Facultad de Matemática, Astronomía, Física y Computación
Universidad Nacional de Córdoba

Física Computacional
Laboratorio 1a

Gastón Briozzo

E-mail: gbriozzo@mi.unc.edu.ar

Fecha de entrega: viernes 2 de abril del 2021

Resumen
El objetivo de este trabajo es emplear métodos de integración numérica para estudiar el comportamiento del error de algoritmo y la contribución del error de truncamiento de la máquina.
Para esto, se escribieron en fortran 90 una serie de programas para calcular una integral definida por los métodos del trapecio, de Simpson y por la cuadratura de Gauss, y para resolver un sistema de ecuaciones diferenciales de segundo orden a valores iniciales por los métodos de Euler y de Runge-Kutta de segundo y cuarto orden. Los programas se corrieron en Ubuntu, Linux, y los resultados fueron graficados empleando el software gnuplot. A partir de estos, se estimaron las leyes de potencias que rigen el comportamiento del error de cada método, como también el de la máquina, encontrando de esta forma el régimen óptimo de funcionamiento de cada método y determinando la eficiencia de estos.
Se observó que el error desciende de manera determinista al aumentar el número de particiones del intervalo de integración hasta alcanzar su valor mínimo, para después crecer estocásticamente debido a la contribución del error de truncamiento de la máquina. También se comprobó que, al aproximarse a la apreciación de la máquina, el error comienza a comportarse caóticamente.

1. Introducción

Se llama integración numérica, o cuadratura numérica, al conjunto de fórmulas y técnicas provistas por la matemática y la computación, generalmente en forma de algoritmos, empleados para calcular de forma aproximada el valor de integrales definidas. El término también se usa, por extensión, para describir algoritmos numéricos empleados para la resolución de ecuaciones diferenciales.

Estos métodos resultan de gran utilidad ante la imposibilidad de realizar la integración de forma analítica, ya sea porque son muy complejas y consumen demasiados recursos o porque directamente no existe una solución analítica. La solución numérica nos daría una solución aproximada, con un error que dependerá tanto del algoritmo que se emplee como de la precisión de la máquina que lo ejecute.

Al ser aproximaciones, cada algoritmo introduce un error característico ε_a. Idealmente, este error se puede hacer tan pequeño como se desee incrementando la fineza del algoritmo, del mismo modo que la suma de Riemann converge a la integral exacta cuanto dx tiende a 0. Si n es el número de iteraciones que realiza el algoritmo, el error acumulado resulta generalmente de la forma ε_a/n^{α}, donde el exponente α es también característico del algoritmo. Sin embargo, al aumentar el número de iteraciones, estamos incrementando también el tiempo requerido, de modo que una solución exacta requeriría infinito tiempo computacional.

Al mismo tiempo, dado que las maquinas cuentan con una memoria limitada, rara vez pueden guardar un número exacto. Al introducir

un número, este es truncado, conservando solo algunas cifras significativas. Existe incluso un límite a cuan pequeño puede ser un número antes de ser considerado cero, llamado *apreciación de la máquina* ε_m. Éste será característico de la máquina con la que trabajemos. Dado que el valor del error de truncamiento es aleatorio, y que este oscila entre más y menos ε_m, el error total acumulado describe una caminata al azar unidimensional, resultando su valor más probable en $\varepsilon_m\sqrt{n}$, donde n es el número de truncamientos realizados (en la práctica, n es indistinguible del número de iteraciones).

