Find Actual Solutions for Nonlinear Equations in Form of $y''(x) + a \cdot [y'(x)]^2 + b = 0$ *and* $[y'(x)]^2 + a \cdot \cos[y(x)] + b = 0$

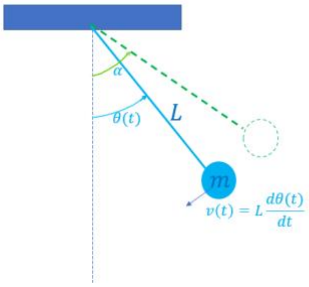
Motivation and Research

Motivation

(1) Skydiving is beautiful, heart-racing, and exhilarating on film. I admire freefall cameramen's skill to control the diving speed under gravitational acceleration and still shoot good movies. It is intriguing to find IB examiners hope students to learn more about the speed change of skydiving by putting it as an examination question. (*ref.1*)

To understand it in depth, I found a good article in College Physics (*ref. 2*) well demystifying how skydivers control their terminal speed by changing their cross sections facing ground. However, the nonlinear differential equation, $\frac{d^2S(t)}{dt^2} = -g + \frac{1}{2m} C\rho A \left(\frac{dS(t)}{dt}\right)^2$ (where S: height g: gravitational acceleration C: drag coefficient A: cross section facing ground ρ air density m: person's mass), is really daunting. I looked up college calculus but failed to find the resolution.

(2) As I tried to build a pendulum of true isochronism for my Physics IA, I derived, based on energy conservation, the differential equation describing the time function of swing angle, $\theta(t)$, for a pendulum. It is $[\frac{d\theta(t)}{dt}]^2 = \frac{2g}{L}[cos\theta(t) - cos\alpha]$ *or* $\frac{d\theta(t)}{dt} = \pm\sqrt{\frac{2g}{L}[cos\theta(t) - cos\alpha]}$. (where θ: swing angle, α: initial swing angle, g: gravitational acceleration, L: pendulum's length)



By energy conservation:

$K + U = constant$:

$\Delta K = -\Delta U$

$\frac{1}{2}mv^2 = -mg\Delta h$

$\frac{1}{2}m(L\frac{d\theta(t)}{dt})^2 = mg(\ L\cos\theta(t) - L\cos\alpha)$

$(\frac{d\theta(t)}{dt})^2 = \frac{2g}{L}\ [\cos\theta(t) - \cos\alpha]$

$\frac{d\theta(t)}{dt} = \pm\sqrt{\frac{2g}{L}\ [\cos\theta(t) - \cos\alpha]}$

It looks like a neat 1st order differential equation, much simpler than the 2nd order one of skydiving dynamics. However, it seems none of resolutions I have learned from Math HL is applicable to resolve this. A paper from *University of Connecticut* (*ref.3*) suggests it be resolved by *"Elliptic Integral of the First Kind"*.

To get rid of the trauma that even ordinary engineering questions can be only handled by mathematicians, I need to figure out a way to depict what the solutions of those differential equations look like.

In most of the rest, I will often use general forms to abstractly express the Skydiving equation and Pendulum equation as below.

Skydiving Equation:

$$\frac{d^2S(t)}{dt^2} = -g + \frac{1}{2m}C\rho A\left[\frac{dS(t)}{dt}\right]^2 \rightarrow (general\ form): y''(x) + a \cdot [y'(x)]^2 + b = 0$$

Pendulum Equation:

$$[\frac{d\theta(t)}{dt}]^2 = \frac{2g}{L}[cos\theta(t) - cos\alpha] \rightarrow (general\ form): [y'(x)]^2 + a \cdot \cos\big(y(x)\big) + b = 0$$

Research

(i) Methods to resolve 1st differential equation

Through IB Diploma Programme math HL, I have learned several skills and procedures to find the solutions for some types of 1st order differential equations that can be written in the forms below. (*ref. 4*)

Type I. $\frac{dy}{dx} = g(x)$ (e.g. $\frac{dy}{dx} = e^{2x}$)
Type II. $\frac{dy}{dx} = \frac{g(x)}{h(y)}$ (called separable) (e.g. $\frac{dy}{dx} = \frac{x}{y^2}$) .
Type III. $\frac{dy}{dx} = g(\frac{y}{x})$ (called homogeneous) (e.g. $\frac{dy}{dx} = \frac{y}{x} - 1$)
Type IV. $\frac{dy}{dx} + P(x)y = Q(x)$ (e.g. $\frac{dy}{dx} + 3x^2y = 6x^2$)

It is easy to get lost in those type-orientated approaches even though the solutions are only for a portion of 1st order differential equations. Besides, meeting one of the 4 types is not sufficient for an equation to be resolved to have a solution in closed form. For instance, the Pendulum equation $\frac{d\theta(t)}{dt} = \pm\sqrt{\frac{2g}{L}[cos\theta(t) - cos\alpha]}$ falls in Type II ($\frac{dy}{dt} = \pm\sqrt{\frac{2g}{L}(cosy - cos\alpha)}$) but we fail to find the integral, $\int \frac{1}{\sqrt{\frac{2g}{L}(cosy - cos\alpha)}}dy$.

