

XUÑO 2023. VOLUME 6



---

# MARCO NUMÉRICO

---

PUBLICACIÓN MATEMÁTICA DO IES MARCO DO CABBALLÓN



XUÑO 2023. VOLUME 6

---

# MARCO NUMÉRICO

---

PUBLICACIÓN MATEMÁTICA DO IES MARCO DO CABBALLÓN

## **MARCO NUMÉRICO**

Xuño 2023. Volume 6

*Marco Numérico* é a publicación matemática do IES Marco do Camballón (Vila de Cruces, Pontevedra). De carácter anual, está aberta a artigos de investigación suxeitos ás esixencias profundidade, rigor e forma (metodolóxica e expositiva) propias dos estándares habituais do ámbito académico.

<b>EDICIÓN</b>	Departamento de Matemáticas
<b>REVISIÓN LINGÜÍSTICA</b>	Departamento de Lingua e Literatura Galega Departamento de Inglés
<b>DESEÑO E IMPRESIÓN</b>	Sacauntos S. Coop. Galega Santiago de Compostela
<b>ISSN</b>	2603-7459
<b>DEP. LEGAL</b>	PO 581-2017
<b>CONTACTO</b>	<a href="mailto:marconumerico@iesmarcodocamballon.com">marconumerico@iesmarcodocamballon.com</a>

Os textos contidos neste volume poden ser difundidos, dados a coñecer e transmitidos en calquera medio ou ámbito comunicativo e da información, sempre que se respecte con veracidade o traballo autorial e non se persigan fins comerciais.

# MARCO NUMÉRICO

PUBLICACIÓN MATEMÁTICA DO IES MARCO DO CABBALLÓN

Xuño 2023. Volume 6



IES MARCO DO CABBALLÓN



# Índice

<i>Limiar</i> .....	7	
<i>Cadeas de Markov na metereoloxía do mundo de Oz</i> .....	9	
Calvo-López, N.; Mazás-Rivela, P.		
<i>Cadeas de Markov nun poema de Rosalía de Castro</i> .....	17	
Sebio-Mazás, R.; Rey-Silva, A.; Méndez-Reimondéz, S..		
<i>Método de Newton-Raphson aplicado á ecuación de Kepler</i> .....	23	
Álvarez-Lodeiro, I.; Amosa-Martínez U.; Casal-Blanco, R.; Ferro-Pérez, J.M.		
<i>Análise dos parámetros da onda estacionaria</i> .....	33	
Fraiz-Casas, A.; López-Calviño, E.; Puga-Silva, L.; Silva-Lodeiro, X.; Verde-Salgado, N.....		33
<i>Dous métodos de integración numérica (I)</i> .....	47	
Sebio-Mazás, R.; Rey-Silva, A.; Calvo-López, N..		
<i>Dous métodos de integración numérica (II)</i> .....	59	
Mazás-Rivela, P.; Méndez-Reimóndez, S.		
<i>Series de tempo en datos de esperanza de vida</i> .....	69	
Rey-Silva, A.; Sebio-Mazás, R.; Calvo-López, N.; Mazás-Rivela, P.; Méndez-Reimóndez, S.		





## Limiar

O primeiro artigo estatístico que redactei permanece fresco no meu recordo, como se o redactara fai pouco tempo. Tratábase de descifrar un fragmento de texto e, neste caso, proviña de *A Esmorga* (1959), unha novela de Eduardo Blanco Amor.

Recentemente, tiven a oportunidade de ler outro libro deste autor titulado *Os Biosbardos* (1962). Aínda que non sei se se trata dunha mera coincidencia, é innegable que o feito é interesante e, incluso, máxico. E por iso, pregúntome: que é a estatística senón a manifestación máis fascinante das matemáticas? E que son os biosbardos máis ca seres fantásticos?

Fiando estes sucesos e pensamentos, creo que se poden comparar os artigos expostos neste libro cos contos do escritor ourensán. Estes últimos exploran a imaxinación e a fantasía, mentres que os artigos estatísticos permiten desentrañar os misterios dos números, revelando paradigmas ocultos dos datos. Ambos son relatos abraiantes, e aínda que non se pode afirmar que os artigos sexan máxicos, queda ao voso xuízo consideralos así.

En definitiva, a estatística e a literatura poden parecer mundos afastados, pero a través da miña experiencia con ambos, teñen máis similitudes do que imaxinamos. Este libro comparte algunhas das impresionantes historias que se poden descubrir a través da análise de datos e como isto pode transformar a maneira en que vemos e entendemos o mundo que nos rodea.

Gustaríame rematar esta breve introdución invitando aos lectores e lectoras a unirse a esta viaxe emocionante cara o mundo da estatística e a ciencia dos datos. Espero que este desorde de palabras sobre os meus comezos coa estatística e o encontro cos contos fantásticos vos espertara o interese e vos anime a difundilo coas persoas que vos rodean, pois este ámbito non é exclusivo e calquera pode gozalo.

Só me queda agradecer poder ser quen introduce estes artigos, fun das primeiras persoas en comezar este proxecto e é todo un pri-

vilexio poder seguir nel. Confío en que os autores e autoras deste volume disfrutaran elaborando os artigos igual ca min no seu momento. E non son meiga, pero augúrolles un porvir estupendo se continúan relatando as historias máxicas dos números.

Noelia García García  
Enxeñeira informática  
Exalumna do IES Marco do Camballón

# Cadeas de Markov na metereoloxía do mundo de Oz

*Calvo-López, N.; Mazás-Rivela, P.*

IES MARCO DO CAMBALLÓN

**Resumo.** Neste artigo inténtase aproximar ao lector ás cadeas de Markov explicando os seus fundamentos teóricos e utilizando un exemplo concreto para facilitar a comprensión.

**Abstract.** In this article we try to approach to lector the Markov chains explaining their fundamentals using a easy example

## 1. Introducción

Neste artigo imos explicar o concepto de cadea de Markov e algunhas das súas aplicacións cun exemplo concreto que é a análise da meteoroloxía do mundo de Oz.

Primeiro necesitamos comprender o que é un proceso estocástico, que se pode definir como unha sucesión de variables aleatorias que cambian en función do tempo. Os valores que adoptan as variables aleatorias reciben o nome de estados e o seu conxunto, espazo de estados. Un exemplo trivial de proceso estocástico sería lanzar un dado cada certo tempo. Neste caso os estados serían cada unha das caras do dado e o espazo de estados serían as 6 en conxunto. No exemplo anterior, o proceso estocástico non nos permite facer predicións porque o estado anterior non afecta ao seguinte, é dicir, que saque un 2 nunha primeira tirada non condiciona o resultado da seguinte. Dado que nun proceso estocástico intervén o azar, sempre traballaremos usando probabilidades.

Un exemplo de proceso estocástico son as cadeas de Markov que teñen a peculiaridade de que un estado está condicionado polo estado anterior, esta característica recibe o nome de propiedade de Markov. En linguaxe matemática defínese como:

$$P[X_{n+1} = x_{n+1}/X_0 = x_0, X_1 = x_1, \dots, X_n = x_n] = P[X_{n+1} = x_{n+1}/X_n = x_n]$$

Andréi Markov (1856-1922) naceu en Riazán (Rusia) e dende pequeno xa amosou talento para as matemáticas. Estudou na Universidade de San Petesburgo e ao acabar a súa tese de doutoramento accedeu como adxunto á Academia de Ciencias de San Petesburgo recomendado polo seu profesor. Dez anos despois gañou o posto de académico regular. Markov foi o profesor que se ocupou dos cursos de teoría de probabilidade. Despois de vinte e cinco anos retirouse da universidade definitivamente. Markov dedicouse a investigar sobre distintas ramas das matemáticas como os traballos sobre fraccións continuas, a demostración do teorema central do límite e a creación das cadeas de Markov.

As cadeas de Markov empregan as matrices como ferramenta matemática para ordenar as probabilidades de transición dos estados nas filas e nas columnas. En cada posición da matriz colócase a probabilidade de transición entre dous estados. A probabilidade de transición do estado  $i$  ao estado  $j$  é valor  $P_{ij}$  na matriz de transición.

Para calcular as probabilidades de transición entre os estados en  $n$  pasos elévase a matriz de transición á potencia  $n$ . É dicir:

$$P^{(n)} = P^n$$

No noso taballo utilizaremos as cadeas de Markov para predecir os cambios metereolóxicos no mundo de Oz.

## 2. Análise dos datos

Neste fragmento da novela, *O marabilloso mago de Oz*, de Lyman Frank Baum, podemos observar os cambios na metereoloxía da zona e que podemos compilar nunha matriz as probabilidades de cada cambio

*A Terra de Oz foi bendicida con moitas cousas, pero non cun bo tempo. Nunca ten dous días asollados seguidos. Se ten un día asollado, é igual de probable que ao día seguinte haxa tanto neve como choiva. Se un día neva ou chove, ao día seguinte hai a mesma probabilidade de que o tempo siga igual ou de que cambie. Se cambia, só a metade das veces cambia a asollado.*

Do texto obtemos a nosa matriz de transición:

$$\begin{array}{c} \\ \\ \\ \end{array} \begin{array}{ccc} \text{S} & \text{C} & \text{N} \\ \text{S} & \begin{pmatrix} 0.00 & 0.50 & 0.50 \end{pmatrix} \\ \text{C} & \begin{pmatrix} 0.25 & 0.50 & 0.25 \end{pmatrix} \\ \text{N} & \begin{pmatrix} 0.25 & 0.25 & 0.50 \end{pmatrix} \end{array}$$

Calcularemos de xeito analítico a matriz estacionaria e avaliáremolo computacionalmente, cun programa de código R. Tamén calcularemos a distribución estacionaria, que é un vector que se mantén invariable ao aplicarlle matriz de transición, é dicir:

$$\pi = P\pi$$

sendo  $\pi$  a distribución estacionaria e  $P$  a matriz de transición. A matriz estacionaria é a matriz que agrega en filas a distribución estacionaria.

A partir de esta igualdade obtemos un sistema de ecuación:

$$\begin{cases} 0.25\pi_2 + 0.25\pi_3 = \pi_1 \\ 0.5\pi_1 + 0.5\pi_2 + 0.25\pi_3 = \pi_2 \\ 0.5\pi_1 + 0.25\pi_2 + 0.5\pi_3 = \pi_3 \end{cases}$$

Resolvemos o sistema utilizando o método de Gauss e obtemos os seguintes resultados:

$$\begin{cases} \pi_1 = 0.2 \\ \pi_2 = 0.4 \\ \pi_3 = 0.4 \end{cases} \Rightarrow \pi = (0.2, 0.4, 0.4)$$

Entón deducimos que a matriz estacionaria é:

$$P_e = \begin{pmatrix} 0.2 & 0.4 & 0.4 \\ 0.2 & 0.4 & 0.4 \\ 0.2 & 0.4 & 0.4 \end{pmatrix}$$

Un coñecido resultado teórico das cadeas de Markov establece que o límite das matrices de transición elevadas á potencia  $n$  cando  $n$  tende a infinito converge á matriz estacionaria:

$$\lim_{n \rightarrow \infty} P^n$$

Para comprobalo computacionalmente, creamos o seguinte código en R:

```
n<-as.integer(readline(prompt= "Expoñente da matriz: "))
vector<-c(0,0.25,0.25,0.5,0.5,0.25,0.5,0.25,0.5)
m<-matrix(vector, nrow=3, ncol=3)
library(Biodem)
p<-mtx.exp(m,n)
print(p)
```

Instalamos a librería *Biodem* que nos permite entre outras cousas elevar matrices a calquera potencia. Creamos un comando para que o programa solicite o valor enteiro ao que queres elevar a matriz. Establecemos un vector  $v$  coas probabilidades de transición e a partir deste dimensionamos a matriz coa función *matrix* establecendo as filas e columnas cos parámetros *nrow=* e *ncol=* respectivamente. Despois creamos unha variable  $p$  á que lle asignamos o valor da matriz elevada á potencia  $n$ , coa función *mtx.exp*. Por último utilizamos o comando *print (p)* para amosar a matriz resultante.

Probando con diversos valores de  $n$ , empezando no 2, elevando a matriz e vemos como os resultados se van achegando a nosa matriz estacionaria a medida que aumentamos os valores. Neste caso ao chegar a  $n=12$  a matriz converxe.

Resultados para:

$$n=2$$

	[,1]	[,2]	[,3]
[1,]	0.2500	0.3750	0.3750
[2,]	0.1875	0.4375	0.3750
[3,]	0.1875	0.3750	0.4375

$$n=6$$

	[,1]	[,2]	[,3]
[1,]	0.2001953	0.3999023	0.3999023
[2,]	0.1999512	0.4001465	0.3999023
[3,]	0.1999512	0.3999023	0.4001465

$$n=12$$

	[,1]	[,2]	[,3]
[1,]	0.2	0.4	0.4
[2,]	0.2	0.4	0.4
[3,]	0.2	0.4	0.4

Con este resultado vemos que efectivamente o límite converge á matriz estacionaria.

### 3. Conclusións

Deste artigo concluímos que o tempo no mundo de Oz pode ser estudado utilizando as cadeas de Markov, polo que é posible dimensionalmente facilmente como unha matriz facer predicións en termos de probabilidades.

Se analizamos a nosa matriz estacionaria observamos que o tempo que faga no instante inicial perde capacidade de influír nun período de tempo superior a 12 días, que é cando converge a matriz estacionaria. O tempo dentro de 20 días se hoxe neva e o resultado son as probabilidades de cada estado en 20 días.

$$(0, 0, 1) \begin{pmatrix} 0.2 & 0.4 & 0.4 \\ 0.2 & 0.4 & 0.4 \\ 0.2 & 0.4 & 0.4 \end{pmatrix} = (0.2, 0.4, 0.4)$$

Estes resultados serían exactamente iguais se o día no que nos atopamos fose soleado ou chuvioso.

Con este exemplo conseguimos predecir certas situacións aplicando as cadeas de Markov. Un caso interesante para analizar sería o tráfico dunha rotonda, estudando as probabilidades de que un coche que vén dun lugar vaia cara ao outro. Para facer un estudo rigoroso deste fenómeno teríamos que ter en conta moitas máis variables que afectan ao tráfico, como a época do ano ou se é festivo ese día, ao ter en conta todo isto aparecen cadeas de Markov moito máis complexas que as vistas neste artigo, abrindo así un enorme campo dentro das matemáticas.



#### **4. Bibliografía / Webgrafía**

Wikipedia. Wiki. Recuperado o 28 de novembro do 2022 do sitio web: [https://es.wikipedia.org/wiki/Cadena\\_de\\_M%C3%A1rkov](https://es.wikipedia.org/wiki/Cadena_de_M%C3%A1rkov)

Wikipedia. Wiki. Recuperado o 28 de novembro do 2022 do sitio web: [https://es.wikipedia.org/wiki/Andr%C3%A9\\_M%C3%A1rkov](https://es.wikipedia.org/wiki/Andr%C3%A9_M%C3%A1rkov)

Estadística DMA. Estadística e procesos estocásticos. Recuperado o 28 de novembro do 2022 do sitio web:

[https://estadistica-dma.ulpgc.es/EyPE/pdf/Tema%204-2\\_Cadenas\\_de\\_Markov.pdf](https://estadistica-dma.ulpgc.es/EyPE/pdf/Tema%204-2_Cadenas_de_Markov.pdf)

Cran project. Biodem: Biodemography Functions. Recuperado o 28 de novembro do 2022 do sitio web:

<https://cran.r-project.org/web/packages/Biodem/Biodem.pdf>



# Cadeas de Markov nun poema de Rosalía de Castro

*Sebio-Mazás, R.; Rey-Silva, A.;  
Méndez-Reimondéz, S.*

IES MARCO DO CAMBALLÓN

**Resumo.** Neste artigo escollemos un poema de Rosalía de Castro (*Negra sombra*) para analizar os patróns estatísticos asociados. Para isto empregamos as cadeas de Markov e mediante matrices de transición analizamos os modelos de combinación (vogal, vogal), (consoante, vogal), (vogal, consoante) e (consoante, consoante).

