Universitat Autònoma de Barcelona

Facultat de Ciències

Pràctica 1

Autors:

Gerard Lahuerta & Ona Sánchez

1601350 — 1601181

28 de Març del 2022

Índex

1 Presentació de la llibreria i informació d'interés

La llibreria de funcions ha estat programada en llenguatge C i consta de dos fitxers, un fitxer capcelera i un amb el codi de les funcions.
Per al seu funcionament precisa de les llibreries de C math.h i limits.h.
També, el fitxer capcelera té definides les constants:

- E_NUM_SUBINTER: missatge d'error per nombre d'intervals introduït com a paràmetre inferior a 1.
- E_FUNC: missatge d'error per apuntador de funció introduïda per paràmetre $NULL$.
- E_NUM_SUBINTER_LEG: missatge d'error per coeficients del nombre d'intervals introduït com a paràmetre no disponible en memòria.

La llibreria té programades quatre funcions que permeten calcular numèricament integrals de variable real. Totes retornen el valor de la integral o, en cas d'error, el valor $LLONG\_MAX$; ambdos casos serà de tipus **double** el valor retornat.
Recalcar a més, que cada funció conté el seu *CDocs* propi que conté la informació més rellevant de la funció de manera resumida: *input, output, descripció i informació rellevant.*

2 Presentació de les funcions

La llibreria creada en aquesta pràctica consta de 4 funcions, cada una amb un mètode d'integració numèrica. El mètodes implementats son els següents:

2.1 Trapezi (Compost)

- Entrada:
  - Funció a integrar: **double** *f (**double**, ***void**).
  - Paràmetres que es passaran a la funció: **void** *params.
  - Valor inferior de la integral: **double** x0.
  - Valor superior de la integral: **double** xn.
  - Nombre de subintervals: **int** n.
- Sortida:
  - Valor de la integral pel mètode del Trapezi: **double** area.
- Funcionament: Un cop cridada la funció amb els paràmetres corresponents, es calcula el valor de la integral segons la següent expressió (mètode d'integració del Trapezi compost):

$$\int_{x_0}^{x_n} f(x)dx \approx \frac{x_n - x_0}{2n}\left(f(x) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right)$$

  En el programa, la funció f ve introduïda com a paràmetre en forma d'atribut pel que qualsevol programa que incorpori la llibreria pot enviar-li la fòrmula a integrar a la funció.

2.2 Simpson (Compost)

- Entrada:
  - Funció a integrar: **double** *f (**double**, ***void**).
  - Paràmetres que es passaran a la funció: **void** *params.
  - Valor inferior de la integral: **double** x0.
  - Valor superior de la integral: **double** x2n.
  - Nombre de subintervals: **int** n.
- Sortida:
  - Valor de la integral pel mètode del Simpson: **double** area.
- Funcionament: Un cop cridada la funció amb els paràmetres corresponents, es calcula el valor de la integral segons la següent expressió (mètode d'integració de Simpson compost):

$$\int_{x_0}^{x_{2n}} f(x)dx \approx \frac{x_{2n} - x_0}{3 \cdot 2n} \left(f(x_0) + f(x_{2n}) + 4 \sum_{i=0}^{n-1} f(x_{2i+1}) + 2 \sum_{i=1}^{n-1} f(x_{2i}) \right)$$

  En el programa, la funció f ve introduïda com a paràmetre en forma d'atribut pel que qualsevol programa que incorpori la llibreria pot enviar-li la fòrmula a integrar a la funció.

2.3 Legendre

- Entrada:
  - Funció a integrar: **double** *f (**double**, ***void**).
  - Paràmetres que es passaran a la funció: **void** *params.
  - Valor inferior de la integral: **double** x0.
  - Valor superior de la integral: **double** xn.
  - Nombre de subintervals: **int** n.
- Sortida:
  - Valor de la integral pel mètode de Legendre: **double** area.
- Funcionament: Un cop cridada la funció amb els paràmetres corresponents, es calcula el valor de la integral segons la següent expressió (mètode d'integració de Legendre):

$$\int_{x_0}^{x_n} f(x)dx \approx \frac{x_n - x_0}{2}\left(\sum_{i=0}^{n-1} f\left(\frac{x_n - x_0}{2}\cdot(x_i) + \frac{x_n + x_0}{2}\right)\cdot w_i\right)$$

  En el programa, la funció f ve introduïda com a paràmetre en forma d'atribut pel que qualsevol programa que incorpori la llibreria pot enviar-li la fòrmula a integrar a la funció.

