Proyecto PNLCC
To the Moon

Esteba Agenjo, Raúl
Ramírez Puertas, Pablo

1 de abril de 2022

«Non est ad astra mollis e terris via»
-
«No hay forma fácil de llegar a las estrellas desde la tierra»

Séneca

Nomenclatura

G	Constante de gravitación universal.
m_E	Masa de la Tierra.
m_M	Masa de la Luna.
m	Masa del Cohete.
ρ_E	Radio de la Tierra.
ρ_M	Radio de la Tierra.
D	Distancia del centro de la Luna al centro de la Tierra.
$s(t)$	Posición del cohete en función del tiempo.
$v(t)$	Velocidad del cohete en función del tiempo.
$a(t)$	Aceleración del cohete en función del tiempo.

1. INTRODUCCIÓN

Sea un cohete, el cual despega de un punto predefinido de la Tierra y aterriza en otro de la Luna. A través de nuestro modelo queremos encontrar cuál sería la trayectoria óptima que debe recorrer dicho cohete, de tal forma que el uso de combustible sea mínimo.

Vamos a partir de un modelo básico donde en vez de un cohete, será una partícula cualquiera. Partimos de un espacio vacío: sin satélites, sin meteoritos, sin basura espacial, tan sólo nuestra partícula en el espacio. Como veremos más adelante, la masa de la partícula resultará despreciable, y por ello suponemos que el uso del combustible tiene relación directa con la aceleración total aplicada.

Y por último, asumimos que no existe ni rotación ni traslación, es decir, la Tierra y la Luna están en reposo.

Como resultados del modelo obtendremos tanto la trayectoria que debe recorrer la partícula, como la aceleración total que debe realizar.

2. MODELO

2.1. Sistema de referencia

Al suponer que la Tierra y la Luna se encuentran en estado de reposo absoluto, podemos realizar cualquier isometría que nos convenga para hacer más cómodo el sistema; en este caso hemos colocar a la Tierra en el origen de coordenadas, así como la Luna desplazada sobre el eje OX.

2.2. Mecánica Clásica

Para desarrollar el modelo que nos proponemos, necesitamos obtener restricciones para conocer el movimiento de la partícula. Nos basaremos en tres resultados fundamentales de la mecánica clásica: la *segunda ley de Newton*, la *ley de gravitación universal* y las *ecuaciones de movmiento*.

De la *segunda ley de Newton* se deduce que la fuerza experimentada por una partícula se puede descomponer como la suma total de las fuerzas que experimenta. $F = \sum_{i=0}^{N} F_i$. También se desprende que la Fuerza experimentada depende de la aceleración.

La *ley de gravitación universal* nos permite conocer la fuerza que experimentará la partícula, debido a cualquiera de los dos astros, en cada punto del espacio.

$$F = -G\frac{Mm}{d}\vec{r}$$

Por último las *ecuaciones de movimiento* nos darán un sistema de ecuaciones diferenciales del que partir para construir las restricciones de movimiento.

$$\ddot{s}(t) = \dot{v}(t) = a(t)$$

2.3. Ecuaciones Diferenciales

Poniendo en común todos los elementos del apartado anterior tenemos que la fuerza experimentada por la partícula viene dada por

$$F(t) = F_{ER}(t) + F_{MR}(t) = ma(t) \tag{1}$$

$$a(t) = \frac{F(t)}{m} = -G\left(\frac{m_M}{\|s(t) - (D,0)\|^{3/2}}(s(t) - (D,0)) + \frac{m_E}{\|s(t)\|^{3/2}}s(t)\right) \tag{2}$$

En este punto podemos observar como la masa del cohete desaparece de las ecuaciones, por ello, la masa se vuelve irrelevante.

Por otro lado, podemos tomar la ecuación de segundo orden proporcionada por la *ecuación de movimiento* y obtener un sistema de primer orden.

$$(3)\begin{cases} \dot{v}(t) &= -G\left(\frac{m_M}{\|s(t)-(D,0)\|^{3/2}}(s(t) - (D,0)) + \frac{m_E}{\|s(t)\|^{3/2}}s(t)\right) + \Lambda(t) \\ \dot{s}(t) &= v(t) \end{cases}$$

, donde $\Lambda(t)$ representa la aceleración *artificial* experimentada por la partícula, es decir, la aceleración que experimentaría el cohete a cuenta de los propulsores.