Vemos que, mientras el error acumulado del algoritmo decrece con n, el de la máquina aumenta. De esta forma, existe un valor de n para el cual el error total es mínimo, llamado n *optimo*, que puede ser calculado como sigue:

$$\varepsilon_T = \varepsilon_a n^{-\alpha} + \varepsilon_m n^{1/2}$$

$$\left.\frac{d\varepsilon_T}{dn}\right|_{n_0} = -\alpha\varepsilon_a n_o^{-\alpha-1} + \frac{\varepsilon_m n_o^{-1/2}}{2} = 0$$

$$\alpha\varepsilon_a n_o^{-\alpha-1/2} = \frac{\varepsilon_m}{2}$$

$$n_o = \left(\frac{\varepsilon_m}{2\alpha\varepsilon_a}\right)^{-1/(\alpha+1/2)} \qquad (1)$$

Al elegir un método de integración numérica, debemos considerar tanto el error total como el tiempo y la capacidad de la máquina de los que disponemos.

El problema que se plantea la integración numérica es encontrar una solución aproximada a la integral definida.

$$I = \int_a^b f(x)dx \qquad (2)$$

La mayoría de los métodos se basan en aproximar la función a integrar $f(x)$ por otra función $g(x)$, generalmente un polinomio, cuya integral exacta es conocida. Para esto, se requiere que $f(x_i) = g(x_i)$ para un dado conjunto $\{x_j : j = 1, \dots, m\}$, perteneciente al intervalo de integración. Cuando los extremos del intervalo a y b forman parte de este conjunto de puntos, la nueva función se llama una interpolación de la función original. En caso contrario, se la llama extrapolación.

Generalmente, la aproximación se puede expresar como

$$I_n = \sum_{i=0}^{n} \omega_i f(x_i) \qquad (3)$$

donde los ω_i, llamados *pesos*, y los x_i dependerán del método.

El *método del trapecio*, o *trapezoide*, consiste en aproximar la función por polinomios de primer orden, es decir rectas, que interpolan la función entre dos x_i consecutivos. El intervalo de integración se divide en $n-1$ segmentos idénticos, de longitud $h = (b-a)/(n-1)$, resultando en n puntos de evaluación x_i equiespaciados, que incluyen a los extremos del intervalo. Se tiene entonces:

$$I_T = \sum_{i=1}^{n-1} \frac{f(x_i) + f(x_{i+1})}{2} h$$

$$\omega_i = \left\{\frac{h}{2}, h, h, \dots, h, h, \frac{h}{2}\right\}$$

$$I_T = h\left[\frac{f_1}{2} + f_2 + \cdots + f_{n-1} + \frac{f_n}{2}\right] \qquad (4)$$

El método de Simpson aproxima la función por polinomios de segundo orden, que interpolan a la función en conjuntos de tres x_i. Un requisito para emplear este método es que el número de puntos a evaluar sea impar. Nuevamente, se toman los n puntos de evaluación x_i equiespaciados, incluyendo los extremos del intervalo. Éste se divide en $(n-1)/2$ segmento de longitud $2h$, cada uno de los cuales es interpolado por una parábola que se empalma con la siguiente. Recordemos que una parábola ajusta perfectamente todo conjunto de tres puntos. Conociendo la integral de estas, obtenemos:

$$I_S = \sum_{i=1}^{(n-1)/2} \frac{f(x_{2i-1}) + 4f(x_{2i}) + f(x_{2i+1})}{6} 2h$$

$$\omega_i = \left\{\frac{h}{3}, \frac{4h}{3}, \frac{2h}{3}, \dots, \frac{2h}{3}, \frac{4h}{3}, \frac{h}{3}\right\}$$

$$I_T = h\left[\frac{f_1}{3} + \frac{4f_2}{3} + \frac{2f_3}{3} + \cdots + \frac{4f_{n-1}}{3} + \frac{f_n}{3}\right] \qquad (5)$$

La cuadratura de Gauss de orden n es un método diseñado para obtener la solución exacta

al integrar cualquier polinomio de grado menor o igual a $2n-1$. Desplazando el intervalo de integración a $[-1,1]$, el punto de evaluación x_i se corresponden con la i-ésima raíz del polinomio de Legendre de orden n, mientras que los pesos se obtienen a partir de la siguiente ecuación:

$$\omega_i = \frac{2}{(1-x_i^2)[P_n'(x_i)]^2}$$

Este método es sumamente eficiente al integrar funciones suaves o moduladas por exponenciales.