  Els pesos del mètode de Legendre, w_i, estan guardats a memòria per al nombre de subintervals $n \in \{2, 5, 10\}$.

2.4 Txebixev

- Entrada:
    - Funció a integrar: **double** *f (**double**, ***void**).
    - Paràmetres que es passaran a la funció: **void** *params.
    - Valor inferior de la integral: **double** x0.
    - Valor superior de la integral: **double** xn.
    - Nombre de subintervals: **int** n.
- Sortida:
    - Valor de la integral pel mètode del Txebixev: **double** area.
- Funcionament: Un cop cridada la funció amb els paràmetres corresponents, es calcula el valor de la integral segons la següent expressió (mètode d'integració de Txebixev):

$$\int_{x_0}^{x_n} f(x)dx \approx \frac{x_n - x_0}{2} \cdot \frac{\pi}{n+1} \cdot \sum_{i=0}^{n+1} \left(\sqrt{1 - t_i^2} \cdot f(x_i) \right)$$

En el programa, la funció f ve introduïda com a paràmetre en forma d'atribut pel que qualsevol programa que incorpori la llibreria pot enviar-li la fòrmula a integrar a la funció.

El valor del paràmatre t_i és el valor dels zeros del polinomi de grau $n+1$ que pertany a la família de polinomis ortogonals de Txebixev; és a dir,

$$t_i \in \left\{ \cos \frac{(2i+1)\pi}{2n+2}, \ \ \forall i \in \{0, \cdots, n\} \right\}$$

3 Control dels errors

Per tal d'assegurar el correcte funcionament dels mètodes implementats a la llibreria a inicis de la funció (abans de començar a fer cap càlcul) es confirmarà la correcta introducció dels paràmetres.

Es comprovarà:

1. Que el nombre de subintervals sigui estrictament major a 0, $n > 0$.
2. Que l'apuntador a funció f sigui diferent a $NULL$.
3. Que els valors dels extrems de l'interval siguin diferents.
4. *(Només per al mètode de Legendre)* Que el valor del nombre de subintervals pertanyi al conjunt $\{2, 5, 10\}$.

En cas d'inclompir alguna d'aquestes condicions el programa actuarà de les següents maneres:

- En cas d'inclomplir els punts 1 o 2 es mostraran per pantalla els missatges d'error que es guarden a les variables definides E_NUM_SUBINTER o E_FUNC respectivament (explicades anteriorment a la secció **1**).
- En cas d'incomplir el punt 3, els mètodes retornen 0 com a valor de la integral sense fer cap càlcul.
- En cas d'incomplir el punt 4, es mostrarà per pantalla el missatge d'error que es guarda a la variable definida E_NUM_SUBINTER_LEG (explicada també a la secció **1**).

Recalcar un altre cop que en cas d'inclomplir aluna condició que impedexi fer els càlculs (cas que no es compleixin les condicions 1, 2 o 4) es retornarà el valor $LLONG\_MAX$.

4 Comprovacions de funcionament

Cadascun dels mètodes que s'inclouen a la llibreria tenen el seu error corresponent, i són exactes fins a polinomis d'un cert grau. A la taula que es mostra a continuació hi ha recopilada aquesta informació:

<table>
<tr><th>Mètode</th><th>Error</th><th>Exacte a</th></tr>
<tr><td>Trapezi</td><td>$\frac{b-a}{12}h^2 \cdot f''(\xi)$</td><td>grau < 2</td></tr>
<tr><td>Simpson</td><td>$\frac{b-a}{180}\frac{h^4}{16} \cdot f^{IV}(\xi)$</td><td>grau < 4</td></tr>
<tr><td>Legendre</td><td rowspan="2">$\frac{(x-x_0)^2 \cdots (x-x_n)^2}{(2n+2)!} f^{2n+2}(\xi)$</td><td>grau $< 2n+2$</td></tr>
<tr><td>Txebixev</td><td>$\times$</td></tr>
</table>