Este sistema es vectorial, luego al descomponerlo por componentes obtendremos las restricciones que definirán el movimiento de la partícula.

2.4. Valores iniciales y finales

Para dar los datos iniciales y finales del problema (posición y velocidad) usaremos por comodidad un sistema polar de referencia. De este modo dado dos ángulos θ_E y θ_M tendremos suficiente para determinar la posición inicial y final respectivamente. Por otro lado, añadiendo los módulos de las velocidades iniciales y finales, v_0, v_N, podremos determinar estas velocidades. Por comodidad supondremos que las velocidades son normales a la superficie de los astros en los puntos de llegada, y que la velocidad final apunta en dirección al centro de la Luna. De todo lo anterior tenemos:

- $s(0) = (\rho_E \cos(\theta_E), \rho_E \sin(\theta_E))$ (Posición inicial)
- $s(n) = (a + \rho_M \cos(\theta_M), \rho_M \sin(\theta_M))$ (Posición inicial)
- $v(0) = (v_0 \cos(\theta_E), v_0 \sin(\theta_E))$ (Velocidad inicial)
- $v(n) = (v_N \cos(\theta_M - \pi), v_N \sin(\theta_M - \pi))$ (Velocidad final)

2.5. Restricciones de colisión

Parece razonable pedir que en el movimiento de la partícula no atraviese ni la Tierra ni la Luna. Esto se podría conseguir con las siguientes restricciones:

$$\begin{cases} s_x(t)^2 + s_y^2 & \geq \rho_E \\ (s_x(t) - D)^2 + s_y^2 & \geq \rho_M \end{cases}$$

2.6. Función objetivo

Tras asumir la relación directa entre la aceleración y el consumo de combustible, podemos establecer nuestra función objetivo como minimizar la *cantidad* de aceleración artificial usada. Una vez discretizamos el sistema, queda como

$$\min \sum_{i=1}^{n} \left(\Lambda_x(t_i)^2 + \Lambda_y(t_i)^2\right)$$

3. PROGRAMACIÓN

Para la programación del problema debemos discretizar el sistema de restricciones y la función objetivo (expresión mostrada en la sección anterior.)

El código se compone de cinco partes. En la primera parte, la entrada 1, encontramos la declaración de las variables. En la segunda, la entrada 2, la construcción de elementos que serán utilizados más adelante para definir las restricciones. Las restricciones de colisión y aceleración se encuentran en la entrada 3 y las restricciones propias del método aplicado para discretizar el sistema de ecuaciones diferenciales se localizan en la entrada 4 y la entrada 5. Por último, encontramos la función objetivo y la llamada a la solución en la entrada 6

Listing 1: Declaración de variables

```
# Definimos las variables:
#SiVar := posicion con respecto a
#la componente i.
#ViVar := velocidad con respecto a
#la componente i.
#Lambdai := aceleracion con respecto a
#la componente i.
## Nota: Hemos definido Lambda en el informe de esta forma, para mejorar
init = 1; end = NN + 1;
sxVar = Table[sx[i], {i, init, end}];
syVar = Table[sy[i], {i, init, end}];
vxVar = Table[vx[i], {i, init, end}];
vyVar = Table[vy[i], {i, init, end}];
LambdaxVar =
 Table[Lambdax[i], {i, init,
   end}];
LambdayVar =
 Table[Lambday[i], {i, init,
   end}];
Vars = Join[sxVar, syVar, vxVar,
  vyVar, LambdaxVar, LambdayVar];
```

