

Herramientas computacionales en el análisis numérico de fenómenos físicos y sistemas dinámicos

Marcelo Gallardo Burga¹ and Jorge R. Chávez Fuentes²

¹Estudios Generales Ciencias PUCP

²Departamento de Matemáticas PUCP

Diciembre 2020

1. Resumen

Con la cuarta revolución industrial, el mundo cambió radicalmente. La presencia de artefactos digitales en nuestra vida cotidiana aumentó considerablemente. Desde el disco duro hasta el microprocesador I9. Es en este contexto que nace la física computacional. Ya hoy en día un físico, ingeniero, matemático o economista, enfrenta en algún momento de su carrera un problema de naturaleza numérica. Esto concierne el análisis de datos, las simulaciones etc. Por otro lado, desaprovechar la potencia del cálculo numérico impediría la resolución de ciertos problemas físicos, sin soluciones analíticas. En este documento presentaremos estas ventajas mediante ejemplos de la física. En particular, estudiaremos el comportamiento de una partícula sometida a la fuerza de Lorentz así como un péndulo agitado con una fuerza eléctrica. Nuestro propósito será ilustrar dichas ventajas con el objetivo de impulsar la creación o/y desarrollo de cursos y material relacionado a este campo de estudios en la PUCP.

Palabras Claves: convergencia numérica, estabilidad, sensibilidad, integradores numéricos.

2. Abstract

With the fourth industrial revolution, our world radically changed. The presence of digital devices in our daily lives, increased considerably. From the hard disc to the I9 microprocessor. It is in this context that computational physics was born. Nowadays, a physicist, engineer, mathematician or economist, faces at some point of his career a problem of a numerical nature. This concerns data analysis, simulations etc. On the other hand, wasting the power of numerical calculus, without profiting of this advantages, would prevent the resolution of problems without analytical solutions. In this document we present these advantages, using examples from physics. In particular, we will study the behaviour, total energy ... of an electric particle submitted to Lorentz force, as well as a forced and damped pendulum. Our purpose is to illustrate these advantages in order to promote the creation or development of courses and related material to this field of study at the PUCP.

Key Words: numerical convergence, stability, sensibility, numerical integrators.

3. Introducción

La pandemia del Coronavirus ha evidenciado en el Perú la brecha tecnológica en el acceso a la educación. En efecto, tanto la falta de conectividad como la de dispositivos de estudio (computadoras, tabletas...) se han hecho notar después de la virtualización de los cursos, para casi la totalidad de las instituciones educativas. No obstante, también ha permitido descubrir nuevas herramientas digitales, y valorar estas últimas. Por otro lado, en laboratorios como el Swiss Plasma Center (Suiza), cada día se simula la interacción de un plasma al interior de un reactor Tokamak. ¿Qué nos muestra esto? La brecha digital, no sólo interna (es decir, entre la regiones de nuestro país), pero también la fuerte diferencia en la valorización y acceso a estas tecnologías con respecto a otros países. Sin ir muy lejos, la inversión en ciencia tecnología en el Perú, es de las más bajas en América Latina. Según CONCYTEC, en el 2017 el Perú invertía 0.08 puntos del PBI en ciencia y tecnología, mientras que nuestros vecinos, Uruguay, Chile, México ... cincuenta veces más (Gago 2017). Es así que, en este contexto, proponemos la implementación y desarrollo de cursos o material relacionados a la física computacional, es decir, los métodos numéricos que permiten simular sistemas dinámicos. Creemos que esto contribuirá a acortar la brecha XX que nos separa de países más avanzados y con acceso a más tecnología.

Para ilustrar la importancia y los beneficios de las herramientas computacionales, usaremos dos casos clásicos de la literatura. El primero es una partícula eléctrica (protón), bajo el efecto de la Fuerza de Lorentz. En este caso, es posible deducir una solución analítica bajo procedimientos estándares de cálculo diferencial e integral. Por otro lado, tomaremos como ejemplo del caso no lineal, uno donde es imposible hallar una solución analítica. Para ejemplificar esto, utilizaremos el caso del péndulo forzado será utilizado. Nos permitirá introducir la teoría del caos y comparar ambos casos, versus las soluciones numéricas. Nuestro interés, en particular, será de analizar la convergencia numérica así como la estabilidad de los integradores numéricos. La sensibilidad a las condiciones iniciales y los límites del cálculo numérico también serán abordados. Si bien numerosos artículos y libros, relacionados a la teoría del caos ya han presentado estos sistemas físicos, nuestro aporte será el de brindar una comparación cualitativa entre diversos integradores numéricos, para evidenciar las ventajas y los límites de las simulaciones. Todas las simulaciones, imágenes, ecuaciones y códigos han sido elaborados por los autores (salvo indicación contraria).

4. Materiales y métodos

4.1. Código C++

Para simular los fenómenos físicos que estudiaremos, usamos las ventajas de la programación orientada objetos en C++. ¹ En efecto, la Programación Orientada a Objetos [1], permite estructurar el código de tal forma que optimizamos el desarrollo de las simulaciones (Chappellier 2020). Primero, creamos clases (estructuras generalizadas, más completas) que contienen los parámetros del sistema (velocidad angular, posición, energía...). En particular, cuando se definen estas propiedades, se usan funciones propias de la clase. Esto facilita su acceso desde un integrador numérico. Es el caso de la función aceleración por ejemplo $\sum F = m\mathbf{a}$. Finalmente, incorporamos diferentes integradores numéricos, con el objetivo de hacer evolucionar el sistema estudiado (esto se detalla en la sección Integradores numéricos). En líneas generales, el código tiene el siguiente aspecto:

```
1 #include <iostream>          // inputs b sicos
2 #include <fstream>           // input output file stream class
3 #include <cmath>              // libreria matematica de base
4 #include <iomanip>            // manipuladores
5 #include <valarray>           // funcion valarray para simplificar operaciones
6 #include <vector>             // funciones vectoriales
7 #include "ConfigFile.h"      // Archivo anexo que contiene el c digo que permite importar las
                              // simulaciones
8
9
10 using namespace std; // para usar la liber a de C++ est ndar
11
12 //La clase engine es la que contiene el motor principal de las simulaciones, en este caso
13 //la clase que nos permite simular el p ndulo
14 class Engine
15 {
```

¹Códigos C++ y Matlab en GitHub

```

16
17 private:
18     //definición de variables internas: propias al sistema dinámico
19
20     double const pi=3.14159265358979323846264338327950288419716939937510582097494459230e0;
21     double tfin=0.e0;    // Tiempo final de la simulación
22     double g=0.e0;       // aceleración gravitacional
23     double mass=1.e0,q=0.e0,L=0.e0; //masa, carga eléctrica, longitud del péndulo
24     double k=0.e0;       // constante de fricción
25     double omega=0.e0,phi=0.e0; // velocidad angular y desfase
26     double E0=0.e0;      // intensidad del campo eléctrico
27     valarray<double> winit=valarray<double>(0.e0,2); // posiciones y velocidades iniciales
28     unsigned int sampling=1; // # de pasos cada cuanto se escribe en el output (t + N *
        samplig dt)
29     unsigned int last;
30     ofstream *outputFile; // puntero hacia el archivo de salida
31
32     //funciones internas a la clase Engine
33
34     void printOut(bool write)
35
36     {
37         if((!write && last>=sampling) || (write && last!=1))
38         {
39             *outputFile << t << " " << w[0] << " " << w[1] << " " \
40             << energy() << " " << nonConservativePower() << " " << \
41             (w[0]+pi)-(2.*pi)*floor((w[0]+pi)/(2.*pi))-pi << endl; // escritura en el output file
42             last = 1;
43         }
44         else
45         {
46             last++;
47         }
48     }
49
50     //Cálculo de las fuerzas no conservativas
51     double nonConservativePower()
52     {
53         double P = L*w[1]*(q*E0*sin(omega*t)*sin(w[0]) - L*k*w[1]);
54
55         return P;
56     }
57
58
59     //energía mecánica del sistema (ver cálculos analíticos)
60     double energy() const
61     {
62         double E = 0.5*mass*pow(L,2)*pow(w[1],2) + mass*g*L*(1-cos(w[0]));
63
64         return E;
65     }
66
67     // iteración temporal: función redefinida en la subclase integrador
68     virtual void step()=0;
69
70 protected:
71
72     //Variables libres, se puede acceder desde otra sección del código que no sea la clase
        Engine
73

```

```

74 double t=0.e0; // tiempo discretizado
75 double dt=1.e5; //paso de tiempo
76 valarray<double> w=valarray<double>(0.0e0,2); //velocidades, posiciones
77 valarray<double> constants=valarray<double>(0.0e0,3);
78
79 //Funcion accessible desde el integrador numrico (para este caso, Stormer Verlet)
80
81 double electricAcceleration(double const& t_,double const& theta_)
82 {
83
84     double a_E= (q/(mass*L))*E0*sin(omega*t_ + phi)*sin(theta_);
85     //se calcula la fuerza del campo el ctrico
86     return a_E;
87 }
88
89 //acceleracion gravitacional
90 double conservativeAcceleration(double const& theta_)
91 {
92     double a_P = -g/L*sin(theta_);
93     // calcular l acceleration de gravite
94     return a_P;
95 }
96
97 //acceleracion por las fuerzas de friccin
98 double dragAcceleration(double const& thetaDot_)
99 {
100     double a_F = (-k/mass)*thetaDot_;
101
102     // calcular l acceleration de la trainee lineaire
103     return a_F;
104 }
105 //acceleracion total
106 double acceleration_totale (double const& t, double const& theta, double const& thetaDot
    ){
107
108     return electricAcceleration(t, theta) + conservativeAcceleration(theta) +
        dragAcceleration(thetaDot);
109
110 }
111 public:
112 //Constructor de la clase:
113 Engine(ConfigFile configFile)
114 {
115     // variable locale
116     tfin    = configFile.get<double>("tfin");
117     dt      = tfin/configFile.get<double>("nsteps");
118     mass    = configFile.get<double>("mass");
119     q       = configFile.get<double>("q");
120     L       = configFile.get<double>("L");
121     g       = configFile.get<double>("g");
122     k       = configFile.get<double>("k");
123     E0      = configFile.get<double>("E0");
124     omega   = configFile.get<double>("omega");
125     phi     = configFile.get<double>("phi");
126     winit[0] = configFile.get<double>("theta0");
127     winit[1] = configFile.get<double>("thetaDot0");
128     sampling = configFile.get<unsigned int>("sampling");
129
130     // Apertura del archivo de salida
131     outputFile = new ofstream(configFile.get<string>("output").c_str());

```

```

132     outputFile->precision(15); //Los nmeros se escriben con 15 decimales de precision
133 };
134
135 // Destructor virtual
136 virtual ~Engine()
137 {
138     outputFile->close();
139     delete outputFile;
140 };
141
142 // Simulacin completa
143 void run()
144 {
145     t = 0.e0; // t0
146     w = winit; // vector posicion, velocidad angular en t0
147     last = 0;
148     printOut(true);
149     while(t<=tfin)
150     {
151         step(); // metodo step: integracion
152         printOut(false);
153     }
154     printOut(true);
155 };
156
157 };
158
159 // Integrador Stormer Verlet:
160 class EngineStormerVerlet: public Engine
161 {
162 protected:
163 public:
164
165     // Constructor
166     EngineStormerVerlet(ConfigFile configFile): Engine(configFile){}
167     void step()
168     {
169         double xmem = w[0] ;
170         w[0]=w[0]+w[1]*dt + 0.5*acceleration_totale(t,w[0],w[1])*dt*dt;
171         double v = w[1] + 0.5*acceleration_totale(t,w[0],w[1])*dt;
172         w[1] = w[1]+0.5*(acceleration_totale(t+dt,w[0],v) + acceleration_totale(t,xmem,v))*dt;
173         t+=dt;
174     }
175 };
176
177 int main(int argc, char* argv[])
178 {
179     string inputPath("configuration.in"); //
180     if(argc>1)
181         inputPath = argv[1];
182
183     ConfigFile configFile(inputPath);
184
185     for(int i(2); i<argc; ++i)
186         configFile.process(argv[i]);
187     string schema(configFile.get<string>("schema"));
188
189     Engine* engine;
190
191     if(schema == "StormerVerlet" || schema == "SV")

```

```

192 | {
193 |
194 |     engine = new EngineStormerVerlet(configFile);
195 | }
196 | else
197 | {
198 |     cerr << "Schema inconnu" << endl;
199 |     return -1;
200 | }
201 |
202 | engine->run();
203 |
204 | delete engine;
205 | cout << "Fin de la simulation." << endl;
206 | return 0;
207 | }

```

4.2. Código en Matlab

Después de haber creado la data para las simulaciones en C++ (un archivo output), un documento texto con nuestras variables en orden (posición angular, velocidad angular, energía...), importamos los datos en Matlab. Presentemos el orden de las operaciones:

- 1) Calculamos analíticamente las ecuaciones de interés (ver sección cálculos analíticos)
- 2) Elaboramos el código C++ de tal forma que recibimos un input (datos θ_0, m, L) y obtenemos un output (texto con los valores numéricos en tiempo discreto)
- 3) Importamos la data a Matlab y pasamos la información a continuo (reconstruimos con la data discreta)

4.3. Integradores numéricos

4.3.1. Método de Euler explícito

La idea de un integrador numérico, es pasar de lo discreto a lo continuo descomponiendo el tiempo en una variable denombrable. Separamos en intervalos Δt . El objetivo es resolver (ver cálculos analíticos) $\frac{d}{dt}\mathbf{y} = \mathbf{f}(\mathbf{y}, t)$. Entonces, escogemos un intervalo $I = [a, b]$ con $t \in I$. Luego, dividimos I en n intervalos de longitud homogénea $\frac{b-a}{n} = \Delta t = t_{n+1} - t_n$. Finalmente, computamos $f(\mathbf{y}_n, t_n)$. Con esta información, obtenemos $\mathbf{y}_{n+1} = \mathbf{y}_n + \mathbf{f}(\mathbf{y}_n, t_n)\Delta t$. [8]

4.4. Euler implícito

El mayor inconveniente con el método de Euler explícito, es la inestabilidad numérica (este fenómeno se estudiará en la sección resultados, para la partícula electromagnética). Es por esta inestabilidad que introducimos el método de Euler implícito, que en vez de usar \mathbf{y}_n , usa \mathbf{y}_{n+1} .