(ii) Methods to solve 2nd differential equation

2nd differential equations are not addressed in IB math HL. I turn to *Thomas' Calculus* (*ref. 5*) for resolution of 2nd order differential equations. Only the two following types of 2nd order differential equation are considered possibly solvable to have closed form solutions:

1. Linear nonhomogeneous differential equation

$$a\frac{d^2y(x)}{dx^2} + b\frac{dy(x)}{dx} + cy(x) = G(x)\ where\ a, b, c\ are\ constants$$

2. Euler Equation

$$ax^2\frac{d^2y(x)}{dx^2}+bx\frac{dy(x)}{dx}+cy(x)=0,\qquad where\ a,b,c\ are\ constants$$

Obviously, the Skydiving equation does not fall into these two categories.

(iii) Power Series for linear differential equation

My math teacher advised me to make good use of Maclaurin Series. I notice a similar "Power Series Method" is also demonstrated in *Thomas Calculus* (*ref.5*).

Next, I will apply Power Series Method to solve the Skydiving equation in general form $y^{(2)}(x)+a\left[y^{(1)}(x)\right]^2+b=0\ldots\ldots\ldots..(Eq.1)$

Express y(x) in Maclaurin Series:

$$y(x)=\sum_{n=0}^{\infty}C_nx^n=C_0+C_1x+C_2x^2+C_3x^3+C_4x^4+C_5x^5+\cdots$$

Then

$$y^{(1)}(x)=\sum_{n=0}^{\infty}nC_nx^{n-1}=C_1+2C_2x+3C_3x^2+4C_4x^3+5C_5x^4+\cdots$$

$$y^{(2)}(x)=\sum_{n=0}^{\infty}n(n-1)C_nx^{n-2}$$
$$=(1\cdot2)C_2+(2\cdot3)C_3x+(3\cdot4)C_4x^2+(4\cdot5)C_5x^3+\cdots$$

With $y^{(1)}(x), y^{(2)}(x)$ in Eq.1 replaced with the expressions above,

$$[(1\cdot2)C_2+(2\cdot3)C_3x+(3\cdot4)C_4x^2+(4\cdot5)C_5x^3+\cdots]$$
$$+a[C_1+2C_2x+3C_3x^2+4C_4x^3+5C_5x^4+\cdots]^2+b=0$$

Handle $[C_1+2C_2x+3C_3x^2+4C_4x^3+5C_5x^4+\cdots]^2$ first, as below

	1	x	x^2	x^3	x^4	$x^5\cdots$
	C_1	$2C_2$	$3C_3$	$4C_4$	$5C_5$	$6C_6\cdots$
$\times$	C_1	$2C_2$	$3C_3$	$4C_4$	$5C_5$	$6C_6\cdots$

$(1C_1)(1C_1)$	$(1C_1)(2C_2)$	$(1C_1)(3C_3)$	$(1C_1)(4C_4)$	$(1C_1)(5C_5)$	$(1C_1)(6C_6)$
	$(2C_2)(1C_1)$	$(2C_2)(2C_2)$	$(2C_2)(3C_3)$	$(2C_2)(4C_4)$	$(2C_2)(5C_5)$
		$(3C_3)(1C_1)$	$(3C_3)(2C_2)$	$(3C_3)(3C_3)$	$(3C_3)(4C_4)$
			$(4C_4)(1C_1)$	$(4C_4)(2C_2)$	$(4C_4)(3C_3)$
				$(5C_5)(1C_1)$	$(5C_5)(2C_2)$
					$(6C_6)(1C_1)$

The resultant coefficient for each x^n for the polynomials on the left side shall be 0.

$(1 \cdot 2)C_2 + a(1C_1 \cdot 1C_1) + b = 0$
$(2 \cdot 3)C_3 + a(1C_1 \cdot 2C_2 + 2C_2 \cdot 1C_1) = 0$
$(3 \cdot 4)C_4 + a(1C_1 \cdot 3C_3 + 2C_2 \cdot 2C_2 + 3C_3 \cdot 1C_1) = 0$
$(4 \cdot 5)C_5 + a(1C_1 \cdot 4C_4 + 2C_2 \cdot 3C_3 + 3C_3 \cdot 2C_2 + 4C_4 \cdot 1C_1) = 0$
……
$\rightarrow (n-1) \cdot n \cdot C_n$
$+a(1C_1 \cdot (n-1)C_{n-1} + 2C_2 \cdot (n-2)C_{n-2} + \cdots + (n-2)C_{n-2} \cdot 2C_2 + (n-1)C_{n-1} \cdot 1C_1)$
$= 0, n \geq 3$

From the above, we know C_n can always be learned from $\{C_0, C_1, \ldots, C_{n-1}\}$ by

$$With\ C_0, C_1\ given, \qquad C_2 = \frac{-a(1C_1 \cdot 1C_1) - b}{1 \cdot 2}, \qquad C_n = \frac{-a\sum_{i=1}^{n-1} iC_i(n-1)C_{n-1}}{(n-1)n}, \forall n \geq 3$$

It means we can sequentially, starting from $C_0\ and\ C_1$, work out all the coefficients of Maclaurin Series.