Listing 2: Derivados

```
En este apartado vamos a definir varias funciones para economizar
el lenguaje.

h = T/NN;

(*-------------------------------------------------------------------*)
    (*Componente direccional cohete en OX*)
rRx = Table[
   sxVar[[k]]/Sqrt[sxVar[[k]]^2 + syVar[[k]]^2], {k, 1, NN + 1}];

(*-------------------------------------------------------------------*)
    (*Componente direccional cohete en OY*)

rRy = Table[
   syVar[[k]]/Sqrt[sxVar[[k]]^2 + syVar[[k]]^2], {k, 1, NN + 1}];

(*-------------------------------------------------------------------*)
    (*Distancia Cohete-Tierra*)

dER = Table[Sqrt[sxVar[[k]]^2 + syVar[[k]]^2], {k, 1, NN + 1}];

(*-------------------------------------------------------------------*)
    (*Distancia Cohete-Luna*)

dMR = Table[Sqrt[(sxVar[[k]] - a)^2 + syVar[[k]]^2], {k, 1, NN + 1}];

(*-------------------------------------------------------------------*)
    (*Componentes direccionales vector Cohete-Luna*)

rMRx = Table[(sxVar[[k]] - rM[[1]])/
    Norm[{sxVar[[k]], syVar[[k]]} - rM, 2], {k, 1, NN + 1}];

rMRy = Table[(syVar[[k]] - rM[[2]])/
    Norm[{sxVar[[k]], syVar[[k]]} - rM, 2], {k, 1, NN + 1}];

(*-------------------------------------------------------------------*)
```

```
    (*Componentes direccionales vector Cohete-Tierra*)
rERx = Table[(sxVar[[k]] - rE[[1]])/
    Norm[{sxVar[[k]], syVar[[k]]} - rE, 2], {k, 1, NN + 1}];

rERy = Table[(sxVar[[k]] - rE[[2]])/
    Norm[{sxVar[[k]], syVar[[k]]} - rE, 2], {k, 1, NN + 1}];

Gx = Table[-((N[mM G]/dMR[[k]]^2)*rMRx[[k]]+ (N[mE G]/dER[[k]]^2)*
      rERx[[k]]), {k, 1, NN + 1}];
Gy =
 Table[-((N[mM G]/dMR[[k]]^2)*rMRy[[k]] + (N[mE G]/dER[[k]]^2)*
      rERy[[k]]), {k, 1, NN + 1}];

Fx[k_] := Gx[[k]] +  LambdaxVar[[k]];
Fy[k_] := Gy[[k]] + LambdayVar[[k]];
```

Listing 3: Restricciones

```
# Agregamos las restricciones:
        (*-------------------------------------------------------*)
        (*Posiciones de salida del cohete*)
        sInit = {sxVar[[1]] ==    roE*Cos[tethaE],
   syVar[[1]] ==    roE*Sin[tethaE]};

        (*-------------------------------------------------------*)
        (*Posiciones de llegada del cohete*)
        sFinal = {sxVar[[NN + 1]] ==  a + roM*Cos[tethaM],
   syVar[[NN + 1]] == roM*Sin[tethaM]};

        (*-------------------------------------------------------*)
        (*Velocidad de salida*)
        v0 = 50; (*m/s*)
        vInit = {vxVar[[1]] == v0*Cos[tethaE],
  vyVar[[1]] ==    v0*Sin[tethaE]};

        (*-------------------------------------------------------*)
        (*Velocidad de llegada*)
```

```
        vN = 0; (*m/s*)
        vFinal = {vxVar[[NN + 1]] == vN*Cos[tethaM],
vyVar[[NN + 1]] == vN*Sin[tethaM]};

        (*Evitar la colision con la Tierra*)
        R1 = Table[sxVar[[k]]^2 + syVar[[k]]^2
              >= roE^2 , {k, 1, NN + 1}];

        (*-------------------------------------------------------*)
        (* Evitar la conlision con la Luna*)
        R2 = Table[(sxVar[[k]] - a)^2 + syVar[[k]]^2
              >= roM^2, {k, 1,
  NN + 1}];
        R71 = Table[LambdaxVar[[k]]
              >= -10, {k, 1, NN + 1}];
        R72 = Table[LambdaxVar[[k]]
              <= 10, {k, 1, NN + 1}];
        R81 = Table[LambdayVar[[k]]
              >= -10, {k, 1, NN + 1}];
        R82 = Table[LambdayVar[[k]]
              <= 10, {k, 1, NN + 1}];

        (*-------------------------------------------------------*)
```

Listing 4: Método de Euler

```
#Para el caso en el cual usemos el Metodo de Euler:

  R3 = Table[vxVar[[k + 1]] == Fx[k]*h
                        + vxVar[[k]], {k, 1, NN}];
        R4 = Table[vyVar[[k + 1]] == Fy[k]*h
                             + vyVar[[k]], {k, 1, NN}];
        R5 = Table[sxVar[[k + 1]] == vxVar[[k]]*h
                             + sxVar[[k]], {k, 1, NN}];
        R6 = Table[syVar[[k + 1]] == vyVar[[k]]*h
                             + syVar[[k]], {k, 1, NN}];
```