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \mathbf{f}(\mathbf{y}_{n+1}, t_{n+1})\Delta t + O(\Delta t)$$

El problema es que \mathbf{y}_{n+1} se calcularía usando \mathbf{y}_{n+1} (lo cuáles una contradicción). Para encontrar \mathbf{y} lo suficientemente cerca (una aproximación), iteramos:

$$\mathbf{y}_{n+1}^{k+1} = \mathbf{y}_n + \mathbf{f}(\mathbf{y}_{n+1}^k, t_{n+1})\Delta t$$

Este nuevo algoritmo evita la inestabilidad.

4.5. Euler-Cromer

El algoritmo de Euler Cromer (así como el de Runge-Kutta), usa, parcialmente, la información en la etapa $n + 1$, con el objetivo de calcular el último parámetro con mayor precisión. En efecto, para nuestras ecuaciones es necesario calcular \mathbf{x}_{n+1} y \mathbf{v}_{n+1} con $\mathbf{x} = (x_x, x_z)$ y $\mathbf{v} = (v_x, v_z)$.

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{x}_{x,n}\Delta t \implies v_{x,n+1} = v_{x,n} + a_x(v_{z,n})\Delta t$$

Sólo después de obtener la información, calculamos $v_{z,n+1} = a_z(v_{z,n})\Delta t$.

4.6. Runge-Kutta orden 2

El algoritmo de Runge-Kutta (orden 2), usa la información en $\frac{\Delta t}{2}$. Primero, calculamos:

$$\mathbf{x}_{t_n+\Delta t/2} = \mathbf{x}_n + \mathbf{f}(\mathbf{x}_n, t_n)$$

Actualizamos \mathbf{f} con la información en $t_n + \frac{\Delta t}{2}$.

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \mathbf{f}(\mathbf{y}_n, t_n)$$

4.7. Boris-Buneman

[3] Finalmente, presentamos el integrador numérico de Boris-Buneman. La ventaja de este integrador es la simetría (provee una información de los parámetros en la totalidad del intervalo $[t_n, t_{n+1}]$). Definimos

$\omega_c = \frac{qB}{m}$ y $\beta = \omega_c \Delta t$:

$$\begin{aligned} \mathbf{x}_- &= \mathbf{x}_n + \mathbf{v}_n \frac{\Delta t}{2}, \quad \mathbf{v}_- = \mathbf{v} + \frac{q\mathbf{E}}{m} \frac{\Delta t}{2}, \quad v_{x,+} = \left[\left(1 - \frac{\beta^2}{4} \right) v_{x,-} - \beta v_{z,-} \right] \times \frac{1}{1 + \frac{\beta^2}{4}} \\ v_{z,+} &= \left[\left(1 - \frac{\beta^2}{4} \right) v_{z,-} + \beta v_{x,-} \right] \times \frac{1}{1 + \frac{\beta^2}{4}}, \quad \mathbf{v}_{n+1} = \mathbf{v}_+ + \frac{q\mathbf{E}}{m} \frac{\Delta t}{2}, \quad \mathbf{x}_{n+1} = \mathbf{x}_- + \mathbf{v}_{n+1} \frac{\Delta t}{2} \end{aligned}$$

5. Cálculos analíticos

5.1. Partícula cargada en un campo electromagnético: ecuaciones diferenciales del movimiento y energía total

En esta sección estudiaremos el comportamiento de una partícula eléctrica (un protón), en un campo electromagnético (fuerza de Lorentz), así como el de un péndulo agitado eléctricamente. Consideremos un protón de carga $1,6022 \times 10^{-19}[C]$ y de masa $1,6726 \times 10^{-27}[kg]$. Usamos como posiciones iniciales $(x_0, z_0) = (0, 0)$ y velocidades iniciales $(v_{0x}, v_{0z}) = (0, v_0)$. La fuerza actuando sobre el sistema (la partícula), es la fuerza de Lorentz:

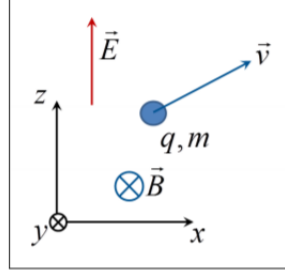


Figura 1: Partícula eléctrica en campo electromagnético

$$\sum \mathbf{F} = m\mathbf{a} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (1)$$

Nuestro objetivo es obtener un sistema de ecuaciones diferenciales de la forma $\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y})$ con $\mathbf{y} = (x, z, v_x, v_z)$. Usando la fórmula estándar del producto vectorial, [6] obtenemos:

$$\mathbf{v} \wedge \mathbf{B} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ v_x & 0 & v_z \\ 0 & B_0 & 0 \end{vmatrix} = \begin{pmatrix} -B_0 v_z \\ 0 \\ B_0 v_x \end{pmatrix} \quad (2)$$

De este modo, notando que $\frac{dx}{dt} = v_x$ y $\frac{dz}{dt} = v_z$:

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}) = \mathbf{f}(x, z, v_x, v_z) = \left(v_x, v_z, \frac{-qB_0 v_z}{m}, \frac{qv_x B_0 + qE_0}{m} \right) \quad (3)$$

Ya habiendo definido las ecuaciones diferenciales del movimiento (3), nos interesamos en la fórmula de la energía mecánica de nuestro sistema. Sabemos que:

$$E_{\text{mec}} = K + U = \frac{1}{2}M\mathbf{v}^2 + U \quad \text{and} \quad \nabla U = -\mathbf{F} \quad (4)$$

Lo que implica:

$$U = - \int_{\infty}^r \mathbf{F} d\mathbf{r} = - \int_{\infty}^r \left(q\mathbf{E} + q\mathbf{v} \wedge \mathbf{B} \right) d\mathbf{r} = -q\langle \mathbf{E}, \mathbf{r} \rangle = -qEz \quad (5)$$

Usando $\langle \mathbf{v} \wedge \mathbf{B}, d\mathbf{r} \rangle = 0$:

$$E_{\text{mec}} = \frac{1}{2}m(v_x^2 + v_z^2) - qEz \quad (6)$$

Debemos probar que la energía mecánica se conserva (al ser la fuerza de Lorentz conservativa):

$$\frac{dE_{\text{mec}}}{dt} = ma_x v_x + ma_z v_z - qE_0 v_z = -qB_0 v_z \cdot v_x + (qv_x B_0 + qE_0) \cdot v_z - qE_0 v_z = 0 \quad (7)$$

Puesto que $v_x \cdot v_z = 0$.

En el caso de una partícula acelerada bajo la fuerza de Lorentz, es posible resolver analíticamente la ecuación del movimiento, i.e, resolver el sistema $\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y})$. Usando los resultados ya obtenidos, tenemos:

$$\begin{cases} -qB_0 v_z = ma_x \\ qv_x B_0 + qE_0 = ma_z \end{cases} \implies \begin{cases} -qB_0 \dot{z} = m\ddot{x} \\ qB_0 \dot{x} + qE_0 = m\ddot{z} \end{cases} \quad (8)$$

Analicemos en un primer lugar el caso $E_0 = 0$. Llamamos $\omega_0 = \frac{qB_0}{m}$:

$$\begin{cases} -\frac{qB_0}{m}\dot{z} = \ddot{x} \\ \frac{qB_0}{m}\dot{x} = \ddot{z} \end{cases} \implies \begin{pmatrix} \ddot{x} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} 0 & -\omega_0 \\ \omega_0 & 0 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{z} \end{pmatrix} = \ddot{\mathbf{Y}}(t) = A\dot{\mathbf{Y}}(t) \quad (9)$$

La solución general (Rumbos, Lomelí, 2001)[7] está dada en términos de los valores y vectores propios de la matriz:

$$\chi_A(t) = \begin{vmatrix} t & \omega_0 \\ -\omega_0 & t \end{vmatrix} = t^2 + \omega_0^2 \quad (10)$$

Con $\lambda_k = \alpha \pm i\beta$ y $\mathbf{r} \pm i\mathbf{s}$ la matriz de vectores propios. Usamos $\mathbf{r} = (0, 1)$ y $\mathbf{s} = (1, 0)$. En efecto, $\mathbf{v}_1 = (i, 1)^T$ y $\mathbf{v}_2 = (-i, 1)^T$. Tomando en consideración esto último, la solución a la ecuación diferencial sería [2]:

$$\dot{\mathbf{Y}}(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{z}(t) \end{pmatrix} = \begin{pmatrix} c_2 \cos(\omega_0 t) - c_1 \sin(\omega_0 t) \\ c_1 \cos(\omega_0 t) + c_2 \sin(\omega_0 t) \end{pmatrix} \quad (11)$$

Usando las condiciones iniciales $\dot{x}(0) = 0$ y $\dot{z}(0) = v_0$ tenemos $c_1 = v_0$, $c_2 = 0$:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{z}(t) \end{pmatrix} = \begin{pmatrix} v_x(t) \\ v_z(t) \end{pmatrix} = \begin{pmatrix} -v_0 \sin(\omega_0 t) \\ v_0 \cos(\omega_0 t) \end{pmatrix} \quad (12)$$

Integrando obtenemos $x(t)$ y $z(t)$:

$$x(t) = \int v_x(t) dt = \frac{v_0}{\omega_0} \cos(\omega_0 t) + C_1 \quad \text{and} \quad z(t) = \int v_z(t) dt = \frac{v_0}{\omega_0} \sin(\omega_0 t) + C_2 \quad (13)$$

Nuevamente, con las condiciones iniciales escogidas, i.e $x(0) = z(0) = 0$ encontramos:

$$\boxed{x(t) = \int v_x(t) dt = \frac{v_0}{\omega_0} \cos(\omega_0 t) - \frac{v_0}{\omega_0} \quad \text{y} \quad z(t) = \int v_z(t) dt = \frac{v_0}{\omega_0} \sin(\omega_0 t)} \quad (14)$$

Si $E_0 \neq 0$, es necesario resolver un sistema de ecuaciones diferenciales lineal pero no homogéneo:

$$\begin{cases} -\frac{qB_0}{m}\dot{z} = \ddot{x} \\ \left(\frac{qB_0}{m}\right)\dot{x} + \frac{qE_0}{m} = \ddot{z} \end{cases} \quad (15)$$

Escogemos la substitución:

$$\begin{cases} u_x = v_x + \frac{E_0 q}{m\omega_0} \\ u_z = v_z \end{cases} \implies \begin{cases} \dot{u}_x = -\omega_0 u_z \\ \dot{u}_z = \omega_0 u_x \end{cases} \implies \begin{cases} \ddot{u}_x = -\omega_0(\omega_0 u_x) = -\omega_0^2 u_x \\ \ddot{u}_z = \omega_0 u_x \end{cases} \quad (16)$$

Por ende:

$$u_x(t) = A \cos(\omega_0 t) + B \sin(\omega_0 t) \quad \text{and} \quad \dot{u}_x = -\omega_0 A \sin(\omega_0 t) + \omega_0 B \cos(\omega_0 t) \quad (17)$$

Usando la condición inicial sobre la variable u_x , heredada de v_x :

$$\begin{cases} u_x(0) = \frac{E_0 q}{m\omega_0} \\ \dot{u}_x(0) = -\omega_0 v_0 \end{cases} \implies \begin{cases} A = \frac{E_0 q}{m\omega_0} \\ B = -v_0 \end{cases} \quad (18)$$

$$u_x(t) = \frac{E_0 q}{m\omega_0} \cos(\omega_0 t) - v_0 \sin(\omega_0 t) \quad (19)$$

$$v_x(t) = \frac{E_0 q}{m\omega_0} \cos(\omega_0 t) - v_0 \sin(\omega_0 t) - \frac{E_0 q}{m\omega_0} \quad (20)$$

$$v_z(t) = \omega_0 \int u_x(t) dt = \frac{E_0 q}{m\omega_0} \sin(\omega_0 t) + v_0 \cos(\omega_0 t) \quad (21)$$

Finalmente, obtenemos la posición para todo tiempo, integrando las ecuaciones de la velocidad en los diferentes ejes, y usando las condiciones iniciales $x(0) = 0$ and $z(0) = 0$ obtenemos: $C_1 = \frac{-v_0}{\omega_0}$:

$$x(t) = \int v_x(t)dt = \frac{E_0 q}{m\omega_0^2} \sin(\omega_0 t) + \frac{v_0}{\omega_0} (\cos(\omega_0 t) - 1) - \frac{E_0 q}{m\omega_0} t \quad (22)$$

Y $C_2 = \frac{E_0 q}{m\omega_0^2}$:

$$z(t) = \int v_z(t)dt = \frac{E_0 q}{m\omega_0^2} (1 - \cos(\omega_0 t)) + \frac{v_0}{\omega_0} \sin(\omega_0 t) \quad (23)$$

Ya habiendo establecido el movimiento de la partícula, procederemos a comparar la solución analítica (22,23), con los diferentes integradores, evaluando su desempeño.

5.1.1. Péndulo agitado eléctricamente bajo el efecto de la fricción

El sistema estudiado consiste en un péndulo de longitud L , sometido a su propio peso $\mathbf{P} = m\mathbf{g}$, a una fuerza eléctrica de excitación, $\mathbf{E}(t) = E_0 \sin(\Omega t)\mathbf{e}_y$, así como a una fuerza de fricción $-\kappa\mathbf{v}$.

