75.12 - Análisis Numérico 1

Trabajo práctico N°2

Integración numérica y problemas de valor inicial no lineales

Nombre	Correo electrónico	Padrón
Gonzalo Ávila Alterach	gonzaloavilaalterach@gmail.com	94950
Nicolás Mariano Fernandez Lema	nicolasfernandezlema@gmail.com	93410

Fecha de entrega: 27 de noviembre
2° cuatrimestre de 2013

Resumen

Se tiene un péndulo ideal, con masa puntual, sin ningún tipo de rozamiento ni pérdida de energía cuyo modelo matemático puede representarse mediante la siguiente ecuación diferencial.
$\frac{d^2\phi}{dt^2} + \frac{g}{L}\sin(\phi) = 0$
Se desea obtener su evolución temporal, teniendo en cuenta las condiciones iniciales $\phi(0) = \frac{737}{1000}\pi = A$ y $\frac{d\phi}{dt}(0) = 0$, y que $L = 1$ y $g = 9,81$. Es decir, al péndulo se lo suelta con velocidad inicial nula y con un ángulo inicial de 132.7° respecto de la vertical.

Cálculo del período mediante integración numérica

El movimiento del péndulo analizado es periódico, y su período puede calcularse mediante la siguiente integral definida:

$$T = \int_0^{\pi/2} 4\sqrt{L/G}(1 - sen^2(A/2)sen^2(\varphi))^{-0,5}d\varphi$$

El valor calculado de dicho período es de $T = 3,0087 \pm 0,00005$ segundos.

Para calcular dicha integral se usó el método *trapecio compuesto*, y para lograr 5 dígitos significativos se utilizó un paso de $\frac{\pi}{700}$.
Para obtener el valor necesario de dicho paso se utilizó la fórmula que da una cota del error global en trapecio:

$$E_g \leq \frac{(b-a)}{12}h^2\frac{d^2f}{d\varphi^2}(\xi) \leq \frac{\pi}{24}\frac{\pi^2}{700^2}17 \simeq 0{,}000044822 \leq 0,5 \cdot 10^{-4}$$

Siendo 17 una cota superior del módulo de la derivada segunda de la función a integrar. Dicha derivada segunda fue hallada utilizando WolframAlpha, y es la siguiente:

$$\frac{d^2f}{dx^2} = 1{,}2771((cos((263\pi)/2000)^2cos(x)^2)/(1 - cos((263\pi)/2000)^2sin(x)^2)^{3/2} -$$

$$(cos((263\pi)/2000)^2sin(x)^2)/(1 - cos((263\pi)/2000)^2 * sin(x)^2)^{3/2} +$$

$$(3cos((263\pi)/2000)^4sin(x)^2cos(x)^2)/(1 - cos((263\pi)/2000)^2sin(x)^2)^{5/2}))$$

Su grafico se puede observar en la Figura 1.
Con un análisis grafico se puede apreciar que dentro del intevalo $[0; \pi/2]$, el módulo de la función no toma valores mayores que 17, valor que fue utilizado como cota.

Verificación de la integral para períodos pequeños

Cuando la amplitud inicial es menor a 10 o 15 grados, es posible aproximar la integral como $2\pi\sqrt{\frac{L}{g}}$. Para verificar la correcta implementación de la integral, se calculó la misma para una amplitud de 0,01 radianes, que dio como resultado 2,0061, idéntico a lo esperado utilizando la aproximación mencionada.