Taula 1: Informació dels errors dels mètodes de la llibreria

Definim: f funció a integrar, $a, b \in Dom(f)$ extrems de l'interval a integrar, $n \in \mathbb{N}$ nombre de subintervals, $h = \frac{b-a}{n}$, $\xi \in [a, b]$ un punt desconegut.

Mitjançant la informació mostrada a la taula, adjuntem l'output d'un seguit de proves dutes a terme per a polinomis de grau entre 0 i 22 a un programa senzill fet exclusivament per mostrar la viabilitat de la llibreria. S'ha utilitzat l'interval $[0, 2]$ per fer les proves i un nombre de subintervals $n = 10$.

Recalcar abans de l'anàlisis dels resultats la importància de tenir en compte els possibles errors d'aproximació de resultats a l'hora de fer els càlculs computacionalment (degut a la precisió dels nombres flotants en double que és de 1 bit de signe, 11 bits pel valor de l'exponent i 53 per al valor de la mantissa) i que alguns valors han sigut aproximats i guardats en memòria per així augmentar la velocitat de còmput, com podrien ser els valors del pesos pel mètode de Legendre o el valor de π que és definit a la llibreria math.h.

- Polinomi de grau 0: $f(x) = 1$.

  Mostrem ara l'anàlisis de la informació otinguda:

Output	**Valor real**	**Diferència**
`Area Trapezi --> 2.000000000000000000`	2	0
`Area Simpson --> 2.000000000000000000`	2	0
`Area Legendre--> 2.000062142200000004`	2	$\sim 6 \cdot 10^{-5}$
`Area Txebixev--> 2.006813453956884263`	2	$\sim 6.81 \cdot 10^{-3}$

  Observem que tant el mètode del Trapezi com el de Simpson aproximen de manera exacta l'àrea sota la funció $f(x) = 1$ mentre que els mètodes de Legendre y Txebixev tenen petits errors d'aproximació. Cal recalcar que el mètode de Txebixev no és exacte per cap polinomi, sinó que sempre té un error, cosa que es pot observar al llarg de les proves fetes amb els diversos polinomis.

- Polinomi de grau 1: $f(x) = x$.

  L'output obtingut del programa és el següent:

Output	**Valor real**	**Diferència**
`Area Trapezi --> 2.000000000000000444`	2	$\sim 4 \cdot 10^{-16}$
`Area Simpson --> 2.000000000000000444`	2	$\sim 4 \cdot 10^{-16}$
`Area Legendre--> 2.000115899313752887`	2	$\sim 1.16 \cdot 10^{-4}$
`Area Txebixev--> 2.006813453956883375`	2	$\sim 6.81 \cdot 10^{-3}$

  Observem que tant el mètode del Trapezi com el de Simpson aproximen de manera molt exacta l'àrea sota la funció $f(x) = x$ mentre que els mètodes de Legendre y Txebixev tenen petits errors d'aproximació.

- Polinomi de grau 2: $f(x) = x^2$.

  L'output obtingut del programa és el següent:

Output	Valor real	Diferència
`Area Trapezi --> 2.680000000000000604`	$2,\bar{6}$	$\sim 1.3 \cdot 10^{-1}$
`Area Simpson --> 2.666666666666666519`	$2,\bar{6}$	$\sim 1.47 \cdot 10^{-16}$
`Area Legendre--> 2.666830886162896874`	$2,\bar{6}$	$\sim 1.63 \cdot 10^{-4}$
`Area Txebixev--> 2.680392726600613784`	$2,\bar{6}$	$\sim 1.37 \cdot 10^{-2}$

  Observem que, tal com s'esperava, el mètode del Trapezi ja no dona el valor exacte de l'àrea sota la funció $f(x) = x^2$, sinó que comet un error d'aproximació bastant alt, ja que només és exacte per polinomis de grau menor a 2 (a partir d'ara i per aquesta causa només mostrarem els mètodes restants). Simpson segueix donant el valor exacte, amb un error de l'ordre de 10^{-16} a causa de la precisió del double.

