Instrucciones

Para realizar este certamen considere los siguientes puntos:

- Puede programar en su lenguaje de programación preferido. Si lo encuentra pertinente, puede usar códigos de varios lenguajes para realizar su certamen (a modo de ejemplo, puede llamar R en python o viceversa).
- Puede utilizar todas las funciones **para generar números aleatorios desde distribuciones univariadas** que su lenguaje de programación le entregue.
- El certamen debe ser entregado en Latex. La redacción del certamen es personal.
- Debe realizar un archivo de código por pregunta, debidamente comentado.
- Debe adjuntar sus códigos, y citarlos en el texto (e.g. *...los resultados fueron obtenidos con el archivo pregunta_1a.py*). **No debe escribir los códigos en el documento de entrega**.
- Fecha de entrega: 08 de Octubre del 2023, hasta las 23:59.
- Si usted lo requiere, puede realizar preguntas vía mail a *francisco.cuevas@usm.cl*.
- La entrega se debe hacer mediante AULA.
- Responder todo lo que pueda.

I. Sea $\mathcal{M}_p(\mathbb{R})$ el espacio de las matrices simétricas de dimensión $p \times p$ y con valores propios positivos. Se desea simular desde la siguiente distribución

$$f(\boldsymbol{X}|k,\boldsymbol{V}) = \frac{\det(\boldsymbol{X})^{(k-p-1)/2}}{2^{kp/2}\det(\boldsymbol{V})^{k/2}\Gamma_p\left(\frac{k}{2}\right)} e^{-\frac{1}{2}\mathrm{Tr}(\boldsymbol{V}^{-1}\boldsymbol{X})}, \text{ para } \boldsymbol{X},\boldsymbol{V} \in \mathcal{M}_p(\mathbb{R}). \qquad (1)$$

donde $\det(\boldsymbol{X})$ y $\mathrm{Tr}(\boldsymbol{X})$ son el determinante y la traza de la matrix $\boldsymbol{X}$ respectivamente y $k > p-1$. Sea $\Gamma(\cdot)$ la función Gamma, se define Γ_p por

$$\Gamma_p\left(\frac{k}{2}\right) = \pi^{p(p-1)/4}\prod_{j=1}^{p}\Gamma\left(\frac{k}{2} - \frac{j-1}{2}\right).$$

Escogiendo algún método de simulación, genere un reporte que al menos contenga:

- Una descripción del esquema de simulación, especificando las configuraciones realizadas.
- Gráficos donde se estudien los tiempos de realización respecto a la dimensión de la matriz p.
- Gráficos donde se justifique porque su método funciona.
- El hardware utilizado.
- Cualquier otra información que usted encuentre relevante.
- Utilize su método para simular matrices tales que sus valores propios se encuentren entre $[0, \lambda_{max}]$, donde λ_{max} es un valor a fijar.

Obs 1: En el reporte **NO** incluya la semilla, ni las iteraciones realizadas.

Obs 2: Debe realizar el reporte de manera tal que el lector pueda reproducir sus simulaciones y su código.

Obs 3: Para facilitar su reporte, los siguientes resultados podrían ser de utilidad:
Si $\boldsymbol{X}$ es una variable aleatoria con densidad dada por (1), entonces

- $\mathbb{E}[\boldsymbol{X}] = k\boldsymbol{V}$
- Sean X_{ij} y v_{ij} las entradas (i,j) de las matrices $\boldsymbol{X}$ y $\boldsymbol{V}$ respectivamente, entonces se tiene que $\mathrm{Var}[\boldsymbol{X}_{ij}] = k(v_{ij}^2 + v_{ii}v_{jj})$
- $X_{ii} \sim v_{ii}\chi_k^2$

II. En esta pregunta se explorarán los elementos básicos de lo que es la *simulación perfecta*, los cuales se basan en algoritmos recursivos. A modo de ejemplo, considere a $\boldsymbol{Algo1}(\alpha)$ descrito en Algoritmo 1.