**Abstract.** In this article we chose a poem by Rosalía de Castro (*Negra sombra*) through which we wanted to demonstrate the influence of statistics in literature. To do so, we took Markov chains, using transition matrices to analyze the patterns of combination (vowel, vowel), (vowel, consonant) and (consonant, consonant).

## 1. Introducción

Defínese como proceso estocástico ao conxunto de variables aleatorias que evolúen de xeito aleatorio ao longo do tempo. Un tipo de proceso estocástico son as cadeas de Markov.

As cadeas de Markov defínense coma unha secuencia de variables aleatorias que representan os estados dun determinado sistema durante unha serie de intervalos de tempo, de modo que o estado do sistema no intervalo actual depende do seu estado no intervalo inmediato anterior. Cada valor que toma unha variable aleatoria chámase estado, e dependendo da relación entre eles, xéranse distintos tipos de matrices.

Andrei Andreyevich Markov naceu o 14 de xuño de 1856 en Ri-azán, (Rusia) e morreu o 20 de xullo de 1922 en Petrogrado, o actual San Petersburgo. Graduouse na Universidade de Física e Matemáticas de San Petersburgo en 1878 e obtivo o doutoramento en 1884. En 1900 Markov comezou a interesarse pola teoría da probabilidade, materia na que obtivo resultados moi brillantes, entre eles a creación en 1907 das cadeas que levan o seu nome.

As cadeas de Markov teñen numerosas e variadas utilidades. Pódense empregar na física estatística e termodinámica, na predición da meteoroloxía, para calcular a evolución dunha pandemia ou para predicir probabilidades en certos xogos de azar.

Neste traballo pretendemos aplicarlas na busca dun patrón literario no poema *Negra Sombra* de Rosalía de Castro, clasificando as combinacións entre as letras tendo en conta a precedente e creando unha matriz de transición asociada, tal e como fixera Markov co poema *Eugene Onegin* de Pushkin.

O método que empregou Markov consistía en recompilar as primeiras 20.000 letras do poema e organizalas en 200 bloques de 10·10 letras, contando as vogais e consoantes en cada fila e cada columna. Calculou medias e varianzas para obter as medidas de dispersión e a continuación clasificounas en pares de letras sucesivas. A conclusión que sacou foi clara: as letras non son independentes no poema e cada unha mostra unha dependencia do tipoloxía da letra anterior.

## 2. Análise dos datos

Para comezar, analizaremos as probabilidades de transición entre as variables aleatorias que se establecen entre as 325 letras do poema *Negra Sombra* de Rosalía de Castro, onde podemos observar catro situacións distintas:

*Situación 1* (V-V) transición dunha vogal a outra vogal.

*Situación 2* (V-C) transición dunha vogal a unha consoante.

*Situación 3* (C-V) transición dunha consoante a unha vogal.

*Situación 4* (C-C) transición dunha consoante a outra consoante.

En función destes resultados crearemos unha **matriz de transición** cos datos obtidos no poema. Realizamos unha estimación frecuentista das transicións, sen contar os cambios de versos, cuxos resultados nos darán a seguinte matriz de transición (P) :

$$P = \begin{matrix} & \begin{matrix} V & C \end{matrix} \\ \begin{matrix} V \\ C \end{matrix} & \begin{pmatrix} \frac{30}{146} & \frac{116}{146} \\ \frac{119}{164} & \frac{45}{164} \end{pmatrix} \end{matrix} = \begin{pmatrix} 0,21 & 0,79 \\ 0,73 & 0,27 \end{pmatrix}$$

A continuación introduciremos estes datos en R:

```
P = matrix(c(0.21,0.79,0.73,0.27),nrow = 2,byrow = TRUE)
```

```
library(matrixcalc)
```

Neste código introduciremos a matriz (P) mediante a librería *matrixcalc*. Organizaremos a matriz en 2 columnas mediante o comando *nrow* e en dúas columnas co comando *byrow*.

Calculamos o vector de estado estacionario, obtendo (0.5,0.5). A continuación, implementamos un código que calcule a matriz estacionaria en R. Para iso optamos por multiplicamos a matriz estacionaria por si mesma, a fin de estabilizar o seu comportamento ao longo do tempo.

```
library(matrixcalc)
library(markovchain)
P = matrix(c(0.21,0.79,0.73,0.27),nrow = 2,byrow = TRUE)
R=matrix.power(P,100)
print(R)
```

Co comando *matrix.power* elevamos a matriz a 100 e a continuación imprimímolos na pantalla a través do comando *print*. Ao elevala sucesivamente obtemos aproximamos a matriz estacionaria (que ten, por filas, o vector estacionario), e canto maior sexa o exponente, maior será a exactitude dos resultados. Elevando a matriz á centésima obteremos estes resultados.

```
      [,1]      [,2]
[1,] 0.4802632 0.5197368
[2,] 0.4802632 0.5197368
```

### 3. Conclusións

Tras recopilar as relacións entre vogais e consoantes e facer unha matriz de transición coas probabilidades de cambio, concluímos que, tendo en conta só o suceso anterior, existe unha maior probabilidade de atopar un grupo de vogal/consoante e viceversa que atopar os grupos con letras do mesmo tipo.

Porén, ao facer a matriz estacionaria observamos que as relacións entre os patróns de combinación tenden a igualarse. É dicir, se elevamos a matriz de transición a un número elevado para simular un poema máis largo e coa mesma matriz de transición inicial, as probabilidades de combinación de calquera dos patróns posibles é a mesma. A matriz alcanza uns valores de 0.5 en cada un dos seus estados.

A diferenza dos patróns achados no poema de Rosalía, Markov determinou no poema *Eugene Onegin* distintas relacións de combinación que difiren un pouco das nosas, xa que no seu poema hai unha maior probabilidade de cambios entre vogal/consoante (0.87) e menor entre consoante/vogal (0.66).

Sería interesante analizar outro poema en ruso e outro en galego de outros poetas diferentes para poder concretar se estas diferenzas se deben ao léxico de cada idioma ou á diferenza estilística dos autores.

#### **4. Bibliografía / Webgrafía**

- Biografía de Andréi Markov. Recuperado o 21 de novembro do 2022 :  
<https://www.biografiasyvidas.com/biografia/m/markov.htm>
- INFOAMÉRICA | Andrei A. Markov. Recuperado o 21 de novembro do 2022 :  
<https://www.infoamerica.org/teoria/markov1.htm>
- DEFINICIÓN DE MATRIZ. Recuperado o 21 de novembro do 2022 :  
[http://recursostic.educacion.es/descartes/web/materiales\\_didacticos/Calculo\\_matricial\\_d3/defmat.htm](http://recursostic.educacion.es/descartes/web/materiales_didacticos/Calculo_matricial_d3/defmat.htm)
- Capítulo 10 Cadenas de Markov. Recuperado o 21 de novembro do 2022 :  
[https://www.ugr.es/~bioestad/\\_private/cpfund10.pdf](https://www.ugr.es/~bioestad/_private/cpfund10.pdf)
- De cómo la poesía dio lugar a las cadenas de Markov (III) – Antonio Gómez Corral. Recuperado o 21 de novembro do 2022 : <https://blogs.mat.ucm.es/agomez-corral/2021/08/12/de-cómo-la-poesía-dio-lugar-a-las-cadenas-de-markov-iii/>
- Distribución Límite de una Cadena de Markov en Tiempo Discreto. Recuperado o 21 de novembro do 2022 : <https://www.gestiondeoperaciones.net/cadenas-de-markov/distribucion-limite-de-una-cadena-de-markov-en-tiempo-discreto/>
- CONCEPTOS | INVESTIGACIÓN DE OPERACIONES II . Recuperado o 21 de novembro do 2022: <https://investigaciondeoperaciones2.wordpress.com/2010/11/15/conceptos-2/>
- Proceso estocástico - Qué es, definición y concepto | 2022 | Economipedia. Recuperado o 21 de novembro do 2022:  
<https://economipedia.com/definiciones/proceso-estocastico.html>
- Matrices de transición y patrones de variabilidad cognitiva. Recuperado o 21 de novembro do 2022 :  
[http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S1657-92672013000200021#:~:text=T%C3%A9nicamente%2C%20una%20matriz%20de%20transici%C3%B3n,Ap%C3%A9ndice%20%2C%20Ecuaci%C3%B3n%201](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1657-92672013000200021#:~:text=T%C3%A9nicamente%2C%20una%20matriz%20de%20transici%C3%B3n,Ap%C3%A9ndice%20%2C%20Ecuaci%C3%B3n%201)
- Capítulo 10 Cadenas de Markov





# *Método de Newton-Raphson aplicado á ecuación de Kepler*

*Álvarez-Lodeiro, I.; Amosa-Martínez, U.;  
Casal-Blanco, R.; Ferro-Pérez, J.M.*

IES MARCO DO CABBALLÓN

Resumo. Neste artigo traballaremos coa ecuación de Kepler, que describe o movemento dos astros arredor do Sol. Para resolvela empregaremos o método de Newton-Raphson, que acha as raíces de funcións reais. Todo isto faremo a través dun código creado no programa R.

Abstract. In this article, we will try to work with the Kepler's equation, which describe the movement of the celestial bodies around the Sun. To solve it we will use the Newton-Raphson method, which find the roots of real functions. This will be done with a code made in R program.

## **Introdución**

O método de Newton-Raphson recibe este nome gracias ás dúas persoas ás que se debe a súa elaboración, os matemáticos Isaac Newton (que o incluíu nunha publicación escrita no ano 1671 e que viu a luz no 1736) e Joseph Raphson (que o incluíu nun libro publicado no ano 1691).

Trátase dun procedemento matemático que permite achar as raíces dunha función, partindo dun valor próximo á mesma. En xeral, o método resolve unha ecuación do tipo  $f(x)=0$ . Este é un método de rápida converxencia, isto é, que require, polo xeral, poucas repeticións para obter o resultado que se busca.

O método baséase no proceso iterativo definido pola seguinte expresión:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Porén, ao principio Newton tan só aplicaba o método a polinomios e non tiña en conta as sucesivas aproximacións  $x_n$ , errando ao ver o método como simplemente alxébrico e non atoparlle conexión co cálculo.

O método trata de achegarse á solución mediante a recta tanxente da función en determinados puntos  $x_n$ , partindo do valor inicial  $x_0$  que nós introducimos e collendo como  $x_{n+1}$  o punto onde a recta tanxente en  $x_n$  corta ao eixo de abscisas. Así pois, o proceso sería o seguinte:

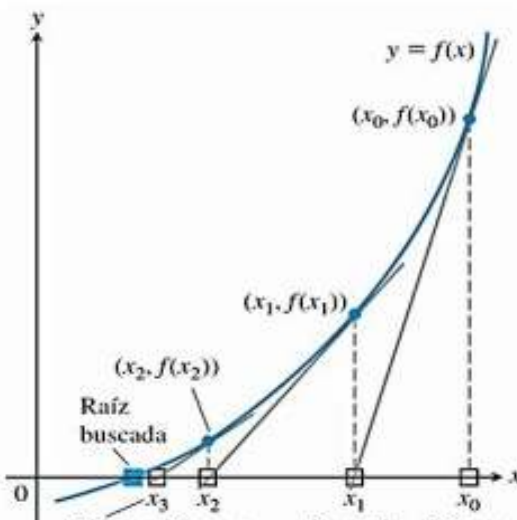


Fig.3 Proceso gráfico do algoritmo de Newton-Raphson.

Como se dixo anteriormente, o método chega con relativa rapidez á raíz buscada na maioría de ocasións. Con todo, existen casos onde aparecen numerosos problemas. Un deles é cando a función ten múltiples raíces, pois o algoritmo tan só é capaz de localizar unha delas. Cando o valor está moi próximo a un punto de inflexión da función, isto provoca que a recta tanxente corte o eixo de abscisas nun punto alonxado da solución que se está a buscar. Tamén cando o valor coincide cun máximo ou un mínimo da función, pois a recta tanxente será horizontal e nunca cortará o eixo de abscisas.

Para evitar estes problemas cómpre comezar a iteración cun valor inicial o suficientemente próximo á raíz.

Unha vez introducido o método de Newton-Raphson, é preciso coñecer tamén que é a ecuación de Kepler, que é a que imos resolver aplicando este método, e cales foron os resultados que o levaron a enunciála. Johannes Kepler foi un matemático e astrónomo alemán de finais do século XVI e comezos do século XVII e é un dos pais da astronomía. Entre os seus logros destacan melloras nos telescopios da época e sobre todo as súas tres leis sobre o movemento dos planetas arredor do Sol, as cales deduciu a partir das observacións de Tycho Brahe.

A primeira lei de Kepler dinos que os planetas seguen unha órbita elíptica nun plano, estando o Sol nun dos focos da elipse. A excentricidade de dita elipse vén dada pola expresión

$$e = \frac{c}{a}$$

onde  $c$  é a semidistancia focal (isto é a distancia desde o foco ao centro da elipse) e  $a$  é o semiexco maior (a maior distancia entre o centro da elipse e a propia elipse). A maioría das órbitas planetarias teñen un valor moi pequeno de excentricidade, é dicir,  $e \approx 0$ . Isto significa que a nivel práctico poden considerarse órbitas circulares descentradas.

A segunda lei de Kepler, ou lei de áreas, dinos que o raiovector que une o planeta co Sol varre áreas iguais en tempos iguais. Esta lei implica que a velocidade areolar, que se define como a área varrida polo vector de posición dun corpo por unidade de tempo, sexa constante:

$$\vec{v}_A = \frac{d\vec{A}}{dt} = cte$$

Ademais, esta lei é equivalente á conservación do momento angular  $\vec{L}$ . Dita magnitude física vén descrita da seguinte forma:  $\vec{L} = m \cdot \vec{r} \times \vec{v}$ . A conservación da mesma implica que o momento angular no afelio (punto máis alonxado do Sol) é igual que no perihelio (punto máis próximo ao Sol). Dado que a masa é a mesma en

ambos puntos, o astro debe ter maior velocidade onde a distancia é menor para que se verifique esta igualdade. Nos casos de astros con órbitas de moi pouca excentricidade como a do planeta Marte ( $e = 0.0093$ ) a diferenza de velocidades é pequena: a velocidade no perihelio é 1,21 veces a súa velocidade no afelio. Noutros astros coma no cometa Halley ( $e = 0.967$ ), a segunda lei de Kepler fai que as velocidades varíen moito, pois a súa velocidade no perihelio é 61,4 veces a velocidade no afelio.

A terceira lei de Kepler, tamén coñecida como harmónica ou de períodos, relaciona os períodos (é dicir, o tempo que tardan os astros en completar unha volta ao redor do Sol) co raio medio da súa órbita. Para un planeta dado, o cadrado do período orbital é proporcional ao cubo da súa distancia media ao sol:

$$T^2 = k \cdot r^3$$

onde  $T$  é o período do planeta,  $k$  é unha constante igual para todos os corpos e  $r$  é a distancia media ao Sol.

Esta terceira lei tería unha gran importancia na física, pois foi a partir do seu estudo o que lle permitiu a Newton formular a súa lei de gravitación universal.

A partir destas leis, sería o propio Kepler o que formulou unha ecuación que predí a situación exacta dun planeta cunha órbita elíptica e que cunha velocidade areolar constante. Dita expresión é a seguinte:

$$M = E - e \cdot \text{sen}(E)$$

onde  $M$  é a anomalía media (é dicir, o ángulo que percorrería un planeta ficticio que realiza un movemento circular uniforme por unha órbita circular de raio igual ao semieixo maior),  $e$  é a excentricidade da órbita do astro e  $E$  é a anomalía excéntrica (isto é, o ángulo que varre a proxección do planeta real sobre a órbita circular ficticia).

