. Está integrado por los módulos, SymPy (para cálculos simbólicos), numpy (cálculos numéricos), matplotlib (gráficos) entre otros. En este curso sólo usaremos SymPy.

SymPy es una biblioteca de Python para matemática simbólica. Su objetivo es convertirse en un sistema de álgebra computacional (SAC) completo, manteniendo el código lo más simple posible para que sea comprensible y fácilmente extensible. SymPy está escrito enteramente en Python y no requiere de ninguna biblioteca externa.

1.2 Instalación

Son muchas las componentes requeridas para poder ejecutar los programas con los que trabajaremos en esta asignatura. Hay que instalar un interprete de python, los módulos que utilizaremos (sympy, matplotlib), es útil utilizar entornos integrados de desarrollo (IDE), que facilitan al usuario editores de código fuente (especializados con la sintáxis de python), consolas de comandos mejoradas (ipython, qt, etc). Otro recurso que se dispone son las notebooks, de las cuales hablaremos más adelante. Sería engorroso instalar todas estas componentes, que muchas veces tienen orígenes en desarrolladores diferentes, de manera independiente. Para nuestra fortuna existen, las así llamadas, *distribuciones*. Estas son programas que instalan todas las coponentes necesarias, o al menos muchas de ellas, de un determinado paquete de software. Recomendamos las siguientes distribuciones.

1.2.1. Anaconda

La versión de código abierto de Anaconda es una distribución de alto rendimiento de Python y R e incluye más de 100 de los paquetes científicos más populares asociados a estos lenguajes. Además, se puede acceder a más de 720 paquetes que pueden ser fácilmente instalados con Conda, un programa incluído en Anaconda para la gestión de paquetes. Anaconda tiene licencia BSD que da permiso para utilizar Anaconda comercialmente y para su redistribución. Al día que se escriben estas líneas, anaconda parece la opción más sencilla y completa para instalar todos los recursos para desarrollar los contenidos de estas notas. Existen versiones para linux, OS X y Windows.

Hay distribuciones específicas para distintos sistemas operativos

1.2.2. Windows

La distribución python(x,y) instala el interprete de python y todos los módulos de scipy. Además el entorno de desarrollo integrado (IDE) spyder.

1.2.3. linux

Aquí todo es más sencillo, el interprete de python suele venir con la distribución del SO y se pueden instalar los módulos, SymPy, NumPy, etc, recurriendo al administrador de paquetes o tipeando la sentencia adecuada en la línea de comandos.

1.2.4. Android

Qpython es una aplicación que permite ejecutar código python y una versión básica de sympy desde tablets y smartphones. Se descarga desde la plataforma google play.

1.2.5. Otros recursos de utilidad:

SageMath es un sistema de software de matemáticas, libre, de código abierto bajo la licencia GPL. Es construído sobre muchos paquetes de código abierto existentes: NumPy, SciPy, matplotlib, SymPy, Maxima, GAP, FLINT, R y muchos más. Se acceda a su poder combinado a través de un lenguaje común, basado en Python. Puede instalarse bajo linux o usarse en línea desde cualquier plataforma de manera remota, por ejemplo desde el sitio SageMathCloud.

Entre las útilidades destinadas a editor de código fuente para python, sobresale emacs. Esta herramienta de software libre puede extenderse, ampliarse y permite la edición de código fuente de muchos lenguajes de programación, incluídos los que más populares dentro de la matemática, python, LaTeX, Octave (lenguaje m). Claro está que provee la comunicación con los respectivos interpretes o compiladores y en el caso de lenguajes intepretados que pueden ser ejecutados desde una consola de manera interactiva, provee una consola.

1.3 Forma de trabajo: por medio de scripts e interactiva

Se puede trabajar de dos formas

1. Interactivamente, ingresando sentencias, de a una por vez, en la línea de comandos y obteniendo respuestas.

2. Haciendo un script (programa) donde se guardan todas las sentencias que se desea ejecutar. Posteriormente este script se puede ejecutar, ya sea desde la línea de comandos o en desde un IDE (spyder) oprimiendo un botón de ejecución.

1.4 Características del Lenguaje

Seguiremos en esta exposición a Wikipedia (2016) de manera cercana. Las principales características del lenguaje son:

- Interpretado. Es necesario un conjunto de programas, el interprete, que entienda el código python y ejecute las acciones contenidas en él.

- Implementa tipos dinámicos.

- Multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

- Multiplataforma.

- Es comprendido con facilidad. Usa palabras donde otros lenguajes utilizarían símbolos. Por ejemplo, los operadores lógicos `!, || y \&\&` en Python se escriben not, or y and, respectivamente.

- El contenido de los bloques de código (bucles, funciones, clases, etc.) es delimitado mediante espacios o tabuladores.

- Empieza a contar desde cero (elementos en listas, vectores, etc).

1.5 Elementos del Lenguaje

1.5.1. Comentarios

Hay dos formas de producir comentarios, texto que el interprete no ejecuta y que sirve para entender un programa.

La primera, para comentarios largos es utilizanda la notación `''' comentario '''`.

La segunda notación utiliza el símbolo `#`, no necesita símbolo de finalización pues se extienden hasta el final de la línea.

```
'''
Comentario  largo en un script de Python
'''
print "Hola mundo" # Comentario corto
```

El intérprete no tiene en cuenta los comentarios, lo cual es útil si deseamos poner información adicional en nuestro código como, por ejemplo, una explicación sobre el comportamiento de una sección del programa.

```
x = 1
x = "texto" # Esto es posible porque los tipos son asignados \
dinamicamente
```

1.5.2. Variables

Las variables se definen de forma dinámica, lo que significa que no se tiene que especificar cuál es su tipo de antemano y que una variable puede tomar distintos valores en distintos momentos de un programa, incluso puede tomar un tipo diferente al que tenía previamente. Se usa el símbolo = para asignar valores a variables. Es importante distinguir este = (de asignación) con el igual que es utilizado para definir igualdades en sympy, para ecuaciones por ejemplo.

1.5.3. Tipo de datos

Python implementa diferentes tipos de datos. Para la noción de *tipos de datos* en general ver Wikipedia (2017). A continuación describimos sumariamente algunos de los tipos más comunes presentes en Python. Cuando se utilizan módulos específicos (p. ej. sympy) la diversidad de tipos de datos se expande, con la incorporación de tipos Con significación matemática, p.ej. matrices, expresiones algebraicas, etc.

Tipo	Clase	Notas	Ejemplo
`str`	Cadena	Inmutable	`'Cadena'`
`list`	Secuencia	Mutable, puede contener objetos de diversos tipos	`[4.0, 'Cadena', True]`
`tuple`	Secuencia	Inmutable, puede contener objetos de diversos tipos	`(4.0, 'Cadena', True)`
`dict`	Mapping	Grupo de pares clave:valor	`{'key1': 1.0, 'key2': False}`
`int`	Número entero	Precisión fija, convertido en *long* en caso de overflow.	`42`
`long`	Número entero	Precisión arbitraria	`42L` ó `456966786151987643L`
`float`	Número decimal	Coma flotante de doble precisión	`3.1415927`
`complex`	Número complejo	Parte real y parte imaginaria *j*.	`(4.5 + 3j)`
`bool`	Booleano	Valor booleano verdadero o falso	`True` o `False`

Se clasifican en:

Mutable si su contenido puede cambiarse.