Otro tipo de problemas es el encontrar una aproximación numérica a las soluciones de *Ecuaciones Diferenciales Ordinarias* (ODEs) con condiciones iniciales, de la forma

$$y'(t) = f(t, y(t)), \quad y(0) = y_0 \qquad (6)$$

donde f es conocida, t es la variable independiente e y es la dependiente. La mayoría de los métodos se basan en hacer avanzar t de forma casi diferencial, aproximando la derivada por una división sobre un intervalo de tiempo muy pequeño, y resolviendo la ecuación resultante para obtener el valor de y en el instante siguiente. Este procedimiento se repite de forma iterativa para obtener una aproximación a la curva descrita por $y(t)$.

El método de Euler parte de la siguiente aproximación:

$$\frac{dy(t)}{dt} \approx \frac{y(t+h)-y(t)}{h}$$
$$y(t+h) \approx y(t) + h\frac{dy(t)}{dt}$$

$$y(t+h) \approx y(t) + hf\big(f, y(t)\big) \qquad (7)$$

Aquí, h es el paso de integración, o *step size*, que debe ser pequeño comparado con los periodos característicos de la función. Notemos que en el límite en que h tiende a cero obtenemos la solución exacta. Empleando este método, podemos dividir el intervalo de tiempo a integrar en segmentos de tamaño h y encontrar el valor de y en cada instante en función de su valor en el instante anterior.

Sin embargo, si bien $y'(t)$ se corresponde con la pendiente de y en el instante t, no tiene por que ser representativa de la variación de y entre t y $t+h$. En vista de esto, métodos más complejos y rigurosos como los de Runge-Kutta evalúan la función f en tiempos entre t y $t+h$, para valores de y entre $y(t)$ y el $y(t+h)$ obtenido por Euler, con el objetivo de encontrar una pendiente que aproxime en mejor medida la variación de y.

El método de Runge-Kutta de segundo orden procede como se muestra a continuación:

$$k_1 = hf\big(t, y(t)\big)$$
$$k_2 = hf(t+h, y(t)+k_1)$$

$$y(t+h) = y(t) + \frac{k_1+k_2}{2} \qquad (8)$$

Por su parte, el método de Runge-Kutta de cuarto orden obedece el siguiente algoritmo:

$$k_1 = hf\big(t, y(t)\big)$$
$$k_2 = hf(t+h/2, y(t)+k_1/2)$$
$$k_3 = hf(t+h/2, y(t)+k_2/2)$$
$$k_4 = hf(t+h, y(t)+k_3)$$

$$y(t+h) = y(t) + \frac{k_1+2k_2+2k_3+k_4}{6} \qquad (9)$$

Estos métodos pueden extenderse a ecuaciones diferenciales de orden n, dado que estas son equivalentes a n ecuaciones diferenciales de primer orden. Para esto, debemos extender a las variables y, f, y k_i a vectores de dimensión n.

2. Procedimiento experimental

Considerando la siguiente integral definida

$$I = \int_0^1 e^{-x}dx = 1 - e^{-1} \approx 0{,}63212\ldots \qquad (10)$$

se escribieron en fortran 90 una serie de programas para realizar dicha integral por los métodos del Trapecio, de Simpson y de Gauss. Estos programas dividieron el intervalo de integración en $n = 2^i$ segmentos idénticos, barriendo valores enteros para i desde 2 hasta 30. Los resultados obtenidos numéricamente se compararon con el resultado analítico para obtener sus respectivos errores relativos. Estos últimos se graficaron en función de n en escala logarítmica, empleando el software gnuplot. A

partir de estos gráficos se lograron estimar las leyes de potencia que rigen el error de cada algoritmo, así como el error de la máquina, encontrando su dependencia con n.