Power Series Method could solve Skydiving equation:

$$Equation: y^{(2)}x + a\left[y^{(1)}x\right]^2 + b = 0$$

$$Solution: y(x) = C_0 + C_1x + \frac{-aC_1{}^2 - b}{2}x^2 + (-a) \cdot \sum_{n=3}^{\infty} \frac{\sum_{i=1}^{n-1} iC_i(n-i)C_{n-i}}{(n-1)n}x^n$$

$$, where\ C_0, C_1\ are\ given\ initial\ condition\ and\ a = -\frac{1}{2m}C\rho A, b = g$$

Below is the python code to easily get the coefficients, up to n=10, with given C_0, C_1

```
c[0] = 5000 #initial value
c[1] = 0 #initial value
c[2] = 1/2*(-a*c[1]**2-b)
k = 10
n = 3
while n <= k:

    index = 1
    sum = 0
    while index < n:
        c[n] += -a*(index*(n-index)*c[index]*c[n-index])/((n-1)*n)
        index += 1
    n+=1
```

Striking a hot iron, let us try out Power Series Method on Pendulum equation.
$[y^{(1)}(x)]^2 + a \cdot \cos[y(x)] + b = 0 \ldots\ldots\ldots\ldots (Eq.2)$

$Express\ y(x)\ in\ Maclaurin\ Series$:

$$y(x) = \sum_{i=0}^{\infty} C_i x^i = C_0 + C_1 x + C_2 x^2 + C_3 x^3 + C_4 x^4 + C_5 x^5 + \cdots$$
$$y^{(1)}(x) = \sum_{n=0}^{\infty} nC_n x^{n-1} = C_1 + 2C_2 x + 3C_3 x^2 + 4C_4 x^3 + 5C_4 x^4 \ldots$$
$$y^{(2)}(x) = \sum_{n=0}^{\infty} n(n-1)C_n x^{n-2} = (1 \cdot 2)C_2 + (2 \cdot 3)C_3 x + (3 \cdot 4)C_4 x^2 + (4 \cdot 5)C_5 x^3 + \cdots$$

With $y^{(0)}(x), y^{(1)}(x)$ in Eq.2 replaced with the expressions above,
$$[C_1 + 2C_2 x + 3C_3 x^2 + 4C_4 x^3 + 5C_4 x^4 \ldots]^2$$
$$+a \cdot \cos[C_0 + C_1 x + C_2 x^2 + C_3 x^3 + C_4 x^4 + C_5 x^5 + \cdots] + b = 0$$

Recall that Power Series Method works on the base that all the coefficients of the resultant polynomials should be made zero. This method might fail this equation since $\cos(C_0 + C_1 x + C_2 x^2 + C_3 x^3 + C_4 x^4 + C_5 x^5 + \cdots)$ cannot be easily broken down into neat polynomials.

(iv) "Proprietary(?) Deep Differentiation" Method for differential equations of infinitely differentiable functions

When Power Series Method is applied to get coefficients of Maclaurin Series,

$$y(x) = y(0) + \frac{y^{(1)}(0)}{1!}x + \frac{y^{(2)}(0)}{2!}x^2 + \frac{y^{(3)}(0)}{3!}x^3 + \cdots$$

it is to substitute the polynomial forms of $y(x), y^{(1)}(x), y^{(2)}(x)$, ...into a differential equation and then to make zero the coefficients of each resultant x^n term to find $C_n (= \frac{y^{(n)}(0)}{n!})$. ***Then it came to my mind: Why not directly find $\boldsymbol{y^{(n)}(0)}$?***

So next, I will work out $\boldsymbol{y^{(n)}(0)}$ for the tenacious Pendulum equation:
$$[y^{(1)}(x)]^2 + a \cdot \cos[y(x)] + b = 0 \ldots\ldots\ldots (Eq.2)$$

Rearrange the equation to have
$$[y^{(1)}(x)]^2 = -a \cdot \cos\big(y(x)\big) - b$$
By differentiating both sides
$$2y^{(1)}(x)y^{(2)}(x) = a \cdot \sin\big(y(x)\big)\, y^{(1)}(x)$$
$$\rightarrow y^{(2)}(x) = \frac{a}{2} \cdot \sin\,(y(x))$$
$$\rightarrow y^{(3)}(x) = \frac{a}{2} \cdot \cos\,(y(x))y^{(1)}(x)$$
$$\rightarrow y^{(4)}(x) = -\frac{a}{2} \cdot \sin\,(y(x))[y^{(1)}(x)]^2 + \frac{a}{2} \cdot \cos\,(y(x))y^{(2)}(x)$$
$$\rightarrow y^{(5)}(x) = \cdots$$

We can keep differentiating it to sequentially get all $y^{(n)}(x), \forall n \in N$