Como pode advertirse facilmente, o principal problema desta ecuación é que non ten unha resolución sinxela por procedementos usuais, polo que se fai necesario recorrer a métodos numéricos.

## 2. Análise dos datos

En primeiro lugar, precisamos crear o código do método de Newton-Raphson para poder resolver calquera ecuación. Comezamos por gardar nunha variable  $f$  a ecuación que queremos resolver coa función *expression*, para que a detecte como unha expresión alxebrica.

Agora precisamos calcular a derivada da función para poder traballar con ela. Para iso usamos a función propia  $D(f, 'x')$ , que nos permite derivar expresións. Neste caso queremos derivar respecto de  $x$  (a nosa incógnita) a función gardada previamente en  $f$ , por iso introducimos dentro da paréntese eses parámetros.

```
f<-expression(x-5)
df<-D(f, 'x')
```

Unha vez asignamos un valor inicial a  $x$ , creamos un bucle *while* onde se vai repetir o algortimo de Newton-Raphson ata atopar o resultado, polo que a condición para finalizar o bucle será  $eval(f) \neq 0$  (isto quere dicir que o bucle rematará cando a función avaliada no punto  $x$  usado nesa iteración sexa 0). Secasí, como en ocasións é moi complicado chegar á solución exacta, utilizamos unha condición menos esixente:  $eval(f) > 0.001$  no caso de que nos acheguemos pola parte positiva da función, e  $eval(f) < -0.001$  no caso de achegarse por valores negativos.

Unha vez dentro do bucle *while* precisamos que o programa realice o algoritmo do método de Newton-Raphson, polo que introducimos a fórmula adaptándoa ao código de R:

```
while(eval(f) != 0){
  x<-x-(eval(f)/eval(df))
```

onde  $eval(f)$  é a función avaliada no punto  $x$  e  $eval(df)$  a súa derivada avaliada nese mesmo punto.

Por último só queda pedir que nos devolva a raíz mediante a función *print*.

```
f<-expression(x-5)
df<-D(f, 'x')

x<-10

while(eval(f)!=0){
  x<-x-(eval(f)/eval(df))
}

print(x)
```

Á hora de traballar coa ecuación de Kepler  $M = E - e \cdot \text{sen}(E)$ , precisamos primeiro adaptala á forma  $f(x)=0$ . Daquela, a función coa que imos traballar será  $f(E) = M + e \cdot \text{sen}(E) - E$ , onde  $E$  é a nosa variable e  $M$  e  $e$  son datos coñecidos de distintos astros.

Así, segundo os datos que achamos na rede sobre a anomalía media ( $M$ ) e a excentricidade ( $e$ ) dos astros do Sistema Solar, obtivemos os valores da anomalía excéntrica ( $E$ ), cos que elaboramos a seguinte táboa:

Planeta	Ano. Media	Excentr.	Ano. Excént.
Mercurio	3,0368732333	0,20563069	3,054715
Venus	0,8726647222	0,00677323	0,8778758
Terra	6,2308261167	0,01671123	6,229936
Marte	0,233	0,09	0,7963221
Xúpiter	0,3141593	0,04839266	0,3298329
Saturno	5,5850542222	0,05648	5,543263
Urano	2,4783678111	0,04440559	2,504773
Neptuno	4,6600296167	0,00858587	4,651449
Plutón	0,2443461222	0,244	0,3214319

Ademais decidimos traballar tamén con obxectos do Sistema Solar que presentan valores extremos de excentricidade da órbita, co propósito de forzar posibles problemas no algoritmo do método. Para eles obtivemos os seguintes resultados:

Corpo celeste	Ano. Media	Excentr.	Ano. Excént.
Tritón	6,143559644	2E-05	6,143556
11/Oumuamua	0,436332361	1,19951	1,633486
Halley	0,669159309	0,96799	1,635146

Para medir o número de iteracións requiridas polo código á hora de atopar unha raíz, creamos unha nova variable á que chamamos *cont* e que imos usar a modo de contador. Para iso asignamos como valor inicial o 0 e dentro do bucle *while* pedimos que vaia sumando 1 cada vez que se realiza unha iteración. Por último pedimos que nos devolva o valor do contador coa función *print*.

```
f<-expression((0.213) *pi+0.96799* sin(x)-x)
df<-D(f, 'x')
x<-pi

w<-c()
cont<-0
while(eval(f)!=0){
  cont<-cont+1
  x<-x-(eval(f)/eval(df))
  w<-append(w,x)
}

print(x)
print(cont)
```

Así pois, cos resultados elaboramos a seguinte gráfica onde vemos o valor inicial (no noso caso é  $\pi$ , pero podería ser calquer outro valor), a raíz e as iteracións precisas para chegar a ela.

Planeta	Ano. Media	Excentr.	Ano. Excént.	Valor Inicial	Iteracións
Mercurio	3,0368732333	0,20563069	3,054715	3,14159265359	3
Venus	0,8726647222	0,00677323	0,8778758	3,14159265359	4
Terra	6,2308261167	0,01671123	6,229936	3,14159265359	4
Marte	0,233	0,09	0,7963221	3,14159265359	4
Xúpiter	0,3141593	0,04839266	0,3298329	3,14159265359	5
Saturno	5,5850542222	0,05648	5,543263	3,14159265359	4
Urano	2,4783678111	0,04440559	2,504773	3,14159265359	3
Neptuno	4,6600296167	0,00858587	4,651449	3,14159265359	3
Plutón	0,2443461222	0,244	0,3214319	3,14159265359	6

Corpo celeste	Ano. Media	Excentr.	Ano. Excént.	Valor Inicial	Iteracións
Tritón	6,143559644	2E-05	6,143556	3,14159265359	3
11/Oumuamua	0,436332361	1,19951	1,633486	3,14159265359	6
Halley	0,669159309	0,96799	1,635146	3,14159265359	6

Realizando estes estudos demos cun dos problemas xa citados do método de Newton-Raphson. Para o caso do cometa 11/Oumuamua obtemos a seguinte función a partir da súa ecuación de Kepler:

$$f(x) = (5/36)\pi + 1.19951 \operatorname{sen}(x) - x$$

Dita función presenta un punto de inflexión en  $x=6,128552$  (punto marcado sobre ela), polo que ao comezar o algoritmo do método cun  $x_0=2\pi$  (liña vertical de puntos), a recta tanxente corta o eixo de abscisas no punto  $x=35.58925$ , polo que ao realizar unha iteración alonxámonos do punto de corte da función co eixo. Neste caso, sería preciso realizar as 8 iteracións fronte ás 6 que precisa ao ter  $x_0=\pi$ .

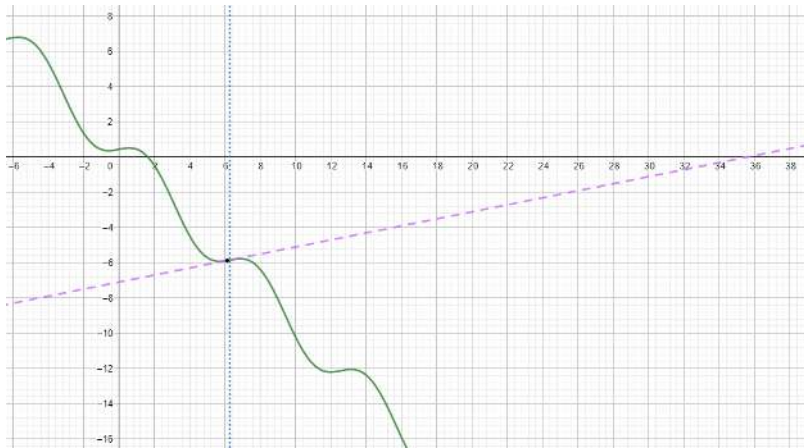


Fig.12 Caso particular do 11/Oumuamua.



### 3. Conclusións

Dado que a ecuación de Kepler ten a forma  $M = E - e \cdot \text{sen}(E)$ , a natureza da función asociada á ecuación vai depender moito do valor da excentricidade  $e$ . Para corpos cunha excentricidade máis pequena, a función vai semellar unha recta, pois vai facer que o comportamento sinusoidal sexa case insignificante. Por outra banda, con valores moi elevados da mesma a función pasa a ter un comportamento sinusoidal máis acusado.

Resolvendo a ecuación de Kepler con distintos valores tanto de anomalía media como sobre todo de excentricidade, podemos sacar algunhas conclusións sobre o método de Newton-Raphson para resolver ecuacións.

En primeiro lugar, cando se trata de ecuacións semellantes a unha recta (cun valor de excentricidade moi pequeno) o método chega a resultados exactos sen necesidade de reiterar moitas veces o algoritmo e podendo comezar cun valor de  $x_0$  relativamente alonxado da raíz.

Porén, cando a función ten un comportamento sinusoidal o método comeza a dar certos problemas, pois ou ben non chega a resultados tan exactos ou ben require dun maior número de iteracións para chegar a eles e ademais precisamos comezar cun valor inicial bastante próximo á raíz que buscamos. Isto sucede debido a que é máis probable que o noso  $x_0$  se atope preto dun punto de inflexión ou un máximo, facendo que a recta tanxente se alonxe da raíz que buscamos.

## 4. Webgrafía

Dinámica celeste (2016). *El problema de Kepler*

Recuperado o 28 de abril do 2022, do sitio web:

<http://www.sc.ehu.es/sbweb/fisica3/celeste/kepler/kepler.html>

Veníamos abiertas al universo(s.d.). *Kepler: El método y las leyes*

Recuperado o 28 de abril do 2022, do sitio web:

<https://media4.obsmp.fr/public/VAU/masa/gravitacion/kepler/kepler-leyes/OB-SERVER.html>

Fiscalab(2022). *Leyes de Kepler*

Recuperado o 28 de abril do 2022, do sitio web:

<https://www.fiscalab.com/apartado/leyes-kepler>

Vargas Cantero Jose Enrique, *Método de Newton Raphson* (2018)

Recuperado o 28 de abril do 2022, do sitio web:

<http://personal.cimat.mx:8181/~julio/courses/progra01/clase18/MetododeNewton-Raphson.pdf>

Miana Pedro J. ; Romero Natalia, (2010) *La historia de la conjetura de Kepler*; Recuperado o 28 de abril do 2022, do sitio web:

<https://www.divulgameteo.es/fotos/meteoroteca/Conjetura-Kepler.pdf>

Wikipedia (2022) . *Método de Newton*

Recuperado o 28 de abril do 2022, do sitio web:

[https://es.wikipedia.org/wiki/M%C3%A9todo\\_de\\_Newton](https://es.wikipedia.org/wiki/M%C3%A9todo_de_Newton)

Porque todo tende a infinito gaussianos (2013), *La historia del método de Newton Raphson y otro caso más de mala documentación en el cine*

Recuperado o 28 de abril do 2022, do sitio web:

<https://www.gaussianos.com/la-historia-del-metodo-de-newton-raphson-y-otro-caso-mas-de-mala-documentacion-en-el-cine/>

Bussons Gordo, Javier (2008) *Ecuación de Kepler*

Recuperado o 28 de abril do 2022, do sitio web:

<https://webs.um.es/bussons/ecKepler.pdf>

# *Análise dos parámetros da onda estacionaria*

*Fraiz-Casas, A.; López-Calviño, E.;  
Puga-Silva, L.; Silva-Lodeiro, X.;  
Verde-Salgado, N.*

IES MARCO DO CABBALLÓN

**Resumo.** No seguinte estudo resolvemos a ecuación da onda estacionaria en función dos parámetros que a definen, concretamente a súa velocidade e a súa lonxitude. Para isto, creamos empregaremos o método de Newton-Raphson un código escrito no programa de análise estadístico R no que se desenvolve o método de Newton-Raphson.

**Abstract.** In the following research we resolve the stationary wave equation based on the parameters that define it, especially its speed and length. For this purpose, we create a code using the statistical analytics program R using the Newton-Raphson method.

## **1. Introducción**

O movemento ondulatorio ou movemento de propagación da onda defínese como unha perturbación que se propaga dun punto a outro sen que exista transporte neto de materia, pero si transmisión de enerxía. Un exemplo no que se demostra isto prodúcese cando unha pedra cae a un estanque e se xera unha perturbación. Esta alteración é unha onda que se estende en forma de círculos concéntricos situados a certa distancia do punto no que cae a pedra.



Fig. 1. Representación do movemento ondulatorio

De forma xeral, as ondas réxense por unha serie de características comúns a todas elas. En primeiro lugar, é preciso unha fonte ou foco emisor que actúe como orixe da perturbación. A enerxía deste é transmitida ao medio de propagación nas súas inmediacións. No noso caso, a pedra que cae transfere a súa enerxía ás moléculas de auga que se encontran próximas. Por outro lado, debe existir un medio de propagación que, a medida que é atravesado pola perturbación, experimenta unha variación temporal e reversible nalgunhas das súas propiedades físicas. No exemplo proposto, cando a onda acada as partículas de auga do estanque, que estaban en repouso, estas empezan a vibrar alterando a súa posición cara a ambos lados do seu punto de equilibrio, voltando, ao cabo dun anaco, ao seu estado inicial. Finalmente, cada punto do medio transmite a perturbación a puntos veciños, polo que podemos afirmar que o fenómeno ondulatorio é unha forma cooperativa de propagación da enerxía que se transmite entre o foco e os puntos acadados.

Segundo o tipo de criterio empregado existen distintas clasificacións para as ondas. Por unha banda, en función do tipo de enerxía propagada distinguimos entre as ondas mecánicas, nas que se transmite enerxía mecánica como o exemplo anterior do estanque; e as electromagnéticas, nas que se propaga a enerxía electromagnética producida por oscilacións de campos eléctricos e magnéticos como a luz que ten como medio de propagación máis favorable o baleiro. Por outra banda, en función da dirección de propagación, diferéncianse entre as ondas lonxitudinais, nas que a dirección de vibración coincide coa dirección de propagación; e as transversais, nas que a dirección de transmisión é perpendicular entre si. Ademais, existen ondas en función do número de dimensións de propagación. Estas poden ser unidimensionais, é dicir, a enerxía propágase principalmente nunha dimensión como a dunha corda; bidimensionais, isto é, a enerxía propágase principalmente en dúas dimensións, por exemplo, na superficie da auga dun estanque e as ondas tridimensionais nas que a enerxía se transmite principalmente en tres dimensións, como a luz ou o son.

## 2. Caracterización da onda estacionaria

A onda estacionaria é un caso particular de interferencia que se produce cando se superpoñen dúas ondas da mesma dirección, amplitude e frecuencia, pero sentido contrario. Nunha onda estacionaria os distintos puntos que a conforman oscilan ao redor da súa posición de equilibrio denominado nodo e os puntos que flutúan e chegan a amplitude máxima chámanse ventres. A medida que transcorre o tempo o patrón da onda non se move, de aí o seu nome. Non todos os medios de propagación son capaces de producir de xeito natural ondas estacionarias.