Por otro lado, se escribieron en fortran 90 programas para resolver por los métodos de Euler, RK2 y RK4 la siguiente ecuación diferencial de segundo orden,

$$\ddot{x} = -kx \qquad (11)$$

$$x(0) = 1\,m, \quad \dot{x}(0) = 1\,m/s, \quad k = 1\,s^{-2}$$

cuya solución exacta corresponde a un oscilador armónico de la forma

$$x(t) = \sqrt{2}\sin(ts^{-1} + \pi/4)\,m$$
$$v(t) = \sqrt{2}\cos(ts^{-1} + \pi/4)\,m/s$$

Para estimar el n optimo, se calculó el error global cometido por cada método en función de n, que tomo valores iguales a 2^i, para i entero entre 2 y 30. Los valores resultantes se graficaron en escala logarítmica en gnuplot con el objetivo de encontrar las leyes de potencias que rigen los errores de cada algoritmo. Una vez encontrados los n_o, se compararon los resultados de cada método. También se evaluó la energía cinética y potencial del oscilador en función del tiempo, comparando los resultados obtenidos para cada método en su régimen óptimo. Recordemos que, para un oscilador armónico, la energía total es constante, mientras que sus componentes cinética y potencial resultan

$$E_c = \frac{1}{2}mv^2, \quad E_p = \frac{1}{2}kx^2$$

3. Resultados y discusión

3.1 Integral definida

El error relativo de integración de cada método se encuentra graficado en función de n en la *Figura 1*. En este gráfico se encuentran también los ajustes realizados con leyes de potencias para estimar cada error, incluyendo el error de la máquina.

Lo primero que observamos es que, antes de superar el n óptimo, el error de cada método describe casi perfectamente una ley de potencias, reflejada en el gráfico *log-log* como una recta. Decimos entonces que, en esta región, el error es *determinista*, dado que podemos predecir su valor exacto a partir de la teoría. Por otro lado, una vez superado el n óptimo, el error parece tomar valores aleatorios, aunque siempre próximos a otra ley de potencia dada por el error de redondeo. En este punto, el error pasa a ser *estocástico*, de modo que no podemos predecir exactamente cual será su valor, pero podemos saber que posibilidades tenemos de encontrarlo apartado una cierta distancia del valor mas probable, correspondiente al de una caminata al azar unidimensional.

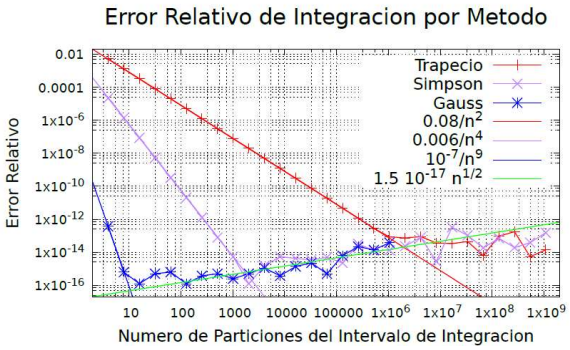

Figura 1: Error relativo por método en función de n.

Podemos notar que la cuadratura de Gauss es el método más eficiente para resolver esta integral, llegando a su régimen óptimo en tan solo 2^4 particiones y alcanzando un error relativo de 10^{-16}, apenas por encima de la precisión de la máquina. Sin embargo, al decaer tan rápido el error, contamos con demasiados pocos puntos para ajustar la ley de potencia. Por este motivo, se decidió evaluar nuevamente su error relativo, para valores de n entre 1 y 16. Los resultados se muestran en la *Figura 2*.

