INFORME TAREA 1

Antonia Labarca Sánchez
RUT:20.072.551-4
Github: @antolabarca

1. Introducción

El objetivo de este informe es presentar un cálculo del percentil para un experimento con la siguiente función de densidad de probabilidad:

$$pdf(x) = \chi^2(x) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} e^{-x/2} \tag{1}$$

$$\text{Con } \Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx \text{ y } k = 4.551$$

En particular se desea calcular el percentil evaluado en 0.95, es decir, resolver la siguiente ecuación:

$$0.95 = \int_{-\infty}^{a} pdf(x)dx \tag{2}$$

Para realizar estos cálculos, en primer lugar se debe calcule numéricamente $\Gamma(z)$.

Teniendo eso, se programará una función que calcule $\chi^2(x)$, con la que luego se podrá calcular

$$f(a) = \int_0^a \chi(x)^2 dx \tag{3}$$

Como el dominio de χ^2 es positivo, se tendrá la siguiente equivalencia:

$$\mathbb{P}(x < a) = \int_0^a \chi(x)^2 dx \tag{4}$$

Por lo que simplemente se debe ocupar un algoritmo para buscar el valor de a tal que $f(a) = 0.95$, y resolver el problema.

En este informe se detallaran los pasos realizados y los resultados obtenidos, junto con uns discusión sobre su convergencia

2. Desarrollo

Lo primero es regularizar la integral. Para ello se utilizará el cambio de variable $u = \frac{1}{x+1}$, con lo que la integral queda:

$$\Gamma(z) = \int_1^0 (\frac{1}{u} - 1)^{z-1} \ e^{-(\frac{1}{u}-1)} \ (-1)\frac{1}{u^2}du = \int_0^1 (\frac{1}{u} - 1)^{z-1} \ e^{1-\frac{1}{u}} \ \frac{1}{u^2}du \tag{5}$$

Se llamará $g_\Gamma(x,z)$ a la función original a integrar. Su forma (con $z = \frac{k}{2}$) se puede ver en la Figura 1.

Se llamará $g(u,z)$ a la función a integrar luego del cambio de variable. Su forma (con $z = \frac{k}{2}$) se puede ver en la Figura 2.

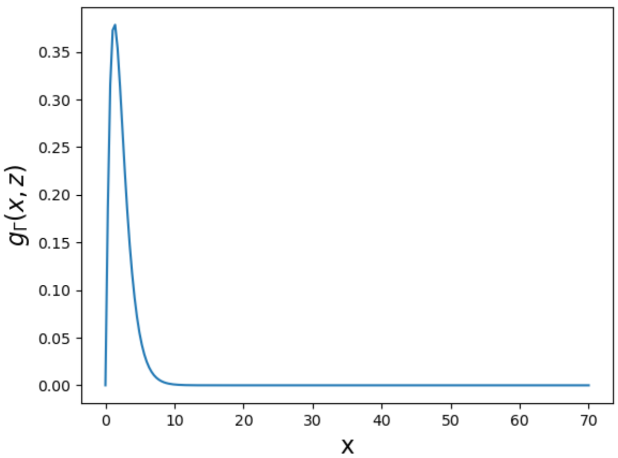

Figura 1: $g_\Gamma(x,z)$ con $z = \frac{k}{2}$ fijo, $x \in [0,70]$

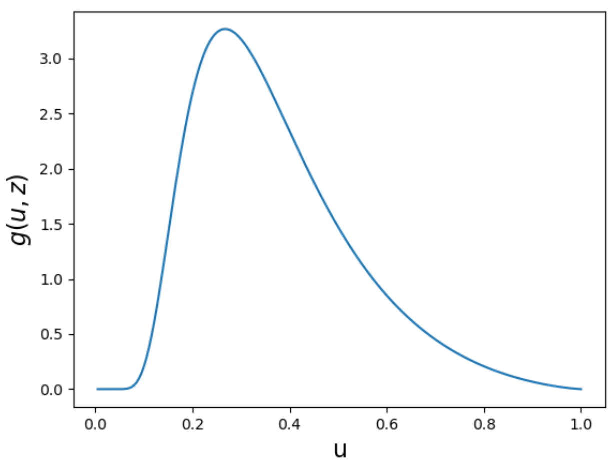

Figura 2: $g(u,z)$ con $z = \frac{k}{2}$ fijo y $u \in [0,1]$

Se utilizó $z = \frac{k}{2}$ para evaluar la forma de las funciones, ya que por la forma de $\chi^2(x)$ este es el único valor para el cuál se calculará Γ.