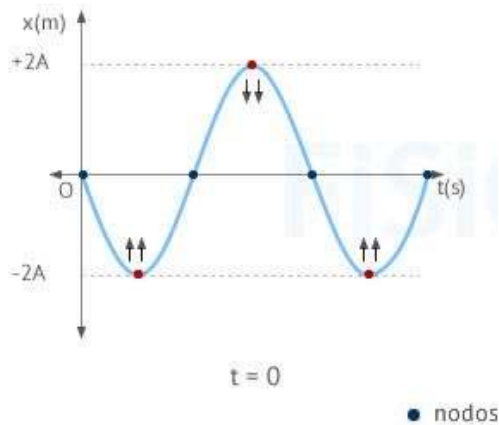


Fig. 2. Representación onda estacionaria

A ecuación pola que se rexe esta onda é a seguinte:

$$y = 2 \cdot A \cdot \sin(k \cdot x) \cdot \cos(\omega \cdot t)$$

sendo  $k = 2\pi/\lambda$  e  $\omega = 2\pi\nu/\lambda$

O problema que nos atinxe focalízase nas ondas do mar. Estas compórtanse de xeito análogo ás ondas sísmicas e propáganse entre a atmosfera e o océano. Cando unha onda pasa por augas profundas, as moléculas de auga voltan case ao mesmo lugar onde orixinalmen-

te se atopaban. A partir disto, podemos constatar que se produce un pequeno desprazamento da auga na dirección de propagación, dado que en cada oscilación unha molécula non volta exactamente ao mesmo punto, senón que o fai a outro lixeiramente máis adiantado.

No momento en que a onda se aproxima cara a costa, as súas características modifícanse cando a profundidade da auga empeza a ser menor que a semilonxitude da onda. Tamén se ve modificada polos efectos da refracción, isto é, a velocidade e a lonxitude da onda diminúen, mentres que a pendente da onda se esaxera facéndose inestable ata romper. Ademais, ao atoparse cun obstáculo na superficie, modifícase segundo os fenómenos de difracción e reflexión, e mesmo pode cambiar por efecto dun obstáculo mergullado.

Deseguido, centrarémonos nos parámetros que se empregan para así poder comprender a ecuación da onda do mar. A parte máis alta desta coñécese como “crista” mentres que a máis profunda da depresión entre dúas ondas chámase “val”. Por outro lado, a distancia entre dúas cristas denomínase “lonxitude da onda” ( $\lambda$ ). A amplitude ( $A$ ) é a metade da distancia entre a crista e o val. A altura da onda comeza sendo igual á amplitude e vaise reducindo segundo esta se amortece. A pendente ( $\delta$ ) é o cociente da altura e a lonxitude da onda réxese pola seguinte fórmula:

$$\delta = H / \lambda$$

Chámase ( $T$ ) ao tempo que tarda en percorrer unha lonxitude da onda. A velocidade da onda, é dicir, a velocidade de propagación, calcúlase dividindo a lonxitude da onda polo período:

$$v = \lambda / T$$

Cómpre destacar que en augas profundas a velocidade da onda é proporcional á lonxitude da onda; pola contra, en augas moi superficiais depende só da profundidade. Dun xeito gráfico, a onda queda representada do seguinte xeito:

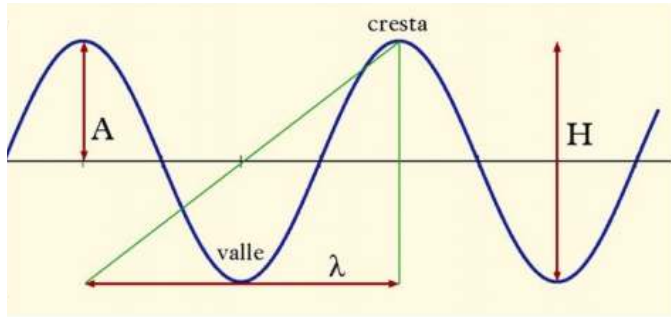


Fig. 3. Gráfica coas partes da onda estacionaria

A ecuación da onda de mar é a seguinte:

$$h = h_0 \left[ \text{sen} \left( \frac{2\pi x}{\lambda} \right) \cos \left( \frac{2\pi t v}{\lambda} \right) + e^{-x} \right]$$

Fig.4. Ecuación onda estacionaria

### 3. Método de Newton-Raphson

Debido á natureza da expresión anterior, que non permite a súa resolución con métodos de cálculo ordinarios, é necesario o uso dun algoritmo que permita a aproximación das raíces ou ceros de ecuación da forma  $f(x)=0$ . Neste caso usaremos o método de Newton-Raphson.

Este método foi descrito por Isaac Newton (1643-1727) en *De analysi per aequationes número terminorum infinitas*, escrito en 1669 e publicado en 1711 por William Jones e en *Methodus fluxionum et serierum infinitorum*, escrito en 1671, traducido e publicado en 1736 por John Colson. Newton aplicou o método só a polinomios e non consideraba as aproximacións sucesivas  $x_n$ , senón que calculaba unha secuencia de polinomios para chegar á aproximación da raíz  $x$ .

Este método é un algoritmo eficiente para encontrar aproximacións dos ceros ou raíces dunha función real e está baseado na serie de Taylor. O polinomio de Taylor é unha aproximación polinómica

dunha función  $n$  veces derivable nun punto concreto, é dicir, é unha suma finita de derivadas locais avaliadas nun punto concreto. O método de Newton tamén é empregado para achar o máximo e o mínimo dunha función, atopando os ceros na súa primeira derivada. Trátase dun método aberto, polo que a súa converxencia global non está garantida e a única maneira de acadala é seleccionar un valor inicial o suficientemente próximo á raíz buscada. Polo tanto, débese comezar a iteración cun valor próximo a cero, denominado como punto de arranque ou valor suposto. A relativa proximidade do punto inicial á raíz depende da natureza da propia función; no caso de que esta presente múltiples puntos de inflexión ou pendentes grandes no entorno da raíz, as posibilidades de que o algoritmo diverxa aumentan. Polo tanto, é necesario seleccionar un valor próximo á raíz. Na seguinte figura pódese observar que, aínda partindo dun punto próximo á raíz buscada, existe un caso no que o método converge e noutro non:

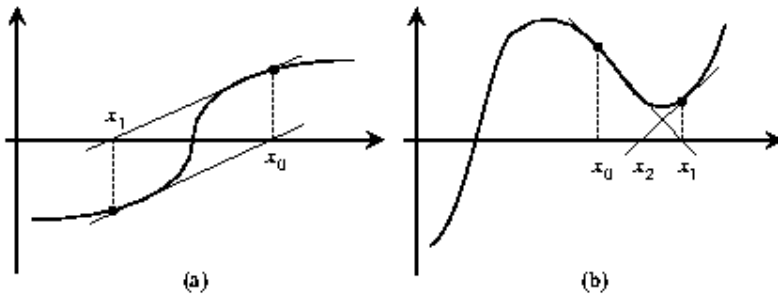


Fig. 5. Gráficas nas que o método de Newton-Raphson non funciona adecuadamente.

Isto ocorre debido a que a súa converxencia é cadrática, polo que se nalgún momento o erro é menor ou igual a 0.1, a cada nova iteración dobramos, aproximadamente, o número de decimais exactos. Polo tanto, tómase o erro relativo como se a última aproximación fora un valor exacto e detense o proceso iterativo no momento no que este erro é aproximadamente menor que a cantidade fixada anteriormente.

Unha vez feito isto, o método aproxima a función pola recta tan-



xente nese valor e a abscisa no orixe da recta será, segundo este método, unha mellor aproximación da raíz que o valor anterior. Finalmente, realízanse sucesivas iteracións ata que o método converxa. A continuación, podemos observar nas seguintes representacións gráficas diferentes supostos:

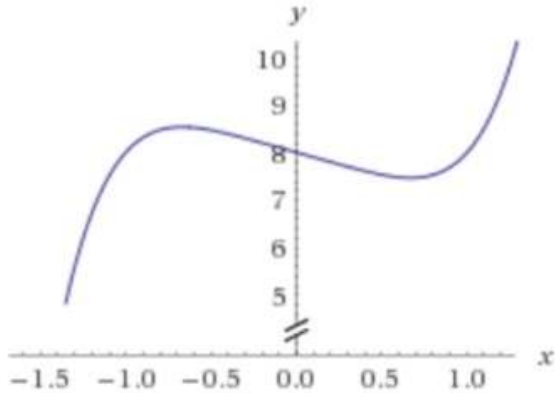


Fig. 6. Gráfico cun caso de éxito onde  $f(x) = x^5 - x + 8$

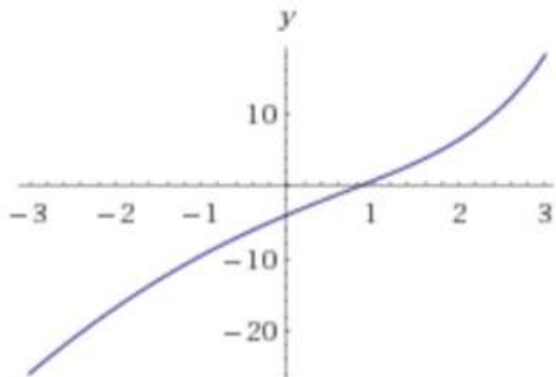


Fig. 7. Gráfico cun caso onde falla  $f(x) = \ln(x+1) + 1$

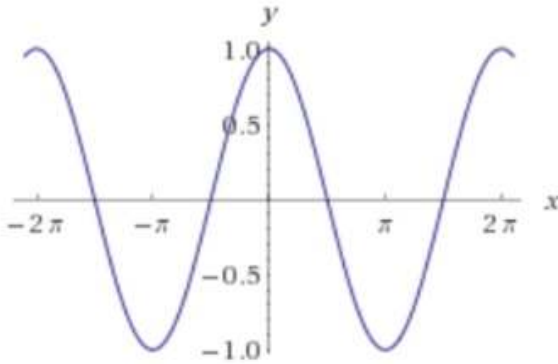


Fig. 8. Gráfico caso fronteira onde  $f(x)=\cos(x)$

Outro método de resolução de equações é o método da secante, que se trata dunha simplificación do método de Newton-Raphson. En vez de tomar a derivada da función da raíz que se quere encontrar, aproxímase por unha recta secante á curva, e a pendente é semellante á derivada no punto inicial. A principal diferenza co método anterior é coñecer dous puntos da función para poder xerar dita recta. Sexan  $x_0$  e  $x_1$  pertencentes a certa  $f(x)$ , podemos aproximar o valor da derivada a través da pendente da recta secante á curva nos puntos  $(x_0, f(x_0))$  e  $(x_1, f(x_1))$ , do seguinte xeito:

$$f'(x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Fig. 9. Fórmula do método de Newton-Raphson

Se substituímos esta expresión na ecuación do método de Newton, obtemos a expresión do método da secante:

$$X_{n+1} = x_n - \frac{f(x_n) (x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Fig. 10. Fórmula método da secante



corresponde a amplitude inicial (h0) e a “h” para referirse a altura a uns determinados metros. Todos os datos destas variables vanse ir modificando para poder estudar o comportamento da onda.

Antes de empregar a fórmula, creamos un vector “Data” onde imos ir almacenando os resultados.

### 4.2. Estudo da onda estacionaria

A continuación, introducimos a fórmula da ecuación da onda estacionaria, despezándoa previamente para igualala a 0 (na forma f(x)=0), empregando o comando “expression” para que así o programa a recoñeza como unha función.

```
f<-expression( h - A*( sin( (2*pi*x)/L ) * cos( (2*pi*T*v)/L ) ) + (2.718282)^(-x) )
```

Fig. 12. Estrutura fórmula onda estacionaria

Deseguido, usamos o comando “D” para derivar a expresión introducida anteriormente.

```
df<-D(f, "x")
```

Fig.13. Estrutura comando “D”

Para poder ir variando as velocidades, empregamos o comando “for” que nos permite crear un bucle no que estas van ir aumentando nun valor que determinaremos para cada unha das experiencias. O intervalo empregado no bucle ditamina cantos valores queremos, de modo que se pode cambiar segundo conveña.

Despois, aplicamos o método de Newton-Raphon a través do comando “repeat” denominando a función como f1 e a derivada desta como f2 para, posteriormente, utilizar o procedemento de Newton.

Finalmente, cando a función dada (f1) se aproxime a 0 (10-3), almacenaremos o valor de “x” na variable “Data” creada previamente e empregamos o comando “break” para que pare e reinicie este procedemento, esta vez cun valor diferente para o parámetro.

```

for (i in 1:6) {
  v <- v + 2
  x <- 7

repeat{
  f1 <- eval(f)
  f2 <- eval(df)
  x <- x - f1/f2

  if(f1 < 10^-3)
  {
    Data <- append(Data, x)
    break
  }
}

```

Fig. 14. Estrutura dos comandos empregados

## 5. Tratamento dos datos

Unha vez feito o programa, centrarémonos en analizar os datos resultantes da nosa ecuación. Para iso, fixamos as variables e imos cambiando unha delas para resolver a ecuación  $f(x)=0$ , é dicir, a distancia que percorre a onda ata  $h=h_0$ . Tamén, estudaremos o seu percorrido ata que  $h=h_0/2$  ou  $h=h_0/3$  da altura inicial ou amplitude. Para conseguir isto, variamos a velocidade e a lonxitude da onda xa que cambiar o tempo non resultaría útil porque ao ser unha función periódica obteríamos resultados redundantes. A continuación, tomamos valores extremos aos comúns nunha onda deste tipo para observar os comportamentos máis esaxerados, mentres que para as variables fixas empregamos valores normais dunha onda.

### 5.1. Velocidade da onda

Comezamos variando a velocidade da onda. Para iso, fixamos a súa lonxitude, a amplitude e o tempo segundo a seguinte proporción: 16 metros, 4 metros e 12 segundos respectivamente. Como a velocidade normal dunha onda é de 36m/s é necesario tomar valores pequenos comprendidos entre 2 e 12 m/s e máis grandes entre 36 e 46 m/s. Como resultado obtemos a seguinte táboa:

**L = 16 m      T = 12 s      A = 4 m**

h = h <sub>0</sub>	V = 2	V = 4	V = 6	V = 8	V = 10	V = 12
	11.991033	4.399534	11.991033	4.399534	11.991033	4.399534
h = h <sub>0</sub> /2	V = 2	V = 4	V = 6	V = 8	V = 10	V = 12
	9.331773	6.670393	9.331773	6.670393	9.331773	6.670393
h = h <sub>0</sub> /3	V = 2	V = 4	V = 6	V = 8	V = 10	V = 12
	8.864068	7.138189	8.864068	7.138189	8.864068	7.138189
h = h <sub>0</sub>	V = 36	V = 38	V = 40	V = 42	V = 44	V = 46
	4.399534	11.991033	4.399534	11.991033	4.399534	11.99103
h = h <sub>0</sub> /2	V = 36	V = 38	V = 40	V = 42	V = 44	V = 46
	6.670393	9.331773	6.670393	9.331773	6.670393	9.331773
h = h <sub>0</sub> /3	V = 36	V = 38	V = 40	V = 42	V = 44	V = 46
	7.138189	8.864068	7.138189	8.864068	7.138189	8.86406

## 5.2. Lonxitude da onda

Agora, variamos a lonxitude da onda e voltamos a tomar valores extremos aos da media que é de 16 metros. Polo tanto, traballamos con valores comprendidos entre 2 e 10m, e 22 e 30m. Fixamos a velocidade en 36m/s, o tempo en 12s e amplitude en 4m. Con estes datos, a onda non sempre consegue o obxectivo de que se faga igual á altura, a 1/2 ou a 1/3. Refléxase esta análise na seguinte táboa:

$h = h_0$	L = 2	L = 4	L = 6	L = 8	L = 10
	2.360607	1.445986	2.002065	3.000000	-*
$h = h_0/2$	L = 2	L = 4	L = 6	L = 8	L = 10
	2.124240	1.784476	2.760866	6.694528	1.253072
$h = h_0/3$	L = 2	L = 4	L = 6	L = 8	L = 10
	2.066.264	1.883.914	3.013.800	7.926.038	3.272.888
$h = h_0$	L = 22	L = 24	L = 26	L = 28	L = 30
	-*	7.354193	-*	-*	-*
$h = h_0/2$	L = 22	L = 24	L = 26	L = 28	L = 30
	0.203674	10.0002	0.2300813	0.2445933	0.2430814
$h = h_0/3$	L = 22	L = 24	L = 26	L = 28	L = 30
	0.3120734	10.702009	0.6314407	0.6158292	0.6506015

\*Nestes casos, a ecuación carece de solución.