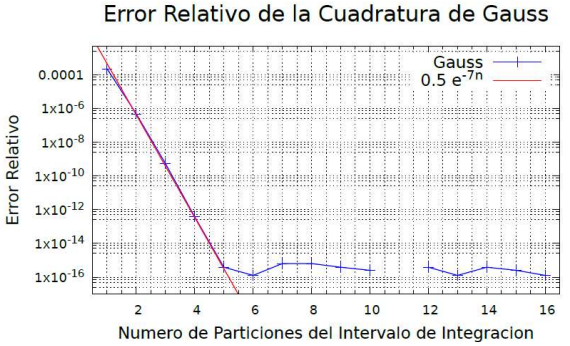

Figura 2: Error relativo de Gauss para n pequeño.

Observando el gráfico, podemos notar que la cuadratura gaussiana es tan eficiente que su error

no obedece una ley de potencia, sino que decae exponencialmente con n, necesitando de tan solo 5 particiones para alcanzar su régimen óptimo.

El siguiente método más eficiente es el de Simpson, cuyo n óptimo es de 2^{10} , resultando en un error relativo de $5 \cdot 10^{-15}$. Si bien para $n = 2^{11}$ se obtuvo un error menor, ya pertenece al régimen estocástico, por lo que este valor se debe al azar, no a la efectividad del método, y no podemos garantizar que vuelva a repetirse. El ajuste sugiere que el error es proporcional a n^{-4}, lo que se condice con el valor teórico.

El método más impreciso resulto ser el del Trapecio, con un n óptimo de 2^{20} y un error relativo de 10^{-13}. El ajuste indica que su ley de potencia es proporcional a n^{-2}, que es lo esperado para este método.

Por otro lado, pudimos estimar la apreciación de la máquina en $\varepsilon_m = 1{,}5 \cdot 10^{-17}$, y comprobar que el error de redondeo va como $\varepsilon_m \sqrt{n}$. A partir de este resultado, y de la *Ecuación 1*, podemos realizar las siguientes estimaciones:

$$n_T = \left(\frac{1{,}5 \cdot 10^{-17}}{2 \cdot 2 \cdot 0.08}\right)^{-1/(2+1/2)} = 3{,}40 \cdot 10^6$$
$$n_T = 3{,}2 \cdot 2^{20}$$

$$n_S = \left(\frac{1{,}5 \cdot 10^{-17}}{2 \cdot 4 \cdot 0{,}006}\right)^{-1/(4+1/2)} = 2789.9$$
$$n_S = 2{,}7 \cdot 2^{10}$$

$$n_G = \left(\frac{1{,}5 \cdot 10^{-1}}{2 \cdot 9 \cdot 10^{-7}}\right)^{-1/(9+1/2)} = 14.66$$
$$n_G = 0{,}92 \cdot 2^4$$

Vemos que los valores obtenidos experimentalmente para el n óptimo de cada método coinciden con en valor predicho teóricamente dentro de un orden de magnitud.

3.2 *ODEs. Error global y n óptimo*

En la *Figura 3* se muestra el error global resultante de cada método en función de n. Nuevamente, observamos que el error desciende de forma determinista hasta alcanzar su valor mínimo en el n óptimo, para después subir de forma estocástica. Las leyes de potencias encontradas se encuentran también graficadas, siendo proporcionales a n^{-1} para el método de Euler, n^{-2} para RK2 y n^{-4} para RK4, coincidiendo con los valores predichos teóricamente. Además, la apreciación de la máquina y el error de truncamiento coinciden con los obtenidos en la sección anterior.

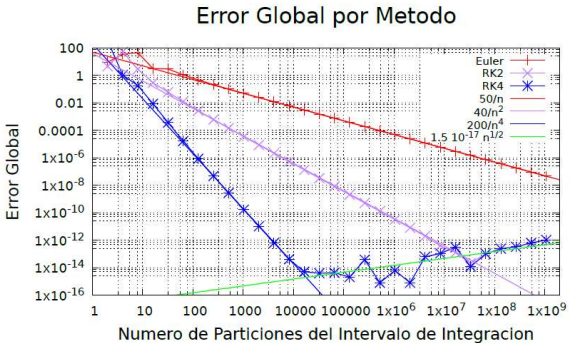

Figura 3: Error global por método en función de n.