With $y(0)$ given and $x = 0$ substituted into the formulas above, we will get $y^{(n)}(0)$ as shown below

$$[y^{(1)}(0)]^2 = -a \cdot \cos\big(y(0)\big) - b$$

$$\rightarrow y^{(1)}(0) = \pm\sqrt{-a \cdot \cos\big(y(0)\big) - b}$$
$$\rightarrow y^{(2)}(0) = \frac{a}{2} \cdot \sin\big(y(0)\big)$$
$$\rightarrow y^{(3)}(0) = \frac{a}{2} \cdot \cos\big(y(0)\big)\, y^{(1)}(0)$$
$$\rightarrow y^{(4)}(0) = -\frac{a}{2} \cdot \sin\big(y(0)\big)[y^{(1)}(0)]^2 + \frac{a}{2} \cdot \cos\big(y(0)\big)\, y^{(2)}(0)$$
$$\rightarrow y^{(5)}(0) = \cdots$$

Therefore, we can find out all the $y^{(n)}(0)$ sequentially to make the actual solution for Pendulum Equation in Maclaurin Series.

This method is applicable to all differential equations of infinitely differentiable functions. Therefore, we can use it to resolve the Skydiving equations as well.

Exploration: Point-by-Point Method

Even though I have solved Skydiving and Pendulum equations in Maclaurin Series, it does not really ease my concern. Excel could be used for calculation of coefficients, but it is still tedious. Besides, Maclaurin Series has infinite terms --- any truncation causes inaccuracy which I do not have a way to estimate. This is probably why a solution in infinite series is not considered as in closed form though it is made of ordinary polynomials. I do need to figure out alternatives.

I scrutinize the fundamental of differentiation:

$$y^{(1)}(x_0) = \frac{dy(x)}{dx}|_{x=x_0} = \lim_{h\to 0}\frac{y(x_0+h)-y(x_0)}{h}$$

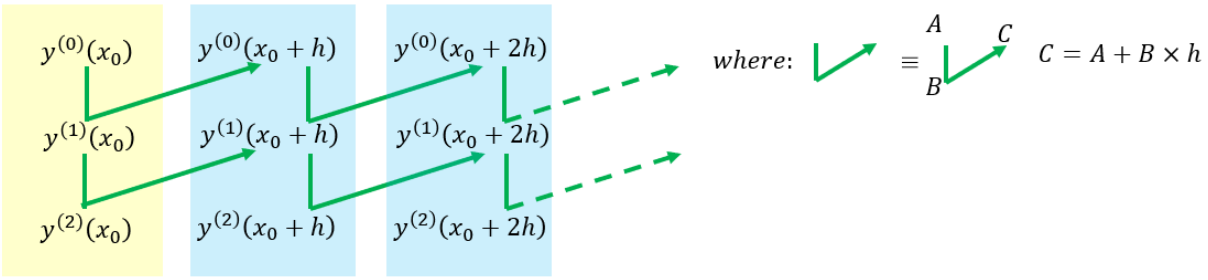
After rearrangement, it tells : $y^{(0)}(x_0+h) = y^{(0)}(x_0) + y^{(1)}(x_0)\cdot h \quad ,where\ h \to 0$

Similarly,

$$y^{(2)}(x_0) = \frac{dy^{(1)}(x)}{dx}\Big|_{x=x_0} = \lim_{h\to 0}\frac{y^{(1)}(x_0+h)-y^{(1)}(x_0)}{h}$$

equivalently says $y^{(1)}(x_0+h) = y^{(1)}(x_0) + y^{(2)}(x_0)\cdot h$

I draw a picture to have a concrete idea of the mathematical expressions above



For Skydiving equation

$$y^{(2)}(x) + a[y^{(1)}(x)]^2 + b = 0, \forall x$$

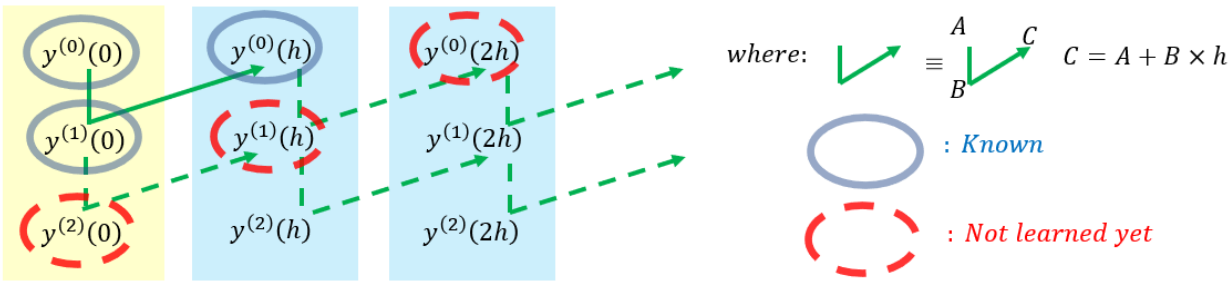
,with $y^{(0)}(0), y^{(1)}(0)$ given, we can know $y^{(0)}(h)$ by $y^{(0)}(h) = y^{(0)}(0) + y^{(1)}(0)\cdot h$.