Inmutable si su contenido no puede cambiarse.

Se usa el comando `\type` para averiguar que tipo de dato contiene una variable

```
>>> x=1
>>> type(x)
<type 'int'>
>>> x='Ecuaciones'
>>> type(x)
<type 'str'>
```

1.5.4. Listas y tuplas

- Es una estructura de dato, que contiene, como su nombre lo indica, listas de otros datos en cierto orden. Listas y tuplas son muy similares.
- Para declarar una lista se usan los corchetes [], en cambio, para declarar una tupla se usan los paréntesis (). En ambos casos los elementos se separan por comas, y en el caso de las tuplas es necesario que tengan como mínimo una coma.
- Tanto las listas como las tuplas pueden contener elementos de diferentes tipos. No obstante las listas suelen usarse para elementos del mismo tipo en cantidad variable mientras que las tuplas se reservan para elementos distintos en cantidad fija.
- Para acceder a los elementos de una lista o tupla se utiliza un índice entero (empezando por "0", no por "1"). Se pueden utilizar índices negativos para acceder elementos a partir del final.
- Las listas se caracterizan por ser mutables, mientras que las tuplas son inmutables.

```
>>> lista = ["abc", 42, 3.1415]
>>> lista[0] # Acceder a un elemento por su indice
'abc'
>>> lista[-1] # Acceder a un elemento usando un indice negativo
3.1415
>>> lista.append(True) # Agregar un elemento al final de la lista
>>> lista
['abc', 42, 3.1415, True]
>>> del lista[3] # Borra un elemento de la lista usando un indice
>>> lista[0] = "xyz" # Re-asignar el valor del primer elemento
>>> lista[0:2] # elementos del indice "0" al "2" (sin incluir ultimo)
['xyz', 42]
>>> lista_anidada = [lista, [True, 42L]] # Es posible anidar listas
>>> lista_anidada
[['xyz', 42, 3.1415], [True, 42L]]
>>> lista_anidada[1][0] # Acceder a un elemento de una lista dentro de
    otra lista
True
```

```
>>> tupla = ("abc", 42, 3.1415)
>>> tupla[0] # Acceder a un elemento por su indice
'abc'
>>> del tupla[0] # No es posible borrar ni agregar
( Excepcion )
>>> tupla[0] = "xyz" # Tampoco es posible re-asignar
( Excepcion )
>>> tupla[0:2] # elementos del indice "0" al "2" sin incluir
('abc', 42)
>>> tupla_anidada = (tupla, (True, 3.1415)) # es posible anidar
>>> 1, 2, 3, "abc" # Esto tambien es una tupla
(1, 2, 3, 'abc')
>>> (1) #  no es una tupla, ya que no posee al menos una coma
1
>>> (1,) # si es una tupla
(1,)
>>> (1, 2) # Con mas de un elemento no es necesaria la coma final
(1, 2)
>>> (1, 2,) # Aunque agregarla no modifica el resultado
(1, 2)
```

1.5.5. Diccionarios

- Para declarar un diccionario se usan las llaves {}. Contienen elementos separados por comas, donde cada elemento está formado por un par clave:valor (el símbolo : separa la clave de su valor correspondiente).
- Los diccionarios son mutables, es decir, se puede cambiar el contenido de un valor en tiempo de ejecución.

- En cambio, las claves de un diccionario deben ser inmutables. Esto quiere decir, por ejemplo, que no podremos usar ni listas ni diccionarios como claves.
- El valor asociado a una clave puede ser de cualquier tipo de dato, incluso un diccionario.

```
>>> dicci = {"cadena": "abc", "numero": 42, "lista": [True, 42L]}
>>> dicci["cadena"] # Usando una clave, se accede a su valor
'abc'
>>> dicci["lista"][0]
True
>>> dicci["cadena"] = "xyz" # Re-asignar el valor de una clave
>>> dicci["cadena"]
'xyz'
>>> dicci["decimal"] = 3.1415927 # nuevo elemento clave:valor
>>> dicci["decimal"]
3.1415927
>>> dicci_mixto = {"tupla": (True, 3.1415), "diccionario": dicci}
>>> dicci_mixto["diccionario"]["lista"][1]
42L
>>> dicci = {("abc",): 42} # tupla puede ser clave pues es inmutable
>>> dicci = {["abc"]: 42} # No es posible que una clave sea una lista
( Excepcion )
```

1.5.6. Listas por comprensión

Una lista por comprensión es una expresión compacta para definir listas. Al igual que el operador lambda, aparece en lenguajes funcionales. Ejemplos:

```
>>> range(5) #  "range" devuelve una lista, empezando en 0 \
y terminando con el numero indicado menos uno
[0, 1, 2, 3, 4]
>>> [i*i for i in range(5)]
[0, 1, 4, 9, 16]
>>> lista = [(i, i + 2) for i in range(5)]
>>> lista
[(0, 2), (1, 3), (2, 4), (3, 5), (4, 6)]
```

1.5.7. Funciones

- Las funciones se definen con la palabra clave `def`, seguida del nombre de la función y sus parámetros. Otra forma de escribir funciones, aunque menos utilizada, es con la palabra clave `lambda` (que aparece en lenguajes funcionales como Lisp). Generalemente esta forma es apropiada para funciones que es posible definir en una sola línea.
- El valor devuelto en las funciones con `def` será el dado con la instrucción `return`.

```
>>> def suma(x, y = 2): # el argumento y tiene un valor por defecto
...     return x + y # Retornar la suma
...
>>> suma(4) # La variable "y" no se modifica, siendo su valor: 2
6
>>> suma(4, 10) # La variable "y" si se modifica
14
```

```
>>> suma = lambda x, y = 2: x + y
>>> suma(4) # La variable "y" no se modifica
6
>>> suma(4, 10) # La variable "y" si se modifica
14
```

1.5.8. Condicionales

Una sentencia condicional (`if condicion`) ejecuta su bloque de código interno sólo si `condicion` tiene el valor bolleano `True`. Condiciones adicionales, si las hay, se introducen usando `elif` seguida de la condición y su bloque de código. Todas las condiciones se evalúan secuencialmente hasta encontrar la primera que sea verdadera, y su bloque de código asociado es el único que se ejecuta. Opcionalmente, puede haber un bloque final (la palabra clave `else` seguida de un bloque de código) que se ejecuta sólo cuando todas las condiciones fueron falsas.

```
>>> verdadero = True
>>> if verdadero: # No es necesario poner "verdadero == True"
...     print "Verdadero"
... else:
...     print "Falso"
...
Verdadero
>>> lenguaje = "Python"
>>> if lenguaje == "C":
...     print "Lenguaje de programacion: C"
... elif lenguaje == "Python": # Se pueden agregar "elif" como se quiera
...     print "Lenguaje de programacion: Python"
... else:
...     print "Lenguaje de programacion: indefinido"
...
Lenguaje de programacion: Python
>>> if verdadero and lenguaje == "Python":
...     print "Verdadero y Lenguaje de programacion: Python"
...
Verdadero y Lenguaje de programacion: Python
```

1.5.9. Bucles

El bucle `for` es similar a otros lenguajes. Recorre un objeto iterable, esto es una lista o una tupla, y por cada elemento del iterable ejecuta el bloque de código interno. Se define con la palabra clave `for` seguida de un nombre de variable, seguido de `in`, seguido del iterable, y finalmente el bloque de código interno. En cada iteración, el elemento siguiente del iterable se asigna al nombre de variable especificado:

```
>>> lista = ["a", "b", "c"]
>>> for i in lista: # Iteramos sobre una lista, que es iterable
...     print i
...
a
b
c
>>> cadena = "abcdef"
>>> for i in cadena: # Iteramos sobre una cadena, que es iterable
...     print i, # una coma al final evita un salto de linea
...
a b c d e f
```

```
>>> numero = 0
>>> while numero < 10:
...     print numero
...     numero += 1,  #un buen programador modificara las variables de
    control al finalizar el ciclo while
...
0 1 2 3 4 5 6 7 8 9
```

Bibliografía

Wikipedia (2016). Python. Available from: https://es.wikipedia.org/wiki/Python.