Se implementó el cálculo de $\Gamma(z)$ utilizando la Regla de Simpson. Dado que la función $g(u,z)$ no diverge en ningún punto del dominio a evaluar, no habrá problemas. Además, no posee alto contraste, por lo que el resultado obtenido será una muy buena aproximación.

Se programaron 2 funciones: `gamma_n(z,n)` que calcula $\Gamma_n(z)$ (el valor aproximado de $\Gamma(z)$ con

n puntos en total) y `gamma(z,tol)` que calcula $\Gamma(z)$ duplicando la cantidad de puntos en cada iteración, hasta llegar a que

$$\left|\frac{\Gamma_n(z)-\Gamma_{2n}(z)}{\Gamma_n}\right| < tol$$

El resultado de `gamma(n,tol)` se comparó con `math.factorial(n-1)` para $n \in \{1, \dots 12\}$, obteniéndose diferencias muy pequeñas. De esto se concluyó la correctitud de `gamma(n,tol)`.

Posteriormente, se programó la función `chi2(x)`, que calcula el valor de $\chi^2(x)$ considerando una precisión de la función Γ de `tol=0.0000001`. Se escogió este valor ya que al probar `gamma(n,tol)` para $n \in \{2, 3\}$ (los números enteros más cercanos a $k/2$, el valor para el que se calculará Γ) la diferencia con $(n-1)!$ era relativamente pequeña, pero se lograba converger en un tiempo corto. Como en χ^2 siempre se considera $\Gamma(\frac{k}{2})$, este valor se pre-calculó para no tener que volver a realizar el cálculo de una integral, que es bastante costoso si se desea una alta precisión, para cada iteración posterior al integrar χ^2.

La forma de la función `chi1(x)` se observa en la figura 3. Nuevamente, es una función sin discontinuidades ni alto contraste, por lo que se escogió calcular la integral con la Regla de Simpson.

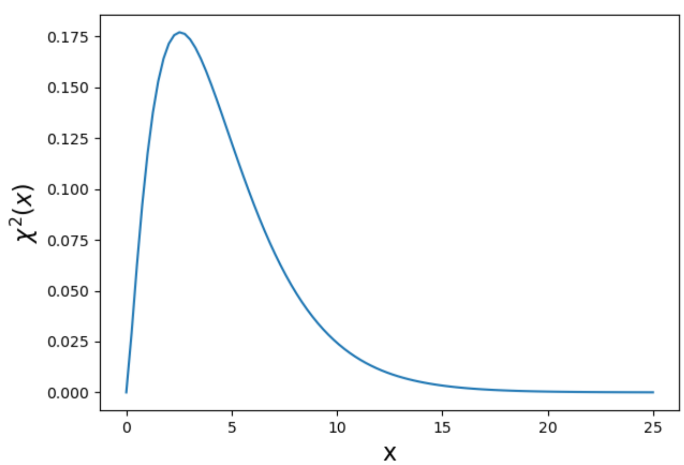

Figura 3: $\chi^2(x)$ para $x \in [0, 25]$

Con esto se pudo programar dos funciones que calculen $f(a)$ de la misma forma que al integrar Γ: en primer lugar, `f_n(a,n)` que calcula la integral usando `n` puntos, y teniendo esa función, `f(a,tol)` que calcula $f(a)$ usando `f_n(a,n)` y duplicando el número de puntos hasta alcanzar la convergencia relativa `tol`.

Definiendo $f_{0.95}(a) = f(a) - 0.95$, es fácil ver que si se encuentra a tal que $f_{0.95}(a) = 0$, el mismo valor de a cumplirá que $f(a) = 0.95$.

Entonces, se programó `f_95(a,tol)` que simplemente calcula el valor de `f(a,tol) - 0.95`.

La forma de $f_{0.95}(a)$ se observa en la Figura 4

Haciendo zoom, se puede ver que el 0 de la función anterior es bastante cercano al 4. Esto se ve en la figura 5.