- Polinomi de grau 3: $f(x) = x^3$.

  L'output obtingut del programa és el següent:

Output	Valor real	Diferència
`Area Simpson --> 4.000000000000000000`	4	0
`Area Legendre--> 4.000247331057168232`	4	$\sim 2.47 \cdot 10^{-4}$
`Area Txebixev--> 4.027551271888075490`	4	$\sim 2.75 \cdot 10^{-2}$

  Observem que el mètode de Simpson aproxima de manera exacta l'àrea sota la funció $f(x) = x^3$, mentre que els mètodes de Legendre y Txebixev tenen petits errors d'aproximació.

- Polinomi de grau 4: $f(x) = x^4$.

  L'output obtingut del programa és el següent:

Output	Valor real	Diferència
`Area Simpson --> 6.400026666666668085`	6.4	$\sim 2 \cdot 10^{-5}$
`Area Legendre--> 6.400341752823407226`	6.4	$\sim 3.4 \cdot 10^{-4}$
`Area Txebixev--> 6.455305343617610880`	6.4	$\sim 5.53 \cdot 10^{-2}$

  Observem que el mètode de Simpson ja no aproxima de manera exacta l'àrea sota la funció $f(x) = x^4$, sinó que comet, igual que Legendre i Txebixev, petits errors d'aproximació, ja que deixa de ser exacte en polinomis de grau major o igual a 4 (per aquest motiu ja només mostrarem els dos mètodes restants).

- Polinomi de grau 10: $f(x) = x^{10}$.

  L'output obtingut del programa és el següent:

Mètode	Valor real	Diferència
`Area Legendre--> 186.152831255796655796`	$186,\bar{18}$	$\sim 2.90 \cdot 10^{-2}$
`Area Txebixev--> 189.802435902868097628`	$186,\bar{18}$	~ 3.62

- Polinomi de grau 11: $f(x) = x^{11}$.

  L'output obtingut del programa és el següent:

Mètode	Valor real	Diferència
`Area Legendre--> 341.260709606603484190`	$341,\bar{3}$	$\sim 7.26 \cdot 10^{-2}$
`Area Txebixev--> 348.602740413383571649`	$341,\bar{3}$	~ 7.27

  Observem que el mètode de Txebixev cada cop comet errors d'aproximació més grans al calcular l'àrea sota la funció $f(x) = x^{11}$, ja que aquest mètode no és exacte per cap polinomi. El mètode de Legendre comet petits errors d'aproximació (mencionem una altre cop que en el nostre cas, les proves s'han fet amb n=10).

- Polinomi de grau 21 : $f(x) = x^{21}$.

  Només es mostra l'output de Legendre per comprovar l'exactitud del mètode (com n=10, ha de ser exacte fins grau 22):

Mètode	**Valor real**	**Diferència**
`Area Legendre--> 190454.669746665487764403`	$190650,1818$	~ 195.51

  Aquest mètode utilitza variables guardades a memòria amb 10 decimals, per lo que la diferència que hi ha entre el valor que el mètode mostra per pantalla i el valor real és degut a errors d'aproximació i d'arrodoniment de les variables utilitzades, així com de les aproximacions que fa el double.

- Polinomi de grau 22 : $f(x) = x^{22}$.

  Només es mostra l'output de Legendre per comprovar l'exactitud del mètode:

Mètode	**Valor real**	**Diferència**
`364311.713383253605570644`	364722.087	~ 410.38

  Com s'esperava, Legendre ja no calcula de forma exacta l'àrea sota la funció $f(x) = x^{22}$ ja que la diferència entre el valor que retorna el mètode i el valor real és bastant significativa.