Listing 5: Método del trapecio

```
#Si usamos el Metodo del trapecio
    R3 = Table[vxVar[[k + 1]] == vxVar[[k]] + .5 h (Fx[k] + Fx[k + 1])
                              ,{k, 1,NN}];
    R4 = Table[vyVar[[k + 1]] == .5(Fy[k] + Fy[k + 1])*h + vyVar[[k]]
                              ,{k, 1,NN}];
    R5 = Table[sxVar[[k + 1]] == (vxVar[[k]] + vxVar[[k + 1]])*h/2 +
         sxVar[[k]],{k, 1, NN}];
    R6 = Table[syVar[[k + 1]] == (vyVar[[k]] + vyVar[[k + 1]])*h*.5 +
         syVar[[k]],{k, 1, NN}];]
```

Listing 6: Función Objetivo

```
#Por ultimo, definimos la funcion objetivo:
          Objetive =
  Sum[Lambdax[tn]^2 + Lambday[tn]^2, {tn, 1,
    NN + 1}];

          time = Timing[Sol = FindMinimum[{Objetive, Constraints},
                                          Vars]];
```

4. Resultados

Durante los primeros intentos para obtener resultados del problema nos encontramos con varios problemas, entre ellos:

- El número de puntos de discretización del sistema es reducido. El tiempo de cómputo resultaba *muy elevado* (por encima de los 300 segundos) para valores por debajo de 30 puntos.
- Al conseguir convergencia, *Mathematica* devolvía errores de *Overflow*.
- Las soluciones obtenidas cuando las restricciones de colisión están activas *pierden el sentido*

Para intentar resolver estos problemas tomamos dos medidas:

- **Reescalar el problema**. El problema originalmente estaba siendo tratado con los datos astronómicos reales, es decir, trabajábamos con valores del orden de 10^6 hasta 10^{22}. Este aspecto podía explicar la aparición del error de *overflow*, por tanto intentamos *reescalar* el problema.

A falta de poder encontrar otros valores de un orden menor que pudiera darnos una intuición del comportamiento que debería tener a nivel físico, tomamos *a ojo* otros valores mucho más pequeños que de algún modo, representen las proporciones.

Gracias a esta resolución conseguimos evitar el problema de *overflow* y aumentar el número de puntos de discretización bajo un tiempo de cómputo *más razonable*. No obstante el problema sigue presentando una gran dificultad resolutiva para *Mathematica*

- **Prescindir de la colisión**. Aún bajo las condiciones del reescalado, las restricciones de colisión siguen presentando un problema a la hora de obtener soluciones *razonables*, por tanto decidimos eliminar estas restricciones del cómputo.

4.1. Casos particulares

En este apartado mostraremos algunos resultados para casos particulares. En la figura 1 encontramos la relación entre los valores iniciales, finales y el tiempo de ejecución; en términos del método del trapecio.

Casos	θ_E	θ_M	v_0	v_N	t_{20}	t_{40}
1	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	50	0	20.512	411.572
2	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	20	0	17.260	478.731
3	$\frac{\pi}{2}$	$\frac{3\pi}{2}$	20	0	76.384	505.640
4	$\frac{\pi}{2}$	$\frac{\pi}{2}$	20	0	80.569	460.298
5	$\frac{\pi}{6}$	$\pi + \frac{\pi}{6}$	20	0	8.745	351.244

Figura 1: Tabla de resultados

Como se puede apreciar en los distintos gráficos, $N = 20$ representa una partición insuficiente para obtener resultados que pudiésemos considerar como razonables. Con $N = 40$ las soluciones se vuelve más comprensibles y se presenta una mejora general en el uso de aceleración.

Para el caso 1, se han añadido las gráficas relativas a las particiones de $N = 70$ y $N = 100$ puntos. En el primero de los casos se puede observar una *irregularidad* en la trayectoria, que vuelve a recuperarse en la siguiente partición. Se puede observar como la trayectoria seguida por la partícula no presenta cambios significativos conforme al aumento de la partición ($N > 40$).