Wikipedia (2017). Tipo de dato --- wikipedia, la enciclopedia libre. [Internet; descargado 25-febrero-2017]. Available from: https://es.wikipedia.org/w/index.php?title=Tipo_de_dato&oldid=97149088.

Capítulo 2

Generalidades

2.1 ¿Que son las ecuaciones diferenciales?

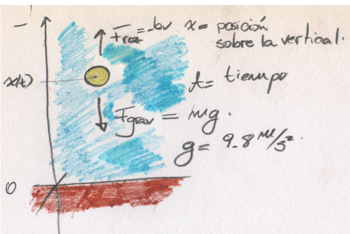
Definición 1 (Ecuación diferencial, definición informal).

Es una o varias relaciones entre una o varias variables dependientes y sus tasas de cambio respecto a ciertas variables independientes.

El problema básico asociado a las ecuaciones diferenciales es hallar las variables dependientes que las resuelven.

Las ecuaciones diferenciales son usadas muy a menudo en matemática aplicada, puesto que muchas leyes (de la física por ejemplo) se expresan atraves de este tipo de ecuaciones.

Ejemplo 1. [Caída libre] Modelizar matemáticamente el movimiento de un cuerpo de masa m en las proximidades de la superficie terrestre, asumiendo que su movimiento es sobre la vertical y que las fuerzas que sobre él actúan son la gravedad y el rozamiento con el aire.



$x(t) =$ posición a lo largo de la vertical, relativo a un eje de coordenadas
$v(t) = \frac{dx}{dt} =$ velocidad
$F_{grav} =$ fuerza debida a la gravedad $= mg$ donde $g = 9{,}8m/s^2$ $F_{roz} =$ fuerza de rozamiento, proporcional a la velocidad y de sentido contrario $= -cv$, $c > 0$.

Usamos la Segunda Ley de Newton, esto es la suma de las fuerzas totales que actuan sobre un cuerpo de masa m es igual al producto de la masa m y la aceleración $a(t)$. Recordemos que la aceleración es la tasa de cambio de la velocidad, es decir la derivada segunda de la posición.

Usando todas las relaciones mencionadas

$$ma(t) = mv'(t) = F_{\text{total}} = F\text{grav} + F\text{roz} = mg - cv$$

vale decir

$$\boxed{x''(t) + \frac{c}{m}x'(t) = g.} \tag{2.1}$$

$$\boxed{v'(t) + \frac{c}{m}v = g.} \tag{2.2}$$

2.2 Algunos conceptos relacionados con ecuaciones diferenciales

Definición 1 (Orden).

El índice de la mayor derivada interviniente en la ecuación.

Por ejemplo la ecuación (2.1) es de orden 2 y la ecuación (2.2), si bien está estrechamente relacionada con la anterior, es de orden 1. Como regla casi general, cuanto menor es el orden, más fáciles de estudiar y/o resolver las ecuaciones son.

Definición 2 (Solución).

Una función que satisface la relación que indica la ecuación.

Por ejemplo

$$v(t) = \frac{m}{c}g + ke^{-\frac{c}{m}t}, \tag{2.3}$$

resuelve (2.2), para todo $C \in \mathbb{R}$. No deberíamos perder tiempo en chequear una cuestión tan sencilla, pero aprovechemos la ocasión para usar SymPy.

```
>>>from sympy import *
>>>m,g,c,k,t=symbols('m,g,c,k,t')
>>>v=m/c*g+k*exp(-c/m*t)
>>>simplify(v.diff(t)+c/m*v)
g
```

Notar que las líneas que comienzan con el signo del prompt >>> indican entradas por línea de comandos y las que comienzan sin este signo son las respuestas del interprete.

Rara vez utilizaremos las siguientes funcionalidades de sympy, pero es oportuno decir que `SymPy` puede encontrar la solución a una ecuación diferencial

```
>>>v=symbols('v',cls=Function)
>>>EqCaida=Eq(v(t).diff(t)+c/m*v(t),g)
>>>Vel=dsolve(EqCaida,v(t))
>>> Vel
v(t) == (g*m + exp(c*(C1 - t/m)))/c
```

La solución obtenida es la ya conocida. Es instructivo averiguar que tipo de dato tiene la variable `Vel`

```
>>> type(Vel)
<class 'sympy.core.relational.Equality'>
```

A este tipo de cuestiones hay que prestar atención cuando se trabaja con sympy, pues existe la tendencia a confudir los conceptos matemáticos con los propios del lenguaje. Por ejemplo, matemáticamente una solución es una función. Sin embargo, en este caso, cuando le solicitamos una solución a sympy nos entrega un objeto de tipo ``relación de igualdad``.

Definición 3 (Ecuación diferencial ordinaria (EDO)).

Es una ecuación donde las variables dependientes sólo dependen de una única variable independiente.

Las ecuaciones (2.1) y (2.2) son ejemplo de ello, la variable independiente es el tiempo.

Definición 4 (Ecuación en derivadas parciales (EDP)).

Es una ecuación donde las variables dependientes dependen de más de una variable independiente.

Ejemplo de este tipo de ecuación es la ecuación de Laplace

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

Puede ocurrir también que dispongamos de varias ecuaciones diferenciales que se deben satisfacer simultaneamente. En estos casos, el conjuntos de ecuaciones se suele escribir como una única ecuación vectorial. Por este motivo, muchas veces, cuando dispongamos de una única ecuación diremos que tenemos una *ecuación escalar*.

Definición 5 (Sistema de ecuaciones).

Es un conjunto de ecuaciones diferenciales que se deben satisfacer simultaneamente.

En ese caso es de esperar que tengamos varias incognitas en nuestro problema. En general una ecuación escalar determina sólo una incognita. De hecho aquí ocurre, a semejanza con ecuaciones algebraicas, que es frecuente necesitar tantas ecuaciones como incognitas.
Ejemplo 2. [Ecuación del péndulo] El sistema de ecuaciones:

$$\begin{cases} x' &= y \\ y' &= -\operatorname{sen}(x) \end{cases}$$

es muy conocido pues modeliza el movimiento de un péndulo.

Muy a menudo hablaremos de resolver una ecuación, pero es oportuno discutir que queremos significar con esto.

Definición 6 (Resolver una ecuación).

Es expresar la solución como combinaciones algebraicas y composiciones de funciones que consideramos elementales.