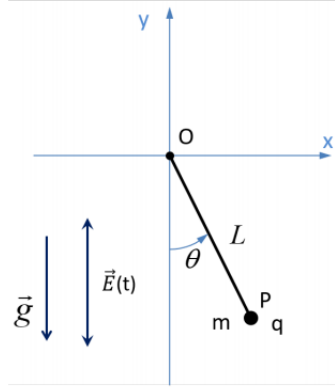


Figura 2: Pendulum Numerical Physics Laurent Villard 2020

Usando la fórmula de la velocidad y aceleración en coordenadas cilíndricas (Ricci 2019) [?]:

$$\vec{v}_P = \underbrace{\dot{p}}_{=0} \vec{e}_p + p\dot{\theta}\vec{e}_\theta + \underbrace{\dot{z}}_{=0} \vec{e}_z \quad (24)$$

$$\vec{a} = \left(\underbrace{\ddot{p}}_{=0} - p\dot{\theta}^2 \right) \vec{e}_p + \left(2\dot{p}\dot{\theta} + p\ddot{\theta} \right) \vec{e}_\theta + \underbrace{\ddot{z}}_{=0} \vec{e}_z \quad (25)$$

Nótese que hemos usado el hecho que $\dot{p} = \ddot{p} = 0$ ya que $\forall t : p = L$. Ya que no hay aceleración en \mathbf{e}_r , nos concentramos en el eje \mathbf{e}_θ . Proyectando las fuerzas en el sistema de coordenadas polares:

$$\sum \vec{F} \cdot \vec{e}_\theta = m\vec{a} \cdot \vec{e}_\theta = mL\ddot{\theta} = -mg \sin(\theta) + qE_0 \sin(\Omega t) \sin(\theta) - \kappa L\dot{\theta} \quad (26)$$

$$\ddot{\theta} = -\frac{g}{L} \sin(\theta) + \frac{q}{mL} E_0 \sin(\Omega t) \sin(\theta) - \frac{\kappa}{m} \dot{\theta} \quad (27)$$

Toda fuerza conservativa satisface:

$$\oint_{\Gamma} \mathbf{F} d\mathbf{r} = 0 \Leftrightarrow \mathbf{F} = -\nabla U$$

Sabemos que la energía potencial del peso es conservativa. Sin embargo, la fuerza eléctrica no deriva de un potencial puesto que depende del tiempo. En efecto, no podemos encontrar una solución a U_α a la ecuación $\vec{F}_\alpha = -\nabla U_\alpha$ con $\vec{F}_\alpha = \{\vec{F}_f, \vec{F}_E\}$. La fuerza eléctrica provee una contribución positiva a la energía

potencial. La fuerza de fricción claramente no es conservativa, ya que disipa energía en todo momento. De este modo, obtenemos la siguiente fórmula para la energía mecánica: [6]

$$\boxed{E(t) = \underbrace{\frac{1}{2}mL^2\dot{\theta}^2}_{=K} + \underbrace{mgL(1 - \cos(\theta))}_{=U}} \quad (28)$$

Tomando $(0\mathbf{e}_x, -L\mathbf{e}_y)$ como el punto de referencia para la energía potencial. De este modo, aplicando el teorema de la energía mecánica

$$\int_A^B \frac{d}{dt} E dt = \int_A^B P_{NC} dt \implies \frac{d}{dt} \int_A^B \frac{d}{dt} E dt = \frac{d}{dt} \int_A^B P_{NC} dt$$

Queremos obtener $\frac{d}{dt} E = P_{NC}$:

$$\frac{d}{dt} E(t) = \frac{d}{dt} \left(\frac{1}{2} mL^2 \dot{\theta}^2 + mgL(1 - \cos(\theta)) \right) = mL^2 \ddot{\theta} \dot{\theta} + mgL \sin(\theta) \dot{\theta}$$

Como $\theta \equiv \theta(t)$. Usamos (27) con $\ddot{\theta}$:

$$\begin{aligned} mL^2 \left(-\frac{g}{L} \sin(\theta) \dot{\theta} + \frac{1}{mL} qE_0 \sin(\Omega t) \sin(\theta) \dot{\theta} - \frac{\kappa}{m} \dot{\theta}^2 \right) + mgL \sin(\theta) \dot{\theta} \\ = \cancel{-gmL\dot{\theta}\sin(\theta)} + LqE_0 \sin(\Omega t) \sin(\theta) L\dot{\theta} - \kappa L^2 \dot{\theta}^2 + \cancel{(gmL\sin(\theta)\dot{\theta})} \\ = \boxed{qE_0 \sin(\Omega t) \sin(\theta) L\dot{\theta} - \kappa L^2 \dot{\theta}^2} \end{aligned} \quad (29)$$

Por otra parte, tomando $\mathbf{v}_P = L\dot{\theta}$:

$$P_{NC} = \mathbf{F} \cdot \mathbf{v}_P = \boxed{\left(qE_0 \sin(\Omega t) \sin(\theta) - \kappa L\dot{\theta} \right) \cdot L\dot{\theta}} \quad (30)$$

De este modo, tal y como buscábamos:

$$\frac{dE}{dt} = P_{NC}$$

5.2. Posiciones de equilibrio

Si tomamos $E = \kappa = 0$ and $|\theta - \theta_{eq}| \ll 1$, es posible halla una solución analítica $\ddot{\theta}, \theta$ (es incluso posible sin considerar $|\theta - \theta_{eq}| \ll 1$, [5]. De este modo, usando la información anterior, tenemos $\sum \mathbf{F} = 0 \implies \theta = \theta_{eq} = 0, \pi$:

$$\ddot{\theta} = -\frac{g}{L} \sin(\theta) \sim -\frac{g}{L} \theta$$

Más adelante estudiaremos los límites de esta aproximación $\sin(\theta) \sim \theta$. Entonces:

$$\ddot{\theta} + \frac{g}{L} \theta = \ddot{\theta} + \omega_0^2 \theta = 0$$

Con $\omega_0 = \sqrt{\frac{g}{L}}$. La solución general es:

$$\boxed{\theta(t) = \theta(0) \cos(\omega_0 t) - \frac{\dot{\theta}(0)}{\omega_0} \sin(\omega_0 t)} \quad (31)$$

Para el segundo caso $\pi = \theta$ usamos $\sin(\alpha + \beta) = \sin(\alpha) \cos(\beta) + \sin(\beta) \cos(\alpha)$ y consideramos la ecuación ($\theta \equiv \pi + \tilde{\theta}$):

$$\ddot{\theta} = \ddot{\tilde{\theta}} = -\frac{g}{L} \sin(\pi + \tilde{\theta})$$

Con $\tilde{\theta}(0) \sim 0$:

$$\ddot{\tilde{\theta}} + \frac{g}{L} \left(\sin(\pi) \cos(\tilde{\theta}) + \sin(\tilde{\theta}) \cos(\pi) \right)$$

Puesto que $\cos(\tilde{\theta}) \sim 1$, $\sin(\tilde{\theta}) \sim \tilde{\theta}$, $\sin(\pi) = 0$ and $\cos(\pi) = 1$, obtenemos:

$$\ddot{\tilde{\theta}} - \frac{g}{L}\tilde{\theta} = \ddot{\tilde{\theta}} - \omega_0\tilde{\theta} = 0$$

La solución general tiene la forma: [2]

$$\tilde{\theta}(t) = Ae^{\omega_0 t} + Be^{-\omega_0 t}$$

Recordemos que $\tilde{\theta} = \theta - \pi$. De este modo:[2]

$$\theta(t) = Ae^{\omega_0 t} + Be^{-\omega_0 t} + \pi$$

En este caso, la amplitud aumenta exponencialmente con el tiempo:

$$\lim_{t \rightarrow \infty} \theta(t) = \infty$$

No obstante, esto no es cierto en el laboratorio. Las fuerzas de fricción compensan este efecto. Además, la aproximación $\sin \theta \sim \theta$ se limita a ciertos momento $|\theta - \sin \theta| < \varepsilon$, con ε una variable de control.

6. Resultados

6.1. Análisis de la convergencia

Después de resolver analíticamente las ecuaciones diferenciales del movimiento, y haber definido los integradores numéricos en C++, podemos comparar el desempeño de la integración efectuando un estudio de convergencia. Con esta data (numérica y analítica), graficamos:

$$\varepsilon = \max \|x(t_j) - x_{th}(t_j)\|_2^2 \text{ y } \epsilon = \max \|v(t_j) - v_{th}(t_j)\|_2^2 \quad (32)$$

Para diferentes pasos de tiempo $\Delta t = f(N)$. Este es el error máximo para cada parámetro. Con la solución analítica, es posible medir $\varepsilon = \varepsilon(\Delta t)$. Asumiendo que $\Delta t \ll 1$ mediante una expansión de Taylor para $\varepsilon(\Delta t)$, obtenemos:

$$\varepsilon(\Delta t) = \alpha_0 + \alpha_1 \Delta t + \alpha_2 (\Delta t)^2 + \dots + \alpha_n (\Delta t)^n + o(\Delta t)^{n+1} \quad (33)$$

Sea α_k el primer término no nulo, entonces: $\varepsilon(\Delta t) = \alpha_k (\Delta t)^k + o(\Delta t)^{k+1}$. Aplicando $\log(\theta)$ en ambos lados:

$$\log(\varepsilon(\Delta t)) = \log(\alpha_k (\Delta t)^k) = \log(\alpha_k) + k \log(\Delta t) \quad (34)$$

Así, obtenemos una función lineal con k la pendiente de la curva. Es por esto, que k es el orden de convergencia (mientras más elevado, mejor). En efecto:

$$\Theta(\Delta t^{n+1}) \ll \Theta(\Delta t^n) \quad \forall |\Delta t| < 1 \quad (35)$$

6.2. Partícula eléctricamente cargada: convergencia

Consideremos los siguientes parámetros para nuestras simulaciones $t_f = 10\pi/B_0 q \sim 6,6 \times 10^{-8}[s]$ (5 periodos de rotación, ω_0). Aquí $E_0 = 0$ voltios por metro y $B = 5$ Teslas. El objetivo es analizar la convergencia para los diferentes integradores (comparar la convergencia). [?]

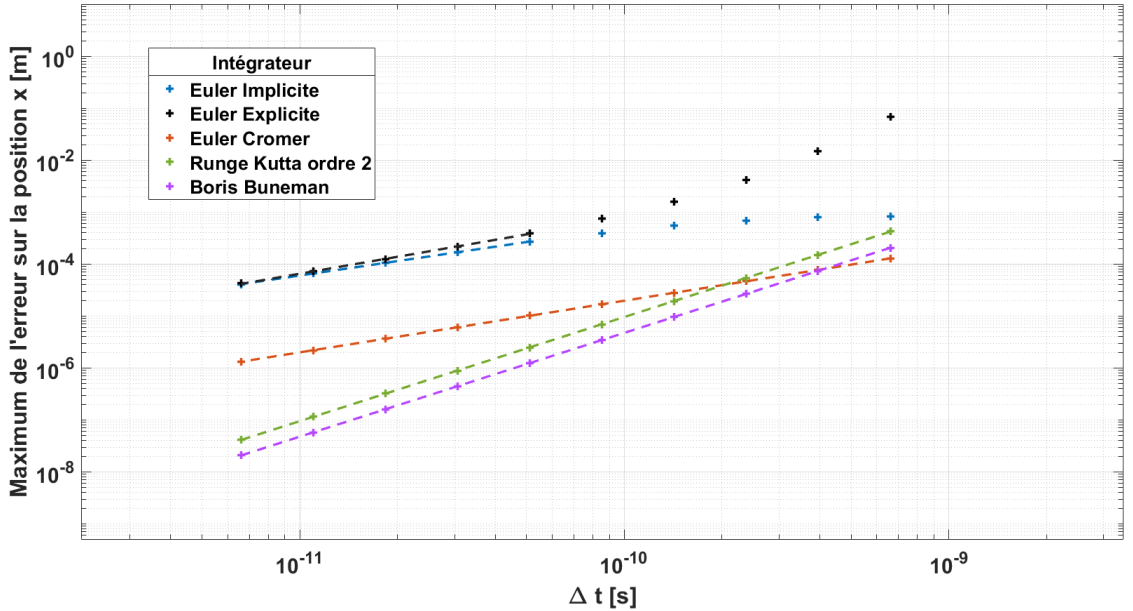


Figura 3: Error en la convergencia partícula electromagnética, posición en x

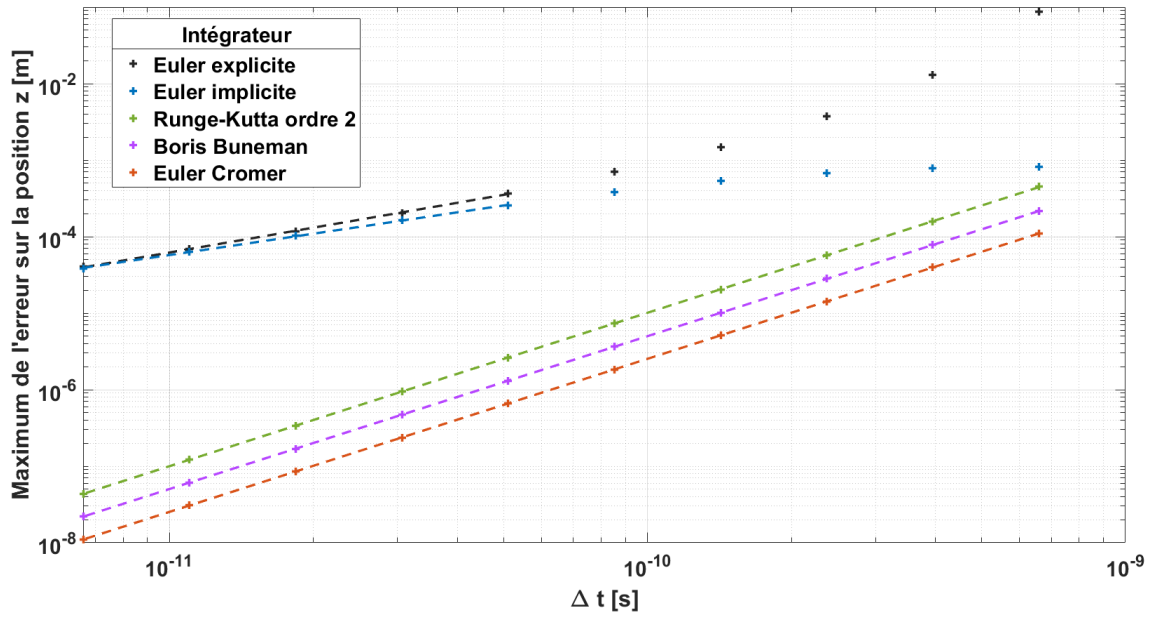


Figura 4: Error en la convergencia partícula electromagnética, posición en z

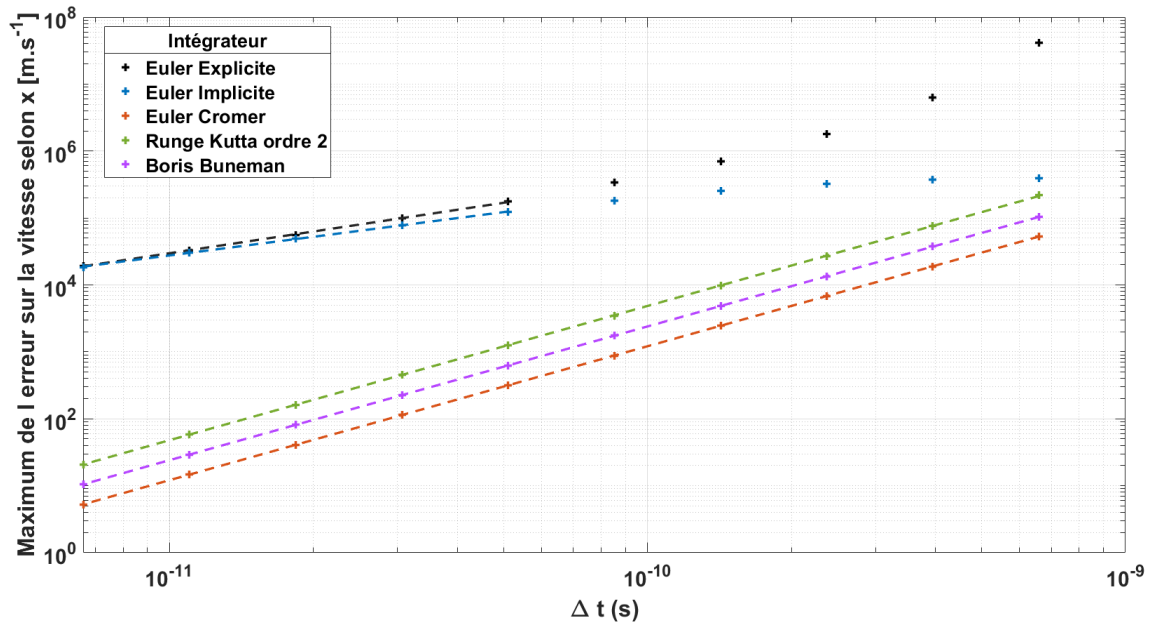


Figura 5: Error en la convergencia partícula electromagnética, posición en velocidad en x