Ens percatem que la fòrmula de la integral numèrica per cadascun dels mètodes explicats a l'apartat **2** depèn directament del nombre d'intervals escollits, de manera que com més gran és el valor de n, més precisió obtenim.
Ho comprovem calculant el valor de l'integral en l'interval $[0, 2]$ per la funció $f(x) = x^7$ amb diferents valor de n:

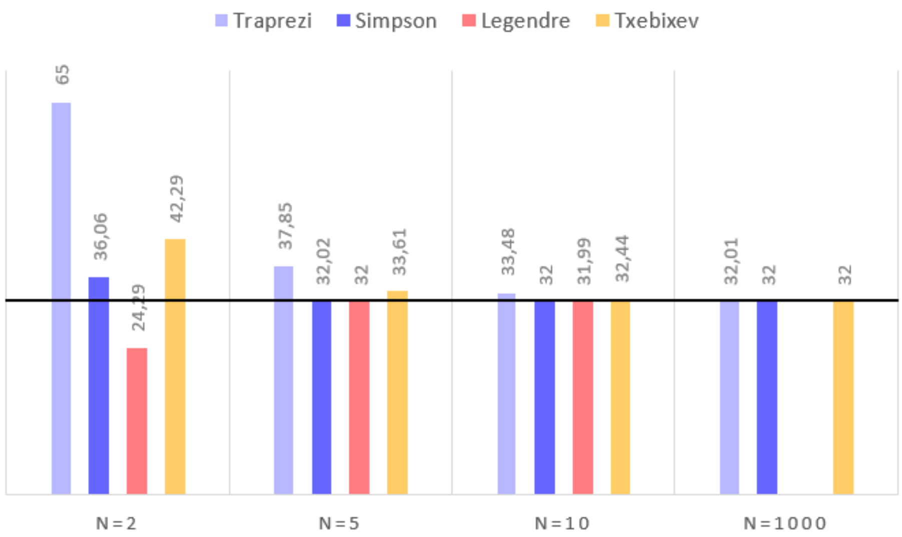

Figura 1: Resultats depenent del nombre de subintervals

La linia negra senyala el valor real de l'integral; és a dir 32.
Els resultats obtinguts per al valor de $n \in \{2, 5, 10, 1000\}$ demostren que, efectivament, la precisió dels mètodes augmenta a mida que augmenta el valor del nombre d'intervals.
Fer èmfasi que per a un valor de $n \notin \{2, 5, 10\}$ el mètode de Legendre no funciona, per falta d'informació en memòria sobre els pesos i els valors de les x a escollir, pel que amb $n = 1000$ no apareix el seu resultat.
Resaltar, a més, que sembla ser que el mètode que millor s'adapta per a calcular els valors de les integrals per a cualsevol valor de n és el mètode de Simpson Compost; ja que veiem com s'apropa ràpidament al valor real per a un valor de n relativament petit (com ho és el valor de $n = 5$).

5 Compilació i execució

5.1 Compilació del Programa

5.1.1 Makefile

Per facilitar la compilació del programa s'ha creat un fitxer *makefile* que inclou tant les comandes per crear l'executable com altres comandes associades (*clean* i *clean_all* que explicarem més endavant) per la correcta gestió dels fitxers que s'obtenen de l'execució del makefile.
Per compilar el programa i generar l'executable (*gauss*) només cal escriure a la terminal on es troben tots els fitxers (inclós el *makefile*) la comanda make:

```
$make
```

Com hem comentat anteriorment, el fitxer makefile també conté comandes per la correcta gestió dels fitxers resultants d'obtenir l'executable, aquestes comandes són:

```
$make clean
$make clean_all
```

La comanda *clean* elimina tots el fitxers *.o* del directori mentre que la comanda *clean_all* elimina tant l'executable com els fitxers amb terminació *.o*.
Mencionar que no es poden utilitzar les dues comandes de manera seguida ja que donarà error.
Important: en cas de no utilitzar el sistema operatiu *Linux* (o semblants) o *IOS* cal modificar la variable *DELETE* de l'arxiu *makefile* per a poder utilitzar-lo (substituir per la comanda *del* en cas d'utilitzar Windows).