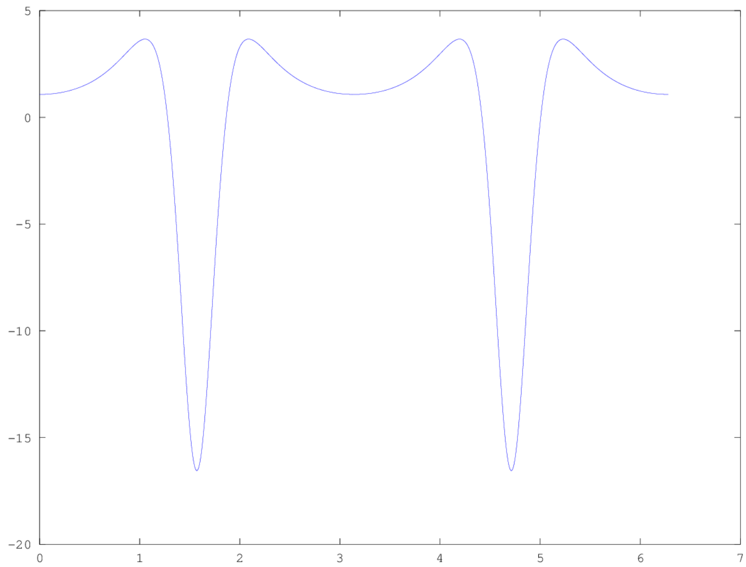

Figura 1: Derivada segunda de $4\sqrt{L/G}(1 - sen^2(A/2)sen^2(\varphi))^{-0,5}$

Integración del problema de valores iniciales

Para calcular la solución numérica al problema de tener datos del ángulo en función del tiempo, se implementó un programa que utiliza los métodos de Euler, Runge-Kutta de orden 2 y Runge-Kutta de orden 4 para integrar el problema a valores iniciales.

Debido a que el problema originalmente es de orden 2, es necesario hacer un cambio de variables para poder utilizar los métodos mencionados, que solamente sirven para ecuaciones de orden 1. Finalmente, queda un sistema de dos ecuaciones diferenciales de primer orden:

$$\phi' = f_1(\phi, \omega, t) = \omega$$

$$\omega' = f_2(\phi, \omega, t) = -\frac{g}{L} sen(\phi)$$

Dichas funciones se mantuvieron separadas en el programa (no se hizo el reemplazo en las llamadas) para mantener el programa más genérico e independiente del sistema a resolver. Como paso se utilizaron fracciones enteras del período calculado con la integral anterior.

Como era de esperar, la solución calculada numéricamente cumple con lo esperado: es periódica de período aproximadamente 3, y se parece a una senoidal (aunque no lo es, ya que la solución exacta involucra funciones elípticas).

Análisis de error en amplitud

Para analizar el error en amplitud de cáda método, se midieron las amplitudes ϕ cuando el tiempo pasaba por múltiplos del período esperado. Además, sabiendo que el problema es del tipo conservativo, por la ausencia de elementos que disminuyan la energía del sistema, se calculó la energía inicial y final, de la siguiente forma:

$E_{mec} = E_{pot} + E_{cin} = gL(1 - cos(\phi)) + \frac{1}{2}(\omega L)^2$

Por ejemplo, para 300 pasos por período se obtuvieron los siguientes resultados:

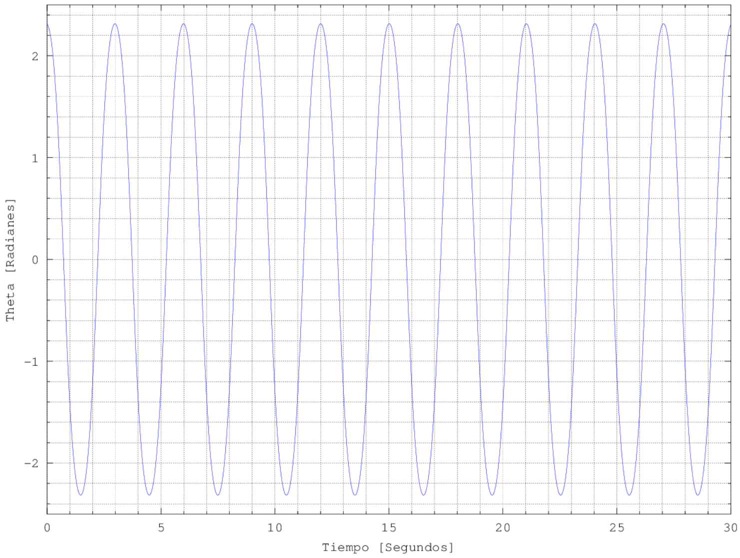

Figura 2: Resultado de integración de la ecuación diferencial con 150 pasos por período, con RK4