Esta definición contiene una vaga apelación a ciertas ``funciones elementales''. El universo de funciones que se considera elemental es una cuestión política, no matemática. En principio, consideraremos elementales a las potencias, exponenciales, logarítmos, trigonométricas y trigonométricas inversas. No obstante esta lista se puede expandir con muchas funciones especiales. Las operaciones permitidas para combinar estas funciones también están sujetas a convenciones. Por ejemplo, admitiremos como válida una expresión que contenga una integral, al menos en el caso que no sea claro como resolver esta integral.

Casi todo este curso trata con la discusión de métodos para resolver ecuaciones. Sin embargo resolver ecuaciones no es quizás el problema principal relacionado con las ecuaciones diferenciales. No importa tanto lograr una expresión formal de la solución, como, por ejemplo, conocer las propiedades que poseen las soluciones. Al fin y al cabo, uno conoce una función a través de sus propiedades.

Definición 7 (Solución general).

Usualmente una ecuación presenta infinitas soluciones. Una solución general es una expresión que representa todas estas soluciones. Es habitual que una solución general contenga parámetros. Cada elección de estos parámetros determina una solución distinta.

Por ejemplo (2.3) es la solución general de (2.2). La afirmación anterior requiere una demostración puesto que sólo hemos mostrado que (2.3) es solución, pero no que toda solución se expresa con (2.3). Para demostrar la afirmación, hay que multiplicar ambos miembros de (2.2) por $e^{\frac{c}{m}t}$ y luego integrar respecto a t

$$\frac{mg}{c}e^{\frac{c}{m}t} = \int g e^{\frac{c}{m}t} dt = \int v'(t)e^{\frac{c}{m}t} + \frac{c}{m}e^{\frac{c}{m}t} v dt = e^{\frac{c}{m}t} v + C.$$

Despejando v del primer y último miembro obtenemos (2.3) con $k = -C$.

Ejemplo 3. En algunas ocasiones sólo podemos dejar una relación implícita entre las variable dependientes e independientes. Por ejemplo

$$x = e^y + y + C \quad \text{para } C \in \mathbb{R} \tag{2.4}$$

es solución general de

$$y'(e^y + 1) = 1.$$

Vamos a chequear sólo que (2.4) es solución, dejando la justificación que toda solución tiene esa forma para más adelante. Derivando (2.4)

$$1 = e^y y' + y'$$

Luego $y' = 1/(1 + e^y)$. Reemplazando esta relación en la ecuación diferencial corroboramos que es solución.

2.3 Definición formal

Definición 1 (Ecuación diferencial).

Una ecuación diferencial ordinaria de orden n es una relación de la forma

$$\boxed{F(x, y(x), y'(x), \ldots, y^{(n)}(x)) = 0}.$$

donde $F : (a, b) \times \Omega \to \mathbb{R}$, Ω es abierto de $\mathbb{R}^{n+1}$ y (a, b) un intervalo de $\mathbb{R}$.

Ejemplo 4. El problema de hallar una primitiva de una función es una ecuación diferencial que, como ya has visto en cursos iniciales de análisis, se relaciona con el concepto de integral. Supongamos $f : (a, b) \subset \mathbb{R} \to \mathbb{R}$ una función continua. Consideremos la ecuación diferencial

$$y'(x) = f(x).$$

Sea $x_0 \in (a, b)$, integrando respecto a x entre x_0 y x

$$y(x) = y(x_0) + \int_{x_0}^{x} f(t)dt$$

Que es una solución general. Quedaría determinada una única solución si, por ejemplo, conociecemos $y(x_0)$.
Ejemplo 5. Supongamos $f : (a, b) \subset \mathbb{R} \to \mathbb{R}$ como antes. Consideremos la ecuación diferencial

$$y''(x) = f(x).$$

Tomemos una integral indefinida respecto a x

$$y'(x) = C_1 + \int f(t)dt$$

Ahora deberemos tomar una integral indefinida más

$$y(x) = C_2 + C_1 t + \int \left(\int f(x)dx \right) dx.$$

Ahora quedan dos constantes C_1 y C_2.

Definición 2 (Principio de Hadamard).

Un problema se dice bien planteado segun Hadamard si satisface que

1. El problema admite solución
2. La solución es única
3. La solución depende de manera continua de los datos numéricos del problema.

Como hemos visto, una ecuación diferencial no determina una única solución, por consiguiente no sería un problema bien planteado. Debemos agregar relaciones a nuestro problema para que sea bien planteado. Es así que aparecen condiciones iniciales, problemas de contorno, etc.

Definición 3 (Problemas de valores iniciales).

Sea $x_0 \in (a,b)$, $F : (a,b) \times \Omega \to \mathbb{R}$ e $y_0, y_0^1, \dots, y_0^{n-1} \in \mathbb{R}$. Las siguientes relaciones se denominan problema de valores iniciales

$$\begin{cases} F(x,y,y', & \dots, y^{(n)}) = 0 \quad x \in (a,b) \\ y(x_0) & = y_0 \\ y'(x_0) & = y_0^1 \\ & \vdots \\ y^{(n-1)}(x_0) & = y_0^{n-1} \end{cases}$$

Definición 4 (Ecuaciones de primer orden).

La ecuación general de primer orden tiene la forma

$$F(x, y(x), y'(x)) = 0,$$

donde $F : (a,b) \times \Omega \to \mathbb{R}$ y Ω abierto de $\mathbb{R}^2$. Con frecuencia asumiremos que y' se despeja de la relación anterior, es decir que existe $f : \Omega' \to \mathbb{R}$, Ω' abierto de $\mathbb{R}^2$, tal que

$$y' = f(x,y).$$

Bajo esta suposición, si $(x_0, y_0) \in \Omega'$ el problema de valores iniciales se escribe

$$\begin{cases} y' & = f(x,y) \\ y(x_0) & = y_0 \end{cases}$$

2.4 Familias paramétricas de funciones

En las secciones siguientes vamos a describir problemas, matemáticos y físicos que se reducen a un problema de ecuaciones diferenciales.

Problema 1.

Dada una familia paramétrica de funciones

$$y = y(x,c), \tag{2.5}$$

dependiente del parámetro $c \in \mathbb{R}$, ¿Será posible hallar una ecuación para la cual la familia sea la solución general?

Familias paramétricas de funciones En líneas generales la respuesta es si. Nos conviene expresar (2.5) como una ecuación implícita

$$f(x,y,c) = 0. \tag{2.6}$$

Derivando esta ecuación respecto a x

$$\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} y'(x,c) = 0. \tag{2.7}$$

Ahora es posible eliminar c de (2.6) y (2.7) al costo de quedarnos con una sola ecuación.
Ejemplo 6. Encontrar la ecuación que satisface la familia paramétrica

$$x^2 + y^2 = c^2.$$

Derivamos

$$2x + 2yy' = 0.$$

Ya está!!!

Ejemplo 7. Idem $x^2 + y^2 = 2cx$. Derivando

$$2x + 2yy' = 2c.$$

Eliminamos c de las dos relaciones

$$\frac{x^2 + y^2}{x} = 2x + 2yy' \Rightarrow \boxed{y' = \frac{y^2 - x^2}{2xy}}.$$

Problema 2.