Se puede observar que la convergencia numérica es superior para los métodos de Runge-Kutta y Boris Buneman. Esto evidencia la importancia del integrador numérico. Si bien el método de Euler es el más conocido y usado, por su simplicidad, no es óptimo. Esto se aprecia con mayor detalle en el estudio de la estabilidad.

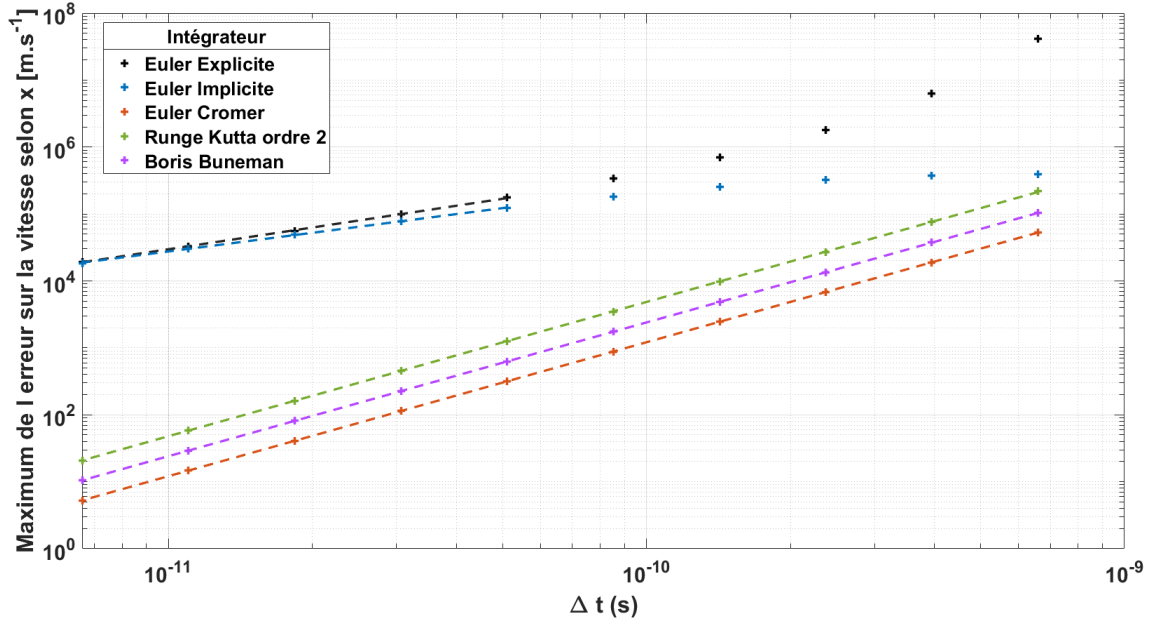


Figura 6: Error en la convergencia partícula electromagnética, posición en velocidad en z

6.3. Péndulo en la posición de equilibrio estable : comparación

Nuestro objetivo es demostrar la eficiencia y las ventajas del cálculo numérico. Para empezar, puesto que tenemos la solución analítica para la aproximación $||\theta|| \ll 1$, comparamos la solución obtenida en (31) con lo obtenido gracias al integrador Stormer-Verlet (anexos), tanto para la posición como para la velocidad angular.

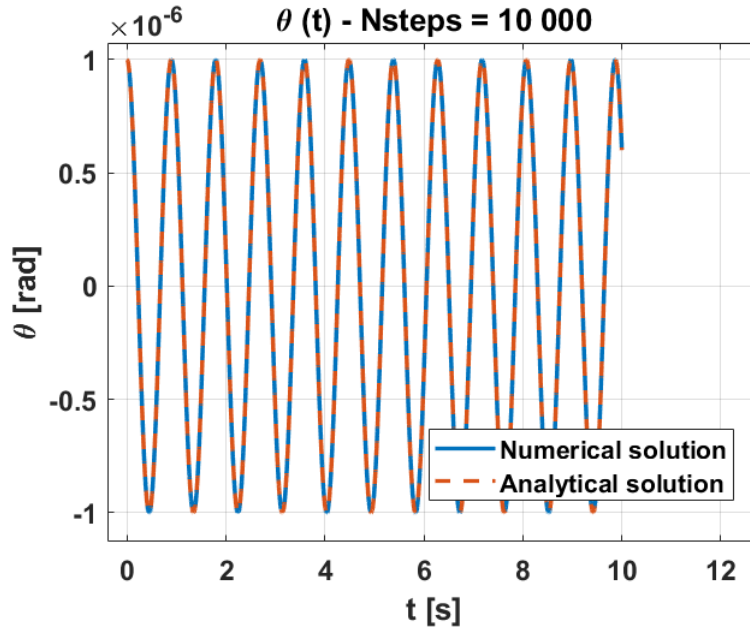


Figura 7: Péndulo, posición en función del tiempo, con la aproximación $|\theta| \ll 1$

Se puede apreciar que la integración numérica es casi perfecta (errores del orden de 10^{-10}). Esto se puede estudiar, como para el caso de la partícula electromagnética, mediante el estudio de la convergencia.

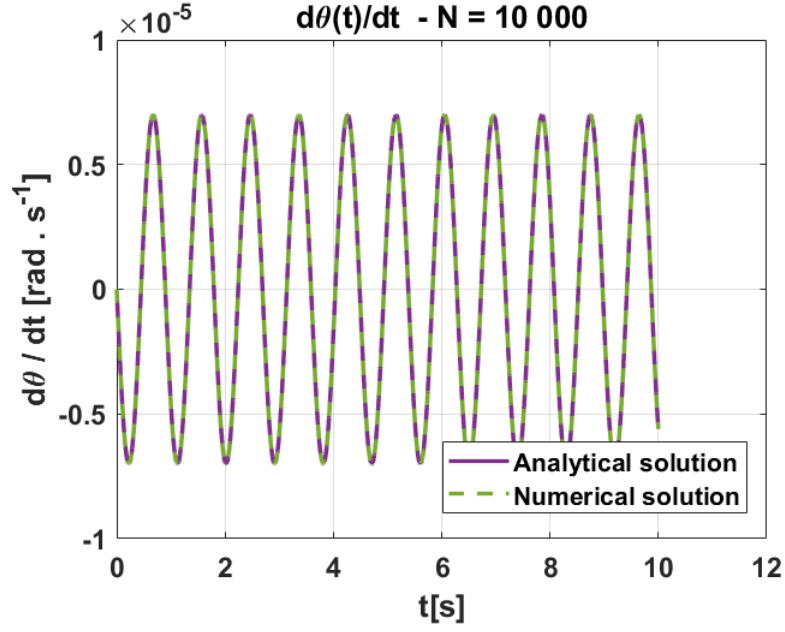


Figura 8: Péndulo, velocidad angular, con la aproximación $|\theta| \ll 1$

6.4. Convergencia péndulo

Así como analizamos la convergencia para la partícula electromagnética, aplicamos el mismo métodos a la convergencia del péndulo, para el caso lineal (solución a la ecuación diferencial lineal de orden 2).

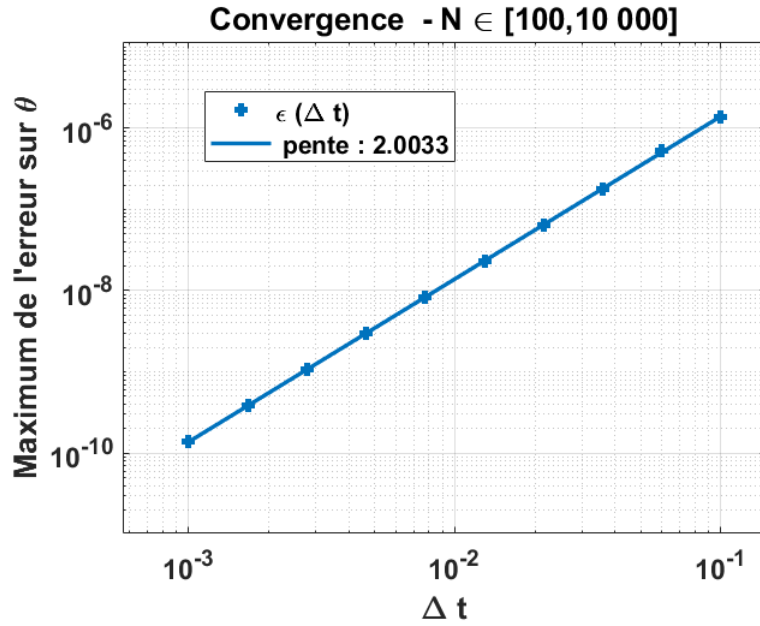


Figura 9: Convergencia péndulo θ sin fricción y campo eléctrico

Lo que nos muestran estos gráficos es que la convergencia es mayor para un número de pasos mayor (Δt más chico).

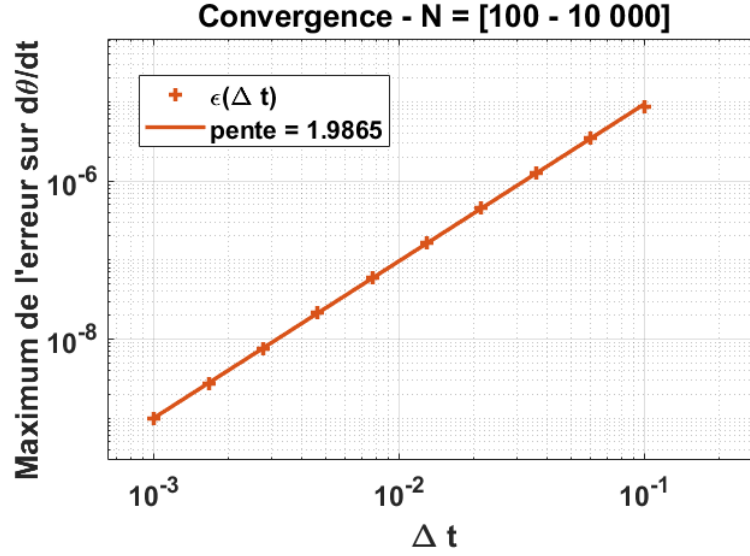


Figura 10: Convergencia péndulo $d\theta/dt$ sin fricción y campo eléctrico

6.5. Estabilidad: energía mecánica

Hemos presentado la convergencia numérica para los dos sistemas dinámicos. Observamos que la solución numérica se acerca, en una buena primera instancia, a la solución analítica (el error llega a ser en el mejor de los casos del orden de 10^{-9}). Observamos igualmente, que la convergencia es mejor para ciertos integradores. Ahora, nuestro objetivo es estudiar la estabilidad de estos integradores en función del tiempo de la simulación y el paso de tiempo. Usando nuevamente Matlab, obtenemos:

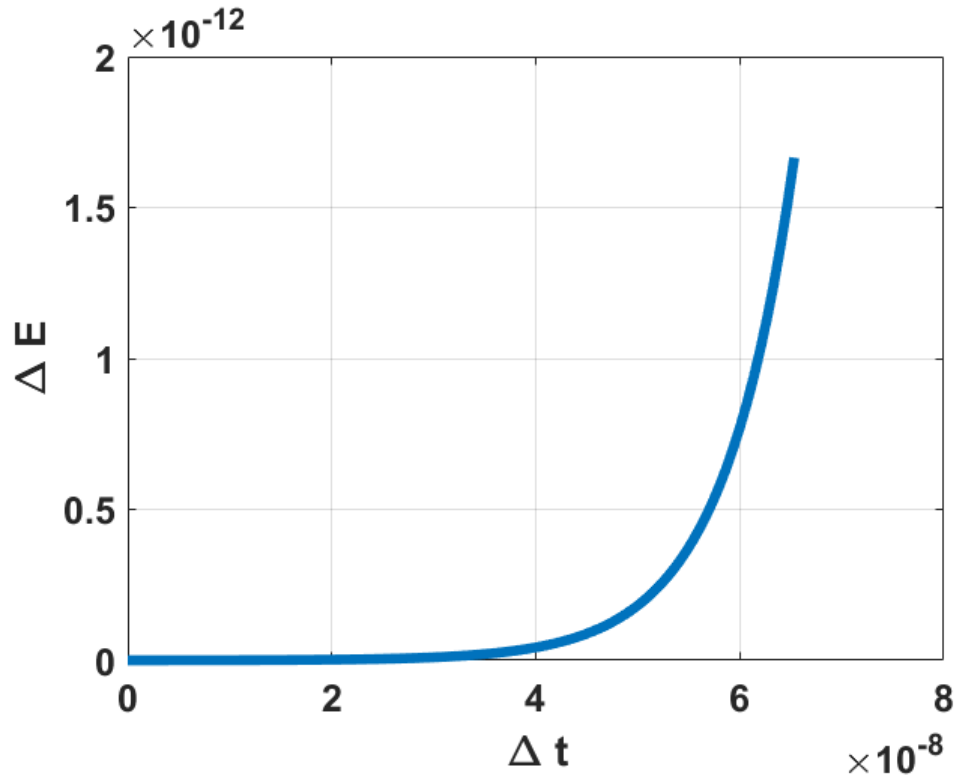


Figura 11: Euler explícito, $t_f = 10^{-8}$, $N = 10^4$

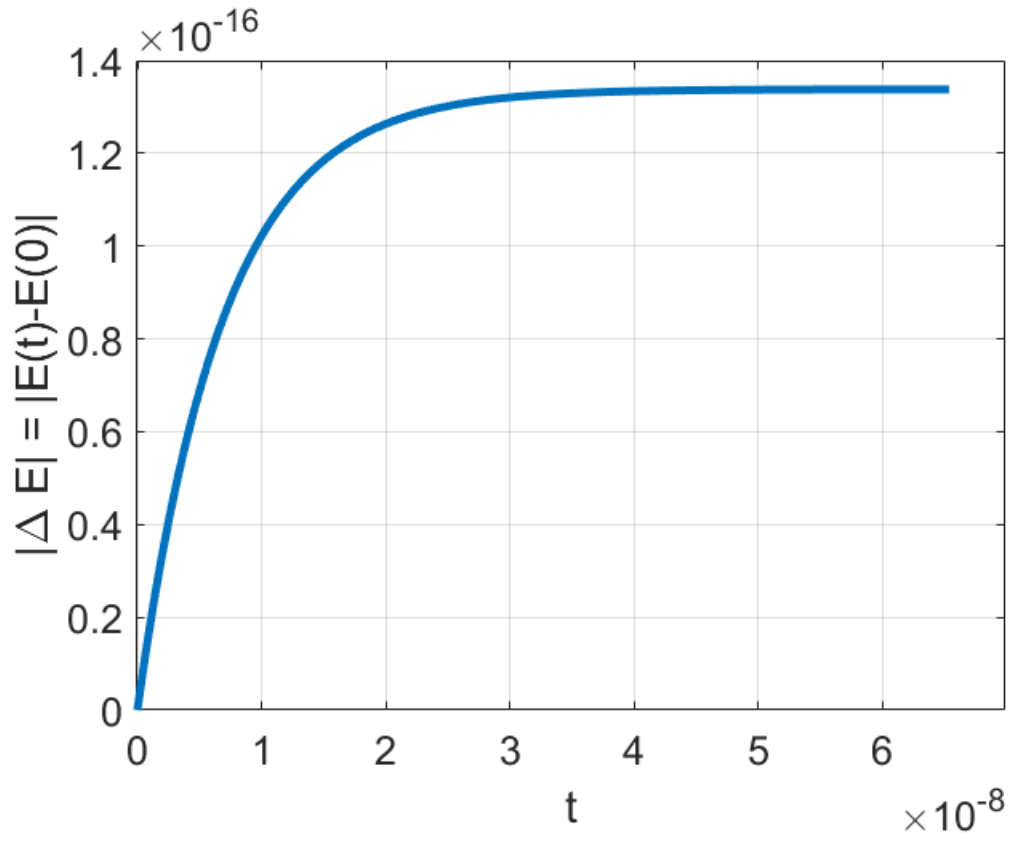


Figura 12: Euler implícito, $t_f = 10^{-8}$, $N = 10^4$

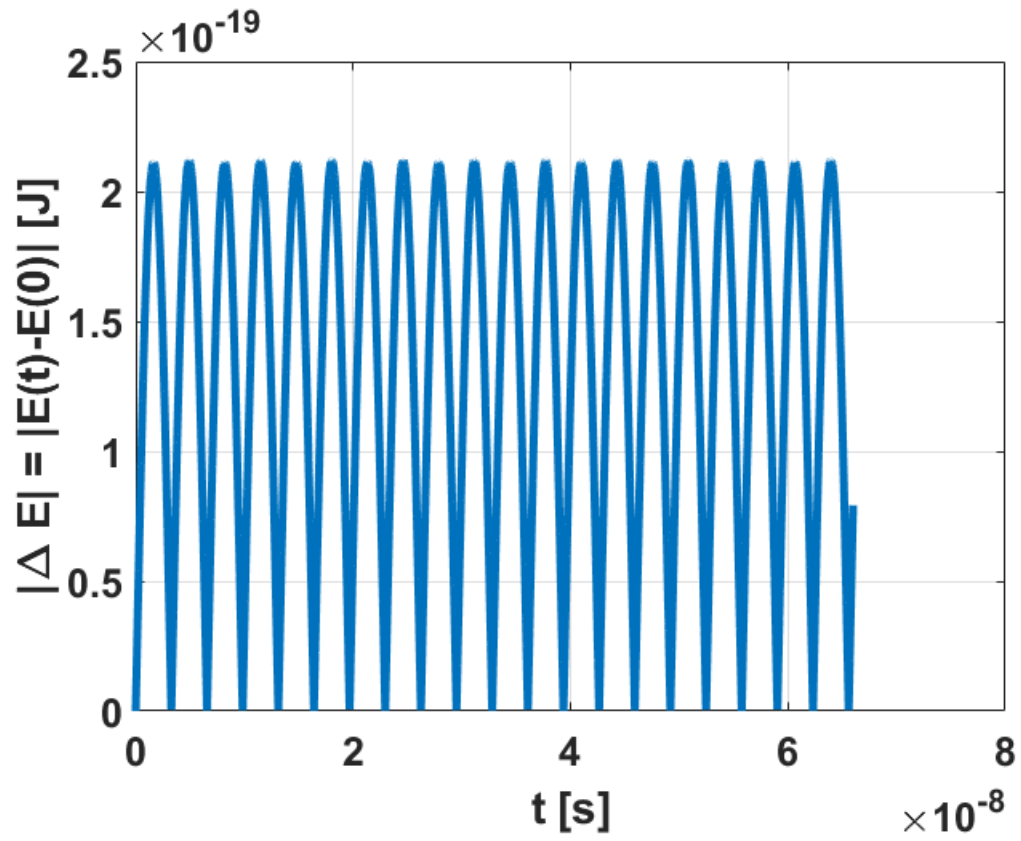


Figura 13: Euler Cromer, $t_f = 6,8 \times 10^{-8}$, $N = 10^4$

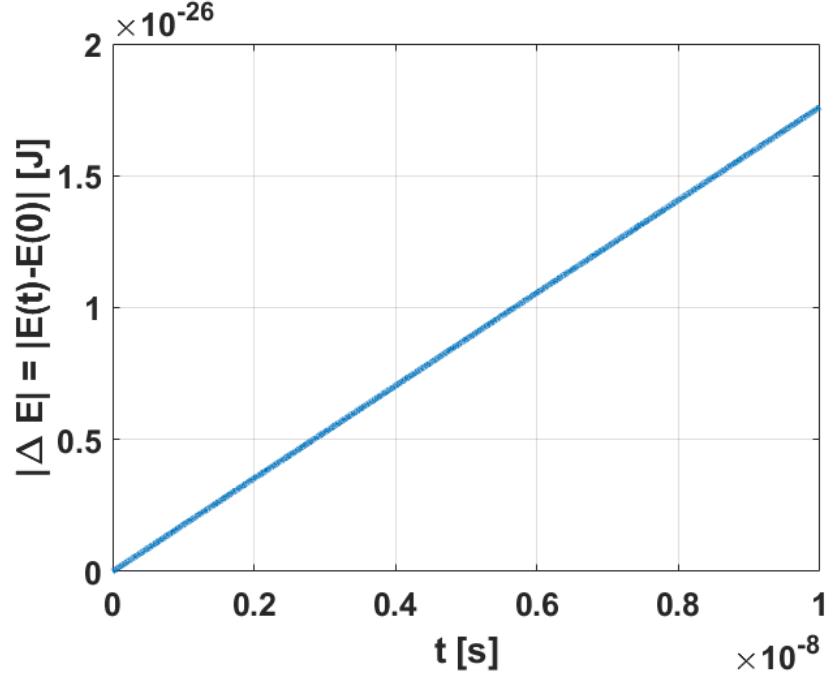


Figura 14: Runge-Kutta orden 2 $t_f = 10^{-8}$, $N = 10^4$

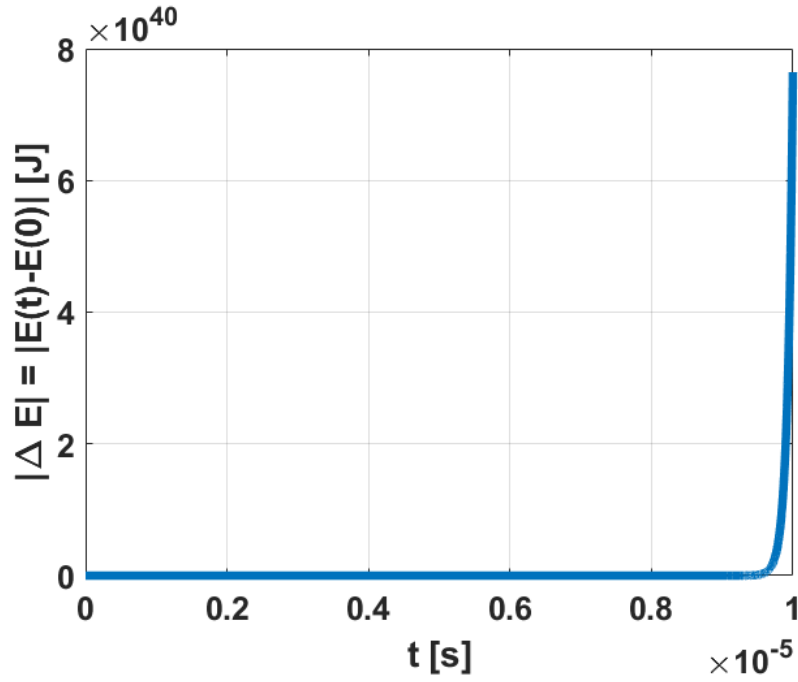


Figura 15: Runge-Kutta orden 2 $t_f = 10^{-5}$, $N = 10^4$

De estas figuras podemos sacar las siguientes conclusiones:

- 1) A mayor tiempo en la simulación, mayor inestabilidad. Por ejemplo, en el caso del integrador numérico Runge-Kutta de orden 2, tomando $t_f = 10^{-5}$ la energía mecánica explota. Sin embargo tomando $t_f = 10^{-8}$, esta se conserva.
- 2) El algoritmo Euler Implícito conserva la energía al largo plazo, el crecimiento es logarítmico. En paralelo, el algoritmo Euler explícito, explota con tal tiempo.
- 3) El integrador Numérico Boris-Buneman presenta un excelente desempeño describiendo la partícula cargada. La conservación de la energía y sus orden de convergencia elevado, respaldan esta observación.

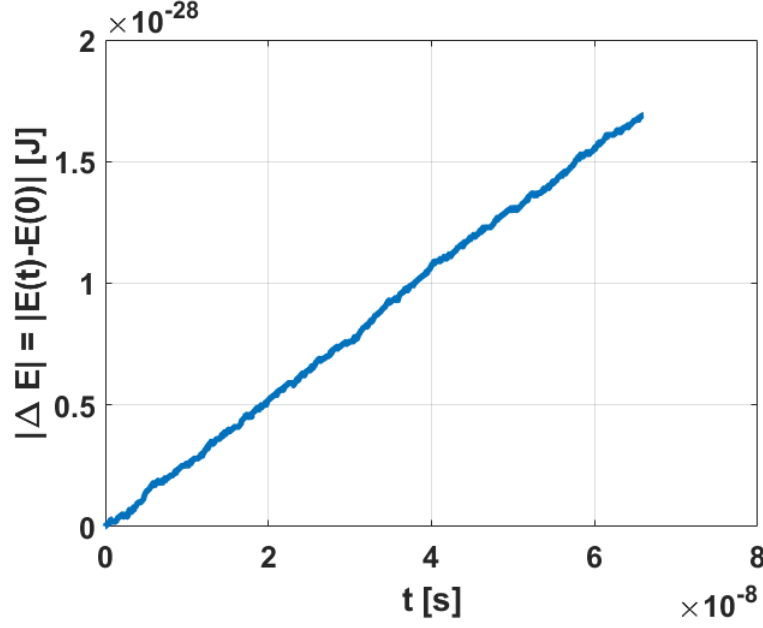


Figura 16: Boris Buneman $t_f = 6,8 \times 10^{-8}$, $N = 10^4$

6.6. Resonancia y dinámica no lineal

En las dos primera secciones, hemos determinado la potencia de los integradores numéricos (convergencia, estabilidad). Sin embargo, en dichos casos, la solución analítica podía ser determinada. En el siguiente caso, para el péndulo, tomaremos $E_0, \kappa \neq 0$. Con estas condiciones, no podemos determinar una solución analítica al problema. Es ahora que evidenciaremos las ventajas de la solución numérica ya que no es posible determinar una solución explícita $\forall t$. Lo primero, es estudiar el fenómeno de resonancia. Después, introduciremos $\theta, \dot{\theta}$ en función del tiempo. Finalmente, presentaremos los diagramas de fase $\{\theta, \dot{\theta}\}$.