Caso 1

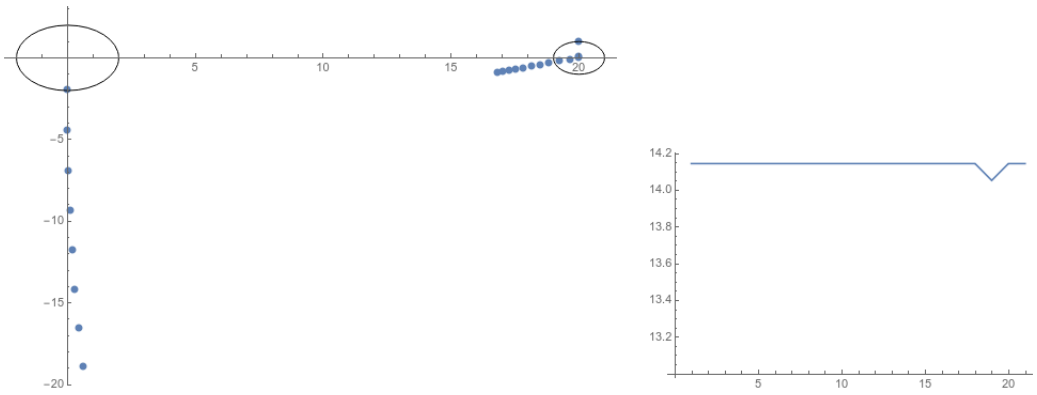

Figura 2: Caso 1. N=20

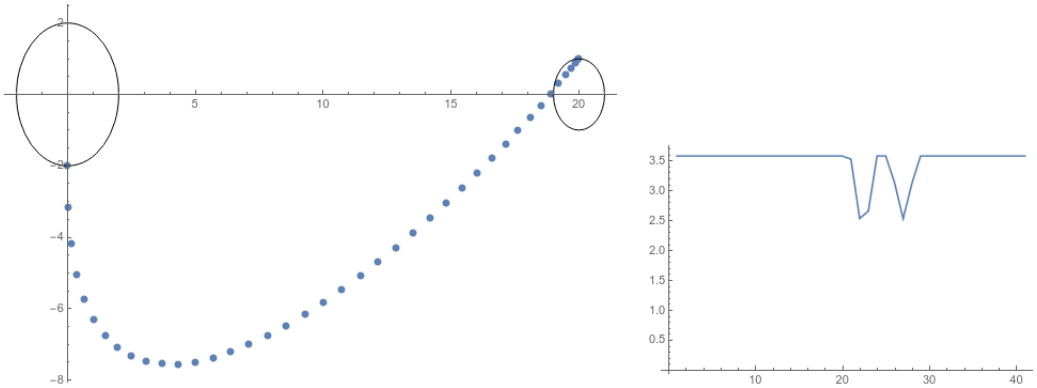

Figura 3: Caso 1. N=40

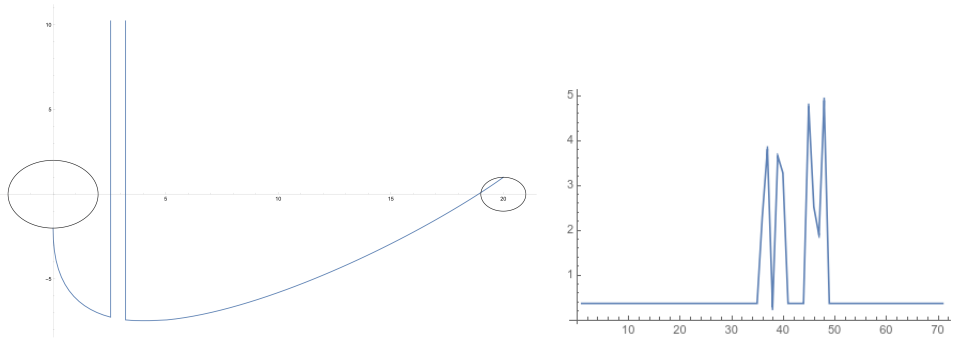

Figura 4: Caso 1. N=70. t = 512.188

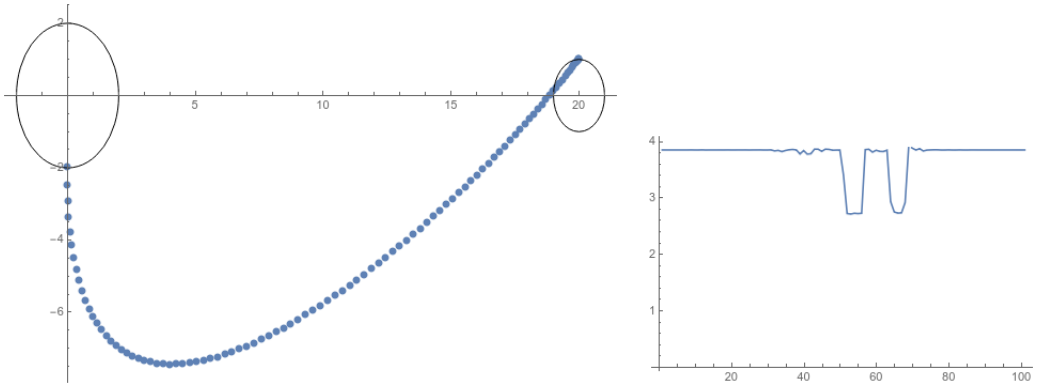