Dada una familia paramétrica

$$f(x, y, c) = 0,$$

encontrar otra

$$g(x, y, d) = 0$$

tal que los ángulos que forman los gráficos entre las funciones de una y de otra familia sean rectos en cada punto de corte entre ellos.

Para resolver este problema se completan estos pasos

- Se encuentra la ecuación diferencial que satisface la familia dada, digamos

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

- Se resuelve

$$y' = -\frac{1}{h(x, y)}.$$

Ejemplo 8. Encontrar la familia de curvas ortogonales a la flia de circunferencias

$$x^2 + y^2 = c^2$$

Hallamos antes que la ecuación que satisfacen estas curvas es

$$y' = -\frac{x}{y}.$$

Luego deberíamos resolver

$$y' = \frac{y}{x}.$$

Que no sabemos pero SymPy si!!!

Función SymPy (`dsolve`).

Sintaxis (documentación SymPy)
`dsolve(eq, f(x), hint)`
`eq`: Ecuación (posicional)
`f(x)`: función incognita (posicional)
`hint`: Método a emplear. Argumento con nombre `hint='cadena'`.

Ejemplo 9.

```
x=symbols('x')
y=Function('y')(x)
MiEcua=Eq(y.diff(x),y/x)
f=dsolve(MiEcua,y)
```

Resultado: Flía rectas por el origen.
La instrucción
`f=dsolve(MiEcua,y,hint='separable')`
produce el mismo resultado.

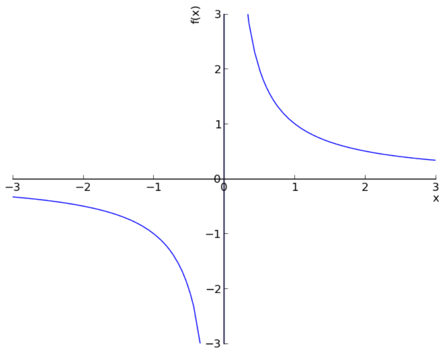
Función SymPy (`plot`).

Sintaxis (documentación `SymPy`)
Para un gráfico simple
`plot(expr,rango,opcionales(claves))`
`expr`: Expresión a graficar
`rango`: Conjunto donde varia la variable independiente
`opcionales` Argumentos que modifican la apariencia del gráfico. Generalemente de la forma de clave=valor

Ejemplo 10.

```
x=symbols('x')
f=plot(1/x,(x,-3,3),ylim=(-3,3))
```

Resultado:



Ejemplo 11. Grafiquemos un familias paramétrica de funciones.

```
from sympy import *
x,y=symbols('x,y')
Rango=range(21)
L=[tan(pi*k/21.0) for k in Rango]
p=plot(L[0]*x,(x,-2,2),show=False ,xlim=(-2,2),\
ylim=(-2,2),aspect_ratio=(1,1))
for pend in L[1:]:
    p1=plot(pend*x,(x,-2,2),show=False ,\
xlim=(-2,2),ylim=(-2,2),aspect_ratio=(1,1))
    p.append(p1[0])
for r in range(1,10):
    p1=plot_implicit(Eq(x**2 + y**2, 0.2*r),\
show=False ,aspect_ratio=(1,1),xlim=(-2,2),ylim=(-2,2))
    p.append(p1[0])
```

Resultado:

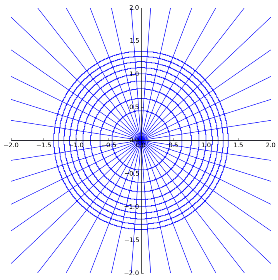


Figura 2.1: Familia curvas ortogonales

Problema 3 (Familias paramétricas de funciones en coordenadas polares)**.**

En ocasiones la ecuación de la familia de curvas esta dada en otras coordenadas. Por ejemplo supongamos que tenemos la flia de curvas dadas por una EDO en coordenadas polares

$$\frac{dr}{d\theta} = f(r, \theta),$$

y queremos hallar su flia ortogonal.

Solución: Calculemos dy/dx para las curvas en la familia dada.

$$\frac{dy}{dx} = \frac{dy/d\theta}{dx/d\theta} = \frac{r_\theta \operatorname{sen}\theta + r\cos\theta}{r_\theta\cos\theta - r\operatorname{sen}\theta} = \frac{f\operatorname{sen}\theta + r\cos\theta}{f\cos\theta - r\operatorname{sen}\theta},$$

donde $r_\theta = dr/d\theta$. La flia ortogonal tiene que satisfacer

$$\frac{dy}{dx} = -\frac{f\cos\theta - r\operatorname{sen}\theta}{f\operatorname{sen}\theta + r\cos\theta}$$

Luego

$$\frac{r_\theta \operatorname{sen}\theta + r\cos\theta}{r_\theta\cos\theta - r\operatorname{sen}\theta} = -\frac{f\cos\theta - r\operatorname{sen}\theta}{f\operatorname{sen}\theta + r\cos\theta}.$$

Si despejamos r_θ llegamos a la ecuación de la flia de curvas ortogonales en coordenadas polares

$$\boxed{\frac{dr}{d\theta} = -\frac{r^2}{f}.} \tag{2.8}$$

2.5 Separación de variables

Definición 1 (Ecuaciones en variables separadas)**.**

Se dice que en una ecuación de primer orden

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

separan las variables, si es posible la factorización $f(x, y) = g(x)h(y)$.

Una ecuación en la que se separan variables se puede resolver siguiendo los siguientes pasos. Como, asumiendo $h(y) \neq 0$,

$$\frac{y'}{h(y)} = g(x),$$

Si H y G son primitivas de $1/h$ y g respectivamente por la regla de la cadena tenemos

$$\frac{d}{dx}H(y) = \frac{d}{dx}G(x)$$

Luego

$$H(y) = G(x) + C, \quad C \in \mathbb{R}.$$

Por último, si podemos encontrar la inversa de H

$$\boxed{y(x) = H^{-1}\left(G(x) + C\right).} \tag{2.9}$$

será candidata a solución general. No podemos estar seguros de esta afrirmación, sobre todo porque la deducción de esta fórmula estuvo sujeta a suposiciones, como $h(y) \neq 0$.

Ejemplo 12. Resolver

$$y' = \frac{y}{x}.$$

Es común emplear el método de la siguiente forma

$$\begin{aligned}\frac{dy}{dx} = \frac{y}{x} \Longrightarrow \frac{dy}{y} = \frac{dx}{x} \Longrightarrow \int \frac{dy}{y} = \int \frac{dx}{x} \\ \Longrightarrow \ln|y| = \ln|x| + C \Longrightarrow |y| = k|x|, \text{ con } k > 0 \\ \Longrightarrow y = kx, \text{ con } k \in \mathbb{R}\end{aligned}$$

Función SymPy (`classify_ode`: clasificación de ecuaciones).

Sintaxis

`classify_ode(eq, f(x))`

Ejemplo 13.

```
x=symbols('x')
y=Function('y')(x)
MiEcua=Eq(y.diff(x),y/x)
tipo=classify_ode(MiEcua,y)
```

Resultado:

```
('separable', '1st_exact', '1st_linear',
'almost_linear', 'lie_group',  etc)
```

2.6 Galeria de Ejemplos

Vamos a describir algunos ejemplos. algunos de ellos llevan a problemas matemáticos muy simples. No obstante es oportuno discutirlos por dos motivos, habituarnos a la utilización de la matemática para resolver problemas de otras ciencias y sentar las bases para discutior problemas más relevantes desde una óptica matemática.