Algoritmo 1 $\boldsymbol{Algo1}(\alpha)$

Generar $X \sim U(\{1, 2, \ldots, 10\})$ ▷ X es la distribución uniforme discreta
if $X \in \{1, 2, 3, 4, 5\}$ **then**
  Return X
else
  $X = \boldsymbol{Algo1}(\alpha)$
  Return X
end if

Para realizar un estudio formal, se deben introducen las siguientes definiciones:

Definición (Esquema recursivo probabilístico) *Se dice que un algoritmo* $\boldsymbol{Algo}(\alpha)$ *es un esquema recursivo probabilístico si el algoritmo hace elecciones aleatorias y, mientras el algoritmo corre, es capaz de llamarse a si mismo con diferentes parámetros* α.

Definición (Algoritmo de simulación perfecta) *Dado un conjunto de parámetros* $\mathcal{P}$ *y una familia de distribuciones objetivos* $\{\pi_\alpha : \alpha \in \mathcal{P}\}$*, un esquema recursivo probabilistico* $\boldsymbol{Algo}(\alpha)$ *es un algoritmo de simulación perfecta si, para todo* $\alpha \in \mathcal{P}$*, el algoritmo termina en tiempo finito con probabilidad* 1 *y regresa una muestra desde* π_α.

Considere ahora $\boldsymbol{Algo2}(\alpha)$, la siguiente variante de $\boldsymbol{Algo1}(\alpha)$:

Algoritmo 2 $\boldsymbol{Algo2}(\alpha)$

Generar $X \sim U(\{1, 2, \ldots, \max(\alpha, 5)\})$ ▷ X es la distribución uniforme discreta
if $X \in \{1, 2, 3, 4, 5\}$ **then**
  Return X
else
  $X = \boldsymbol{Algo2}(X)$
  Return X
end if

Diferente del Algoritmo $\boldsymbol{Algo1}(\alpha)$, el Algoritmo $\boldsymbol{Algo2}(\alpha)$ se llama a si mismo pero con un parámetro diferente en cada iteración. Este tipo de algoritmos son útiles para realizar busquedas locales, pero se vuelve más difícil de estudiar. Para esto, se introduce la siguiente definición:

Definición (Algoritmo localmente correcto) *En una iteración de* $\boldsymbol{Algo}(\alpha)$ *a un esquema recursivo probabilístico, suponga que cualquier llamado recursivo es de la forma* $\boldsymbol{Algo}(\beta)$ *puede ser reemplazado por generar de manera exacta un número aleatorio desde una distribución* π_β*. Si el resultado de hacer este reemplazo genera un algoritmo que simula datos desde la distribución* π_α*, entonces se dice que el algoritmo es localmente correcto con respecto a* $\{\pi_\alpha : \alpha \in \mathcal{P}\}$*.*

Para probar que un algoritmo es de simulación perfecta, se presenta el siguiente Teorema:

Teorema (Teorema fundamental de la simulación perfecta) *Suponga que* $\boldsymbol{Algo}(\alpha)$ *es un esquema recursivo probabilístico que satisface las siguientes propiedades:*

A.- Termina en tiempo finito con probabilidad 1 para todo α*,*

B.- Es localmente correcto con respecto a $\{\pi_\alpha : \alpha \in \mathcal{P}\}$*.*

Entonces el algoritmo simula exactamente desde π_α

Con estos ingredientes, se pide:

a) Pruebe que $\boldsymbol{Algo1}(\alpha)$ es un algoritmo de simulación perfecta.

b) Demuestre que $\boldsymbol{Algo2}(\alpha)$ es un algoritmo de simulación perfecta que simula la distribución uniforme.

c) Se puede demostrar (*no es necesario que lo haga*) que $\boldsymbol{Algo3}(g)$, en efecto, un algoritmo de simulación perfecto que simula datos desde h si $g \geq h$:

Algoritmo 3 $\boldsymbol{Algo3}(g)$

Generar $X \sim g$
Generar $U \sim U[0,1]$
if $U < h(X)/g(X)$ **then**
  Return X
else
  $Y = \boldsymbol{Algo3}(g)$
  Return Y
end if

Programe y compare $\boldsymbol{Algo3}(g)$ con el algoritmo de aceptación-rechazo cuando corresponda. Para esto, genere un reporte que incluya:

- El total de iteraciones (o equivalente) hasta aceptar una simulación,
- tiempo total de simulación,
- distribución resultante en ambos algoritmos.