Puede verse que en el gráfico no se muestra el n óptimo para el método de Euler. Esto se debe a que su valor estimado es de $3 \cdot 10^{12}$, mientras que el tiempo computacional estimado para extender el gráfico hasta ese valor es de 599 240 horas, o 68,4 años. De esta forma, concluimos que este es el método más ineficiente.

El método RK2, por su parte, consiguió un error mínimo de $2 \cdot 10^{-14}$ con 2^{25} particiones, mientras que su n óptimo teórico es de $1{,}2 \cdot 2^{25}$.

Finalmente, el método más eficiente resultó ser el RK4, con un n óptimo experimental de 2^{14}, bastante próximo a su valor teórico de $1{,}7 \cdot 2^{14}$, y logrando un error relativo de $5 \cdot 10^{-15}$.

3.3 *ODEs. Posición y velocidad*

En las *Figuras 4* y *5* se muestran respectivamente la posición y la velocidad en función del tiempo, obtenidas por cada método en su régimen óptimo.

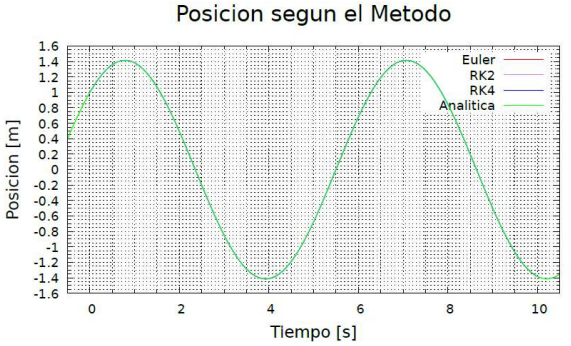

Figura 4: Posición en función del tiempo por método.

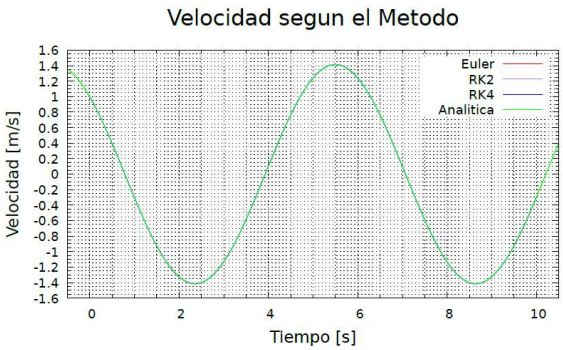

Figura 5: Velocidad en función del tiempo por método.

Aclaramos que, para no excedernos con el tiempo computacional, el método de Euler se corrió con $n = 2^{25}$.

Podemos notar que las curvas obtenidas numéricamente resultan prácticamente indistinguibles de la curva predicha analíticamente. Por este motivo, se decidió graficar solo el error absoluto de cada método en función del tiempo. Para comparar mejor los métodos, este procedimiento se realizó primero corriendo todos los métodos para un mismo n, y luego corriendo cada método en su n optimo respectivo. Los resultados obtenidos se muestran en las *Figuras 6*, *7*, *8* y *9*.

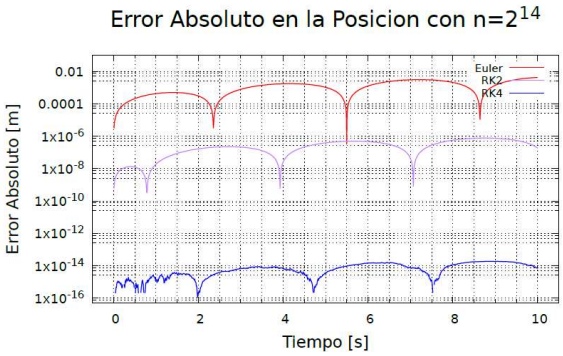

Figura 6: Error absoluto en la posición para $n = 2^{14}$.