5.1.2 Compilació manual

En cas de voler compilar el programa de manera manual (comanda a comanda) utilitzar en ordre les comandes al terminal (una vegada ubicat al directori on es troben els fitxers).

```
$gcc -c integracio.c integracio.h -lm
$gcc -c gauss.c  integracio.c -lm
$gcc gauss.o  integracio.o -lm -o gauss
```

5.2 Execució del programa

Un cop s'ha compilat el programa, per executar-lo cal escriure a la terminal la comanda `$./gauss` μ σ^2 `x`; on els valors μ, σ^2 son els els paràmetres d'una distribució aleatòria Normal ($X \sim N(\mu, \sigma^2)$).
Per altra banda, la x és el valor pel qual es calcularà $P(|X| > x)$.

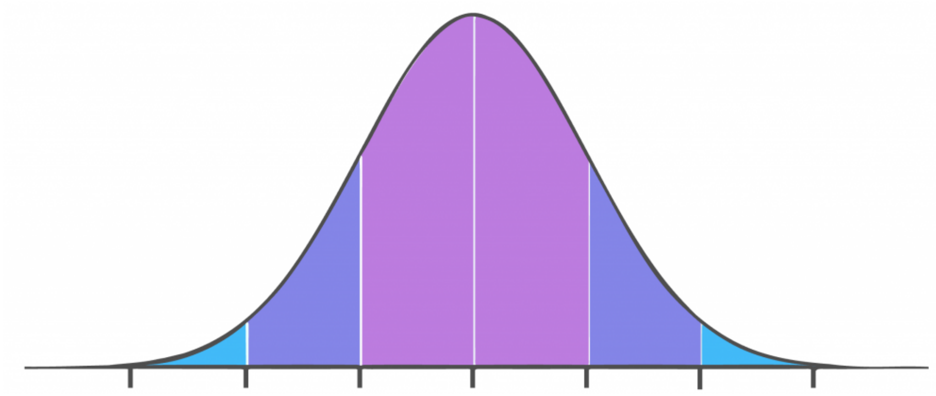

Figura 2: Representació de l'àrea de les cues de la distribució normal

Si s'executa el programa amb aquesta comanda, el valor per defecte de n (nombre de subintervals que s'utilitza per calcular la integral) és 10.
El programa està pensat per poder executar-se també amb la comanda: `$./gauss` μ σ^2 `x n`, on n és el nombre de subintervals, per si es vol calcular amb un valor diferent a 10.
Si s'introdueixen menys dels 3 paràmetres imprescindibles (μ, σ^2 i `x`) es mostrarà un missatge d'error, mentre que si s'introdueixen més de 4 valors (μ, σ^2, `x` i `n`), el programa ignorarà els que hi hagi de més.
Una vegada executat el programa es mostrarà per `stdout` el valor de la probabilitat (calculada pels quatre mètodes que inclou la llibreria).

Cal fer especial menció en que el programa gauss.c només gestiona que els valors introduïts per l'equació de la distribució gaussiana tinguin sentit (que s'hagin introduït $\mu \in \mathbb{R}$, $x, \sigma^2 \in \mathbb{R}^+$); que el valor del nombre de subintervals introduït sigui correcte és gestionat per la llibreria.

6 Explicació Gauss

Recordem que el programa Gauss (mitjançant la llibreria integracio.h ja esmentada) calcula l'àrea dels extrems d'una distribució normal amb paràmetres μ, σ^2 (és a dir; si $X \sim N(\mu, \sigma^2)$, retorna $P(|X| > x)$; per a $x \in \mathbb{R}^+$ el valor desitjat).
Per a l'execució i introducció de les dades es recomana llegir l'apartat anterior (secció 5.2).
A continuació és mostraran diversos resultats obtinguts per a la comprovació del correcte funcionament del programa i la seva deguda anàlisi.

6.1 Exemplificació del programa `gauss.c`

Suposarem en aquest apartat 3 casos pels que serveix i és útil la nostra llibreria.