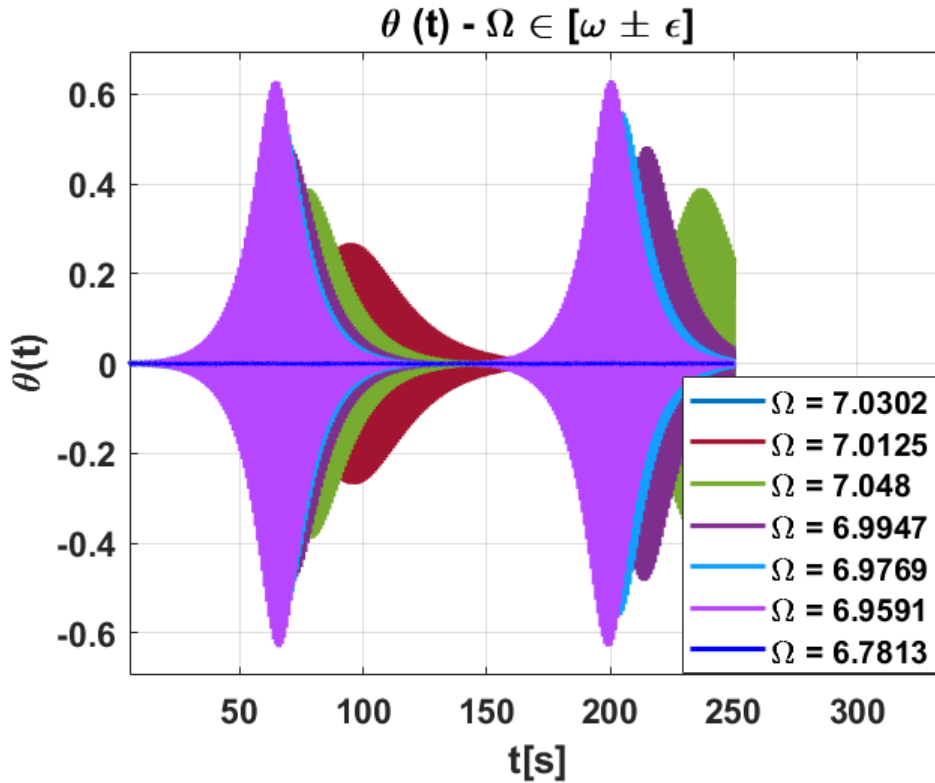


Figura 17: Resonancia $\theta(\Omega)$

En este gráfico, estudiamos θ para $\Omega \in [\omega_0 \pm \delta]$, con $\delta > 0$ y ω_0 la frecuencia de resonancia. Observamos que la máxima amplitud no coincide explícitamente con ω_0 . Esto se explica por el efecto del campo eléctrico. En efecto, la frecuencia de resonancia ω_0 se aplica para pequeños ángulos. Cuando $\theta \not\approx \sin \theta$, esta aproximación ya no es válida, y por ende, la ecuación que nos brinda la frecuencia de resonancia, ya no tiene porque ser ω_0 . Estos resultados no pueden ser determinados por una ecuación explícita.

Ahora, analicemos $\theta(t)$ y $\dot{\theta}(t)$. Consideremos el caso lineal. Tomamos $\theta_0 = \theta - \epsilon$ (posición de equilibrio 2) $\epsilon = 10^{-3}$, $\dot{\theta}(0) = 0$, $\kappa = 0,05$, kg s^{-2} y $t_f = 30\text{s}$. Si $E_0 = 0$, es posible resolver analíticamente, y obtenemos la ecuación diferencial lineal:[2]

$$\ddot{\theta} = -\frac{g}{L} \sin(\theta) - \frac{\kappa}{m} \dot{\theta}$$

La solución general está dada por:

$$\theta(t) = Ae^{-\beta t} \cos(\omega_0 t + \phi)$$

Como el polinomio característico sería $\lambda^2 + 2\beta\lambda + \omega_0^2 = 0$ con $2\beta = \frac{\kappa}{m}$. Las raíces están dadas por $\lambda_j = \frac{-\beta \pm 2\sqrt{\omega_0^2 - \beta^2}}{2}$. Esta solución da una función oscilante (coseno o seno) con decaimiento exponencial. Esto se logra observar en las simulaciones.

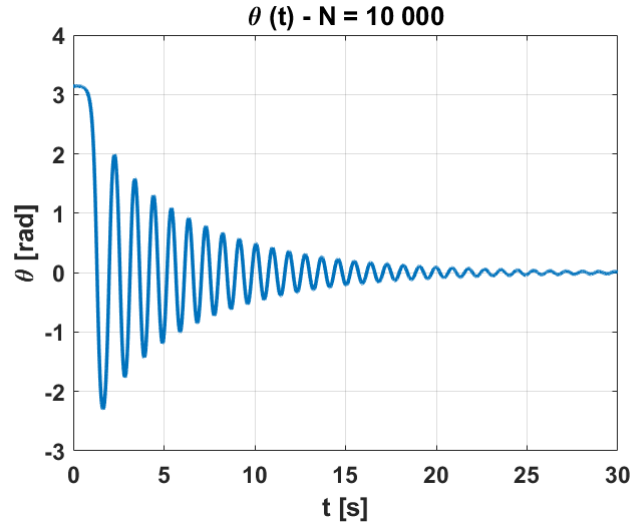


Figura 18: $\theta(t)$, $E_0 = 0$, $\kappa \neq 0$

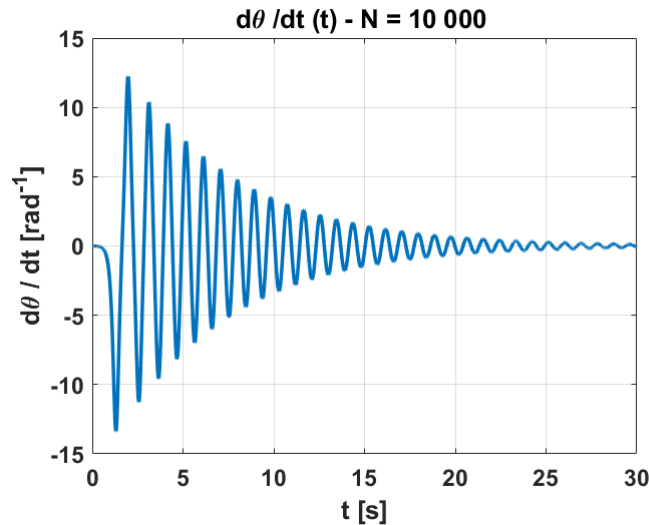


Figura 19: $d\theta(t)/dt$, $E_0 = 0$, $\kappa \neq 0$

En el caso de del diagrama de fases, podemos apreciar la convergencia y la gráfica de un espiral logarítmica:

$$(x, y) \rightarrow (a \cos(t)e^{-bt}, a \sin(t)e^{-bt})$$

Esta curva presenta una geometría similar a la que se puede visualizar en el diagrama de fases y concuerda con la solución más general al péndulo amortiguado.

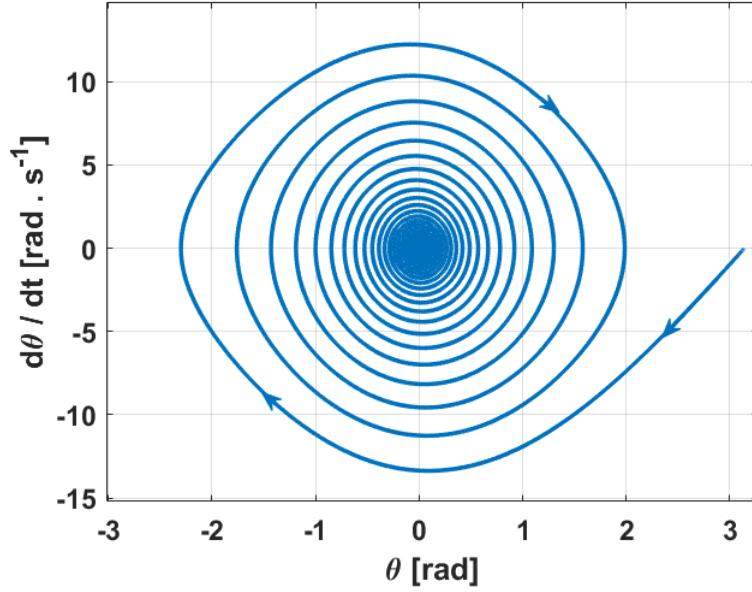


Figura 20: $\{\theta, \dot{\theta}\}$

Si consideramos que $E_0 \neq 0$, tendremos un comportamiento similar, pero con oscilaciones más pronunciadas. Al introducir la fuerza del campo eléctrico, ya no es posible resolver la ecuación diferencial, incluso, para los pequeños ángulos. Por ende, la simulación es más completa. Observamos el mismo comportamiento que el esperado, lo cual ilustra la eficiencia y pertinencia del integrador numérico (en este caso, Stormer Verlet).

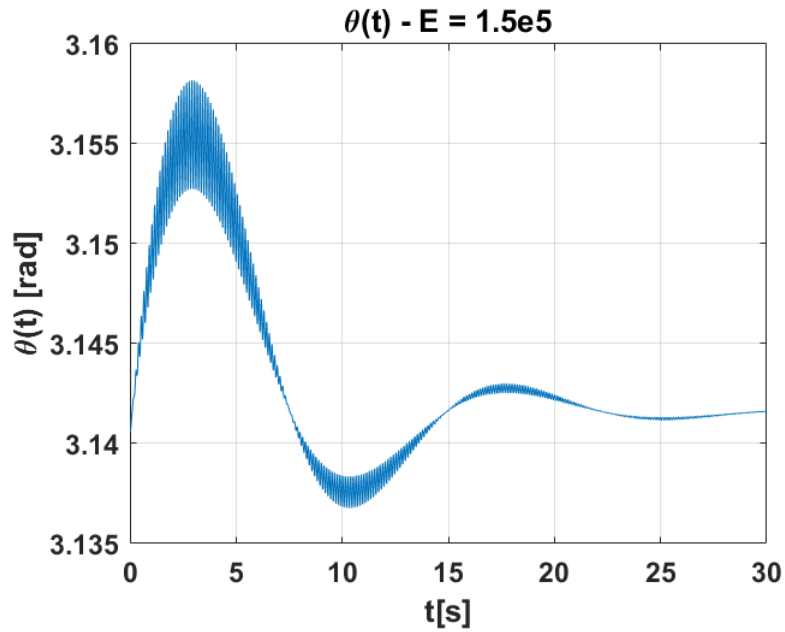


Figura 21: $\theta(t), E_0 \neq 0, \kappa = 0$

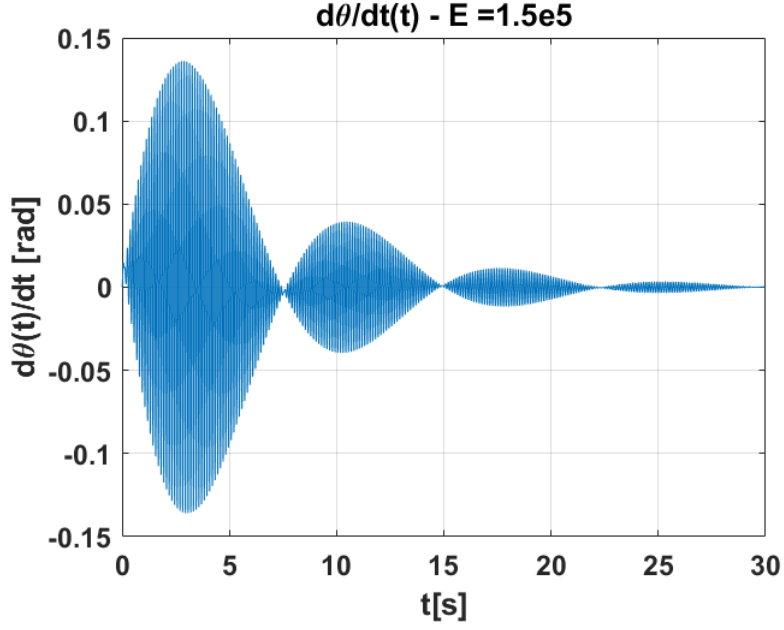


Figura 22: $\dot{\theta}(t)$, $E_0 \neq 0$, $\kappa = 0$

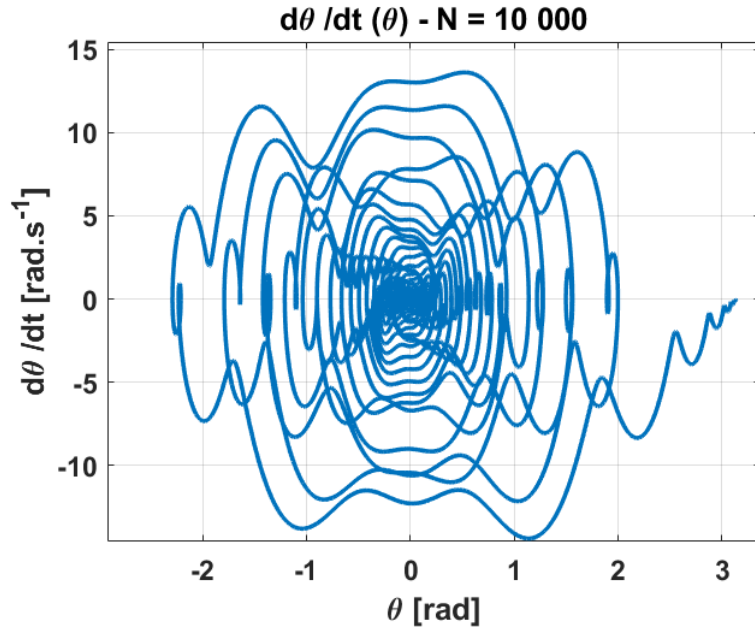


Figura 23: $\{\theta, \dot{\theta}(t)\}$, $E_0 = 10^3$, $\kappa \neq 0$

Apreciamos una convergencia más pronunciada (figura 24) con un campo eléctrico más fuerte. El análisis cualitativo para el caso no lineal es únicamente posible para la solución numérica. Hemos ilustrado entonces, lo importante que es el análisis numérico. En la última sección, enfatizaremos este punto introduciendo brevemente el análisis de sistemas caóticos.