## 6. Conclusións

Neste estudo decatámonos de que a partir da ecuación da onda estacionaria podemos extraer dous resultados con respecto á distancia que esta percorre ata que cumpra unha das nosas condicións. Por unha banda, no caso da velocidade deducimos que cando esta é un valor enteiro e impar o resultado da ecuación nunca será cero. A onda non vai chegar a ter un valor da altura igual a 1/2 ou a 1/3 da amplitude. Por outra banda, a lonxitude da onda para os valores de 10,11,22,26,28 e 30 metros nunca vai ser igual a amplitude. Nestes casos a ecuación segue tendo a mesma periodicidade, pero cando o eixe das abscisas non se achega ao cero senón que toma valores mínimos de 1,4, tal e como se reflexa na seguinte gráfica:

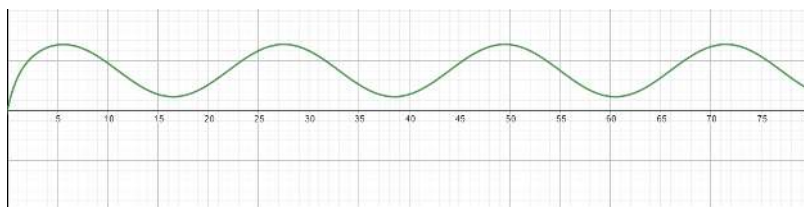


Fig. 15. Representación gráfica cuando esta non ten solución.

## Bibliografía/Webgrafía

Fiscalab (2022). *Ondas estacionarias*. Recuperado o 28 de Abril de 2022, do sitio Web: <https://www.fiscalab.com/apartado/ondas-estacionarias>

StuDocu (S.D.). *Ondas estacionarias*. Recuperado o 28 de Abril de 2022, do sitio Web: <https://www.studocu.com/es/document/bachillerato-espana/fisica-ii/ondas-estacionarias-resumen-y-explicacion-de-ecuacion-de-una-onda-estacionaria-nodos-y-vientres/8025533>

Laboratorio de Física, CC Físicas, UCM (2013/2014). *Ondas estacionarias. Corda vibrante*. Recuperado o 28 de Abril de 2022, do sitio Web: [prac26-1314 \(ucm.es\)](http://prac26-1314.ucm.es)

Glosario Geografía General (2018). *Olas sísmicas*. Recuperado o 28 de Abril de 2022, do sitio Web: [Olas sísmicas \(Geografía General\) \(servidor-alicante.com\)](http://Olas sísmicas (Geografía General) (servidor-alicante.com))

J. Barrio Gómez de Agüero (2000). *Movemento Ondulatorio*. Recuperado o 28 de Abril de 2022, do sitio Web: <https://www.edu.xunta.gal/centros/zonapontevedrad6/system/files/Op%20Sec%20t%2018%20ondas%20y%20energia.pdf>

*Movemento Ondulatorio* (S.D.). Recuperado o 28 de Abril de 2022, do sitio Web: <https://www.edu.xunta.gal/espazoAbalar/sites/espazoAbalar/files/datos/1372178152/contido/unidade02/arquivos/apuntes.pdf>

Manuel Alonso Sánchez (S.D.). *Ondas estacionarias*. Recuperado o 28 de Abril de 2022, do sitio Web: <http://rsefalicante.umh.es/TemasOndas/Ondas.pdf>

Métodos Numéricos 3: Raíces de ecuaciones: Métodos de Newton-Raphson y de la secante (S.D.). Recuperado o 28 de Abril de 2022, do sitio Web:

Métodos Numéricos 3: Raíces de ecuaciones: Métodos de Newton-Raphson y de la secante (ulpgc.es)

Engineering the world (2011). *Método: Newton-Raphson*. Recuperado o 28 de Abril de 2022, do sitio Web: [Metodo: Newton-Raphson | Rama Estudiantil del IEEE de la UCSA \(wordpress.com\)](http://Metodo: Newton-Raphson | Rama Estudiantil del IEEE de la UCSA (wordpress.com))

José Enrique Vargas Cantero (2018). *Método de Newton Raphson*. Recuperado o 28 de Abril de 2022, do sitio Web: [MetododeNewtonRaphson.pdf \(cimat.mx\)](http://MetododeNewtonRaphson.pdf (cimat.mx))

Sara Ruiz (2017). *Método Newton-Raphson*. Recuperado o 28 de Abril de 2022, do sitio Web: [Método Newton-Raphson | Portafolio Métodos Numéricos \(wordpress.com\)](http://Método Newton-Raphson | Portafolio Métodos Numéricos (wordpress.com))



# *Dous métodos de integración numérica (I)*

*Sebio-Mazás, R.; Rey-Silva,  
A.; Calvo-López, N..*

IES MARCO DO CAMBALLÓN

**Resumo.** Neste artigo pretendemos comparar a eficacia de dous métodos de integración numérica, sumas de Riemann e Montecarlo. Para isto, tomamos os valores dunha función nun intervalo pechado e, coa axuda dun código, calculamos a súa integral definida con cada un dos ditos métodos para posteriormente comparar os resultados e medir cal é máis exacto.

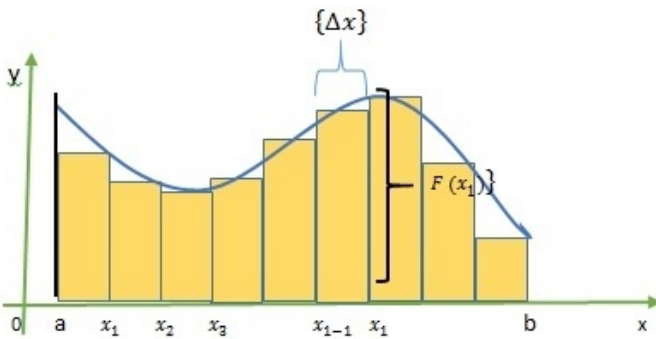
**Abstract.** In this article we aim to compare the effectiveness of two numerical integration methods, Riemann sums and Montecarlo. To do so, we take the values of each function in a given time interval and, according to the code, we calculate the definite integral with each of the two methods in order to compare the results and measure which is the more accurate.

## **1. Introducción**

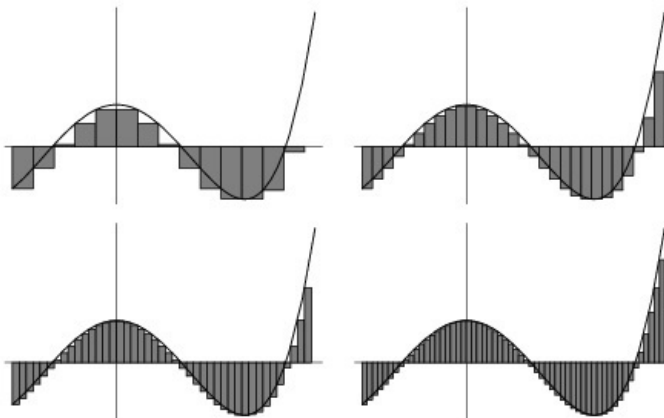
A integración numérica constitúe unha ampla gama de métodos para calcular o valor numérico dunha integral definida e, por extensión, o termo úsase ás veces para describir algoritmos numéricos mediante os cales se resolven ecuacións diferenciais (calquera ecuación que conteña as derivadas dunha ou máis variables dependentes con respecto a unha ou máis variables independentes).

Un destes métodos de integración é o das sumas de Riemann, que deben o seu nome ao matemático Bernhard Riemann. Estas consisten en trazar un número limitado de particións da mesma amplitude dentro dun intervalo acoutado e logo avaliada a función nun dos extremos da partición para despois multiplicar o valor obtido pola amplitude da partición.

Se escollemos o valor superior da partición chamaranse sumas superiores de Riemann e, pola contra, se seleccionamos o valor menor da partición serán sumas inferiores, ambas converxen ao valor da integral. Por último, sumamos todos os resultados obtidos en cada partición para achar así unha aproximación da integral, que é igual a área contida. Graficamente, o que se está facendo é crear rectángulos de igual base que teñen de altura a función avaliada nun dos extremos da base para calcular as áreas individuais de cada rectángulo e sumalas todas.



Cantas máis particións se fagan máis exactitude obteremos. Na seguinte imaxe pódese apreciar este feito:



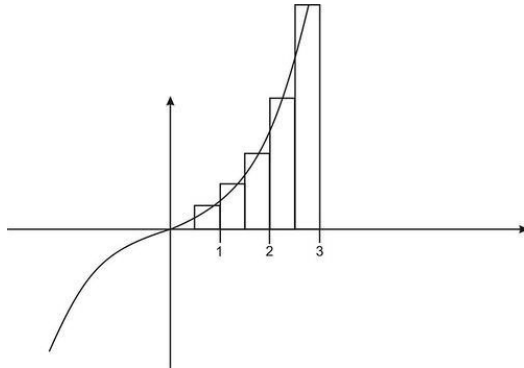
Matematicamente, sendo  $f$  unha función continua e con límites existentes en  $[a,b]$ , dividíndose o intervalo en  $n$  subintervalos de igual ancho ( $\Delta x$ ) con  $x_0 = a, x_1, x_2, \dots, x_n = b$  e cos extremos de cada subintervalo  $\Delta x = \frac{b-a}{n}$ , a integral definida de  $f$  dende  $x = a$  ata  $x = b$  ata  $\Delta x = \frac{1}{2}$  é:

$$\int_a^b f(x) dx = \lim_{n \rightarrow \infty} \sum_{i=1}^n f(x_i) \Delta x$$

Un exemplo da súa aplicación sería:

Calcular a suma de Riemann para  $f(x) = x^3$  no intervalo  $[0,3]$  utilizando  $n = 6$  subintervalos e collendo como puntos de exemplos os puntos medios dos subintervalos.

Se dividimos o intervalo  $[0,3]$  en  $n = 6$  subintervalos iguais, cada intervalo terá unha lonxitude de  $\frac{3-0}{6} = \frac{1}{2}$ , entón teremos



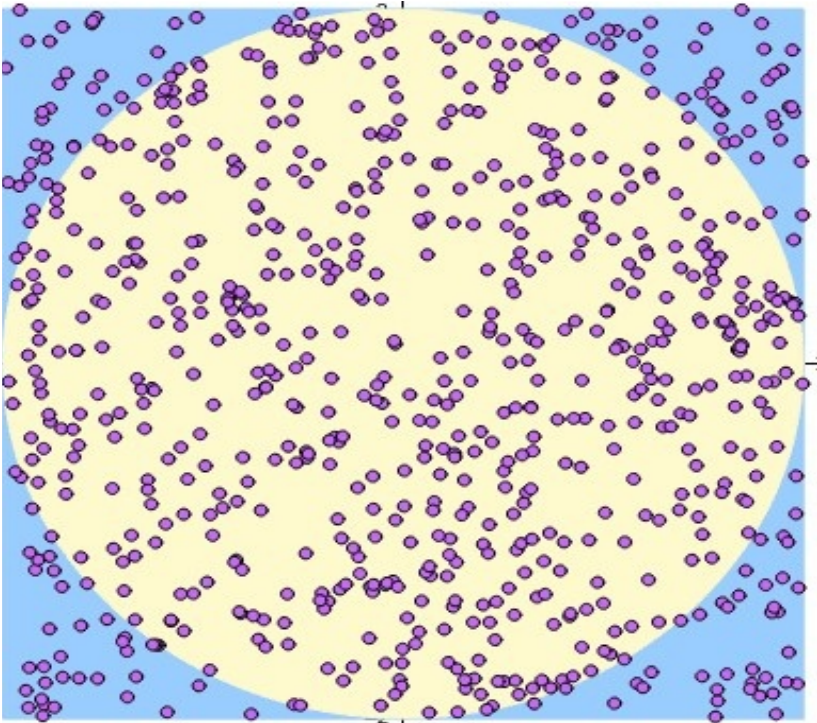
Entón, aplicando a integral deducida nas sumas de Riemann :

$$\begin{aligned} R_6 &= \sum_1^6 f(x_i) \Delta x = f(0,25) \Delta x + f(0,75) \Delta x + f(1,25) \Delta x + f(1,75) \Delta x + f(2,25) \Delta x + f(2,75) \Delta x \\ &= \left(\frac{1}{64}\right) \left(\frac{1}{2}\right) + \left(\frac{27}{64}\right) \left(\frac{1}{2}\right) + \left(\frac{125}{64}\right) \left(\frac{1}{2}\right) + \left(\frac{343}{64}\right) \left(\frac{1}{2}\right) + \\ &\left(\frac{729}{64}\right) \left(\frac{1}{2}\right) + \left(\frac{1331}{64}\right) \left(\frac{1}{2}\right) = \frac{2556}{64} = 39,93 \end{aligned}$$

Outro método de integración numérica é o de Monte Carlo. A súa orixe remóntase a década de 1940, onde matemáticos como John Von Neumann e Stanislaw Ulam aplicaron o método de simulación aleatoria ao campo da experimentación das armas nucleares. A utilización desta técnica como vía de investigación procede do traballo realizado na creación e no desenvolvemento da bomba atómica, durante a Segunda Guerra Mundial. Foi no laboratorio dos Álamos, situado en Estados Unidos, onde se realizou a simulación de cuestións de probabilidade de hidrodinámica (rama da hidráulica que estuda a dinámica dos fluídos), e tamén respecto a difusión dos neutróns (teñen un comportamento aleatorio). Independentemente da súa orixe, este método tamén fai referencia ao Casino de Montecarlo en Mónaco, xa que é considerada a capital dos xogos de azar, destacando a ruleta (xogo de azar que xera resultados aleatorios).

En canto a súa aplicación, consiste en representar aleatoriamente puntos nunha área rectangular (a base vai ter a mesma distancia entre os valores do intervalo no que se quere calcular, e con altura definida polos extremos absolutos da función) para posteriormente seleccionar os valores que se atopan entre a función e os eixos de coordenadas. Desta maneira determínase, dunha maneira aproximada a área de dita función, e a exactitude desta dependerá da cantidade de puntos que se empregaron.

Un exemplo da súa aplicación é a aproximación do número  $\pi$  mediante a representación dun círculo inscrita nun cadrado. O proceso parte da situación inicial de ter un círculo (radio= $R$ ) inscrito nun cadrado (lado= $2R$ ) e consiste en marcar a área da superficie con puntos aleatorios, que aproximarán o valor da área do cadrado e a do círculo. A partir deles créase unha relación entre as áreas (área do círculo/área cadrado), sendo o resultante  $\pi/4$ . Para finalizar, multiplícase o dito resultante por 4 para quitar o valor fraccionario, quedando así o valor de  $\pi$  aproximado. Expresado dunha maneira máis gráfica sería:



$$\frac{\text{Área do círculo}}{\text{Área do cadrado}} = \frac{N^{\circ} \text{ de puntos dentro do círculo}}{N^{\circ} \text{ de puntos totais}}$$

Área cadrado:  $(2R)^2 = 4R^2$

Área círculo:  $\pi R^2$

Despexando a ecuación queda:  $\frac{\pi \cdot R^2}{4R^2} = \frac{\pi}{4}(R^2 \text{ simplifícase,}$

polo tanto queda como resultado o restante na fracción, é dicir,  $\frac{\pi}{4}$ )

A continuación, multiplícase o resultado por 4 para quitar o valor fraccionario e sacar o valor aproximado de  $\pi$ :

$$\frac{\pi}{4} \cdot 4 = \pi ; \text{ o valor achado será aproximado, e a exactitude}$$

deste dependerá da cantidade de puntos empregados.