Como ya se conoce el valor de $\frac{f(a)}{da}$ (es $\chi^2(a)$) y visualmete se observa que alrededor de 4 la función es bastante inclinada (su derivada es muy distinta de 0), se optó por usar el algoritmo de

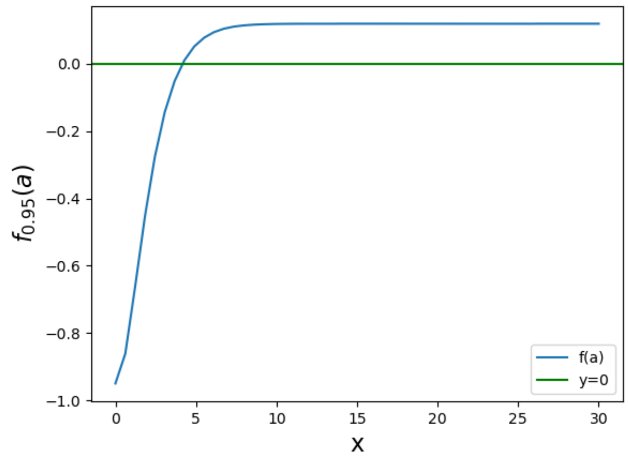

Figura 4: $f_{0.95}(a)$, para $a \in [0, 30]$, ocupando una tolerancia de 0.00001, con una línea horizontal verde a la altura 0

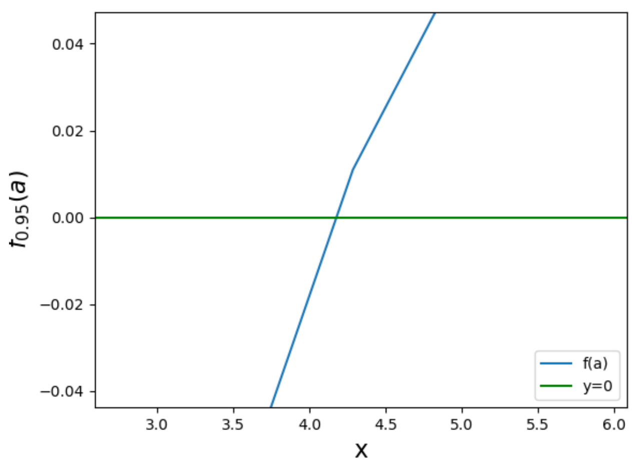

Figura 5: zoom a la figura 4

Newton, debido a su mayor rapidez de convergencia. Como valor inicial se escogió 4 debido a su cercanía a simple vista con el 0.

3. Resultados

El algoritmo para encontrar 0s de la función se corrió para precisiones 10^{-i} con $i \in \{0, \ldots 15\}$, obteniéndose los resultados de la tabla:

El algoritmo converge muy rápidamente a un valor de a cercano a 3.8261.

Podemos observar que el algoritmo es bastante eficiente en la figura 6. Es fácil ver que la

i	precision	a	$f(a)$	iteraciones
0	1	4	-0.0141014	1
1	0.1	4	-0.0141013	2
2	0.01	3.883879	0.0067126	6
3	10^{-3}	3.82441	-0.000159	10
4	10^{-4}	3.82852	-4.003331554380818e-05	14
5	10^{-5}	3.82865	4.4730129199477275e-06	18
6	10^{-6}	3.828609	4.836218107096002e-07	21
7	10^{-7}	3.828609	-8.139185359024026e-08	25
8	10^{-8}	3.82861025274666	2.923029263079968e-09	29
9	10^{-9}	3.8286101993197903	3.8254199807852274e-10	33
10	10^{-10}	3.8286101987424854	-5.331735053459852e-11	37
11	10^{-11}	3.828610199196162	1.414757200279837e-12	41
12	10^{-12}	3.8286101991691797	9.240386233955178e-13	44
13	10^{-13}	3.8286101991625783	-4.8960835385969403e-14	48
14	10^{-14}	3.8286101991632155	-2.55351295663786e-15	52
15	10^{-15}	3.8286101991632155	8.881784197001252e-16	56

cantidad de iteraciones es aproximadamente lineal con respecto a i.