- **Experiment amb resultat $x = \mu + 5\sigma$:**
  Suposem que en un donat experiment obtenim que els resultats pertanyen a l'interval $[-x, x]$ on $x = \mu + 5\sigma$, essent μ, σ^2 els paràmetres d'una distribució normal (més concretament, $\mu = -5, \sigma^2 = 16$; és a dir, $N(-5, 16)$).
  Volem saber si la mostra obtinguda està continguda en un interval amb alta probabilitat o, si per el contrari no ho està (fet que podria suposar algun problema experimental o de plantejament de l'experiment).
  Per tant, volem saber $P(|N(-5, 16)| > -5 + 5 \cdot 4 = 15)$. Els valors obtinguts són:

```
Area Trapezi --> 0.006437551475070247
Area Simpson --> 0.006210149190216119
Area Legendre--> 0.006209368495169398
Area Txebixev--> 0.001042226044340622
```

  Observem doncs que la probabilitat d'haver obtingut valors fora de l'interval és suficientment petita com per afirmar que segueix una distribució normal.

- **Càlcul del p-value**:

  Suposem que tenim dues poblacions independents que segueixen una distribució normal de la qual sabem els valors de σ_x^2 i σ_y^2 i volem fer un test d'hipòtesis per saber si $\mu_x = \mu_y$, amb un nivell de significació $\alpha = 0.05$.

  Rebutjarem la hipòtesis inicial ($\mu_x = \mu_y$) si el p-valor és menor a α.

  Per fer el test d'hipòtesis, lo primer que hauríem de fer és calcular el test estadístic $W(x) \sim N(0,1)$.

  Suposem $W(x) = 2.3$.

  El càlcul del p-valor és la probabilitat d'obtenir un valor més extrem o igual a l'observat $W(x)$, és a dir, $P(|N(0,1)| > 2.3)$.

  Els valors del p-valor obtinguts són:

```
Area Trapezi --> 0.022021479049143755
Area Simpson --> 0.021448511966331518
Area Legendre--> 0.021440498451698042
Area Txebixev--> 0.018104816983536998
```

  Per tant, rebutgem la hipòtesis inicial.

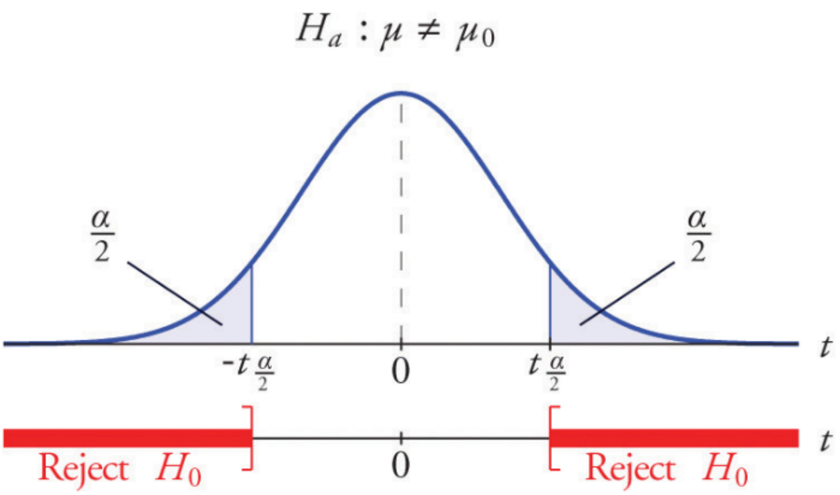

Figura 3: Regió de rebuig d'una distribució normal per a un valor t.

- **Detecció de valors *outliers***:

  Suposem que volem modelitzar un sistema que creiem que segueix una distribució normal amb paràmetres $(\mu, \sigma^2) = (0, 1)$.

  Ens percatem que de les mostres que hem obtingut totes excepte 1 es troben dins l'interval $[-2, 2]$, l'excepció es situa a $x = -2.5$.

  Volem saber quina és la probabilitat de que el fenòmen que provoca aquest resultat sigui suficientent gran com per a tenir-lo en compte; o si pel contrari podem descartar-lo per simplificar la nostra representació.