## 2. Análise dos datos

Agora, imos comparar os métodos expostos para saber cal é o máis exacto, tendo os dous un custo computacional semellante. Para iso, elaboramos dous programas en R:

Primeiro expoñemos o programa para as sumas de Riemann (neste caso superiores):

```
a<-as.double(readline(prompt='Límite inferior a ='))
b<-as.double(readline(prompt='Límite superior b ='))
if (a>=b){
  print(" Os valores teñen que cumprir que a < b")
  a<-as.integer(readline(prompt='Límite inferior a ='))
  b<-as.integer(readline(prompt='Límite superior b ='))
}
er<-c()
it<-c()
n<-1000
while(n<=10000){
  h<-(b-a)/n
  sumsup<-0
  f<-function(x){exp(-(x^2))}
  for (i in 0:(n-1)){
    x<-a+(i+1)*h
    sumsup<-sumsup + f(x)*h
  }
  n<-n+10
  real<-integrate(f, lower=a, upper=b)
  erro<-abs(real$value-sumsup)
  er<-c(er,erro)
  it<-c(it,n)
}
plot(it,er, xlab = "Número de particións", ylab = " Erro ", main = "Gráfica do erro")
```

As primeiras 7 liñas de código teñen a función de determinar os límites de integración da función, sendo estes determinados polo usuario. En caso de que os límites de integración sexan introducidos incorrectamente (o valor inferior debe ser maior ao superior), o código avisa ao usuario e dalle a opción de volver a establecer os valores dos límites.

```
a<-as.double(readline(prompt='Límite inferior a ='))
b<-as.double(readline(prompt='Límite superior b ='))
if (a>=b){
  print(" os valores teñen que cumprir que a < b")
  a<-as.integer(readline(prompt='Límite inferior a ='))
  b<-as.integer(readline(prompt='Límite superior b ='))
}
```

Logo, defínense dous vectores (*er* e *it*) que van almacenar os datos para graficar o erro e a variable *n*, que vai determinar as iteracións que realice o bucle *while*. A finalidade deste bucle é realizar a integral varias veces con distinto número de particións, almacenando o erro para cada un.

```

er<-c()
it<-c()
n<-1000
while(n<-10000){

```

A variable  $h$  vai determinar a amplitude das divisións e o valor desta varía con  $n$ . Por outro lado, *sumsup* almacena as sumas de todas as áreas dos rectángulos. Despois, na variable  $f$  definimos o integrando e abrimos un bucle *for*, cuxa tarefa é resolver a integral sumando as áreas de todas as particións. Dentro do *for*, a variable  $x$  proporciónanos o extremo das divisións para achar a altura do rectángulo. Para iso, utilizamos un resultado que deducimos con anterioridade:  $x_i = a + (i + 1)h$  sendo  $x_i = a + ih$  cada extremo das divisións no caso de facer sumas superiores e  $x_i = \{x_0, x_1, x_2 \dots x_n\}$  e  $x_i$  no caso das inferiores. Recomendáslle ao lector demostrar estes resultados para mellorar a comprensión do método.

```

h<-(b-a)/n
sumsup<-0
f<-function(x){exp(-(x^2))}
for (i in 0:(n-1)){
  x<-a+(i+1)*h
  sumsup<-sumsup + f(x)*h
}

```

Despois á variable  $n$  súmanselle 10 unidades para que o *while* teña fin. Logo faise de novo a integral utilizando unha función predefinida de R chamada *integrate*, para despois restar o valor que nos devolve polo resultado da aproximación (todo en valor absoluto). O erro almacénase no vector *er* e o número de iteracións realizadas na aproximación no vector *it*.

```

n<-n+10
real<-integrate(f, lower=a, upper=b)
erro<-abs(real$value-sumsup)
er<-c(er, erro)
it<-c(it, n)
}

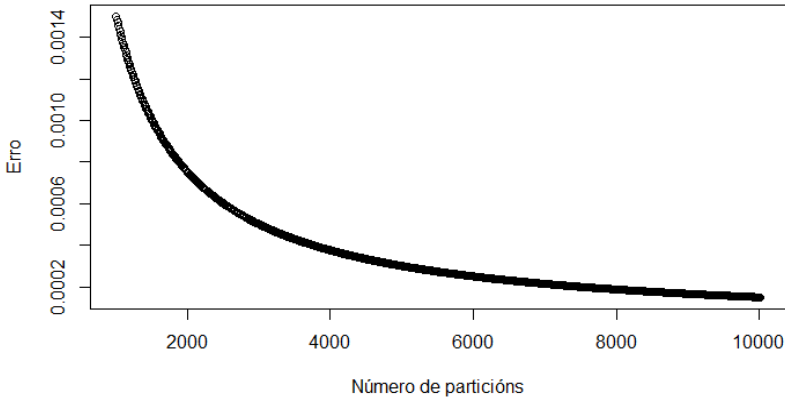
```

Finalmente, empregando o comando *plot* para graficar o erro en función do número de particións.

```
plot(it,er, xlab = "Número de particións", ylab = " Erro ", main = "Gráfica do erro")
```

Un exemplo da gráfica obtida, sendo  $f(x) = e^{-x^2}$  a función a integrar, é:

**Gráfica do erro**



Agora imos expoñer e analizar o código utilizado no método de

Montecarlo:

```
fun<- function(x){ exp(-(x^2)) }
source("newtonec.R")
maxi<-NA
min<-NA
contador<-0
er<-c()
al<-c()
aleatorios<-1000
a<-as.double(readline(prompt='Límite inferior a ='))
b<-as.double(readline(prompt='Límite superior b ='))
if(a>=b){
  print("Introduce ben os límites.")
  a<-as.double(readline(prompt='Límite inferior a ='))
  b<-as.double(readline(prompt='Límite superior b ='))}
fp<-Deriv(fun,"x")
cero<-soluc((b-a)/2)
if(cero<=b & cero>=a){
  if(fp(cero-0.1)*fp(cero+0.1)<0){
    if(fp(cero-0.1)>0){ cero->maxi}else(cero->min)
  }
}
valores<-c(abs(fun(a)),abs(fun(b)),abs(fun(maxi)),abs(fun(min)))
maximo<-max(valores,na.rm=TRUE)
real<-integrate(fun,a,b)
```



```

for(j in 1:100){
  x<-runif(aleatorios,min=a,max=b)
  y<-runif(aleatorios,min=0,max=maximo)
  for(i in 1:aleatorios){
    if(abs(fun(x[i]))> y[i]){
      contador<-contador+1
    }
  }
  area<-(contador/aleatorios)*(b-a)*maximo
  erro<-abs(real$value-area)
  er<-c(er,erro)
  al<-c(al,aleatorios)
  aleatorios<-aleatorios+100
  contador<-0
}
plot(al,er, xlab = " numeros aleatorios", ylab = "erro", main=" Erro do método Montecarlo")

```

Primeiro definimos a función a integrar na variable *fun*. Despois, utilizamos o comando *source* para poder utilizar a función *soluc* que creamos previamente. A súa función é achar as solucións dunha ecuación. Logo, definimos unha serie de variables e vectores que utilizaremos máis adiante. As liñas que van dende a 9-14 teñen como función pedirlle ao usuario que introduza os límites de integración e que, en caso de seren mal introducidos, os corrixa.

```

a<-as.double(readline(prompt='Límite inferior a ='))
b<-as.double(readline(prompt='Límite superior b ='))
if(a>=b){
  print("Introduce ben os límites.")
  a<-as.double(readline(prompt='Límite inferior a ='))
  b<-as.double(readline(prompt='Límite superior b ='))}

```

Para utilizar Monte Carlo temos que acoutar a función para establecer unha rexión na que lanzar os puntos aleatoriamente. Para iso, hai que achar o máximo ou o mínimo da función cuxo valor absoluto sexa maior. Entón, o que facemos é derivar a función e almacénala na variable *fp* para atopar os puntos críticos (máximos, mínimos ou puntos de inflexión) coa función *soluc*. Dentro da función *soluc* poñemos o valor medio do intervalo, xa que a función necesita un punto do cal partir para poder aproximar a solución (véxanse artigos sobre o método Newton-Raphson neste mesmo volume). Unha vez que temos un punto crítico, miramos que este pertenza ao intervalo  $[a,b]$  mediante a función *if*, que compara. Logo, con máis *if*, miramos se a derivada é positiva ou negativa a cada lado das intermediacións para determinar se é un máximo ou un mínimo (Recorde que se a derivada é positiva a función é crecente e, polo contrario,

se é negativa é decrecente). Posteriormente, creamos un vector que nomearemos como *valores*, coa función de almacenar os valores absolutos dos máximos, mínimos e o da función avaliada nos extremos do intervalo a integrar. Coa función *max* achamos o valor máis elevado do vector *valores* e almacenámolo na variable *maximo*. Por outro lado, na variable *real* almacenamos o valor exacto da integral obtido mediante o comando *integrate*. Isto serviranos para comparar o valor que obtemos co valor real, decretando deste xeito o erro.

```
fp<-Deriv(fun,"x")
cero<-soluc((b-a)/2)
if(cero<=b & cero>=a){
  if(fp(cero-0.1)*fp(cero+0.1)<0){
    if(fp(cero-0.1)>0){ cero->maxi}else{cero->min}
  }
}
valores<-c(abs(fun(a)),abs(fun(b)),abs(fun(maxi)),abs(fun(min)))
maximo<-max(valores,na.rm=TRUE)
real<-integrate(fun,a,b)
```

Ata agora, o único que fixo o código foi achar máximos e mínimos para saber entre que rexións numéricas xerar números aleatorios. Logo creamos un *for*, que non vai ter outra utilidade que a de realizar o método de Montecarlo varias veces con distinta cantidade de números aleatorios. Nos vectores *x* e *y*, almacénanse valores numéricos obtidos coa función *runif* a cal xera números aleatorios nun intervalo. Despois, agrupamos os valores de *x* e de *y* segundo o seu orden dentro do vectores correspondentes para dar lugar así a puntos do plano de coordenadas (*x,y*). Para saber se un punto pertence á rexión de interese utilizamos un *if* e se pertence a unha variable chamada *contador* súmaselle unha unidade. Este *if* está dentro dun *for* polo que se vai repetir tantas veces como puntos aleatorios haxa. Sabendo a proporción de puntos que están dentro da área, podemos estimala multiplicando polo valor da área total do rectángulo (almacenándose na variable *area*). Para finalizar, gardamos o valor do erro no vector *er* e no vector *al* almacenamos a cantidade de números aleatorios para obter ese erro. Por último graficamos o erro en función da cantidade de números aleatorios para obter dito erro co comando *plot*.

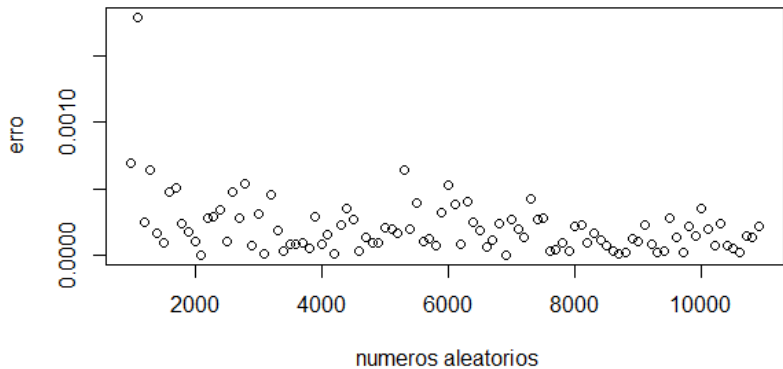
```

for(j in 1:100){
  x<-runif(aleatorios,min=a,max=b)
  y<-runif(aleatorios,min=0,max= máximo)
  for(i in 1:aleatorios){
    if(abs(fun(x[i]))> y[i]){
      contador<-contador+1
    }
  }
  area<-(contador/aleatorios)*(b-a)*máximo
  erro<-abs(real$value-area)
  er<-c(er,erro)
  al<-c(al,aleatorios)
  aleatorios<-aleatorios+100
  contador<-0
}
plot(al,er, xlab =" numeros aleatorios", ylab =" erro", main=" Erro do método Montecarlo")

```

Os resultados obtidos para o método de Montecarlo, introducindo a mesma función que a do método das sumas (  $f(x) = e^{-x^2}$  ), son os seguintes :

**Erro do método Montecarlo**



### 3. Conclusións

Tras o estudo realizado, podemos afirmar que o método de Montecarlo, cuxa dificultade de programación é moito maior, ten unha exactitude análoga ao método das sumas de Riemann, cun custo computacional semellante. Ademais, no método de Montecarlo intervéñ o azar, facendo así que o erro non sexa constante (cunha mesma función calculada varias veces, a gráfica do erro difire); é recomendable calcular máis dunha vez a función con este método para facer unha media do erro.

Por outro lado, determinamos que, no caso de introducir unha función con varios extremos relativos, débese calcular o máximo do intervalo manualmente para asegurarse de que a área do rectángulo a calcular conteña a área contida entre o ancho do intervalo e a altura do máximo, e non a dun máximo relativo.

En conclusión, o método máis axeitado para aproximar integrais definidas entre estes dous é o das sumas de Riemann debido á súa simplicidade de programación e á precisión que nos ofrece.

#### 4. Bibliografía/Webgrafía

<https://latexeditor.lagrida.com/>

<https://flexbooks.ck12.org/cbook/c%C3%A1lculo-2.0/section/5.5/primary/lesson/la-integral-definida%3A-el-l%C3%ADmite-de-una-suma-de-riemann-calc-spn/>

[http://www.sc.ehu.es/sbweb/fisica\\_/numerico/montecarlo/montecarlo.html](http://www.sc.ehu.es/sbweb/fisica_/numerico/montecarlo/montecarlo.html)

<https://ricovictor.com/index.php/2021/04/20/simulaciones-de-montecarlo-en-r/>

# *Dous métodos de integración numérica (II)*

*Mazás-Rivela, P.; Méndez-Reimóndez, S.*

IES MARCO DO CABBALLÓN

**Resumo.** Neste artigo estudaremos a diferenza dos métodos de integración numérica Simpson e trapezio para calcular o valor dunha integral definida. Compararémolos segundo o valor da área que obtemos a partir dos métodos empregados e o seu erro .

**Abstract.** In this article we will study the difference between two methods of numerical integration Simpson and trapeze to calculate the value of a definite integral. Let's compare them according to the value of the area that we obtain from two methods given and their error.

## **1. Introducción**

Neste traballo centrarémolos na regra do trapezio e na regra de Simpson, e compararemos estes dous métodos de integración numérica. A regra do trapezio é un método de integración numérica para calcular aproximadamente o valor dunha integral definida. Esta regra foi inventada entre o séculos XVII e XVIII,. Esta regra funciona creando unha función lineal dende o punto  $(a, f(a))$  ao  $(b, f(b))$ , e aproxímase a área desta función á función previa que pasa por estas coordenadas.

A regra de Simpson é unha forma de aproximar unha integral definida nun intervalo  $[a,b]$  cun método parecido ao da regra do trapezio. A regra de Simpson é un método que funciona tamén para o cálculo de áreas, nomeada así na honra de Thomas Simpson. Recibe tamén o nome de regra de Kepler, xa que a variante máis simple desta regra foi formulada por el no ano 1615, chamada naquel momento como a regra do barril. Johannes Kepler formulou esta re-

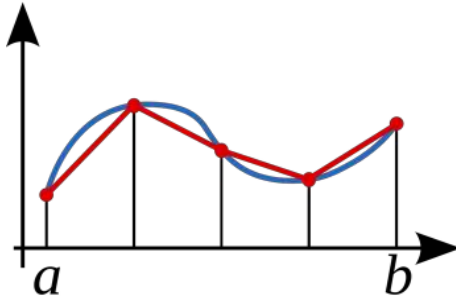
gra mentres buscaba métodos para o cálculo do contido dos barriles de viño . Thomas Simpson naceu o 20 de agosto do 1710 en Inglaterra e morreu o 14 de maio do 1761. Foi un inventor e matemático inglés, recoñecido polos seus traballos sobre a integración numérica ou a interpolación.