Nº Período	ϕ esperado	Amplitudes medidas					
		Euler	$E_r[\%]$	RK2	$E_r[\%]$	RK4	$E_r[\%]$
0	2,3153538	2,3153538	0	2,3153538	0	2,3153538	0
1	2,3153538	2,3153507	-0,0001	2,3154141	0,003	2,3153538	$-2 \cdot 10^{-7}$
2	2,3153538	2,3153476	-0,0003	2,3154741	0,005	2,3153538	$-4 \cdot 10^{-7}$
3	2,3153538	2,3153445	-0,0004	2,3155334	0,007	2,3153538	$-6 \cdot 10^{-7}$
4	2,3153538	2,3153413	-0,0005	2,3155916	0,010	2,3153538	$-8 \cdot 10^{-7}$
5	2,3153538	2,3153380	-0,0007	2,3156479	0,012	2,3153538	$-10 \cdot 10^{-7}$
6	2,3153538	2,3153347	-0,0008	2,3157015	0,015	2,3153538	$-14 \cdot 10^{-7}$
7	2,3153538	2,3153313	-0,0010	2,3157516	0,017	2,3153538	$-13 \cdot 10^{-7}$
8	2,3153538	2,3153279	-0,0011	2,3157970	0,019	2,3153538	$-15 \cdot 10^{-7}$
9	2,3153538	2,3153244	-0,0013	2,3158365	0,021	2,3153537	$-17 \cdot 10^{-7}$

Cuadro 1: Análisis de error en amplitud

E. inicial	E. Final Euler	E_r	E.Final RK2	E_r	E. Final RK4	E_r
16,457712	16,457493	-0,001 %	16,462068	0,026 %	16,457711	$6 \cdot 10^{-6}$ %

Cuadro 2: Análisis de error en energias

Se pueden apreciar varias tendencias:

- Ninguno de los métodos cumplió con la conservación de la energía.
- El método de Euler pierde energía, aunque puede ser considerada despreciable en los 10 períodos medidos.
- RK2 gana energia más rápido de lo que Euler pierde.
- RK4 tiene una pérdida de energía mínima, mucho más chica que la ganancia y pérdida de los otros métodos.

- El cálculo del período parece estar correcto, ya que el mejor método tuvo un error muy bajo, algo que seguramente no hubiera pasado si se hubiera muestrado a la amplitud ϕ en momentos de tiempo incorrectos.

Análisis de error en fase

Para analizar este error, se analizó para qué tiempo el ω cambiaba de signo positivo a negativo, es decir, para qué tiempo el péndulo vuelve a su máxima amplitud. Para obtener resultados más significativos, se redujo la cantidad de pasos a 100 por período.

Nº Período	T esperado	T medidas					
		Euler	$E_r[\%]$	RK2	$E_r[\%]$	RK4	$E_r[\%]$
0	0	0	0	0	0	0	0
1	3,0087	3,0087134	≈ 0	3,0087134	≈ 0	3,0087134	≈ 0
2	6,0174	6,0174267	≈ 0	6,0174267	≈ 0	5,9873396	-0,50
3	9,0261	9,0261401	≈ 0	9,0261401	≈ 0	8,9960529	-0,33
4	12,0348	12,034853	≈ 0	12,034853	≈ 0	12,004766	-0,25
5	15,0435	15,043567	≈ 0	15,073654	0,20	15,01348	-0,20
6	18,0522	18,05228	≈ 0	18,082367	0,17	18,022193	-0,17
7	21,0609	21,060994	≈ 0	21,091081	0,14	21,030906	-0,14
8	24,0696	24,069707	≈ 0	24,129881	0,25	24,03962	-0,13
9	27,0783	27,07842	≈ 0	27,168682	0,33	27,048333	-0,11

Cuadro 3: Análisis del error de fase

Se puede apreciar lo siguiente:

- El método de Euler parece no tener error de fase alguno, o tener mucho menor que los otros dos métodos.
- RK2 comienza correctamente y luego se va enlenteciendo, sus máximos llegan después de lo esperado.
- RK4 solamente se mantiene sin error de fase durante el primer período, luego comienza a acelerarse, y sus máximos vienen antes de lo esperado.

Por lo general, parece que los métodos que mejor conservan las amplitudes tienen más error de fase, y viceversa, parece que a la hora de elegir un método hay que tener en cuenta si es preferible tener menos error temporal o menos error en amplitud.

Análisis de estabilidad

Para este análisis, se disminuyó la cantidad de pasos por período hasta lograr que los métodos se alejen de lo esperado, de forma tanto divergente como convergente. Además, se aumentó la cantidad de períodos integrados, para ver más información.