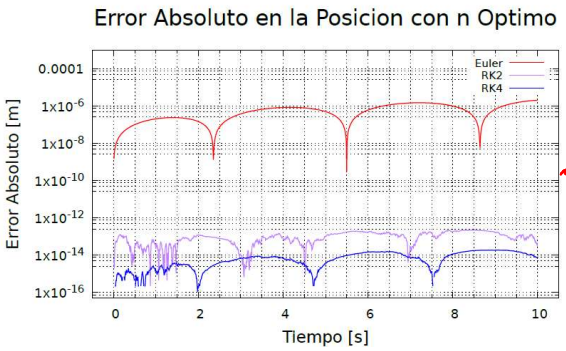

Figura 6: Error abs. en la posición para n óptimo.

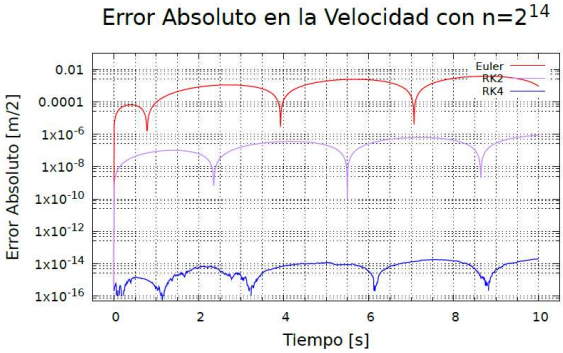

Figura 8: Error abs. en la velocidad para $n = 2^{14}$.

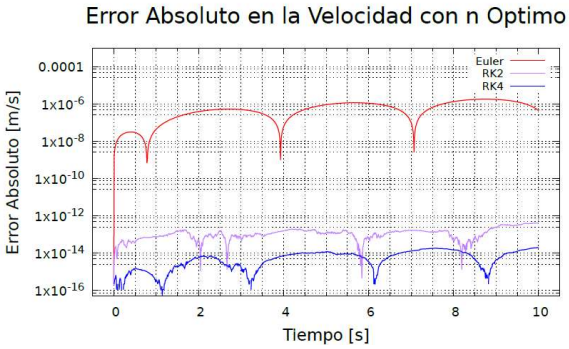

Figura 9: Error abs. en la velocidad para n óptimo.

Puede verse que el método mas eficiente es el de Runge-Kutta de cuarto orden, seguido por el de segundo orden, y resultando el método de Euler el más ineficiente. Aclaramos que se decidió graficar el error absoluto y no el relativo dado que la función original pasa por cero, o tiene valores muy cercanos a este, en muchos puntos. Tomar el error relativo daría como resultado un comportamiento errático alrededor de estos puntos, producto de el error de la máquina al dividir valores tan pequeños, y no representativo de la aproximación numérica conseguida.

Notemos que el error parece crecer y decrecer periódicamente, presentando claros picos de error mínimo. Esto se debe a que las curvas numérica y analítica, si bien no son idénticas, se cortan regularmente con cierta frecuencia. Esta precisión no debería hacernos ilusión; hasta un reloj descompuesto da bien la hora dos veces al día. Lo que estamos observando es solo producto de la naturaleza periódica de la función que intentamos aproximar.

Otra cosa que nos indican los gráficos es que el error "medio" crece lentamente con el tiempo. Si bien esta variación es demasiado pequeña como para intentar ajustar una ley de potencias, es de esperarse que, a medida que el sistema

evoluciona, la solución numérica se aparte de la analítica al ser solo una aproximación.

Finalmente, podemos observar que a medida que n crece, el error absoluto pasa de ser una curva relativamente suave y periódica a otra mucho mas irregular y caótica. Este comportamiento se observa claramente al comparar los resultados del método RK2 para $n = 2^{14}$ y $n = 2^{25}$. Esto podría estarnos advirtiendo sobre la proximidad de la inestabilidad numérica al trabajar con números tan pequeños.