## 2. Análise de datos

A continuación explicaremos detalladamente os métodos introducidos anteriormente.

Na regra do trapecio, partimos dunha integral definida da función  $f(x)$  continua e positiva nun intervalo  $[a,b]$  e dividimos o intervalo en outros máis pequenos, cunha altura  $h$

$$h = \frac{b - a}{n} \quad [a = x_0, x_1], [x_1, x_2], \dots, [x_{n-1}, x_n = b]$$



A regra do trapecio aproxima a integral definida a través da área dos trapecios asociados a cada intervalo, do seguinte xeito:

$$\int_a^b f(x)dx \sim \frac{b - a}{n} \left[ \frac{f(a) + f(b)}{2} + \sum_{k=1}^{n-1} f\left(a + k \frac{b - a}{n}\right) \right]$$

Ou equivalentemente:

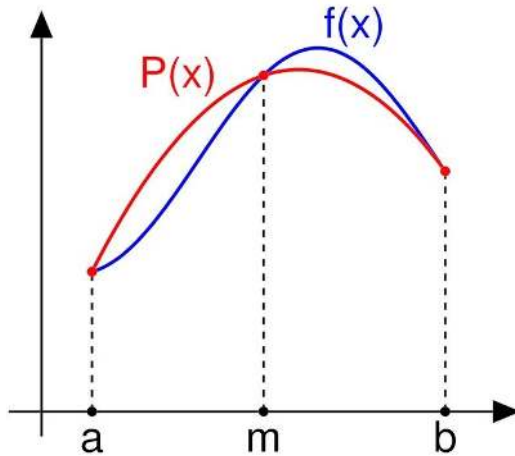
$$\int_a^b f(x) dx \sim \frac{h}{2} [f(a) + 2f(a + h) + 2f(a + 2h) + \dots + f(b)]$$

A aproximación do erro correspóndese con:

$$E_t = -\frac{1}{12} f''(\xi)(b-a)^3$$

sendo  $\xi$  un número comprendido no intervalo entre  $[a,b]$ , e  $f''(x)$  a segunda derivada da función  $f(x)$ .

A fórmula da regra de Simpson simple aproximando a área a través dun polinomio interpolador.



Considerando  $P_2(x)$  como un polinomio interpolador, que aproxima a función  $f(x)$  nun intervalo  $[a,b]$ , sendo  $x_0=a$ ,  $x_1=b$  e  $m=(a+b)/2$

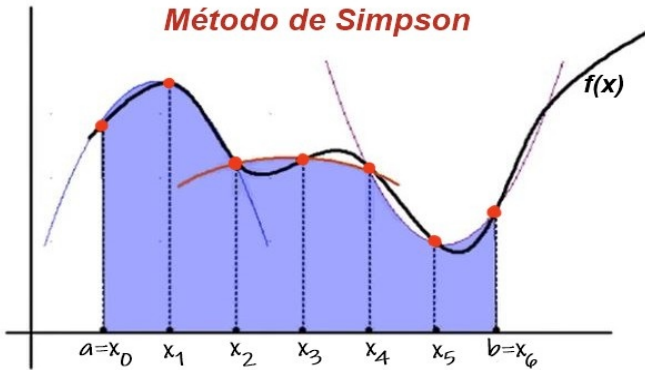
$$P_2(x) = f(a) \frac{(x-m)(x-b)}{(a-m)(a-b)} + f(m) \frac{(x-a)(x-b)}{(m-a)(m-b)} + f(b) \frac{(x-a)(x-m)}{(b-a)(b-m)}$$

Así, a integral buscada equivale á seguinte expresión:

$$\int_a^b f(x) dx$$

na cal  $E(f)$  é o termo de erro. Polo tanto, a integral definida pódese aproximar como:

En adición ás anteriores, empregaremos a fórmula de Simpson composta. A súa explicación é a seguinte :



Partindo dun intervalo  $[a,b]$  , no cal se encontra a integral , divídimo este segmento en  $n$  subintervalos iguais (sendo  $n$  par), de maneira q  $x_i = a + i \cdot h$  , sendo  $h = (b-a)/n$ .

Aplicando a regra de Simpson para cada subintervalo:

$$[x_{j-1}, x_{j+1}], \quad j = 1, 3, 5, \dots, n - 1,$$

Obtemos a seguinte función en cada subintervalo:

$$\int_{x_{j-1}}^{x_{j+1}} f(x) dx \approx \frac{x_{j+1} - x_{j-1}}{6} [f(x_{j-1}) + 4f(x_j) + f(x_{j+1})]$$

Das cales facendo un sumatorio, chegamos a seguinte fórmula:

$$\int_a^b f(x) dx \approx \frac{h}{3} \left[ f(x_0) + 2 \sum_{j=1}^{n/2-1} f(x_{2j}) + 4 \sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n) \right],$$

Na cal o erro máximo ven dado pola seguinte expresión :

$$(b - a) \frac{h^4}{180} \max_{a \leq \xi \leq b} |f^{(4)}(\xi)|.$$

Sendo  $\xi$  un número do intervalo  $[a,b]$  e  $h = (b-a)/n$ .

Para o análise dos métodos expostos, creamos tres códigos en



R. Para esto utilizaremos unha función sinxela:  $f(x)=x^2$ . O primeiro que explicaremos será o código para o funcionamento da regra do trapecio:

```
a<-as.integer(readline(prompt= "Límite inferior a = "))
b<-as.integer(readline(prompt="Límite superior b = "))
if (a>=b){ print(" Ten que cumprir que a < b ")
  a<-as.integer(readline(prompt= "Límite inferior a = "))
  b<-as.integer(readline(prompt="Límite superior b = "))
}
it<-c()
er<-c()
n<-10
while(n<1000){
  h<-(b-a)/n
  trap<-0
  f<-function(x){x^2}
  for( i in 1:n){
    trap<-trap+(((f(a+(i+1)*h)+f(a+(i)*h))/2)*h)
  }
  n<-n+10
  integral<-integrate(function(x) x^2,lower = a, upper = b)
  #str(integral)
  erro<-abs(integral$value-trap)
  it<-c(it,n)
  er<-c(er,erro)}
plot(it,er)
print(trap)
```

En primeiro lugar creamos as variables  $a$  e  $b$  que serán as que delimiten o intervalo no que imos traballar. Para o correcto funcionamento do código, non permitimos que  $a$  sexa maior que  $b$ . A continuación creamos dous vectores sen valores que utilizaremos máis tarde na creación dunha gráfica. Agora creamos a variable  $n$ , que serán os subintervalos nos que se dividirá  $[a,b]$ , que aumentarán a medida que avance o bucle. Este funcionará ata os 1000 intervalos, pero podería ser un número maior ou menor.

Dentro deste bucle definimos algunhas variables: a primeira é  $h$ , que é a altura e a seguinte é a función coa que traballamos. A que prosigue é  $trap$ , que será a variable na que almacenaremos as áreas de cada intervalo  $n$ . Para isto utilizamos un bucle *for* que funcionará ata o valor de  $n$  no momento. Dentro deste bucle utilizamos a fórmula da regra do trapecio, sumándolle o valor das anteriores. A

continuación a  $n$  sumámoslle 10, entón os intervalos son de 10 unidades. O paso subsecuente é facer a integral definida da función entre  $a$  e  $b$  para poder comparar este valor co da fórmula do trapecio e así determinar o erro. Isto levámolo a cabo restando o valor da integral co valor de *trap*, e o resultado almacenámolo na variable *erro*. Os valores de  $n$  e do *erro* dimensinámolos nos vectores sen valor que creamos anteriormente.

Xa fora do bucle utilizamos a función *plot* para que faga unha táboa cos dous vectores antes mencionados e poidamos visualizar o erro en función dos intervalos que haxa. Por último co comando *print*, observamos o valor da área comprendida entre o eixe de ordenadas e a nosa función.

A continuación explicaremos o código que creamos para o análise dos métodos de Simpson, o simple e o composto:

```
a<-as.integer(readline(prompt= "Límite inferior a = "))
b<-as.integer(readline(prompt="Límite superior b = "))
if (a>=b){
  print(" Ten que cumprir que a < b ")
  a<-as.integer(readline(prompt= "Límite inferior a = "))
  b<-as.integer(readline(prompt="Límite superior b = "))
}
m<-(b-a)/2
f<-function(x){x^2}
int<-((b-a)/6)*(f(a)+4*f(a+m)+f(b))
integral<-integrate(function(x) x^2,lower = a, upper = b)
#str(integral)
error<-abs(integral$value-int)
print(int)
print(error)
```

Comezamos como no anterior código, pedindo os valores para  $a$  e  $b$  coas condicións necesarias. No seguinte paso démoslle o valor a  $m$  que antes explicamos. Despois definimos a nosa función e creamos a variable *int*, na que utilizamos a fórmula de Simpson simple. Despois coa función *integrate*, facemos unha integral definida en  $a$  e  $b$  para comparar os resultados co valor de *int*. Este é o seguinte paso, no que na variable *error* metemos o valor da diferenza do valor da integral e a área resultante da fórmula de Simpson.

Para o método de Simpson composto creamos este outro código:

```
a<-as.integer(readline(prompt="Límite inferior a = "))
b<-as.integer(readline(prompt="Límite superior b = "))
if (a>=b){ print(" Ten que cumprir que a < b ")
  a<-as.integer(readline(prompt="Límite inferior a = "))
  b<-as.integer(readline(prompt="Límite superior b = "))
}
sim<-0
f<-function(x){x^2}
n<-1000
for(i in 1:n){
  if((i %% 2) == 0) {
    next}
  h<-(b-a)/n
  xi<-a+(i*h)
  xi1<-a+((i+1)*h)
  xi2<-a+((i-1)*h)
  sim<-sim+(h/3)*(f(xi2)+4*f(xi)+f(xi1))
}
integral<-integrate(function(x) x^2,lower = a, upper = b)
ero<-abs(integral$value-sim)
print(sim)
#str(integral)
print(ero)
```

Como nos outros dous códigos comezamos coa definición dos valores cña variable *sim* cun valor nulo, que utilizaremos no bucle. Definimos a función e creamos a variable *n* que serán as particións que empregaremos para este cálculo. Definimos un bucle *for* dende 1 ata *n* e para que funcione en divisións de dúas unidades de tamaño, coa función *if* puxemos a norma de que deben de ser números impares para que o bucle funcione. É dicir, saltará os números pares no *for*. Dentro desta función creamos *h*, que será as divisións do intervalo. Os seguintes pasos son as definicións de variables que utilizaremos para a fórmula. Creamos a variable *sim* na que facemos a suma da área intervalo que se traballa nese momento máis o valor das áreas dos anteriores. E por último facemos a integral como nos códigos anteriores e comparamos os valores para achar o erro deste método.

Para a comparación dos métodos utilizamos a función  $x^2$ , no intervalo [2,8]. O primeiro código que empregaremos será o da regra do trapecio. Deste obtemos os seguintes datos:

```
Límite inferior a = 2
Límite superior b = 8
[1] "O valor da área é: 168.363893480257"
[1] " o erro é: 0.363893480257275"
```

Despois de obter o resultado mediante a regra do trapecio, empregaremos os códigos de Simpson simple e composto cos mesmos datos, deles concluímos os seguintes resultados:

Simpson simple:

```
Límite inferior a = 2
Límite superior b = 8
[1] "O valor da área é: 168"
[1] " o erro é: 2.8421709430404e-14"
```

Simpson composto:

```
Límite inferior a = 2
Límite superior b = 8
[1] "O valor da área é: 168"
[1] " o erro é: 1.13686837721616e-13"
```

### 3. Conclusións

Tras unha análise destes dous métodos podemos concluír que o método do trapecio é o menos exacto. Se nos centramos no método de Simpson, observamos como, sorprendentemente, o método composto ten un erro maior que o simple. Esta información fainos pensar que o método de Simpson simple é máis eficaz en ecuacións pouco complexas, como o caso de  $x^2$ . Por iso sería interesante avaliar o código cun conxunto amplo de funcións máis complexas, por exemplo, trigonométricas, para unha análise máis profunda destes métodos.

## Bibliografía / Webgrafía

- Gaussianos (2012). *La historia del redescubrimiento de la regla de los trapecios*. Recuperado o 6 de marzo do 2023 do sitio web: <https://www.gaussianos.com/la-historia-del-redescubrimiento-de-la-regla-de-los-trapecios/>
- FjaraboWeb. *Regla del Trapecio*. Recuperado o 6 de marzo do 2023 do sitio web: [https://fjarabo.webs.ull.es/VirtualDoc/Curso%202011-2012/Ingenier%C3%ADa%20Qu%C3%ADmica/2\\_Teor%C3%ADa/Tema\\_6\\_Ingenier%C3%ADa\\_de\\_la\\_Reaccion\\_Quimica/A60/603\\_Integracion\\_grafica\\_por\\_Trapecios.pdf](https://fjarabo.webs.ull.es/VirtualDoc/Curso%202011-2012/Ingenier%C3%ADa%20Qu%C3%ADmica/2_Teor%C3%ADa/Tema_6_Ingenier%C3%ADa_de_la_Reaccion_Quimica/A60/603_Integracion_grafica_por_Trapecios.pdf) .
- Wikipedia. *Regla del Trapecio*. Recuperado o 6 de marzo do 2023 do sitio web: [https://es.wikipedia.org/wiki/Regla\\_del\\_trapecio](https://es.wikipedia.org/wiki/Regla_del_trapecio) .
- FreeCodeCamp. *Regla de Simpson Simple*. Recuperado o 6 de marzo do 2023 do sitio web: <https://www.freecodecamp.org/espanol/news/la-regla-de-simpson-la-formula-y-como-funciona/> .
- Wikipedia. *Regla de Simpson* Recuperado o 6 de marzo do 2023 do sitio web: [https://es.wikipedia.org/wiki/Regla\\_de\\_Simpson](https://es.wikipedia.org/wiki/Regla_de_Simpson) .
- Wikipedia. *Thomas Simpson*. Recuperado o 6 de marzo do 2023 do sitio web: [https://es.wikipedia.org/wiki/Thomas\\_Simpson](https://es.wikipedia.org/wiki/Thomas_Simpson) .
- Teorema Fundamental del Cálculo. *Regla de Simpson*. Recuperado o 6 de marzo do 2023 do sitio web: <http://teoremafuncalculo.blogspot.com/2012/06/regla-de-simpson.html> .



# *Series de tempo en datos de esperanza de vida*

*Rey-Silva, A.; Sebio-Mazás, R.; Calvo-López, N.; Mazás-Rivela, P.; Méndez-Reimóndez, S.*

IES MARCO DO CABBALLÓN

**Resumo.** Neste traballo utilizamos o modelo de regresión lineal para intentar achar a relación de dependencia entre diferentes variables respecto a unha variable dependente.

**Abstract.** In this work we used the linear regression model to try to found a dependency relationship between different variable regarding a dependant variable.