2.6.1. Ley de reproducción normal

En muchos ejemplos de biología, química-física, etc, hay magnitudes que crecen(decrecen) siguiendo una ley que denominaremos Ley de reproducción normal. Según esta ley la cantidad de individuos, sustancia, materia, energía, etc, que se agrega o elimina de una población, cuerpo, etc por unidad de tiempo es proporcional a la cantidad de individuos, sustancia, etc que hay presente. Una población de seres vivos puede reproducirse de esta manera bajo algunas circunstancias especiales, por ejemplo si cuenta con fuente ilimitada de alimentos.

Si $P(t)$ es la cantidad de individuos en el momento t, la ley de reproducción normal establece en este caso la siguiente ecuación diferencial de primer orden

$$P'(t) = kP(t), \quad \text{con } k > 0.$$

La solución es hallada con suma facilidad, siendo ella

$$\boxed{P(t) = Ce^{kt}, \quad C \in \mathbb{R}.}$$

La constante C se puede determinar si tenemos un problema a valores iniciales (pvi), por ejemplo $P(0) = P_0$, siendo $P_0 \in \mathbb{R}$ dado. En ese caso $A = P_0$.

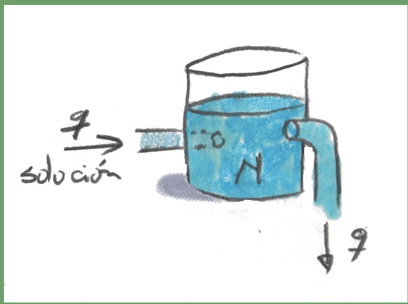
Otro ejemplo de comportamiento similar es la desintegración radiactiva. Algunos átomos de ciertas sustancias, pueden ``desarmarse'' en átomos de otras sustancias. En el proceso suelen emitir radiaciones. La velocidad de desintegración sigue una ley de reproducción normal pero hay que tener en cuenta que la materia radiactiva, es decir la ``población'' en este caso, se pierde. Si $x(t)$ es la masa de materia radiactiva en el momento t, evolucionará acorde a la ley

$$\boxed{x'(t) = -kx(t), \quad \text{con } k > 0.}$$

2.6.2. Soluciones

Problema 1.

Un tanque contiene inicialmente N m^3 de H_2O entre los cuales hay disueltos C kg de sal común $NaCl$. A través de una boca de entrada y una de salida empieza circular la solución, entrando y saliendo al mismo caudal $q\frac{\text{m}^3}{s}$. Se supone que la solución entrante tiene una concentración conocida r. Encontrar la cantidad de sal en el momento t.



Sea $x(t)$ la cantidad de $NaCl$ en el tanque en el momento t. entonces

$$\begin{aligned} x'(t) &= \text{cantidad que entra } - \text{cantidad que sale} \\ &= qr - q\frac{x(t)}{N} \end{aligned}$$

2.6.3. Dinámica del punto

Discusión Teórica

Vamos a recordar algunas temas de la asignatura física. En particular el movimiento de un cuerpo de masa m al que podemos suponer puntual. Lo llamaremos punto masa. Denotamos por $x(t)$ su posición, digamos en $\mathbb{R}^3$. Suponemos que sobre él actúa una fuerza f. Recordemos que $x'(t)$ es la velocidad $v(t)$ y que $x''(t)$ es la aceleración $a(t)$. La segunda ley de Newton implica que

$$\boxed{mx''(t) = f} \tag{2.10}$$

Supongamos que el movimiento del punto masa se realiza entre los momentos t_0 y t_1. Como has visto en Cálculo III la lóngitud de la curva recorrida $s(t)$ se puede calcular por

$$s = \int_{t_0}^{t_1} |x'(t)|dt = \int_{t_0}^{t_1} |v(t)|dt. \tag{2.11}$$

A s se lo suele denominar elemento de arco. Es comun querer utilizar a s como variable independiente en lugar de t, puesto que algunas fórmulas se simplifican de esta forma. Por ejemplo

$$f \cdot v(t)dt = f \cdot \frac{v(t)}{|v(t)|}|v(t)|dt = f_t ds, \tag{2.12}$$

donde f_t denota la proyección de la fuerza f sobre la dirección tangente a la trayectoria.

Si integramos (2.12) entre t_0 y t_1 y usamos (2.10) obtenemos

$$\begin{aligned} W := \int_{s_0}^{s_1} f_t(x(s))ds &= \int_{t_0}^{t_1} f(x(t)) \cdot v(t)dt \\ &= m\int_{t_0}^{t_1} v'(t) \cdot v(t)dt \\ &= \frac{m}{2}\int_{t_0}^{t_1} \frac{d|v|^2}{dt}dt \\ &= \frac{m}{2}|v(t_1)|^2 - \frac{m}{2}|v(t_0)|^2. \end{aligned}$$

A la cantidad $\frac{m}{2}|v|^2$ se la denomina *energía cinética* E_c y a W se lo denomina *trabajo*. Las relaciones obtenidas dicen que la variación de la energía cinética es igual al trabajo realizado $W = \Delta E_c$ por la fuerza f. El trabajo realizado depende de la proyección tangencial de la fuerza f_t. Llamaremos a esta relación Principio de Conservación de la Energía Mecánica.

Vamos a referirnos por *rapidez* al módulo de la velocidad. Si uno quiere incrementar o reducir la rapidez final $|v(t_1)|$ entonces deberá tener fuerzas con una componente tangencial no nula. Dicho de otra forma, si la fuerza es perpendicilar al movimiento, no hay cambio de rapidéz.

Hay fuerzas que siempre actuan en la dirección del movimiento. El ejemplo más conocido son las fuerzas de fricción, resistencia del aire, resistencia a la rodadura, etc. Estás fuerzas, actúan sólo en la dirección del movimiento y se oponen a él.

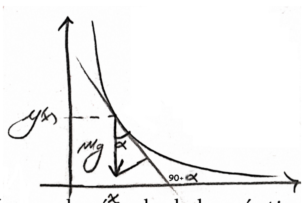
Por el contrario hay otras que actúan perpendiculares al movimiento $f_t = 0$. Ejemplo de ello son las fuerzas que mantienen a un cuerpo moviéndosé a lo largo de una guía. Por ejemplo un niño cayendo por un tobogan. Que el niño no se despegue de la guía (tobogán) se explica por la aparición de una fuerza que se denomina reacción de vínculo que actúa en la dirección perpendicular al movimiento esto es decir a la guía. Está fuerza debe compensar a toda otra fuerza que trata de apartar al cuerpo de la guía. En el caso del tobogán la gravedad trata de apartar al niño de aquel.

Cuerpos cayendo por guías

Analicemos más en detalle el movimiento de un cuerpo cayendo a lo largo de una guía estando además influído por la acción de la gravedad. Supondremos el movimiento en las proximidades de la superficie de la Tierra y por ello, supondremos que la fuerza de la gravedad es la constante mg.