Con 50 pasos por período, a partir de los 110 segundos (36 períodos) aproximadamente RK2 comienza a diverger positivamente, mientras que los otros métodos parecen seguir correctos incluso luego de 100 perídoos

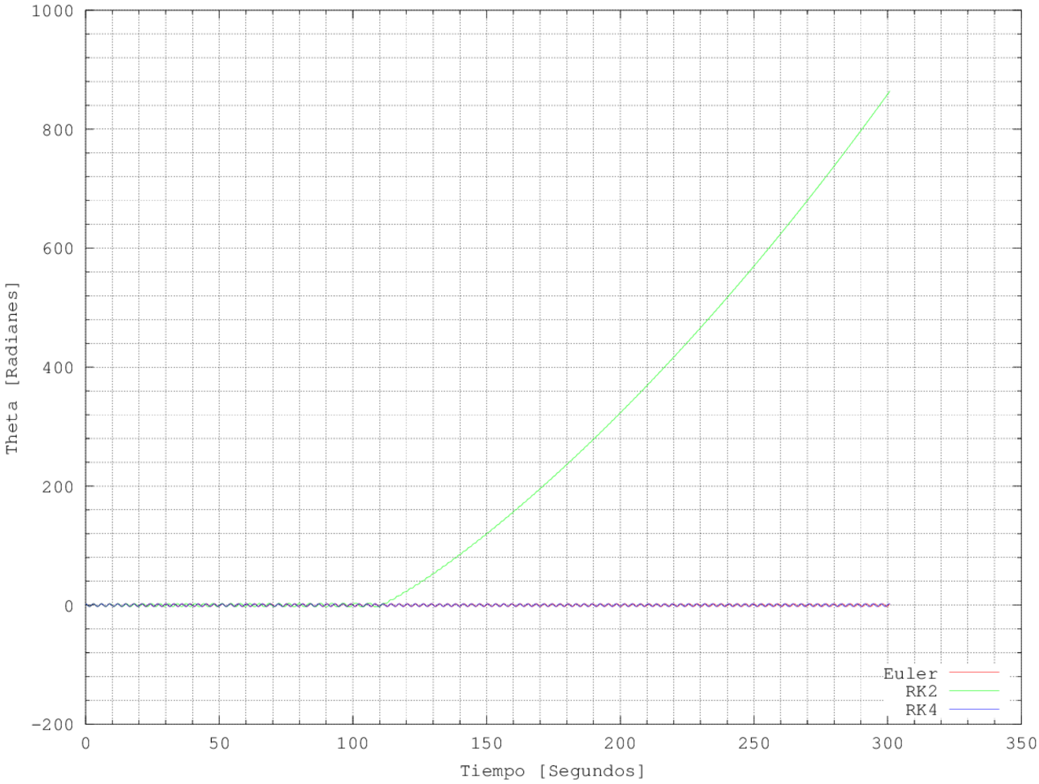

Figura 3: Soluciones para 50 pasos por período

A <8 pasos por período, se puede apreciar que RK4 perdió casi toda su energia cuando llegó de los 150 segundos (50 períodos), mientras que Euler se sigue mantienendo sin mucha pérdida:

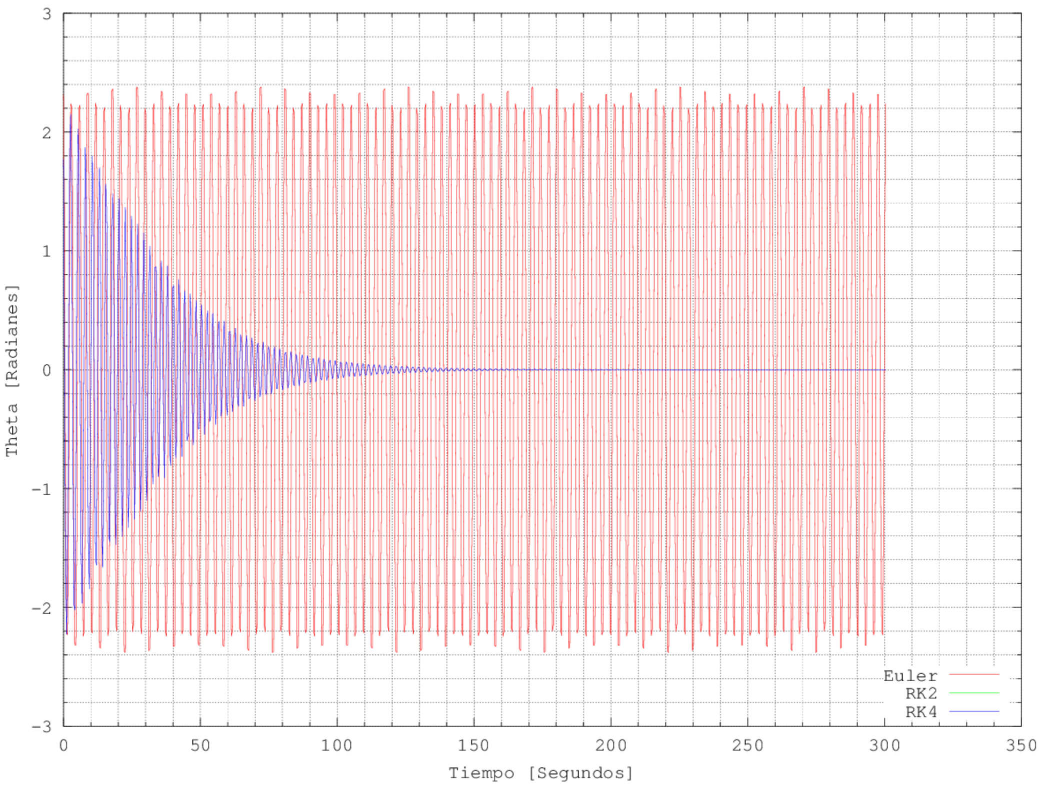

Figura 4: Soluciones para 8 pasos por período

A <7 pasos por período, sucede algo extraño: Euler se va del rango correcto antes de que termine el primer período, pero después se acomoda (suponemos debido a la presencia del seno en la ecuación diferencial) y comienza a oscilar alrededor de otra amplitud, que es la amplitud deseada menos 10π:

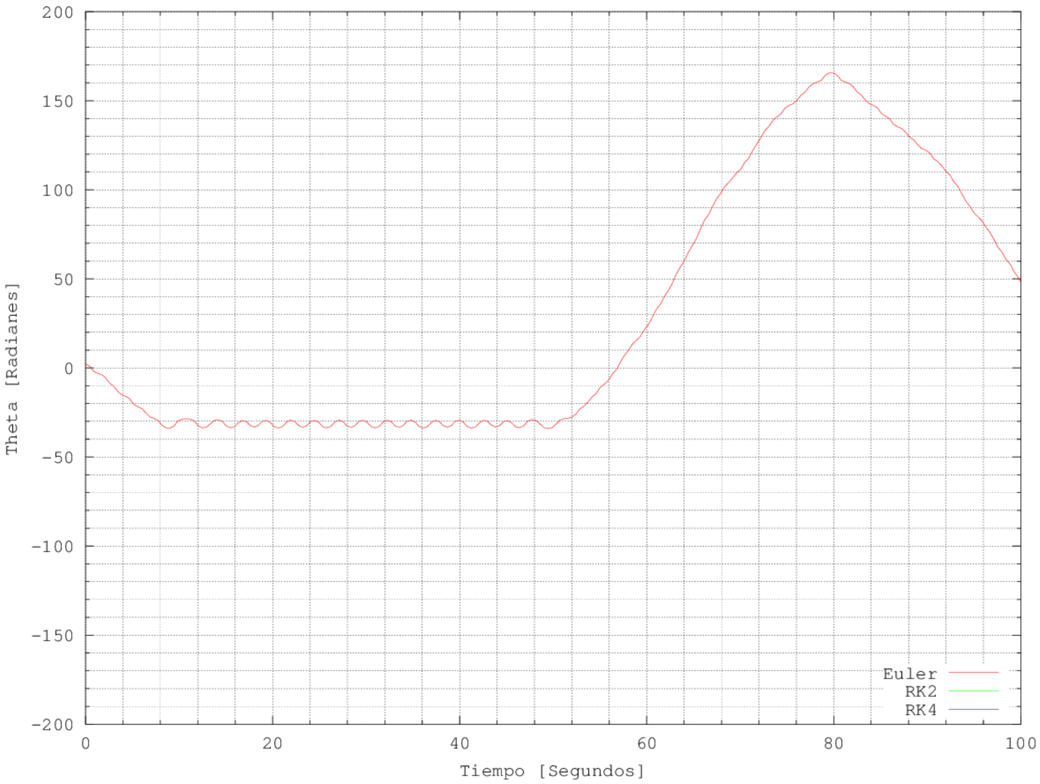

Figura 5: Solución para 7 pasos por período

Conclusiones

- Como primera conclusión, es posible ver que *ninguno* de los métodos utilizados para resolver el problema numérico conservó la energía, ya que son métodos generales no pensados específicamente para problemas conservativos. Si se hubiese aplicado un método conservativo en lugar de los utilizados, se esperaría que el error de amplitud sea casi cero. También es necesario aclarar que la eficacia de los métodos depende mucho del problema a valores iniciales que se quiere integrar. Por ejemplo si las funciones fueran una constantes no tendria sentido utilizar métodos de mayor orden si con métodos más sencillos y con menos evaluaciones se llega a la misma solución.