3.4 ODEs. Energía

Las curvas de energía cinética y potencial en función del tiempo obtenidas por integración numérica para el oscilador armónico se muestran en la *Figura 10*.

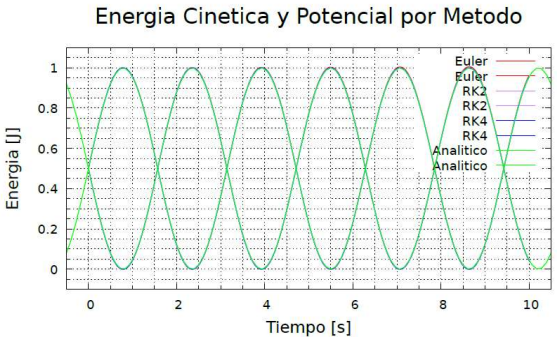

Figura 10: Energía cinética y potencial por método.

Nuevamente, las aproximaciones resultan casi indistinguibles de la solución analítica. Sin embargo, si observamos con cuidado, podemos notar que las curvas numéricas se apartan a medida que el sistema evoluciona, siendo el método de Euler el que presenta diferencias mas evidentes, y el RK4 el que mejor se aproxima, como cabe esperar.

Con el objetivo de estudiar mas detalladamente la evolución de los errores resultantes para cada método, se calculó para cada instante el error relativo en la energía total del sistema. Los resultados se graficaron en función del tiempo, obteniéndose la *Figura 11*. En este gráfico, el error relativo está en escala logarítmica para poder observar los tres métodos a la vez. Sin embargo, podemos notar que este crece linealmente con el tiempo. Para los métodos de Euler y RK2, tenemos que el error describe una recta perfectamente suave, vista el un gráfico semilogarítmico como, justamente, un logaritmo. Este es el comportamiento que cabe esperar para cualquier método.

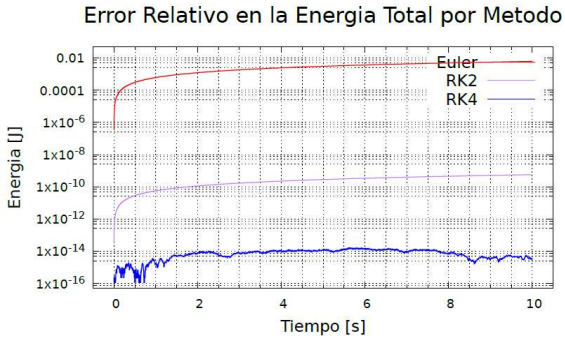

Figura 11: Error relativo en la energía total.

La curva obtenida por RK4 también se aproxima a una recta, pero tiene un comportamiento mucho mas caótico que las otras, sobre todo para tiempos pequeños, mientras que para tiempos largos pareciera que el error comienza a decaer. Este comportamiento puede deberse a que estamos trabajando al borde del error de la máquina, y es posible que ese decaimiento en los últimos instantes se deba solo a una fluctuación aleatoria.

4. Conclusiones

Se observó que el error de cada algoritmo decae de forma determinista al aumentar el número n de particiones, obedeciendo en la mayoría de los casos una ley de potencias, y una exponencial para la cuadratura de Gauss. El error alcanza su valor mínimo y comienza a crecer, presentando ahora un comportamiento estocástico alrededor de un valor medio, dado por otra ley de potencia correspondiente al error de truncamiento de la máquina. Este puede verse como el resultado de una caminata al azar unidimensional.

Gráficamente, los resultados numéricos resultan casi indistinguibles de los analíticos, avalando los métodos empleados.

También se observó que el error crece linealmente con la longitud del intervalo de integración, presentando comportamientos caóticos a medida que se aproxima a la precisión de la máquina.

Bibliografía

[1] S. J. Chapman, Fortran 95/2003 for Scientist and Engineers, 3° edición, McGraw Hill (2007)