Métodos Numéricos

http://www.famaf.unc.edu.ar/∼serra/metodos_numericos.html

Guía 5 – Mayo de 2016

Problema 1: Escriba una función en Fortran 90, que genere números enteros pseudo–aleatorios utilizando el algoritmo congruencial:

$$j_{n+1} = a j_n + c \quad \texttt{módulo} \;\; m.$$

La función debe tener un argumento j_n, que la primera vez que se la llame debe ser la *semilla*, y dar como resultado j_{n+1}. Defina los valores de a, c y m como parámetros. *Ayuda*: utilizar `intent(INOUT)` para el argumento, y modificarlo en la función.

Problema 2: Encuentre algunos ejemplos en los que el generador congruencial del ejercicio anterior no sea de *período completo*, es decir, que tenga un período inferior a m.

Problema 3: Modifique la función del problema 1, para que genere números reales pseudo–aleatorios en el intervalo $[0, 1]$. Utilice los valores de a, c y m recomendados en el teórico. Incorpore esta función a un programa que genere $m/2$ pares de números pseudo–aleatorios, y grafique estos pares (x_i, y_i) usando *gnuplot* para verificar, a simple vista, si están uniformemente distribuidos.

Problema 4: Encuentre a tal que el generador congruencial de la forma:

$$j_{n+1} = a j_n \quad \texttt{módulo} \;\; 7.$$

sea de período completo. Luego, verifique que si se implementase este algoritmo con enteros de 4 bits (rango $[-8, 7]$, por lo tanto, sería de período máximo) usando el método de Schrage se obtienen las secuencias correctas, mientras que si se utiliza `mod()` o `modulo()` se producen errores (hacer en papel, usando las definiciones de `mod()` y `modulo()`).

Problema 5: Escriba una función que genere números reales pseudo–aleatorios en el intervalo $[0, 1]$ que sea del tipo congruencial y tenga período máximo para enteros de 4 bytes ($2^{31} - 1$). Utilice esta función en un programa que genere pares (x_i, y_i), y que **sólo escriba** los pares (x_i, y_i) tal que $x_i \in [0.1, 0.13]$ e $y_i \in [0.1, 0.13]$. Corra el programa para que genere 1×10^9 pares, y grafique en *gnuplot* los pares que cayeron en la región $[0.1, 0.13] \times [0.1, 0.13]$, luego observe la región $[0.1, 0.1005] \times [0.1, 0.1005]$. Qué evidencia este gráfico?

Fortran 90 cuenta con una subrutina intrínseca llamada *RANDOM_NUMBER* para la generación de números pseudo–aleatorios. Experimente su uso y compare con su función congruencial.

Problema 6: *Integrales de Monte Carlo* Escriba un programa para calcular integrales, y sus correspondientes errores estadísticos, en una dimensión utilizando el método de Monte Carlo. El programa debe incluir una `function` donde se definirá la función a integrar. Utilice este código para calcular la integral

$$I = \int_0^1 e^{-t} dt.$$

Elija valores de la cantidad de puntos, N, entre 100 y 30000, y compare los errores relativos, con los obtenidos con el método del trapecio y de Simpson (ver Prob. 7, guía 4). Grafique en escala *log–log* los errores relativos en función de N. Verifica que el error del método de Monte Carlo es proporcional a $N^{-1/2}$? Compare con el error estadístico correspondiente.

Problema 7: Modifique el programa del problema anterior de manera que calcule integrales en 3 dimensiones, y utilícelo para calcular el volumen de la intersección entre una esfera de radio 1, centrada en el origen, y un cilindro infinito de radio $1/2$, centrado en el eje z. Conociendo el valor exacto del volumen, calcule el error relativo de la integral de Monte Carlo en función de N y grafique (recuerde producir gráficos de *buena calidad*, como lo hizo en guías anteriores). Cree un archivo *postscritpt* con el gráfico y luego conviértalo a *PDF* (use el comando de linux *ps2pdf*).