- Comparando los 3 métodos implementados, se puede observar que Euler tuvo el menor error de fase, y un error de amplitud intermedio, mientras que RK2 tuvo error de fase medio y error de amplitud máximo y RK4 tuvo error de fase alto y error de amplitud mínimo. Es decir, en una aplicación donde se pretenda minimizar la pérdida de energía y donde no importe tanto el error temporal se utilizaría RK4.

- Los tres métodos, al ser explícitos, poseen condiciones que hay que garantizar para asegurar la estabilidad. Estas condiciones implican un paso máximo, que equivalen a una cantidad mínima de pasos por período. En nuestros casos, Euler siguió estable incluso con 8 pasos por período, mientras que RK4 y RK2 necesitaron bastantes más pasos por período para no tener tanta pérdida y ganancia de energia, respectivamente.

- En cuanto al esfuerzo de cómputo, el método de Euler solamente requiere una llamada a cada función del sistema de ecuaciones, mientras que RK2 requiere 2 y RK4 requiere 4. En un sistema donde las ecuaciones sean mucho más complejas, está claro que la mayor parte del tiempo se gastaria en las evaluaciones a las funciones, ya que los otros cálculos necesarios son sumas y productos. Para analizar la conveniencia del método RK4 frente a los otros dos, se comparó entre el método de Euler a 300 pasos por período, RK2 a 150 pasos por período y RK4 a 75 pasos por período. Se llegó a que el error de amplitud de RK4 seguía siendo menor que el de Euler, aunque no por tanta diferencia.