Comente sus resultados.

III. En esta pregunta se busca simular funciones de $\mathcal{L}^2(\Omega)$. Para esto, recordamos que una función $f(t) \in \mathcal{L}^2(\Omega)$ se puede escribir mediante una secuencia de bases ortogonales $\{\varphi_i\}_{i=0}^{\infty}$ y una secuencia de números reales $\{\kappa_i\}_{i=0}^{\infty}$ como

$$f(t) := f(t, \{\kappa_i\}_{i=1}^{\infty}) = \sum_{i=0}^{\infty} \kappa_i \varphi_i(t). \tag{2}$$

Considerando la secuencia de variables aleatorias independientes $\{S_k\}_{k=1}^{\infty}$ donde S_k proviene de una distribución $\text{Beta}(1-\alpha, \theta + \alpha k)$. Se puede demostrar que el proceso $W_k = S_k \prod_{i=1}^{k-1}(1 - S_i)$ satisface que $W_k > 0$ para todo número entero k y que además $\sum_{k=0}^{\infty} W_k = 1$. Realice las siguientes simulaciones

a) Simular la secuencia W_k para distintos valores de α y θ. Describa lo observado y fundamente con gráficos apropiados

b) Usando la representación (2), genere un código que pueda generar funciones que satisfacen que:

1) sean periodicas y pares en $\mathcal{L}^2([-1, 1])$.
2) sean estríctamente crecientes en $\mathcal{L}^2([0, 1])$.
3) sean postivas e integren 1 en $\mathcal{L}^2([0, 1])$.

Obs 1: En cada pregunta debe de realizar un experimento de simulación que muestre los efectos de α y θ.
Obs 2: Considere agregar otras distribuciones si es necesario.

IV. Se desea estudiar, via simulaciones, el comportamiento de un interruptor de potencia de alto voltaje. Este interruptor está compuesto por 5 componentes que trabajan en paralelo que reciben daño cuando hay un cortocircuito. Las componentes tienen una vida de 100 puntos los cuales se van gastando en la medida que reciben daños. Para realizar el estudio de simulación, se propone un modelo con las siguientes características:

- El daño es aleatorio y se dividide entre el número de componentes que recibe daño.
- Las componentes menos dañadas son mas propensas recibir daño.
- En el momento que ocurre una falla, el número de componentes que recibe el daño es aleatorio.
- El daño recibido es aleatorio y se modela por la variable aleatoria $Y = 10X$, donde $X \sim \text{Beta}(\alpha, \beta)$.
- Debido a que el cortocircuito ocurre con condiciones aleatorias, se considera que $\alpha \sim U[0{,}75, 1{,}5]$, $\beta \sim U[2{,}75, 3{,}25]$.

a) Escriba en forma de modelo gráfico lo que ocurre al momento de la falla.

b) Realice histogramas que muestren el número de fallas, el total de daño que recibe el equipo, y otro de su interés.

Una componente muere si el daño total recibido es mayor o igual a 100 puntos, mientras que un equipo se declara inoperativo si todas sus componentes están muertas.

c) Proponga una distribución para los tiempos entre falla. Encuentre via simulación el tiempo esperado en que el equipo se declara inoperativo (Proponga dos esquemas de fallas: mucha frecuencia y baja frecuencia).

d) Para mayor protección, una empresa decide instalar varios equipos en **paralelo**, es decir, el equipo sistema se declara inoperativo si todos los equipos están inoperativos. Compare la distribución del tiempo de falla de esta configuración respecto a la de tener un solo equipo funcionando.

Obs 1: Si se debe hacer una simplificación, debe especificarla.