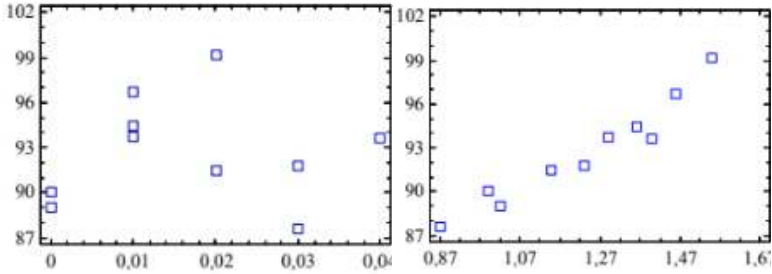
## **1.Introducción**

Neste traballo compararemos diversos países segundo a súa esperanza de vida en relación a tres datos máis: a porcentaxe de gasto do total, o PIB e a taxa de delgadeza entre 1 e 19 anos. Para contrastar estos datos, utilizaremos os modelos de regresión lineal, que son modelos matemáticos que buscan determinar unha función lineal que relacione a variable dependente ( $y$ ), con respecto a outras variables ( $x$ ). No caso de que non se poida aplicar un modelo de regresión, dise que non existe correlación entre as variables estudadas.

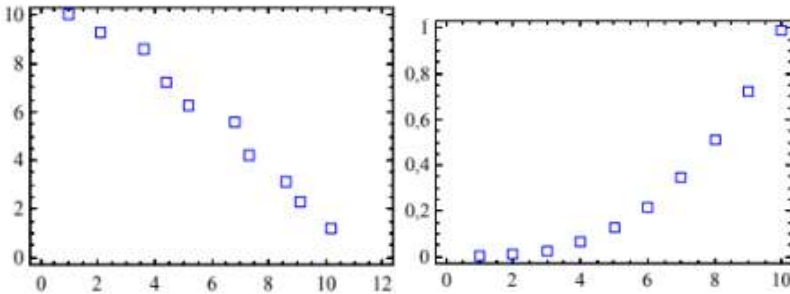
O primeiro estudo que utilizou unha forma de regresión lineal foi o método dos mínimos cadrados, publicado por Legendre no 1805. A primeira vez que o termo regresión foi utilizado realmente foi no estudo de variables antropométricas, no que se comparaban estaturas de proxenitores e os seus descendentes, achando así unha norma. O termo lineal xurdiu para diferenciar este modelo das outras técnicas de regresión que utilizan como base unha función matemática de calquera natureza.

Unha maneira de determinar se existe algunha relación entre as variables coas que estamos traballando é representando os valores das variables nunha gráfica chamada nube de puntos ou diagrama de dispersión. Poden darse catro casos:

- 1. Non existe relación
- 2. Relación lineal positiva



- 3. Relación lineal negativa
- 4. Relación non lineal



## 2. Análise dos datos

Con este traballo tratamos de analizar a relación entre países desfavorecidos con outros máis desenvolvidos. Trátase de: Xapón, Afganistán, Lesotho, Alemania, Suíza, Rumanía, Croacia, Botsswana, Colombia e Etiopía.

Comezaremos explicando os métodos matemáticos empregados. Un deles é o método de regresión lineal, que se expresa da seguinte forma:  $Y = \beta_0 + \beta_1 X + \varepsilon$



A función que pretendemos obter é unha recta co valor da pendente de  $\beta_1$  e  $\beta_0$  como ordenada na orixe. *Epsilon* é o erro cometido. Estes parámetros deben de ser estimados para poder realizar predicións. Para estimar os valores partimos dunha mostra de valores de  $X$  e  $Y$ , e cada un con  $n$  mostrás.

O primeiro que debemos facer é a media de  $X$  e a media de  $Y$  que se calcula da seguinte maneira:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \qquad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Logo temos que calcular dous valores que denotamos con  $SS_{xy}$  e  $SS_{xx}$  que veñen dados por:

$$SS_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \qquad SS_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2$$

Finalmente,  $\beta_1$  e  $\beta_0$  veñen dados polas seguintes expresións:

$$\hat{\beta}_1 = \frac{SS_{xy}}{SS_{xx}} \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Para saber o relacionadas que están as variables, calcúlase o coeficiente de correlación lineal que vén dado por esta fórmula:

$$r = \frac{\sqrt{SS_{xx}}}{\sqrt{SS_{yy}}} \hat{\beta}_1$$

Este coeficiente atópase no intervalo  $(-1,1)$ , canto máis próximo sexa o seu valor a  $-1$  ou a  $1$ , máis relación existirá entre as variables. Se é positivo, o modelo é crecente e se é negativo, o modelo é decrecente.

Moitas veces fanse contrastes de hipótese para tratar de simplificar algún dos parámetros do modelo. Estes consisten en dar unha hipótese ( $b_1$ ) que despois se comproba se é válida ou non mediante un estatístico de contraste que se calcula coa seguinte fórmula:

$$t = \frac{\hat{\beta}_1 - b_1}{\sqrt{\frac{Sr^2}{SS_{xx}}}} \quad \text{sendo } SR^2: \quad Sr^2 = \frac{SS_{yy} - \hat{\beta}_1 SS_{xy}}{n - 2}$$

Despois comparamos o estatístico cos valores tabulados da distribución T de Student. As columnas correspondense aos distintos niveis de significación e as filas cos graos de liberdade. Canto menor sexa o nivel do significación maior será a seguridade á hora de aceptar a hipótese.



Tabla t-Student

Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453

A hipótese rexéitase se:

$$|t| > t_{1-\frac{\alpha}{2}, n-2}$$

onde  $\alpha$  é o nivel de significación e  $n$  é o número de datos.

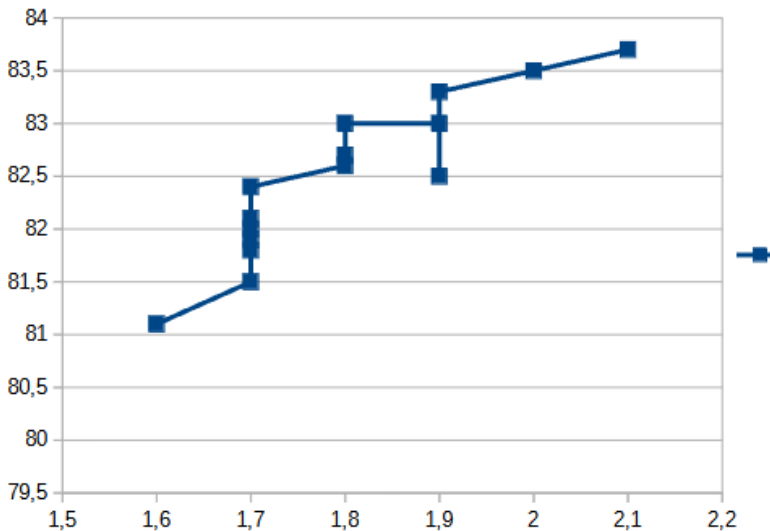
Empregamos datos obtidos da OMS para obter un modelo de regresión lineal tratando de estudar como afectan as distintas variables na esperanza de vida.

Resumimos os nosos resultados, por países, nas seguintes táboas:

Xapón	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	-0,135023750341764	-3,41275302637552E-11	5,01680672268908
$\beta_0$	83,3670521661622	82,6074472686318	79,83234375
SR <sup>2</sup>	4,959904191626	5,15502774579718	4,69061974789916
t1	-0,0297871362562963 Se b1 = -0,14	-0,105204765608736 Se b1 = 0	0,00423264071746795 Se b1 = 5,00
Coefficiente de correlación	-0,196255509879515	-0,0803500596058328	0,920462587940348

Afganistán	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	0,35679464641839	-6,83450219090554E-11	0,169896676833497
$\beta_0$	55,2493021804322	58,2555775971438	- 1300,11116015625
SR <sup>2</sup>	3,58325010813987	3,62821495856365	2,76725829937931
t1	-0,00440564259934006 Se b1 = 0,36	-0,198048800401629 Se b1 = 0	-0,0013638619635250 Se b1 = 0,17
Coefficiente de correlación	0,121689200572706	-0,0409152594407044	0,404621571172044

Nestes dous exemplos, podemos aceptar a nosa hipótese para cada variable (ver táboa), xa que o valor máximo que o estatístico de contraste pode acadar é 1,7459 e todos os valores de t1 son inferiores a este. O valor máximo calculámolo coa táboa T-Student, temos 16 graos de liberdade e o noso nivel de significación é de 0.05. Aínda que todas as hipóteses de b1 sexan aceptables, non hai máis que unha variable que demostre influír sobre a variable Y, esta é a delgadeza de 1 a 19 anos. Podemos deducir isto do coeficiente de correlación obtido, xa que o seu valor é moi próximo a 1, o que nos demostra que existe unha relación real. Esta é a gráfica obtida neste caso:



Alemania	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	0,411320857476676	4,108537340521E-08	-0,0031198640499873
$\beta_0$	79,5112197072117	81,1857128639601	81,1901609144027
SR <sup>2</sup>	8,79936293417395	12,2312375927131	8,81248112860369
t1	-0,0493060521597296 Se b1=0,42	2,0035472247246E-05 Se b1=0,00000004	4,5931040242931E-05 Se b1=-0,004
Coefficiente de correlación	0,529700680132962	0,00020269527228182	-3,693533672323E-05

Lesoto	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	-0,0173513433749452	9,60093465131E-11	-0,123173143668306
$\beta_0$	48,78125	48,7812422415636	49,666556970116
SR <sup>2</sup>	10,5429672946014	10,5460267847018	8,65850318740304
t1	0,00238280739882973 Se b1=2,6E-08	-0,009891768313226 Se b1=-0,018	0,290007770823313 Se b1=-0,2
Coefficiente de correlación	-0,0170325692355442	9,79843592244E-06	-0,123173143668306

No caso de Alemania, salvo o gasto total, as variables non gardan relación coa esperanza de vida. Incluso neste caso a relación é moi pequena. Así mesmo, en Lesoto ningunha das variables garda relación co parámetro  $y$ . Ademais, o PIB, en ambos casos é o que máis hai que crecer para aumentar a esperanza de vida xa que

Colombia	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	0,525692446106793	0,00016760558972	-6,06183745583038
$\beta_0$	70,2319246579158	72,6340775626002	87,2097173144876
SR <sup>2</sup>	0,894799974834744	0,90657921999663 1	0,088606958412612 7
t1	0,0950573749507811 Se b1 = 0,53	1,88807652068367 Se b1 = 0	0,264564809590751 Se b1 = -6,06
Coefficiente de correlación	0,474764973174394	0,46390180393362	-0,96088289531735

Etiopía	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	2,78566645501742	0,0228928560770685	-0,701823366952007
$\beta_0$	45,079853481808	53,2480959915668	64,9421307196354
SR <sup>2</sup>	21,6652544618726	12,9877461971025	16,9676787883139
t1	-0,00973588805035414 Se b1 = 2,8	3,82855915217194 Se b1 = 0,023	-0,007426242834959 Se b1 = -0,7
Coefficiente de correlación	0,464682957808617	0,727991039425946	-0,621237815451612

Nestas dúas táboas, amósanse todas as variables de Colombia e Etiopía, como se apreza na táboa, os valores de  $b_1$  escollidos poden aceptarse agás nos  $b_1$  do PIB, xa que superan o valor de  $t = 1,7459$ . As variables das que podemos dicir que a esperanza de vida dependen, son PIB e delgadeza.

Rumanía	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	-3,206518223	-0,000139203	1,763833428
$\beta_0$	90,91066985	74,53590879	68,31888192
SR <sup>2</sup>	1,800655705	3,717326342	3,435042403
t1	-0,27060605 se b1 = -3 acéptase	-0,90393745 se b1 = 0 acéptase	-0,02983332 se b1 = 1,8 acéptase
Coefficiente de correlación	-0,735274249	-0,227287165	0,351673044

Suíza	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	0,008300238	-6,4923E-06	-5,13068182
$\beta_0$	82,20947368	82,65335148	85,06477273
SR <sup>2</sup>	1,519442365	1,500068971	1,315155303
t1	-0,025931896 se b1 = 0,01 acéptase	-0,458221888 se b1 = 0 acéptase	-0,039033487 se b1 = -5 acéptase
Coefficiente de correlación	0,032678279	-0,11749292	-0,36793071

Podemos ver como estas variables afectan moito menos a Suíza na esperanza de vida que a Rumanía. Ademais, o incremento da esperanza de vida co PIB é nulo xa que aceptamos esa hipótese en ambos casos. Isto parece indicar que o PIB non lle afecta á esperanza de vida case nada. Tamén aparecen resultados aparentemente ilóxicos e contraditorios como por exemplo que canto maior é o gasto de Rumanía menor é a esperanza de vida, isto sabémolo porque  $\beta_1$  é negativo.

Croacia	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	1,45294364995448	- 1,77136772724685E -10	-5,56830031282585
$\beta_0$	65,4750331698993	76,5618194075924	82,044765625
SR <sup>2</sup>	3,95608961715813	4,54629767857111	3,60254985662148
t1	-0,0585456712878551 Se b1 = 1,45 acéptase	-0,87543704187876 Se b1 = 0 acéptase	0,012930022743009 Se b1 = -5,57 acéptase
Coefficiente de correlación	0,411822117605762	-0,389469412515612	-0,899479903271353

Botswana	Gasto total	PIB	Delgadeza 1-19
$\beta_1$	2,05305457480576	2,58000000560317E -10	- 0,028512610808464 4
$\beta_0$	44,7607661567868	55,4411464318742	93,9068750000003
SR <sup>2</sup>	2,17313058325864	3,04835560512327	3,49405442568656
t1	0,0808656967056718 Se b1 = 2,05 acéptase	1,54497975230265 Se b1 = 0 acéptase	0,010631638143237 Se b1 = -0,03 acéptase
Coefficiente de correlación	0,616165094074168	0,095082552093814 3	- 0,013428348631151 5

Nestas táboas, podemos ver que en Croacia e Botswana se aceptan as nosas hipóteses respecto as diferentes variables. A esperanza de vida non varía practicamente en función do PIB porque  $\beta_1$  é practicamente nulo. Algo a destacar é que a delgadeza ten unha maior relación coa esperanza de vida en Croacia que en Botswana. En Croacia canto maior é a delgadeza, maior é a esperanza de vida mais isto aumenta moi lentamente xa que  $\beta_1$  é moi próximo a 0.

### 3. Conclusións

Tras o estudo realizado, podemos afirmar que a esperanza de vida non en xeral está moi relacionada co PIB, coa delgadeza e co gasto total na maioría dos casos, polo menos, se as analizamos por separado dous a dous. Isto é moi probable que sexa porque non tivemos en conta outros factores. Ademais, poderíase facer unha regresión lineal múltiple para ver a influencia de todas as variables na esperanza de vida ao mesmo tempo e obter conclusións máis fiables.

### 4. Bibliografía

Regresión lineal. Wikipedia

Recuperado o 2 de maio do 2023 do sitio web:

[https://es.wikipedia.org/wiki/Regresión\\_lineal](https://es.wikipedia.org/wiki/Regresión_lineal)

Capítulo 9. Regresión lineal

Recuperado o 2 de maio do 2023 do sitio web:

<http://www4.ujaen.es/~dmontoro/Metodos/Tema%209.pdf>

Tabla t de Student

Recuperado o 2 de maio do 2023 do sitio web:

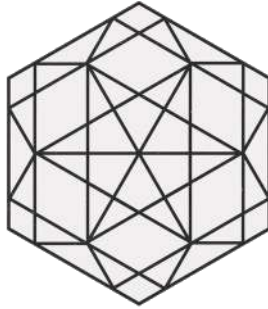
[https://cms.dm.uba.ar/academico/materias/verano2022/probabilidades\\_y\\_estadistica\\_C/tablas/tabla\\_tstudent.pdf](https://cms.dm.uba.ar/academico/materias/verano2022/probabilidades_y_estadistica_C/tablas/tabla_tstudent.pdf)

Life expectancy data

Recuperado o 2 de maio do 2023 do sitio web:

<https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-life-expectancy-and-healthy-life-expectancy>





O sexto número  
de *Marco Numérico*  
saíu do prelo  
en xuño de 2023.





IES MARCO DO CABBALLÓN