If we would like to know $y^{(0)}(2h)$, we need to know $y^{(2)}(0)$; then,

$y^{(1)}(h) = y^{(1)}(0) + y^{(2)}(0)\cdot h$

$y^{(0)}(2h) = y^{(0)}(h) + y^{(1)}(h)\cdot h$

as instructed by the following figure.



Because equation

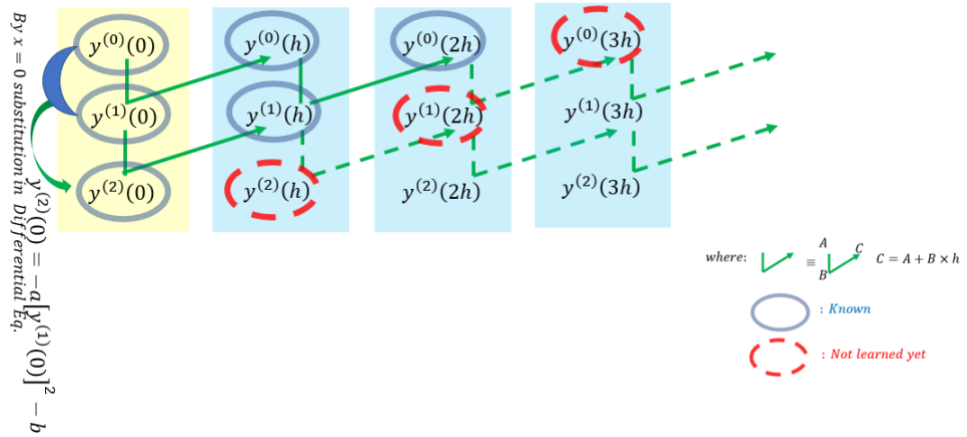
$$y^{(2)}(x) + a[y^{(1)}(x)]^2 + b = 0, \forall x$$

sustains for all x, we can use substitution $x = 0$ to get:

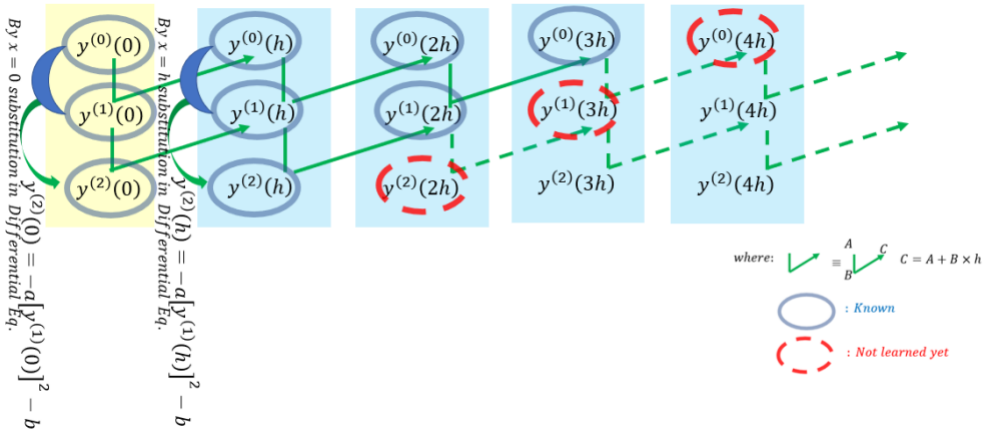
$y^{(2)}(0) + a[y^{(1)}(0)]^2 + b = 0$

$y^{(2)}(0) = -a[y^{(1)}(0)]^2 - b$

$y^{(2)}(0)$ is thus acquired from $y^{(1)}(0)$ and $y^{(0)}(0)$. Then we follow to get $y^{(1)}(h)$ and $y^{(0)}(2h)$ accordingly.



The similar way could be iteratively applied until we get the target y(nh).



It looks feasible to get all the way down to learn $y^{(0)}(nh), \forall n \in N$. Because $h \to 0$, it implies we can get all $y(x), \forall x \geq 0$ as long as $y^{(0)}(0)$ and $y^{(1)}(0)$ given.

Retracing this method, with $y^{(0)}(kh), y^{(1)}(kh)$ known in priori, the operation to get $y^{(0)}[(k+1)h], y^{(1)}[(k+1)h]$ consists of two easy steps below

Step #1: $y^{(2)}(kh) = -a[y^{(1)}(kh)]^2 - b$

(substitute x=kh for $[y^{(1)}(x)]^2 = -a \cdot \cos(y(x)) - b$)

Step #2 $y^{(0)}[(k+1)h] = y^{(0)}(kh) + y^{(1)}(kh) \times h$

$y^{(1)}[(k+1)h] = y^{(1)}(kh) + y^{(2)}(kh) \times h$

Step#2 is from the definition of differentiation and is independent of equation types. This algorithm is apparently applicable to all nonlinear different equations which could make the form of $y^{(n)}(x) = H(\, y^{(n-1)}(x),\ y^{(n-2)}(x),\ldots.\ y^{(1)}(x),\ y^{(0)}(x))$ to facilitate Step#1 .