Supongamos que la guía esta confinada a un plano. Introducimos un sistema de coordenadas ortogonales en dicho plano, con el suelo paralelo al eje x

El elemento longitud de arco s de una curva que es el gráfico de una función $y(x)$ para x en $[x_0, x_1]$ viene dado por



$$s = \int_{x_0}^{x_1} \sqrt{1 + y'(x)^2}dx$$

La fuerza de vínculo de la guía tiene componente tangencial nula, la gravedad tiene una componente tangencial no nula. Su magnitud es $mg\cos\alpha$ (ver dibujo). Vamos a tratar de expresar $\cos\alpha$ en términos de $y'(x)$. Vamos a suponer $\cos\alpha > 0$ e $y'(x) < 0$. Los demás casos quedan como **ejercicio**.

$$\tan^2\alpha = \frac{\operatorname{sen}^2\alpha}{\cos^2\alpha} = \frac{1-\cos^2\alpha}{\cos^2\alpha} = \frac{1}{\cos^2\alpha} - 1$$

y

$$y'(x) = \tan\left(\frac{\pi}{2} + \alpha\right) = -\frac{1}{\tan\alpha}$$

Podemos usar las relaciones anteriores para escribir $\cos\alpha$ en función de $y'(x)$

$$\cos\alpha = -\frac{y'(x)}{\sqrt{1+y'(x)^2}} \tag{2.13}$$

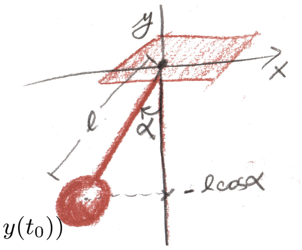
Entonces

$$\begin{aligned} \frac{m}{2}|v(t_1)|^2 - \frac{m}{2}|v(t_0)|^2 &= \int_{s_0}^{s_1} f_t ds = \int_{x_0}^{x_1} f_t \frac{ds}{dx}dx \\ &= -mg\int_{x_0}^{x_1} \frac{y'(x)}{\sqrt{1+y'(x)^2}}\sqrt{1+y'(x)^2}dx \\ &= -mg\,(y_1 - y_0) \end{aligned} \tag{2.14}$$

Esto nos permite escribir la rapidez en función de la altura repecto al piso.

Ejemplo 14. [Péndulo]

Se trata de una masa puntual m suspendida de un punto por medio de una barra de longitud l a la que suponemos sin masa. Equivale al movimiento sobre una guía circular. Usaremos el ángulo α marcado en la figura, como variable dependiente.



Supondremos que el origen del sistema de de coordenadas está sobre el punto de amarre de la barra. Entonces de (2.14) con $t_1 = t$ deducimos

$$\begin{aligned} \frac{m|v(t)|^2}{2} - \frac{m|v(t_0)|^2}{2} &= -mg\,(y(t) - y(t_0)) \\ &= mg\cos\alpha(t) - mg\cos\alpha(t_0). \end{aligned}$$

Ahora la posición de la masa es $x(t) = l(\operatorname{sen}\alpha, -\cos\alpha)$ luego

$$v(t) = l\alpha'(t)(\cos\alpha, \operatorname{sen}\alpha) \Longrightarrow |v(t)|^2 = l^2\alpha'(t)^2.$$

Entonces

$$\frac{ml^2\alpha'(t)^2}{2} = mgl\cos\alpha(t) - mgl\cos\alpha(t_0) + \frac{mv(t_0)^2}{2}.$$

Derivando esta relación

$$ml^2\alpha'(t)\alpha''(t) = -mgl\alpha'(t)\operatorname{sen}\alpha(t).$$

De esto deducimos la ecuación del péndulo

$$\boxed{\alpha''(t) = -\frac{g}{l}\operatorname{sen}\alpha(t)}.$$

Ejemplo 15. [La braquistócrona]

Problema 2.

Dados dos puntos A y B en las proximidades de la superficie terrestre, uno mas abajo respecto al suelo que el otro, queremos diseñar el tobogán óptimo entre los dos, esto es el tobogán que nos lleve de A hasta B en el menor tiempo. La curva solución a este problema se llama curva braquistócrona (braquistos - el más corto, cronos - tiempo).

Este problema fue resuelto por primera vez por Johann Bernoulli y es unos de los problemas precursores de la rama de las matemáticas que se denomina cálculo de variaciones. Vamos a dar la solución de Bernoulli que es muy elegante y está basada en un resultado de óptica llamado el Principio de Mínimo Tiempo de Fermat.

Principio de Mínimo Tiempo de Fermat

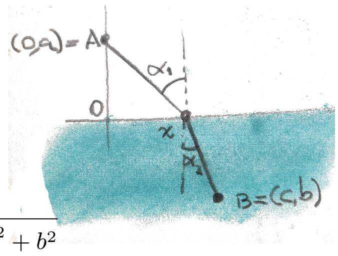
La luz sigue para ir de un punto a otro el recorrido que minimiza el tiempo.

La primera impresión es que ese reccorrido debería ser la línea recta. Si embargo esto no es así debido a que la velocidad de la luz cambia de acuerdo al medio que atraviesa. La velocidad de la luz en el vacío es 299.792,458km/h y en el diamante 124.034,943 km/h. La velocidad de la luz cambia no sólo con la sustancia sino con sus cualidades, como la densidad.

Si la luz se mueve dentro de un medio homogéneo, el camino que sigue es la línea recta. Esto ya no es más así cuando la luz cambia de medio de propagación. Por ejemplo cuando pasa del aire al vidrio.

Supongamos que la luz une los puntos A y B del plano y en el camino atravieza de un medio a otro, siendo la velocidad de la luz en cada uno de ellos v_1 y v_2. Supongamos que $A = (a, 0)$ y $B = (c, b)$ y el eje x es la frontera entre los medios.

Como sabemos que mientras se mueva en un medio homogéneo la luz sigue en línea recta, el tiempo que emplea la luz para ir A a B es



$$t = \frac{\sqrt{a^2 + x^2}}{v_1} + \frac{\sqrt{(c-x)^2 + b^2}}{v_2}$$

Para determinar la trayectoria es suficiente encontrar x, el punto donde la luz choca con la interfaz entre los medios. El principio de Fermat afirma que el tiempo es mínimo de modo que hallaremos un punto crítico de t respecto a x.

$$\frac{dt}{dx} = -\frac{c-x}{\sqrt{(c-x)^2 + b^2 v_2}} + \frac{x}{\sqrt{a^2 + x^2 v_1}} = \frac{\operatorname{sen} \alpha_1}{v_1} - \frac{\operatorname{sen} \alpha_2}{v_2}$$

Deducimos que en un punto crítico

$$\boxed{\frac{\operatorname{sen} \alpha_1}{v_1} = \frac{\operatorname{sen} \alpha_2}{v_2}} \tag{2.15}$$

que se denomina Ley de Snell. El punto crítico es mínimo pues $\left.\frac{dt}{dx}\right|_{x=0} = -\frac{c}{bv_2} < 0$ y $\left.\frac{dt}{dx}\right|_{x=c} = \frac{c}{\sqrt{c^2+a^2 v_1}} > 0$.