  Sabem, per tant, que un fenòmen més extrem a aquest està contingut a l'interval $(-\infty, -2.5] \cup [2.5, +\infty)$; pel que calculem la probabilitat (segons el nostre model) que succeeixi un d'aquests fenòmens extrems, és a dir, $P(|N(0, 1)| > 2.5)$.

  Els valors obtinguts són:

```
Area Trapezi --> 0.012874245540560880
Area Simpson --> 0.012419718839867366
Area Legendre--> 0.012413827204901917
Area Txebixev--> 0.008876008616628128
```

  Ens percatem que la probabilitat d' enregistrar en l'experiment un fenòmen similiar o més extrem al obtingut és suficientment petita (una mica superior al 1%) com per a tractar aquest resultat com *outlier*. Per tant podem no utilitzar-lo per a fer el nostre model més senzill.

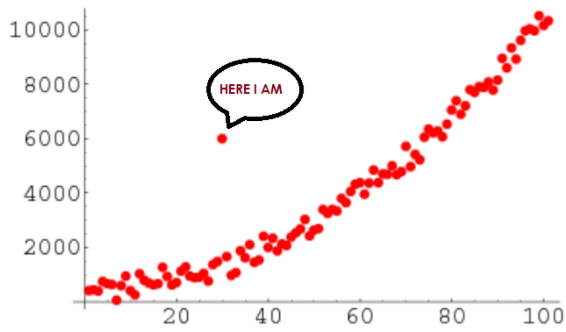

Figura 4: Exemplificació d'un outlier qualsevol.

7 Conclusions

Finalitzem l'informe afirmant (a partir de les dades experimentals dutes als apartats 4 i 6.1 que tant la biblioteca integració.c com el programa `gauss.c` funcionen de manera correcta llevat de petits errors deguts a la precisió de la màquina.
Tot i així les aproximacions de les integrals són suficientment bones com per recomanar l'ús de la llibreria.
Recomanem de la mateixa manera que es calculin les integrals mitjançant els quatre mètodes d'integració per a obtenir una millor aproximació i/o un cert interval de confiança.

7.1 Informació adicional

- Informar que en cas d'obtenir una àrea menor d'una certa tolerància el programa interpretarà aquella àrea com a 0 (la tolèrancia introduïda és de 10^{-10} que és on hem observat experimentalment que existeix més perill d'error d'aproximació amb tendències a desplaçar el valor al semieix negatiu).
- La complexitat dels mètodes són d'ordre lineal; és a dir, $O(n)$ (on n és el nombre de subintervals).
- Responem a la pregunta: **Com generarieu una variable aletòria normal utilitzant les llibreries bàsiques de C?**
  C permet generar variables aleatòries, però segueixen únicament una distribució uniforma. Per poder generar una distribució normal, s'haurà de fer ús de la comanda `rand`.
  Per simular una distribució normal amb mitja μ i variança σ^2 es necessita utilitzar la interpretació del teorema del límit central:
  $$\lim_{N\to\infty} P\left(\frac{\sum_{i=1}^{n} U_i - N\mu}{\sqrt{N}\sigma} \in (a,b)\right) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{\frac{-x^2}{2}} dx$$
  On $U_i \sim Uniforme(\mu)$ (segueix una distribució uniforme d'esperança μ, centrada a μ, que centrarà la distribució normal a aquest valor) i σ^2 és la desviació que volem que tingui la distribució normal.

Per obtenir una distribució normalitzada (de mitja 0 i desviació típica 1) cal seguir els següents passos:

1. Generar n mostres uniformes, amb valors continguts a l'interval $[-0.5, 0.5]$ (és a dir, $P(x \in [-0.5, 0.5]) = 1$ i $P(x \notin [-0.5, 0.5]) = 0$), i una esperança nul·la (centrada a l'origen).
2. Crear una nova variable aleatòria, sumant totes les mostres. Aquesta nova variable aleatòria segueix una distribució normal per valors de n grans.
3. Realitzar aquests passos un nombre gran de cops, de manera que poguem verificar que segueix una distribució normal.