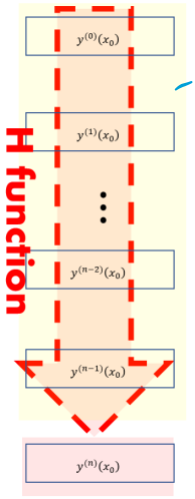
Recapped below is the procedure of Point-by-Point Method.

Point-by-Point Method to Resolve

$$f^{(n)}(x) = H(f^{(n-1)}(x), f^{(n-2)}(x), \ldots, f^{1}(x), f(x), g(x))$$

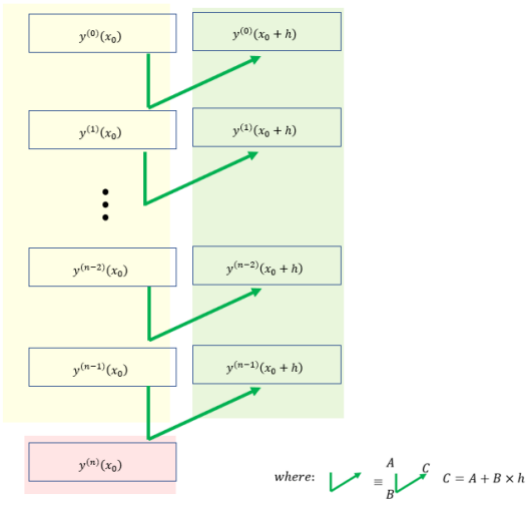
Known in priori: $y^{(0)}(x_0), y^{(1)}(x_0), y^{(2)}(x_0), \cdots, y^{(n-1)}(x_0)$

Step#1: Get $y^{(n)}(x_0)$ by $y^{(n)}(x_0) = H(y^{(0)}(x_0), y^{(1)}(x_0), y^{(2)}(x_0), \cdots, y^{(n-1)}(x_0))$



Step#2 Get

$$y^{(0)}(x_0 + h) = y^{(0)}(x_0) + y^{(1)}(x_0) \times h$$
$$y^{(1)}(x_0 + h) = y^{(1)}(x_0) + y^{(2)}(x_0) \times h$$
$$y^{(2)}(x_0 + h) = y^{(2)}(x_0) + y^{(3)}(x_0) \times h$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$y^{(n-2)}(x_0 + h) = y^{(n-2)}(x_0) + y^{(n-1)}(x_0) \times h$$
$$y^{(n-1)}(x_0 + h) = y^{(n-1)}(x_0) + y^{(n)}(x_0) \times h$$



Step#3: Until we are satisfied with the range swept by x_0 , assign x_0 with new value ($x_0 + h$). Go to Step#2

Worked Examples (A): Skydiving Equation

With parameters: $g = 10ms^{-2}, C = 0.7, \rho = 1.21Kgm^{-3}, A = 0.18m^{-2}$, $intial\ height\ S(0) = 5000\ m\ and\ initial\ speed = 0\ ms^{-1}$ for

$$\frac{d^2S(t)}{dt^2} = -g + \frac{1}{2m}C\rho A\left[\frac{dS(t)}{dt}\right]^2$$

We get $a = -\frac{1}{2m}C\rho A = -0.0010164,\ b = 10, y(0) = 5000, y'(0) = 0$ for

$$y''(x) + a \cdot [y'(x)]^2 + b = 0$$

(A-1) Point-by-Point Method

Python code to implement Point-by-Point Method goes as below

```
y0d[0] = 5000
y1d[0] = 0
for n in range(len(t)-1):
    y0d[n+1] = y0d[n] + y1d[n] * dt
    y1d[n+1] = y1d[n] + y2d[n] * dt
    y2d[n+1] = -a*y1d[n+1]*y1d[n+1]-b
```

(A-2) Power Series Method

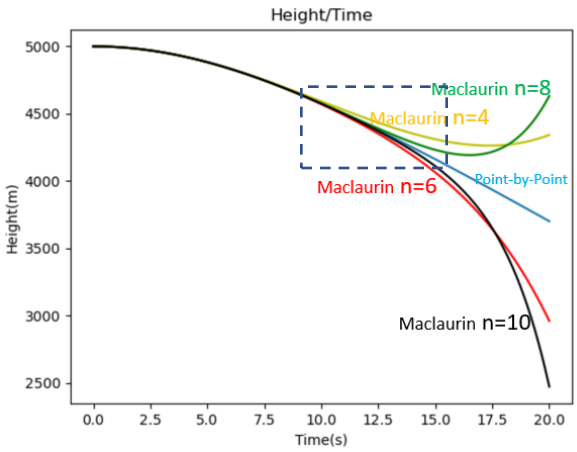
Python code to get Power series Method executed below
(A-2-1) To get the C_n of Power Series calculated first (up to order =10)