A la razón entre la velocidad de la luz dentro de un determinado medio y la velocidad de la luz en el vacio se lo denomina índice de refracción y se lo denota con la letra n. La Ley de Snell se la suele escribir

$$\boxed{n_1 \operatorname{sen} \alpha_1 = n_2 \operatorname{sen} \alpha_2}.$$

Que pasa si la luz atraviesa un medio que va cambiando de manera continua de índice de refracción. Por ejemplo, el índice de refracción en la atmósfera va cambiando de manera continua con la altitud respecto a la superficie terrestre, ya que la densidad del aire va cambiando con la altitud. La Ley de Snell en este caso es

$$\boxed{\frac{\operatorname{sen} \alpha}{v} = \text{cte}}$$

Aquí el ángulo α y la velocidad v cambian respecto a alguna variable/s real/es, por ejemplo la altitud.

¿Que tienen en común el recorrido de la luz y la braquistócrona? Bernoulli se dió cuenta que la situación en los dos casos es la misma, ya que en los dos casos se trata de minimizar el tiempo del recorrido. De modo que la braquistócrona también tiene que satisfacer la Ley de Snell. Ahora supongamos un sistema de coordenadas con origen en el punto A, inicial del recorrido. Además supongamos que el movil parte del reposo. Con estas suposiciones $x(t_0) = 0$ y $v(t_0) = 0$. Por la conservación de la energía

$$\frac{m}{2}|v(t)|^2 = -mgy(t) = mg|y(t)|.$$

Así por la ley de Snell

$$\frac{\operatorname{sen} \alpha}{|v(t)|} = \frac{\operatorname{sen} \alpha}{\sqrt{2g|y|}} = c = \text{ cte.}$$

En (2.13) habíamos expresado el $\cos \alpha$ (en realidad del ángulo opuesto por el vértice, pero es igual) mediante la derivada. Luego

$$\operatorname{sen} \alpha = \sqrt{1 - \cos^2 \alpha} = \frac{1}{\sqrt{1 + y'(x)^2}}.$$

Entonces tenemos

$$\sqrt{2g|y|}\sqrt{1 + y'(x)^2} = c = \text{ cte}$$

Despejando llegamos a la ecuación diferencial

$$\boxed{\sqrt{\frac{y}{c-y}}\, y' = 1}.$$

Es una ecuación con variables separables. La constante c no tiene el mismo valor que en la ecuación anterior.

La solución se obtiene resolviendo la integral

$$x = \int dx = \int \sqrt{\frac{y}{c-y}} dy.$$

lo que no es tan sencillo. Hacemos el cambio de variables

$$\sqrt{\frac{y}{c-y}} = \tan\phi \Longrightarrow y = c \operatorname{sen}^2\phi \Longrightarrow dy = 2c \operatorname{sen}\phi\cos\phi d\phi.$$

Luego

$$x = 2c\int \operatorname{sen}^2\phi d\phi = \frac{c}{2}\left(2\phi - \operatorname{sen} 2\phi\right) + C_1.$$

Como tiene que pasar por $x = 0$ e $y = 0$ debe ser $C_1 = 0$. Tenemos que

$$\begin{cases} y & = c \operatorname{sen}^2\phi \qquad = \frac{c}{2}(1-\cos 2\phi) \\ x & = \frac{c}{2}(2\phi - \operatorname{sen} 2\phi) \end{cases}$$

Conviene llamar $2\phi = \theta$ y $a = c/2$

$$\begin{cases} y & = a(1-\cos\theta) \\ x & = a(\theta - \operatorname{sen}\theta) \end{cases}$$

Que son la ecuaciones paramétricas de una curva conocida con el nombre de cicloide. Podemos usar SymPy para graficar esta curva

```
theta=symbols('theta')
from sympy.plotting import *
plot_parametric(theta-sin(theta),1-cos(theta),(theta,0,10*pi))
```

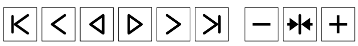


Figura 2.2: Cicloide

Si in tentamos resolver las ecuaciones con SymPy el resultado no es muy alentador.

```
x,c=symbols('x,c')
y=Function('y')(x)
MiEcua=Eq(y.diff(x),sqrt((c-y)/y))
f=dsolve(MiEcua,y,hint='separable')
```

Resultado:

$$\begin{cases} -i\sqrt{c}\sqrt{-1+\frac{1}{c}y(x)}\sqrt{y(x)} - ic\operatorname{acosh}\left(\frac{1}{\sqrt{c}}\sqrt{y(x)}\right) & \text{for } \left|\frac{1}{c}y(x)\right| > 1 \\ \frac{\sqrt{c}\sqrt{y(x)}}{\sqrt{1-\frac{1}{c}y(x)}} + c\operatorname{asin}\left(\frac{1}{\sqrt{c}}\sqrt{y(x)}\right) + \frac{y^{\frac{3}{2}}(x)}{\sqrt{c}\sqrt{1-\frac{1}{c}y(x)}} & \text{otherwise} \end{cases} = C_1 + x$$

Ejemplo 16. [La tautócrona]

Vamos a ver otra propiedad notable de la ciclode. Supongamos que dejamos caer el cuerpo del reposo desde un punto intermedio, digamos en (x_0, y_0). Sea θ_0 el valor del parámetro θ correpondiente a este punto. ¿Cuánto tardara en llegar el cuerpo al punto mínimo de la curva que ocurre cuando $\theta = \pi$?

Tenemos

$$\begin{cases} \frac{dx}{d\theta} & = a(1-\cos\theta) \\ \frac{dy}{d\theta} & = a\operatorname{sen}\theta \end{cases}$$

Como el cuerpo ahora no parte de $(0,0)$ tendremos

$$|v| = \sqrt{2g(y_0 - y)}.$$

La tautócrona Por (2.11) $ds/dt = |v|$. Si llamamos T al tiempo que demanda en llegar a $\theta = \pi$, y llamamos s_0 y s_1 a los arcos correspondientes al punto inicial y final. Tenemos

$$T = \int_0^T dt = \int_{s_0}^{s_1} \frac{dt}{ds} ds = \int_{s_0}^{s_1} \frac{1}{\sqrt{2g(y_0 - y)}} ds.$$

Cambiando la variable de integración a θ. Como

$$\frac{ds}{d\theta} = \sqrt{\left(\frac{dx}{d\theta}\right)^2 + \left(\frac{dy}{d\theta}\right)^2} = \sqrt{2}a\sqrt{1 - cos\theta}.$$

Tenemos

$$T = \sqrt{\frac{a}{g}} \int_{\theta_0}^{\pi} \frac{\sqrt{1 - \cos\theta}}{\cos\theta_0 - \cos\theta} d\theta = \sqrt{\frac{a}{g}} \int_{\theta_0}^{\pi} \frac{\operatorname{sen}\frac{\theta}{2}}{\cos^2\frac{\theta_0}{2} - \cos^2\frac{\theta}{2}} d\theta.$$

Ahora hacemos la sustitución

$$u = \frac{\cos\frac{\theta}{2}}{\cos\frac{\theta_0}{2}} \Longrightarrow du = -\frac{\operatorname{sen}\frac{\theta}{2}}{2\cos\frac{\theta_0}{2}} d\theta.$$

Vemos que

$$T = 2\sqrt{\frac{a}{g}} \int_0^1 \frac{1}{\sqrt{1 - u^2}} du.$$

Que es una expresión independiente de θ_0. En consecuencia el tiempo T que demanda el cuerpo para llegar $\theta = \pi$ es siempre el mismo no importa desde donde se deje caer.

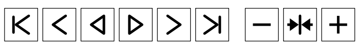


Figura 2.3: Tautócrona