Figura 5: Caso 1. N=100. t = 1183.77

Caso 2

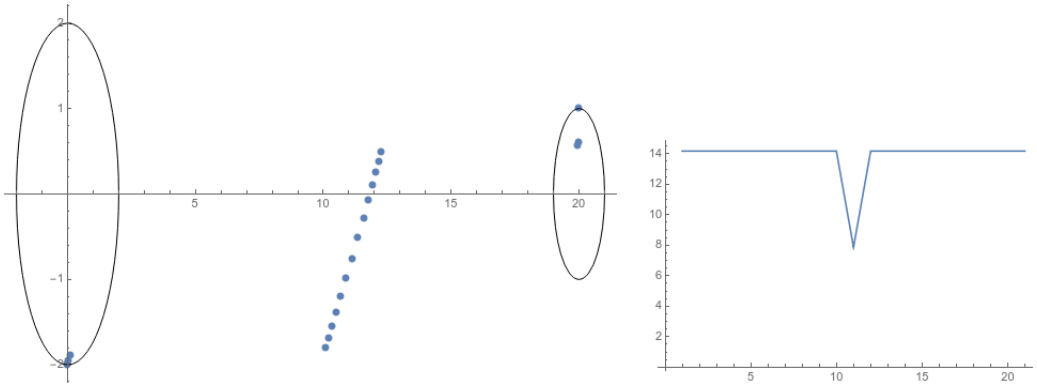

Figura 6: Caso 2. N=20

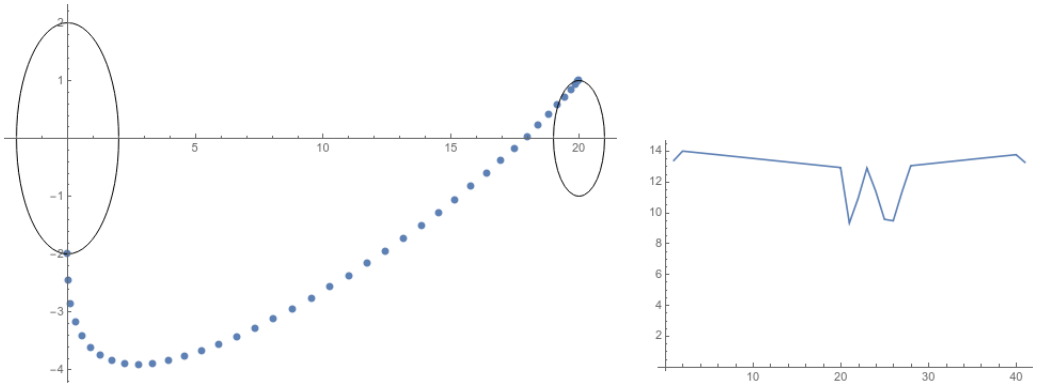

Figura 7: Caso 2. N=40

Caso 3

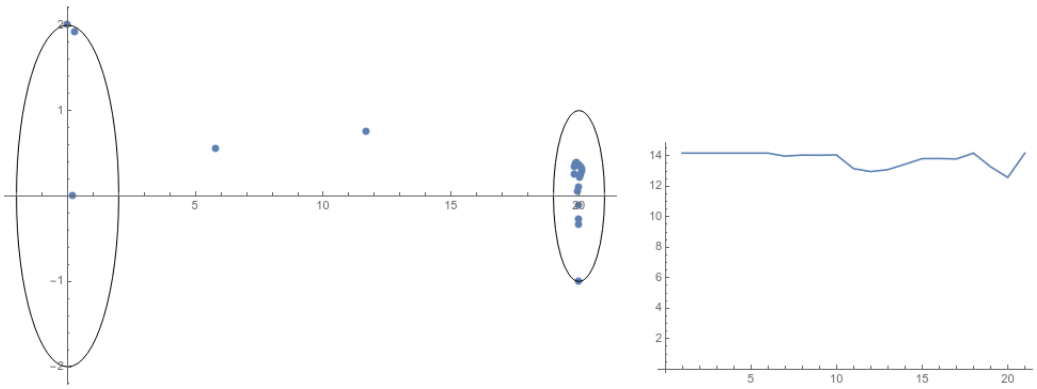

Figura 8: Caso 3. N=20

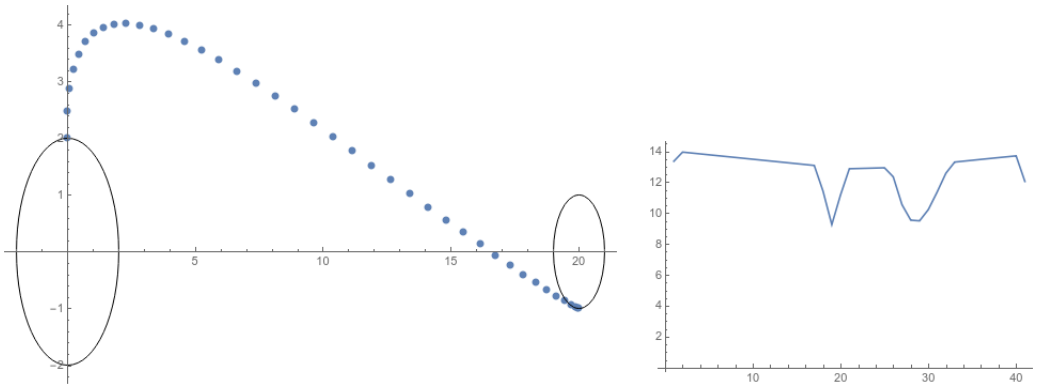