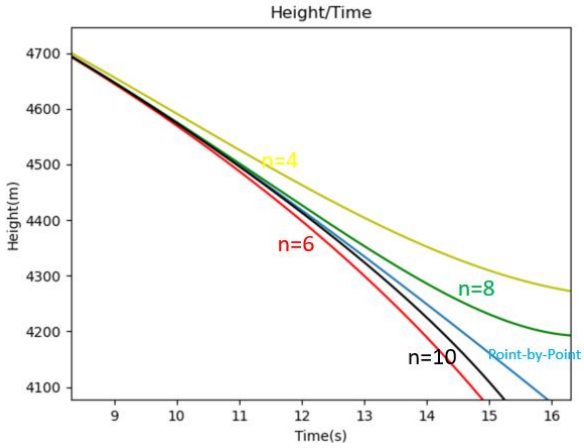
```
c[0] = 5000
c[1] = 0
c[2] = 1/2*(-a*c[1]**2-b)
order = 10
n = 3
while n <= order:
    index = 1
    sum = 0
    while index < n:
        c[n] += -a*(index*(n-index)*c[index]*c[n-index])/((n-1)*n)
        index += 1
    n+=1
```

(A-2-2) Then carry out Maclaurin series with those coefficients

```
y=0
for index in range(11):
    y += c[index]*x**index
```

Result comparison with the region in dashed line zoomed in:





Notice that

(i) As x increases, the waveforms of Maclurin series diverge.

(ii) As order n increases, the traces get closer to the Point-by-Point one. It indicates the Point-by-Point Method is the one of the least error.

Worked Examples (B): Pendulum Equation

With $L = 1m, g = 10ms^{-2} initial\ angle = \pi/3$, for

$$[\frac{d\theta(t)}{dt}]^2 = \frac{2g}{L}[cos\theta(t) - cos\alpha]$$

We have $a = -\frac{2g}{L} = -20, b = \frac{2g}{L} \cdot cos\alpha = 10\ \ and\ y(0) = \pi/3$ for

$$[y'(x)]^2 + a \cdot \cos\big(y(x)\big) + b = 0$$

(B-1) Point-by-Point Method

With

$$y'(x) = \pm\sqrt{-a \cdot \cos\big(y(x)\big) - b}$$

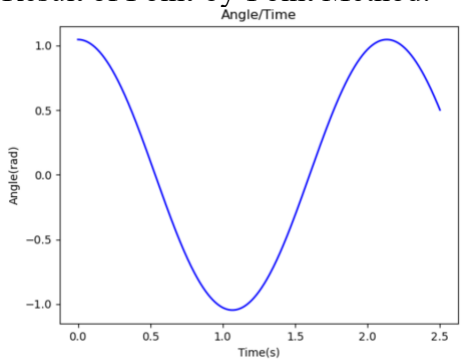
, the python code to implement Point-by-Point Method goes as below

(Due to the intrinsic ambiguity on +/-, it calls for some judgement loop)

```
initialAngle = np.pi/3
a=-2*g/L
b=2*g/L*np.cos(initialAngle)
y0d[0] = initialAngle - 1e-10
y1d[0] = 0
clockwise = 1
for n in range(len(x)-1):
```

```
y0d[n+1] = y0d[n] + y1d[n] * dt
 # If it were not to handle uncertainty of +/-, it would just be neatly
 # y1d[n+1] = -np.sqrt(np.abs(-a * np.cos(y0d[n + 1]) - b))
 # without the following codes
if clockwise == 1:
    if y0d[n+1] > -initialAngle:
        y1d[n+1] = -np.sqrt(np.abs(-a * np.cos(y0d[n + 1]) - b))
    else:
        clockwise = -1
        y1d[n+1] = np.sqrt(np.abs(-a * np.cos(y0d[n + 1]) - b))
else:
    if y0d[n+1] < initialAngle:
        y1d[n+1] = np.sqrt(np.abs(-a * np.cos(y0d[n + 1]) - b))
    else:
        clockwise = 1
        y1d[n+1] = -np.sqrt(np.abs(-a * np.cos(y0d[n + 1]) - b))
```

Result of Point-by-Point Method:



The period 2.132300 seconds matches what is got from the paper from University of Connecticut (*ref.3*)

(B-2) Deep Differentiation Method

I substitute $y(0) = \frac{\pi}{3}$ in those $y^{(n)}(0)$ formula for Pendulum equation to get their numerical values

$$y(0) = \frac{\pi}{3}$$

$$y^{(1)}(0) = \pm\sqrt{-a \cdot \cos\big(y(0)\big) - b} = \pm\sqrt{-(-20) \cdot \cos\left(\frac{\pi}{3}\right) - 10} = 0$$

$$y^{(2)}(0) = \frac{a}{2} \cdot \sin\big(y(0)\big) = \frac{-20}{2} \cdot \sin\left(\frac{\pi}{3}\right) = -5\sqrt{3}$$

$$y^{(3)}(0) = \frac{a}{2} \cdot \cos\big(y(0)\big)\, y^{(1)}(0) = \frac{-20}{2} \cdot \cos\left(\frac{\pi}{3}\right) y^{(1)}(0) = 0$$

$$y^{(4)}(0) = -\frac{a}{2}\cdot\sin\big(y(0)\big)[y^{(1)}(0)]^2 + \frac{a}{2}\cdot\cos\big(y(0)\big)\,y^{(2)}(0)y^{(4)}(0)$$
$$= -\frac{-20}{2}\cdot\sin\left(\frac{\pi}{3}\right)[y^{(1)}(0)]^2 + \frac{-20}{2}\cdot\cos\left(\frac{\pi}{3}\right)y^{(2)}(0) = 25\sqrt{3}$$