A. Codigo (C++)

```cpp
#include <iostream>
#include <fstream>
#include <cstdio>
#include <cmath>
using namespace std;

const double G=9.81;
const double A=M_PI*(7.0*100.0+37.0)/1000.0;
const double L=1.0;
ofstream fout("salida.csv");

double f1(double theta, double omega, double t){
    return omega;
}
double f2(double theta, double omega, double t){
    return -G/L*sin(theta);
}
double energia(double theta, double omega){
    return 0.5*omega*L*omega*L + G*L*(1-cos(theta));
}

double euler(double k, int n, int pasosPorPeriodo, double periodoIntegrado, double theta, double omega){
    int signoOmega = 1, per=0;

    for(int i=0;i<n;i++){
        if(i%pasosPorPeriodo == 0) cout << "\tPeriodo " << i/pasosPorPeriodo << " - Amplitud:" << theta << " ERel%:" << (theta-A)/A *100 << endl;

        theta = theta + k*f1(theta,omega,k*i);
        omega = omega + k*f2(theta,omega,k*i);
        fout << i*k << "\t" << theta << endl;

        if(omega<0 && signoOmega == 1){
            //Paso de positivo a negativo
            cout << "\tMaxima amplitud para T=" << k*i << " Esperado: " << periodoIntegrado*per << " ERel%:" << 100*(k*i-periodoIntegrado*per)/(k*i) << endl;
            per++;
        }
        signoOmega = omega>0?1:0;

    }
    return energia(theta,omega);
}
double rk2(double k, int n, int pasosPorPeriodo, double periodoIntegrado, double theta, double omega){
    int signoOmega = 1, per=0;
    for(int i=0;i<n;i++){
        if(i%pasosPorPeriodo == 0) cout << "\tPeriodo " << i/pasosPorPeriodo << " - Amplitud:" << theta << " ERel%:" << (theta-A)/A *100 << endl;

        double q1_theta = k*f1(theta,omega,k*i);
        double q1_omega = k*f2(theta,omega,k*i);
        double q2_theta = k*f1(theta+q1_theta,omega+q1_omega,k*(i+1.0));
        double q2_omega = k*f2(theta+q1_theta,omega+q1_omega,k*(i+1.0));
        theta = theta + (q1_theta+q2_theta)*0.5;
        omega = omega + (q1_omega+q2_omega)*0.5;

        fout << i*k << "\t" << theta << endl;

        if(omega<0 && signoOmega == 1){
```

```
            //Paso de positivo a negativo
            cout << "\tMaxima amplitud para T=" << k*i << " Esperado: " << periodoIntegrado*per << " ERel%:" << 100*(k*i-periodoIntegrado*per)/(k*i) << endl;
            per++;
        }
        signoOmega = omega>0?1:0;
    }
    return energia(theta,omega);
}
double rk4(double k, int n, int pasosPorPeriodo, double periodoIntegrado, double theta, double omega){
    int signoOmega = 1, per=0;
    for(int i=0;i<n;i++){
        if(i%pasosPorPeriodo == 0) cout << "\tPeriodo " << i/pasosPorPeriodo << " - Amplitud:" << theta << " ERel%:" << (theta-A)/A *100 << endl;

        double q1_theta = k*f1(theta,omega,k*i);
        double q1_omega = k*f2(theta,omega,k*i);
        double q2_theta = k*f1(theta+q1_theta/2.0,omega+q1_omega/2.0,k*(i+0.5));
        double q2_omega = k*f2(theta+q1_theta/2.0,omega+q1_omega/2.0,k*(i+0.5));
        double q3_theta = k*f1(theta+q2_theta/2.0,omega+q2_omega/2.0,k*(i+0.5));
        double q3_omega = k*f2(theta+q2_theta/2.0,omega+q2_omega/2.0,k*(i+0.5));
        double q4_theta = k*f1(theta+q3_theta,omega+q3_omega,k*(i+1.0));
        double q4_omega = k*f2(theta+q3_theta,omega+q3_omega,k*(i+1.0));
        theta = theta + (q1_theta+2.0*q2_theta+2.0*q3_theta+q4_theta)/6.0;
        omega = omega + (q1_omega+2.0*q2_omega+2.0*q3_omega+q4_omega)/6.0;

        fout << i*k << "\t" << theta << endl;

        if(omega<0 && signoOmega == 1){
            //Paso de positivo a negativo
            cout << "\tMaxima amplitud para T=" << k*i << " Esperado: " << periodoIntegrado*per << " ERel%:" << 100*(k*i-periodoIntegrado*per)/(k*i) << endl;
            per++;
        }
        signoOmega = omega>0?1:0;
    }
    return energia(theta,omega);
}

double integrando(double x, double amplitud){
    return 4*sqrt(L/G)*pow(1-pow(sin(amplitud/2.0),2.0)*pow(sin(x),2.0),-0.5);
}

double calcularPeriodo(double amplitud){
    //Trapecio compuesto
    double acc = 0;
    const int n = 350;
    const double b = M_PI/2.0;
    const double HINT = b/n;
    const double fddMax= 17.0; //Cota de la derivada segunda entre 0 y PI/2

    acc += integrando(0,amplitud) * HINT * 0.5;
    acc += integrando(b,amplitud) * HINT * 0.5;

    for(int i=1;i<n;i++)
            acc += integrando(i*HINT,amplitud) * HINT;

    cout << "Error estimado: " << b*HINT*HINT*fddMax/12 << endl;
    return acc;
}

int main(){
```

```
    double theta_0 = A;
    double omega_0 = 0;
    double periodoIntegrado = calcularPeriodo(A);
    int pasosPorPeriodo=75;
    double paso = periodoIntegrado/pasosPorPeriodo;
    int cantPasos = pasosPorPeriodo*10;

    cout.precision(5);
    cout << "Periodo aprox: " << 2*M_PI*sqrt(L/G) << endl;
    cout << "Periodo integrado: " << periodoIntegrado << endl;
    cout << "Periodo para A=0.01: " << calcularPeriodo(0.01) << endl;

    cout.precision(8);
    cout << endl << "Amplitud inicial: " << A << endl;
    cout << "Energia inicial del pendulo: " << energia(theta_0,omega_0) << endl;
    cout << "Euler" << endl;
    cout << "\tEnergia final Euler:" << euler(paso, cantPasos, pasosPorPeriodo, periodoIntegrado,
         theta_0, omega_0) << endl;
    cout << "RK2" << endl;
    cout << "\tEnergia final RK2:" << rk2(paso, cantPasos, pasosPorPeriodo, periodoIntegrado,
        theta_0, omega_0) << endl;
    cout << "RK4" << endl;
    cout << "\tEnergia final RK4:" << rk4(paso, cantPasos, pasosPorPeriodo, periodoIntegrado,
        theta_0, omega_0) << endl;

    return 0;
}
```