Problema 8: Calcule usando el método de Monte Carlo la siguiente integral 20-dimensional:

$$I = \int_0^1 dx_1 \int_0^1 dx_2 \cdots \int_0^1 dx_{20} \, (x_1 + x_2 + \cdots + x_{20})^2$$

Elija el número de puntos, N, de la forma 2^i, con $i = 1, \cdots 20$, y calcule el error relativo para cada N. Grafique el error en función de $1/\sqrt{N}$ e identifique el comportamiento lineal. Utilice una escala conveniente.

Problema 9: *Mínimo global de funciones de varias variables* Use un método de Monte Carlo para localizar un mínimo global de las siguientes funciones:

a) Función *Cross-in-Tray*:

$$f(x,y) \; = \; -\frac{1}{10000}\left(\left|\sin(x)\,\sin(x)\,e^{\left|100-\sqrt{x^2+y^2}/\pi\right|}\right|+1\right)^{1/10}$$

en el cuadrado $0 < x, y < 10$. Nota: esta función tiene multiples mínimos locales en esa región, siendo su mínimo global $f(x = 1.349406685, y = 1.349406685) = -2.06261218$.

b) La función de Shubert:

$$f(x_1, x_2) = \left[\sum_{i=1}^{i=5} i\cos[(i+1)x_1 + i]\right]\left[\sum_{i=1}^{i=5} i\cos[(i+1)x_2 + i]\right], \;\; -10 \le x_i \le 10$$

en el cuadrado $-10 < x, y < 10$. Nota: esta función tiene 760 mínimos locales en esa región, siendo 18 de ellos globales con $f(x_m, y_m) = -186.7309$.

<u>Problema 10</u>: *Caminata al azar.* En una red cuadrada bidimensional implementar un algoritmo que realice una caminat al azar de n pasos. Iniciar la caminata en el sitio central de la red $R(0) = (0, 0)$.

a) Hallar el desplazamiento cuadrático medio $\langle (R(n) - R(0))^2 \rangle$ en función del número de pasos n, promediando ($\langle \ldots \rangle$) sobre $N = 10^6$ realizaciones de la caminata. Se verifica la ley $\langle R^2 \rangle \sim n$?

b) Subdividir la red en cuatro cuadrantes y contabilizar la cantidad de veces que el caminante termina en un dado cuadrante, comparar con el valor esperado $N/4$. Grafique estos resultados en función de n, comparando los resultados de distintos generadores.

<u>Problema 11</u>: *Error de redondeo.* Si asumimos que el error de redondeo es aleatorio, entonces el error de redondeo acumulado en un algoritmo de N pasos sería equivalente a una caminata aleatoria unidimensional de N pasos, con pasos de longitud proporcional a ϵ_M. Resultando:

$$\epsilon^{(N)}_{redondeo} \approx \sqrt{\langle R(N)^2 \rangle} \approx \sqrt{N}\epsilon_m$$

donde se usó que $R(0) = 0$. Verifique este comportamiento para los algoritmos de integración de Trapecio y Simpson. Considere los regímenes donde domina el error de redondeo. Verificar en simple y en doble precisión (defina la precisión a utilizar en un módulo).

Ejercicio Complementario

Problema 12: Compare la cantidad de puntos necesarios para calcular la siguente integral

$$I = \int_0^{2\pi} e^{-x} \sin(x)\, dx\,,$$

con una precisión de 1×10^{-6}, utilizando la regla compuesta de Simpson y el algoritmo de Monte–Carlo. El valor exacto de la integral es 0.499066278634.

Preguntas

- Cuáles son las características principales que debo conocer de un generador congruencial? Nombre algunos "defectos" de éstos generadores.
- Nombre algunas propiedades que debe tener un buen generador de números pseudo–aleatorios.
- Existen generadores de "verdaderos" números al azar (por ej. obteniéndolos de decaímientos radioctivo, ruido térmico, etc.). Nombre algunas ventajas y desventajas de éstos, frente a los generadores de números pseudo–aleatorios.
- En general, cuándo es conveniente utilizar integración de Monte–Carlo?