Then here come the Maclaurin Series:

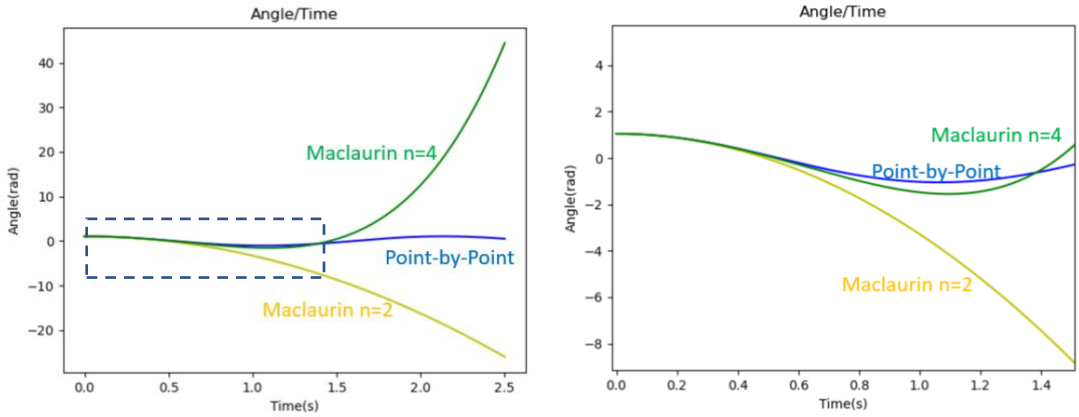
2-order

$$y(x) = y(0) + \frac{y^{(1)}(0)}{1!}x + \frac{y^{(2)}(0)}{2!}x^2 = \frac{\pi}{3} + \frac{-5\sqrt{3}}{2}x^2$$

4-order

$$y(x) = y(0) + \frac{y^{(1)}(0)}{1!}x + \frac{y^{(2)}(0)}{2!}x^2 + \frac{y^{(3)}(0)}{3!}x^3 + \frac{y^{(4)}(0)}{4!}x^4 = \frac{\pi}{3} + \frac{-5\sqrt{3}}{2}x^2 + \frac{25\sqrt{3}}{24}x^4$$

Result comparison with region in dashed line zoomed:



Notice that

(i) As x increases, the waveforms of Maclurin series diverge badly. The solutions in Maclaurin series, up to order 4, don't even show the periodicity of a pendulum.

(ii) As order n increases, the traces get closer to the Point-by-Point one. It indicates the coefficients got from Deep Differentiation should truly work for Maclaurin Series.

Summary:

The actual solution in closed form is difficult to get even for ordinary differential equations in engineering. Pursuing the solution in Maclaurin series seems to provide an alternative, in theory. However, infinite series is impossible to carried out in reality while the error due to truncated terms diverge significantly.

I have developed an iterative way, Point-by-Point Method, to numerically work out the solution for any n-order differential equation in form of

$$y^{(n)}(x) = H(\, y^{(n-1)}(x),\ y^{(n-2)}(x),\ldots.\ y^{(1)}(x),\ y^{(0)}(x)).$$

Without either categorizing equation types and orders or calculating special coefficients, it is simple and straightforward. It turns out accurate without truncation errors.

Working through the research with valuable advices from my teacher and successfully developing an efficient algorithm in this internal assessment, I have expanded my scope a lot and become intrepid, equipped with popular calculation machines nowadays, in handling all kinds of differential equations in the future.

Reference

1. IB Diploma Programme Physics Higher level Paper 1 Friday 6 May 2016 (morning) M16/4/PHYSI/HPM/ENG/TZ0/XX, Question 2
http://ibpastpapers.com/ib-past-papers-12/

2.SUNY(The State University of New York) OER Services University Physics Volume 1- Chapter 6 Applications of Newton's Laws-Section 6.4 Drag Force and Terminal Speed
https://courses.lumenlearning.com/suny-osuniversityphysics/chapter/6-4-drag-force-and-terminal-speed/

3. 3."LARGE-ANGLE MOTION OF A SIMPLE PENDULUM", class notes, Physics 258 - Fall 2004, Univeristy of Connecticut
https://www.phys.uconn.edu/~hamilton/phys258/N/pend.pdf

4.Chapter 25 Differential Equations, "Mathematics Analysis and Approaches HL 2 for use with IB Diploma Programme by Michael Haese, Mark Humphries, Chris Sangwin, Ngoc Vo", from Haese Mathematics

5.Chapter 17 SECOND-ORDER DIFFERENTIAL EQUATIONS, "Thomas' Calculus by George B.Thomas,Jr., Twelfth Edition", from Addison-Wesley