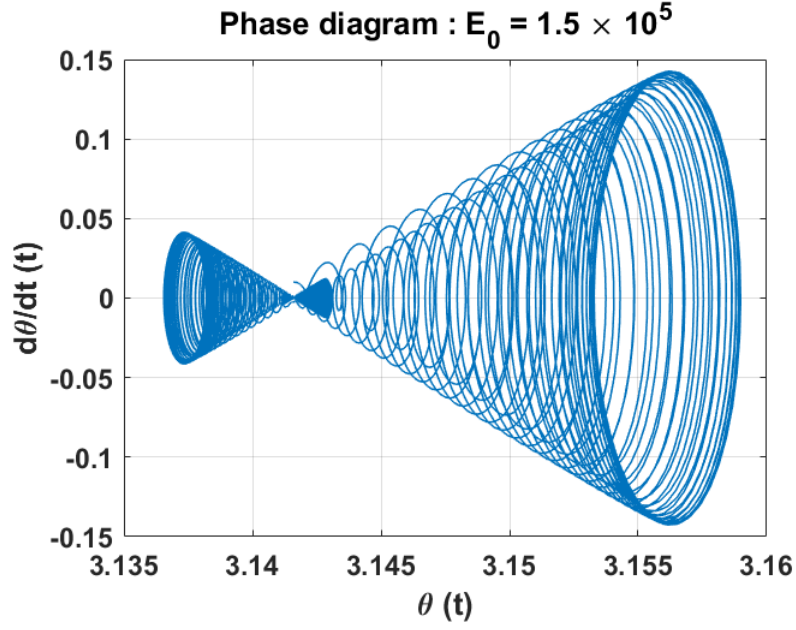


Figura 24: $\{\theta, \dot{\theta}(t)\}, E_0 = 10^5, \kappa \neq 0$

7. Diagramas de Poincare

En el espacio del estado fases, para un sistema dinámico, definimos la transformación de Poincaré, como la restricción de $\{\theta_i, \dot{\theta}_i\}$ para $t_i \in [a, b]$. Se genera por ende una sección transversal del diagrama de fases.[8]

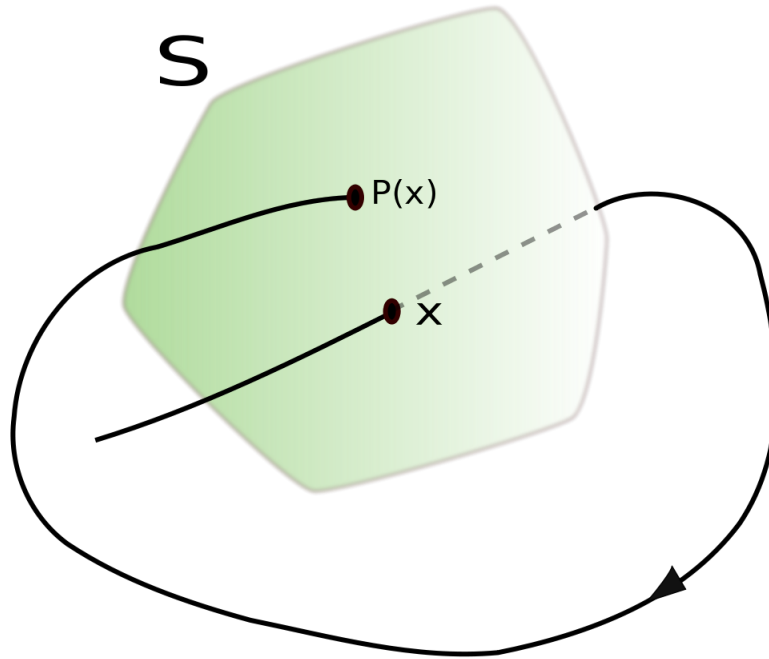


Figura 25: Poincaré map Wikipedia [9]

En la figura 25, S es la sección de Poincare, \mathbf{P} la transformación de Poincaré y \mathbf{x} un punto proyectado por \mathbf{P} hacia S . En nuestro caso (para nuestras simulaciones), definimos:

$$\{\theta(t_k), \dot{\theta}(t_k)\} | t_k = k \frac{2\pi}{\Omega}, k \in [100, 10000]$$

Esto con el propósito de tener un análisis cualitativo de la situación, más coherente (el sistema para el tiempo

cada kT , T el periodo). Defiendo n el número de *stpes* por periodo:

$$\frac{N_{\text{steps}}}{n} \frac{2\pi}{\Omega} = t_f$$

En nuestra simulación, usamos $N = 10^6$, $n = 100$ y $t_f = 8971,402931$. Entonces, para diferentes $\theta(0)$, trazamos la sección de Poincaré. La transformación \mathbf{P} nos muestra que, para:

$$\theta_0 = \{0, \pi/4, \pi/3, \pi/2, 5\pi/8\}$$

las figuras no son caóticas (figura 29). En efecto, apreciamos órbitas bien definidas (la gráfica tiene un orden geométrico). Por otro lado, tomando $\theta_0 = \{\pi, 7\pi/8\}$, obtenemos figuras caóticas (sin ningún orden detrás de $\{\theta(t_i), \dot{\theta}(t_i)\}$). Este análisis muestra la importancia de las condiciones iniciales y su vínculo con el régimen caótico. Sin embargo, nos estamos basando únicamente en la figura obtenida (nótese que esto no constituye prueba alguna, pero nos da una buena primera apreciación del sistema). Para un análisis cuantitativo, usamos las pendientes de Lyapunov, definidas en la siguiente sub-sección.

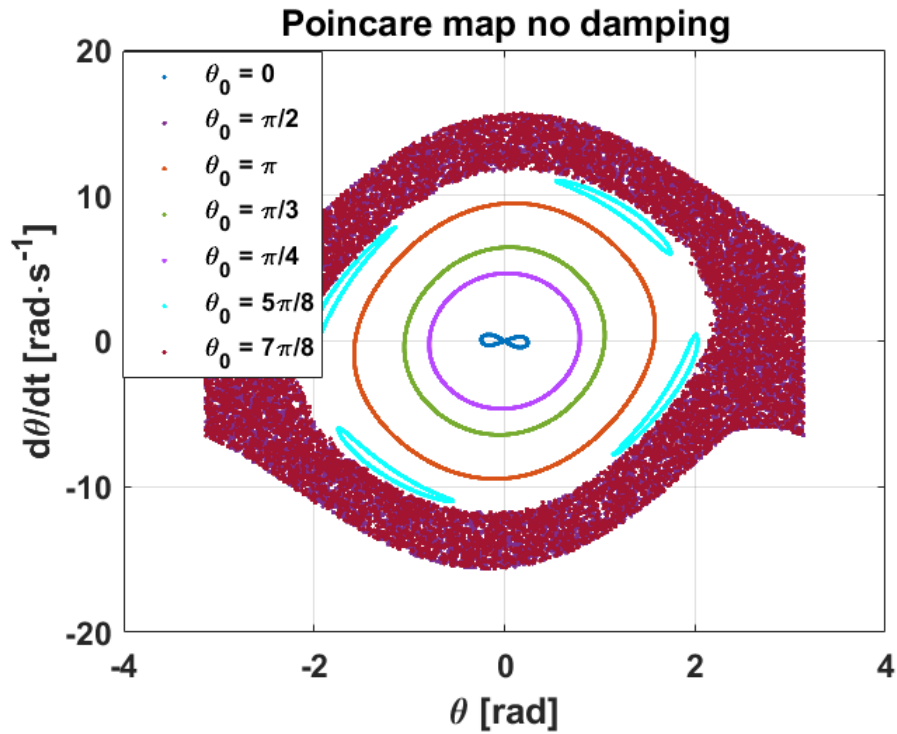


Figura 26: Poincaré map

7.1. Pendiente de Lyapunov

Los diagramas de Poincaré nos permiten analizar cualitativamente la estabilidad de un sistema, revelando los casos caóticos. En el caso de un estudio sin herramientas computacionales, sería imposible obtener la trayectoria del péndulo a lo largo del tiempo. Es por esto que la potencia del cálculo numérico presenta una gran ventaja a la hora de estudiar sistemas dinámicos no lineales. Un análisis más cuantitativo, es estudiar la pendiente de Lyapunov, una métrica que permite medir si un sistema es caótico o no.

$$\|\cdot\|_L = d(t) = \sqrt{(\dot{\theta}_1(t) - \dot{\theta}_2(t))^2 + \Omega^2(\theta_1(t) - \theta_2(t))^2}$$

Aquí θ_2 y θ_1 representan la trayectoria de un péndulo para una diferencia en las posiciones iniciales $\theta_2(0)$ y $\theta_1(0)$ del orden de $10^{-8}[m]$. Si graficamos $\log(d(t))$, y obtenemos una aproximación lineal, el sistema es caótico. Si resulta ser logaritmo o una constante, el sistema no es caótico. [8]

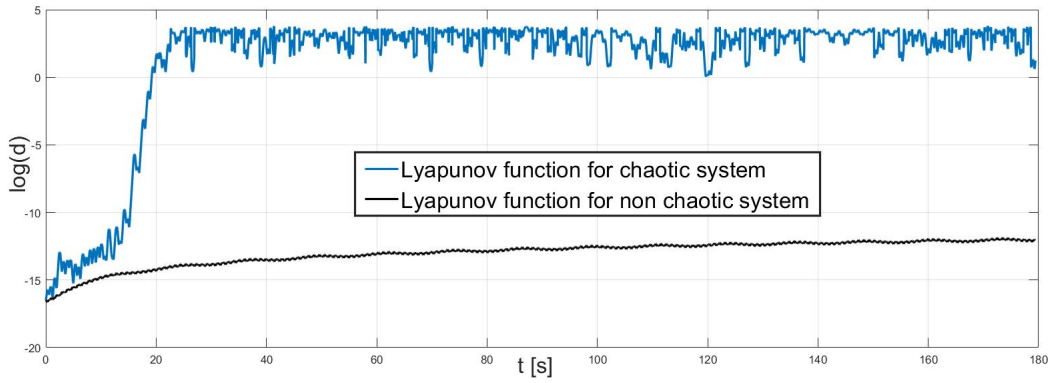


Figura 27: Estable e inestable $\log(d(t))$

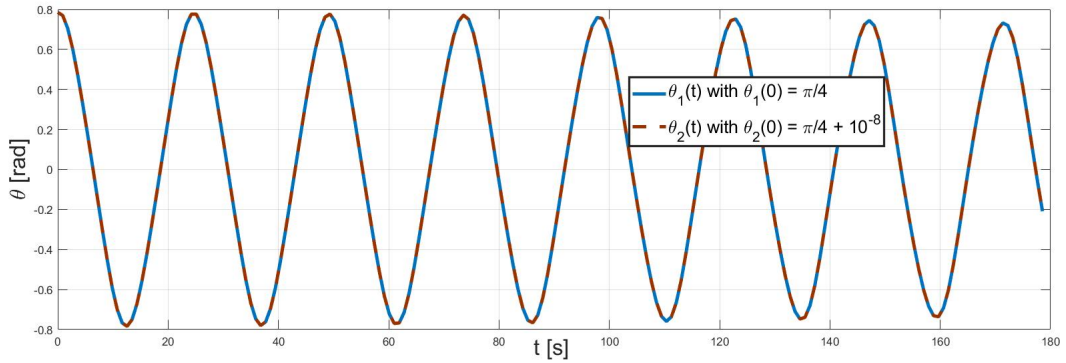


Figura 28: Estable $\theta(t)$

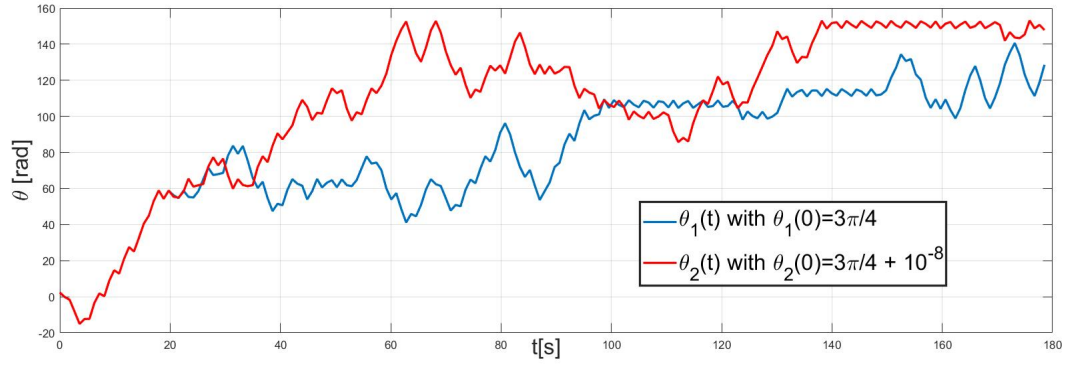


Figura 29: Inestable $\theta(t)$

El objetivo del estudio de las pendientes de Lyapunov es mostrar la alta sensibilidad a las condiciones iniciales en un sistema dinámico. En efecto, para condiciones iniciales en un intervalo $[\theta_0 \pm \delta]$, con δ tan chico como queremos, si nos encontramos en un caso estable, se tardará más tiempo en observar una diferencia en la trayectorias adyacentes. Sin embargo, para un caso inestable, por más pequeño que sea δ , para un tiempo corto, se empezará a notar la diferencia en las trayectorias.

8. Conclusiones

Nuestro objetivo era ilustrar las ventajas y facilidades que nos brinda el cálculo computacional versus la resolución tradicional de ecuaciones diferenciales. En efecto, estas resoluciones se limitan al caso lineal en su mayor parte. Sin embargo, como se pudo ver a través de los ejemplos estudiados, el problema lineal no es el más adecuado para describir los fenómenos de la naturaleza. Es por ello que se recurre al poder de la rama computacional. Los parámetros que hemos considerado para evaluar la optimalidad del método computacional, son **la convergencia numérica, estabilidad y sensibilidad de las condiciones iniciales**. Mediante el estudio de los diagramas de Poincare se lograron establecer los límites de las simulaciones, dando paso a movimientos caóticos.

Con esto, se espera haber demostrado la **importancia, ventajas e intereses** de incorporar material educativo relativo a la física computacional, en la formación de los Estudios Generales Ciencias. Esto brindaría ventajas comparativas a los alumnos, en un mundo que avanza día a día a una gran velocidad en la creación de las nuevas herramientas digitales y de manejo de data. La convergencia numérica nos asegura que los métodos numéricos logran aproximarse adecuadamente al modelo analítico. La estabilidad temporal permite que el fenómeno pueda ser estudiado en diferentes escalas de tiempo. Finalmente, la introducción al análisis de sistemas no dinámicos muestra que las soluciones numéricas pueden brindar un análisis cualitativo de fenómenos que no pueden ser descritos mediante ecuaciones. Sin embargo, nos muestra también que es imposible (de momento), resolver ciertos problemas $\forall t$.