Figura 9: Caso 3. N=40

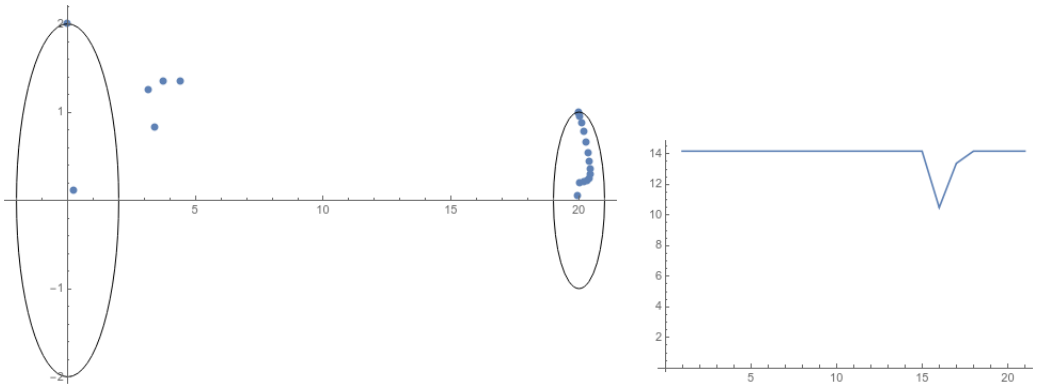

Figura 10: Caso 4. N=20

Caso 4

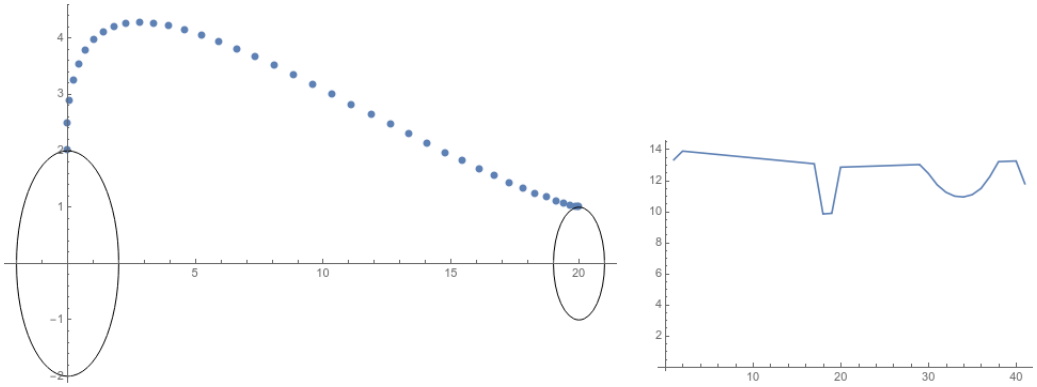

Figura 11: Caso 4. N=40

Caso 5

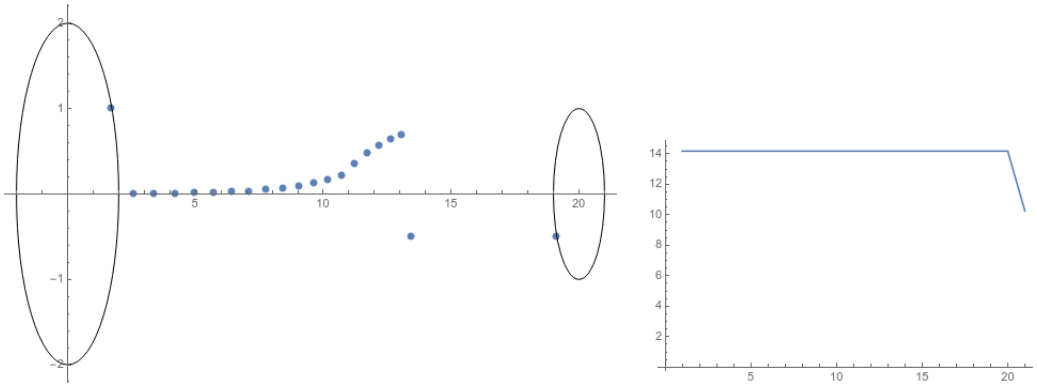

Figura 12: Caso 5. N=20

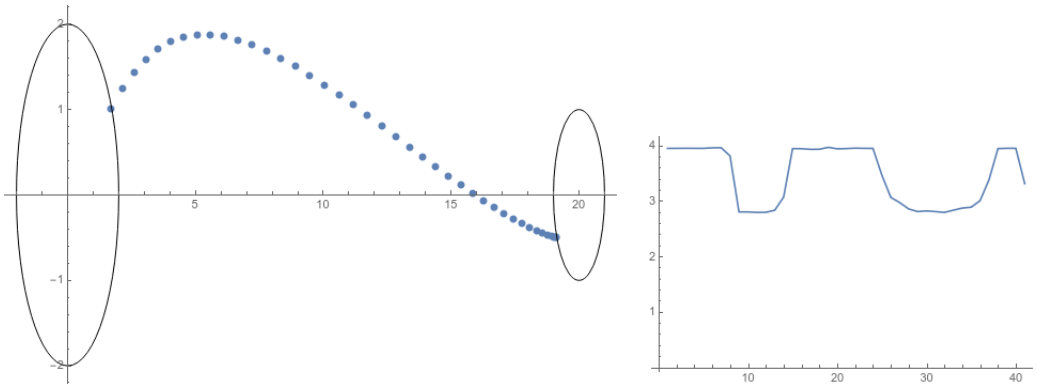

Figura 13: Caso 5. N=40

5. BIBLIOGRAFÍA

- RODRIGUEZ GALVÁN, J.R., **Problemas de valor inicial para ecuaciones diferenciales de primer orden**, 2015
- GIANCOLI, DOUGLAS C.**Física para ciencias e ingeniería. Cuarta edición.**, PEARSON EDUCACIÓN, México, 2008