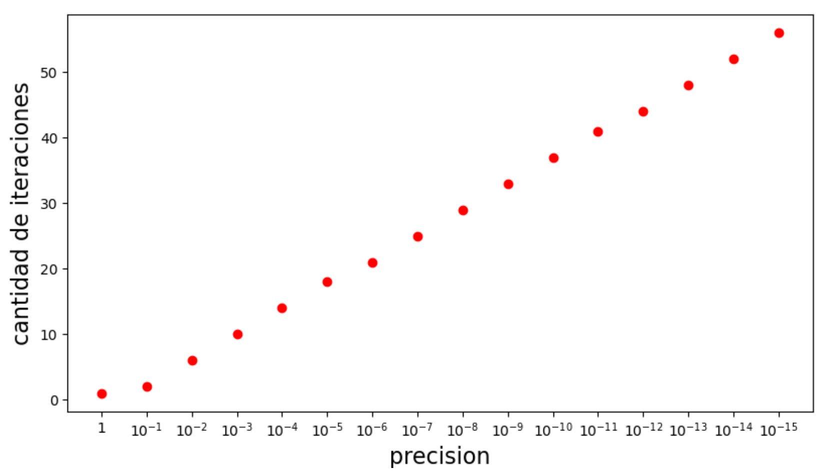

Figura 6: Cantidad de iteraciones realizadas por el algoritmo en función de la precisión pedida.

Además, el algoritmo converge muy rápidamente, como se observa en la figura 7

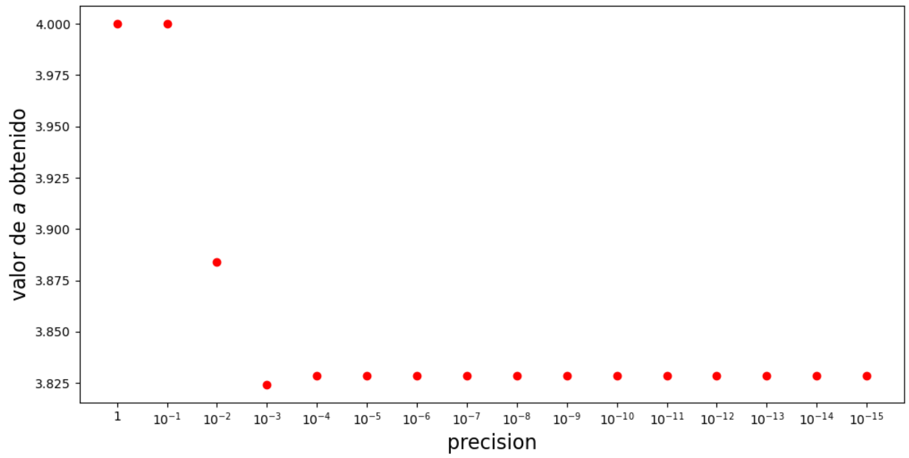

Figura 7: Valor calculado de a en función de la precisión pedida

4. Discusión y Conclusiones

Se logró efectivamente calcular un valor de a tal que la función $f(a)$ **(3)** tuviese un valor muy cercano al 0. Como se dijo previamente, este valor de a es muy cercano a 3.8261.

En términos de los tiempos de ejecución, la cantidad de iteraciones del algoritmo utilizado parece depender linealmente de i si se busca una precisión de 10^{-i}, lo que hace relativamente poco costoso aumentar la precisión del cálculo. De todas formas, podría no ganarse demasiado haciendo esto, ya que el algoritmo converge muy rápidamente y los valores de a obtenidos cada vez son prácticamente iguales. Se contó la cantidad de iteraciones en vez del tiempo total demorado, ya que el tiempo total depende del computador en el que se realicen los cálculos, y para este trabajo no se contó con uno particularmente bueno. Es probable que sería mucho más rápido (en tiempo total) si se hubiesen podido correr los programas en un computador de la Universidad.

El cálculo en general podría ser más exacto fácilmente si se calcula $\Gamma(k/2)$ con mayor precisión. Esto se debería hacer solo una vez, ya que se re-utiliza el mismo valor de Γ en vez de calcularlo para cada vez que se calcula χ^2.

El cambio de variable propuesto resultó en una función bastante suave y sin discontinuidades ni puntos de alto contraste, lo que la hace ideal para calcularla con métodos numéricos. Para las 2 integrales que se debieron realizar se encontró esta propiedad, por lo que ambas se calcularon usando la regla de Simpson, por su mayor precisión.

También se escogió utilizar el algoritmo de Newton para encontrar la raíz de $f(a) - 0.95$. Esta función era ideal para este método, ya que cerca del 0 su curva era bastante empinada, y además poseía exactamente un 0. Otra ventaja fue que por ser la integral de χ^2, su derivada ya se conocía e incluso se tenía implementado un método que la calculaba.

A partir del trabajo realizado se puede concluir que este problema es bastante fácil de resolver numéricamente, en el sentido de que todas las funciones son "fáciles" de calcular por un computador. Un problema es la lentitud de algunos cálculos, pero esto se puede resolver disminuyendo la precisión o contando con mejores computadores.