9. Anexos

Los anexos nos brindan información complementaria al trabajo. Resultados interesantes son tratados. Estos últimos constituyen los fundamentos del análisis de estabilidad (von Newman), la resolución a la ecuación diferencial del péndulo (sin aproximaciones, sin fricción y campo eléctrico), así como los detalles de la implementación del integrador numérico Stormer-Verlet.

9.1. Análisis de von Newman

Brevemente, presentamos el análisis de estabilidad de von Newmann's **Euler explícito**. Sea \mathbf{y}_n el valor de $\mathbf{y} = \begin{pmatrix} x \\ v \end{pmatrix}$ en $t = t_n$ y \mathbf{e}_n el error del método de aproximación de Euler. Entonces $\mathbf{y}_n = \mathbf{y}(t_n) + \mathbf{e}_n$. Nos gustaría determinar el error en t_{n+1} . Usando el método de Euler: [8]

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \mathbf{f}(\mathbf{y}_n)\Delta t$$

En $\mathbf{y}_n = \mathbf{y}(t_n) + \mathbf{e}_n$, entonces:

$$\mathbf{y}_{n+1} = \mathbf{y}(t_n) + \mathbf{e}_n + \mathbf{f}(\mathbf{y}(t_n) + \mathbf{e}_n)\Delta t$$

Por la fórmula de Taylor $\mathbf{f}(\mathbf{y}(t_n) + \mathbf{e}_n)\Delta t \sim \mathbf{f}(\mathbf{y}(t_n)) + \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \mathbf{e}_n$:

$$\mathbf{y}_{n+1} \sim \mathbf{y}(t_n) + \mathbf{e}_n + \left[\mathbf{f}(\mathbf{y}(t_n)) + \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \mathbf{e}_n \right] \Delta t$$

Finalmente, remplazando, $\mathbf{y}_{n+1} = \mathbf{y}(t_{n+1}) + \mathbf{e}_{n+1} \sim \mathbf{y}(t_n) + \mathbf{f}(\mathbf{y}(t_n))\Delta t$, tomamos:

$$\boxed{\mathbf{e}_{n+1} = \left(\mathbf{I} + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right) \mathbf{e}_n} \quad (36)$$

Con \mathbf{I} la matriz identidad, $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^n$. Entonces, $\frac{\partial \mathbf{f}}{\partial \mathbf{y}} \equiv J_{\mathbf{f}}(\mathbf{y})$. Llamamos $\mathbf{G} = \left(\mathbf{I} + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right)$ la matriz de evolución. Usando las propiedades de los endomorfismos (álgebra lineal), si $\exists \lambda_i$, valores propios de \mathbf{G} , tal que $|\lambda_i| > 1$, el integrador de Euler es inestable.

En nuestro modelo, tenemos un sistema de ecuaciones diferenciales de orden 2 (partícula en campo electromagnético):

$$\frac{d}{dt} \mathbf{y} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_0 \\ 0 & 0 & \omega_0 & 0 \end{pmatrix} \begin{pmatrix} x \\ v \\ v_x \\ v_z \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{qE_0}{m} \end{pmatrix}$$

Calculamos $\frac{\partial \mathbf{f}}{\partial \mathbf{y}} = J_{\mathbf{f}}$:

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_0 \\ 0 & 0 & \omega_0 & 0 \end{pmatrix}$$

Por ende, para el método de Euler explícito, la matriz de evolución está dada por:

$$\mathbf{G} = \left(\mathbf{I}_4 + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & -\omega_0 \Delta t \\ 0 & 0 & \omega_0 \Delta t & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & -\omega_0 \Delta t \\ 0 & 0 & \omega_0 \Delta t & 1 \end{pmatrix}$$

Calculamos el polinomio característico para obtener $|\lambda_i|$:

$$\chi_G(t) = \begin{vmatrix} t-1 & 0 & -\Delta t & 0 \\ 0 & t-1 & 0 & -\Delta t \\ 0 & 0 & t-1 & \omega_0 \Delta t \\ 0 & 0 & -\omega_0 \Delta t & t-1 \end{vmatrix} = (t-1)^4 - (t-1)^2 (\omega_0 \Delta t)^2 = (t-1)^2 (t^2 - 2t + 1 - \omega_0^2 \Delta t^2)$$

Las raíces de $\chi_G(t)$ son $\lambda = 1$ y :

$$\lambda_{2,3} = \frac{2 \pm \sqrt{4 - 4(1 - \omega_0^2 \Delta t^2)}}{2} = 1 \pm \omega_0 \Delta t$$

Entonces, existen autovalores $|\lambda| > 1$ ($1 + \omega_0 \Delta t$) y el método es inestable (por eso la energía mecánica diverge). Por otro lado, para el método del integrador implícito de Euler, la matriz de evolución es:

$$\left(\mathbf{I}_4 - \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right)^{-1} = \begin{pmatrix} 1 & 0 & -\Delta t & 0 \\ 0 & 1 & 0 & -\Delta t \\ 0 & 0 & 1 & \omega_0 \Delta t \\ 0 & 0 & -\omega_0 \Delta t & 1 \end{pmatrix}^{-1} =$$

$$\frac{1}{1 + \omega_0^2 \Delta t^2} \cdot \begin{pmatrix} 1 + \omega_0^2 \Delta t^2 & 0 & \Delta t & -\omega_0 \Delta t \\ 0 & 1 + \omega_0^2 \Delta t^2 & \omega_0 \Delta t^2 & \Delta t \\ 0 & 0 & 1 & -\omega_0 \Delta t \\ 0 & 0 & \omega_0 \Delta t & 1 \end{pmatrix}$$

Calculando el polinomio característico, obtenemos los autovalores λ :

$$\chi(\lambda) = \left[\lambda - \left(\frac{1 + \omega_0^2 \Delta t^2}{1 + \omega_0^2 \Delta t^2} \right) \right]^2 \cdot \left[\left(\lambda - \frac{1}{1 + \omega_0^2 \Delta t^2} \right)^2 + \frac{\omega_0^2 \Delta t^2}{(1 + \omega_0^2 \Delta t^2)^2} \right]$$

Finalmente

$$\lambda = 1 \quad \text{or} \quad \lambda = \pm \frac{i \cdot (\omega_0^2 \Delta t^2) + 1}{1 + \omega_0^2 \Delta t^2}$$

Como el módulo de los autovalores son inferiores a 1, concluimos que el método de Euler implícito es estable $\forall \Delta t$.

9.2. Stormer-Verlet

Hemos usado el esquema Stormer-Verlet para la simulación del péndulo. Este integrador numérico incorpora la velocidad en la ecuación de integración numérica. Tomando $\mathbf{a}(\mathbf{x}, \mathbf{v}, t) = \frac{\mathbf{F}(\mathbf{x}, \mathbf{v}, t)}{m} \equiv \ddot{\theta}(\theta, \dot{\theta}, t) = \frac{\mathbf{F}(\theta, \dot{\theta}, t)}{m}$. [6]

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \mathbf{v}_j \Delta t + \mathbf{a}(\mathbf{x}_j, \mathbf{v}_j, t_j) \frac{\Delta t^2}{2}$$

$$\mathbf{v}_{j+1/2} = \mathbf{v}_j + \mathbf{a}(\mathbf{x}_j, \mathbf{v}_j, t_j) \frac{\Delta t}{2}$$

$$\mathbf{v}_{j+1} = \mathbf{v}_j + (\mathbf{a}_1(\mathbf{x}_j, t_j) + \mathbf{a}_1(\mathbf{x}_{j+1}, t_{j+1})) \frac{\Delta t}{2} + \mathbf{a}_2(\mathbf{v}_{j+1/2}) \Delta t$$

9.3. Solución no lineal de un péndulo

Supongamos que el movimiento del péndulo se describe por una ecuación diferencial no lineal del segundo orden: [5]

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\theta = 0$$

Tomamos como condiciones iniciales $\theta(0) = \theta_0$ y $\dot{\theta}(0) = 0$. Es posible reducir el orden de la ecuación diferencial multiplicando por $\frac{d\theta}{dt}$. Esto nos lleva a la ecuación:

$$\frac{d\theta}{dt} \left(\frac{d^2\theta}{dt^2} + \frac{g}{L} \right) = \ddot{\theta}\dot{\theta} + \frac{g}{L} \sin(\theta)\dot{\theta}$$

Usando $\varphi(\gamma(t))' = \gamma'(t) \cdot \varphi'(\gamma(t))$, factorizamos:

$$\frac{d}{dt} \left[\frac{1}{2} \left(\frac{d\theta}{dt} \right)^2 - \frac{g}{L} \cos(\theta) \right] = 0$$

Integrando en ambos lados

$$\int \frac{d}{dt} \left[\frac{1}{2} \left(\frac{d\theta}{dt} \right)^2 - \frac{g}{L} \cos(\theta) \right] dt = C$$

Usando el teorema fundamental del cálculo, simplificamos

$$\frac{1}{2} \left(\frac{d\theta}{dt} \right)^2 - \frac{g}{L} \cos(\theta) = C$$

Dadas las condiciones iniciales $C = -2g/L \cos(\theta_0)$:

$$\left(\frac{d\theta}{dt} \right)^2 = \frac{2g}{L} (\cos(\theta) - \cos(\theta_0))$$

Usando $\cos(\theta) = 1 - 2 \sin^2(\theta/2)$:

$$\left(\frac{d\theta}{dt} \right)^2 = \frac{2g}{L} (1 - 2 \sin^2(\theta/2) - \cos(\theta_0)) = \frac{4g}{L} (\sin^2(\theta_0/2) - \sin^2(\theta/2))$$

Tomando la raíz cuadrada:

$$\frac{d\theta}{dt} = 2\sqrt{\frac{g}{L}} \sqrt{\sin^2(\theta_0/2) - \sin^2(\theta/2)}$$

Aplicando separación de variables:

$$\int \frac{d\theta/2}{\sqrt{\sin^2(\theta_0/2) - \sin^2(\theta/2)}} = \int \sqrt{\frac{g}{L}} dt$$

Llamamos $k = \sin^2(\theta_0/2)$ obtenemos:

$$\sin \frac{\theta}{2} = k \sin(\phi)$$

Por ende:

$$d\left(\sin \frac{\theta}{2}\right) = \cos \frac{\theta}{2} d\left(\frac{\theta}{2}\right) = \sqrt{1 - \sin^2 \frac{\theta}{2}} d\left(\sin \frac{\theta}{2}\right) = \sqrt{1 - k \sin^2(\phi)} d\left(\frac{\theta}{2}\right) = k \cos(\theta) d\theta$$

Así:

$$d\left(\frac{\theta}{2}\right) = \frac{k \cos(\theta) d\theta}{\sqrt{1 - k \sin^2(\phi)}}$$

Usando la nueva variables ϕ :

$$\int \frac{k \cos(\phi) d\phi}{\sqrt{1 - k \sin^2(\phi)} k \cos(\phi)} = \int \frac{d\phi}{\sqrt{1 - k \sin^2(\phi)}} = \sqrt{\frac{g}{L}} \int dt$$

En $\theta = \pi/2$ tenemos $t = \frac{T}{4}$. Por ende:

$$\sqrt{\frac{g}{L}} \int_0^{T/4} dt = \int_0^{\pi/2} \frac{d\phi}{\sqrt{1 - k \sin^2(\phi)}}$$

Esta integral no puede ser expresada como una combinación lineal o composición de funciones estándares. Sin embargo, es posible dar la solución en series de potencias:

$$K(k) = \int_0^{\pi/2} \frac{d\phi}{\sqrt{1 - k \sin^2(\phi)}} = \frac{\pi}{2} \sum_{n=0}^{\infty} \left(\frac{(2n)!}{2^{2n}(n!)^2} \right) k^{2n}$$

Finalmente:

$$T = 2\pi \sqrt{\frac{g}{L}} \sum_{n=0}^{\infty} \left(\frac{(2n)!}{2^{2n}(n!)^2} \right) k^{2n}$$

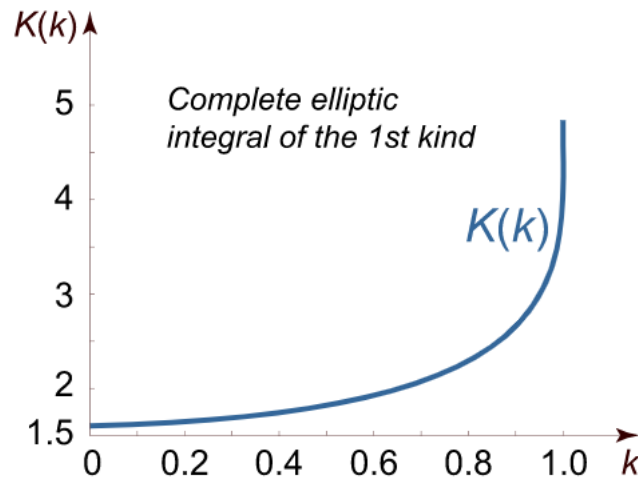


Figura 30: First-Order Elliptic Integral

Referencias

- [1] CHAPPELLIER JEAN-CÉDRIC , *Programación Orientada a Objetos MOOC*, EPFL 2015
- [2] CHÁVEZ RICHARD, *Sistemas dinámicos*, PUCP 2020
- [3] ERNST HAIRER, CHRISTIAN LUBICH, *Energy behaviour of the Boris method for charged-particle dynamics*, UNIVERSITÉ DE GENÈVE <http://www.unige.ch/~hairer/preprints/borisbea.pdf>
- [4] GAGO ALBERTO, *Investigación científica*, 2017 <https://www.youtube.com/watch?v=B0iyNGs2Ya8>
- [5] MATLAB, *Integrales elípticas*, <http://www.sc.ehu.es/sbweb/fisica3/especial/eliptica/eliptica.html>
- [6] RICCI PAOLO, GUINCHARD SALOMON, ROUZAIER LILIAN, *Polycopié Physique Avancée I*, EPFL 2019
- [7] LOMELÍ HÉCTOR, RUMBOS BEATRIZ , *Métodos dinámicos en economía*, ITAM
- [8] Villard Laurent, Lauchli Andreas *Physique Numérique Polycopié* PUCP 2020
- [9] WIKIPEDIA, *Poincare map*, https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_de_Poincar%C3%A9