AULAS 25 E 26: FIS271 - Física Computacional I

Exercício 1. Considere uma população descrita pelo número de indivíduos P_n em um dado tempo n e que evolui segundo a relação de recorrência: $P_{n+1} = P_n(a - bP_n)$ com $n \in \mathbb{N}$. Fazendo $P_n = (a/b)x_n$ e $a = 4r$ obtemos:

$$x_{n+1} = 4rx_n(1 - x_n) \quad ,$$

com $0 \leq x_n \leq 1$. Dada a condição inicial x_0 e a taxa de crescimento $0 < r \leq 1$, os valores x_n para $n = 0, \ldots, N$ definem uma *trajetória* que descreve a evolução do número de indivíduos no sistema. Também nos referimos à relação acima como um *mapa unidimensional* ou *mapa logístico*, definido pela(o) transformação (mapeamento) $x_{n+1} = f_r^{(1)}(x_n)$. Dependendo do valor de r e de x_0 é interessante analisar a convergência da sequência para um valor x^* no limite $n \gg 1$.

a) Considerando as seguintes condições iniciais: $x_0 = 0.1$, $x_0 = 0.3$, $x_0 = 0.5$, $x_0 = 0.7$ e $x_0 = 0.9$, grafique as trajetórias x_n para $r = 0.24$ e $r = 0.26$ com $N = 40$. Quais os valores de $x^* = x_N$ ($N = 500$) para esses valores de r?

b) Utilizando os mesmos valores das condições iniciais do item (a), obtenha trajetórias para encontrar os valores de x^* para $r = 0.33$, $r = 0.43$, $r = 0.53$, $r = 0.63$ e $r = 0.73$.

c) Para certos valores de r é possível observar trajetórias com um comportamento periódico no limite $n \gg 1$ onde p valores, $x_1^*, x_2^*, \ldots, x_p^*$, são repetidos ciclicamente. Tais valores definem um ciclo com período p e podem ser determinados pela relação $x_{n+p} = f_r^{(p)}(x_n) = x_n$. Considerando as condições iniciais $x_0 = 0.1$ e 0.9, grafique trajetórias com $N = 100$ para $r = 0.85$, $r = 0.87$, $r = 0.89$ e $r = 0.96$. A partir dos gráficos, encontre o período p do ciclo para cada um desses valores de r.

d) Considere $r = 0.91$ e duas trajetórias com $N = 120$ partindo de condições iniciais muito próximas, $x_0 = 0.5000$ e $x_0' = 0.5001$. Faça os gráficos das duas trajetória e comente .

Exercício 2. Como veremos, diversas abordagens para resolver problemas na Física requerem o uso de números pseudo-aleatórios, os quais são obtidos por um **gerador** descrito por operações bem definidas. Bons geradores de números pseudo-aleatórios usualmente possuem [2]:

- **aleatoriedade**: números da sequência assumem valores entre 0 e 1 amostrados de acordo com uma distribuição de probabilidades uniforme em uma sequência minimamente correlacionada;
- **período longo**: um número grande de chamadas pode ser feito sem que a sequência se repita;
- **eficiência computacional**: execução rápida e que requer pouca memória;
- **reprodutibilidade**: estado inicial (*semente*) determina totalmente a sequência de números aleatórios;
- **portabilidade**: sequências idênticas para computadores/compiladores/sistemas operacionais diferentes;
- **homogeneidade**: até mesmo os bits são aleatórios.

Por exemplo, é possível obter um "gerador de número pseudo-aleatórios" considerando uma sequência obtida através da equação 1 no regime caótico, isto é, com $r = 0.995$. Neste caso o parâmetro (*semente*) x_0 define o **estado do gerador**, o qual gera uma sequência sequência de números "pseudo-aleatórios" x_1, x_2, x_3, x_4, etc. Infelizmente, o gerador definido pela Eq. 1 é **eficiente**, mas não possui as outras características acima.

a) Assumindo $x_0 = 0.1$, mostre que um gerador baseado na Eq. 1 possui **reprodutibilidade**, isto é, faça um gráfico mostrando que uma sequência de números pseudo-aleatórios x_k com $k = 1, \ldots, 100$ pode ser também obtida em duas etapas, isto é, obtendo primeiro x_k com $k = 1, \ldots, 50$ e depois utilizando x_{50} como *semente* para obter a sequência x_k com $k = 51, \ldots, 100$.

b) Embora o gerador definido pela Eq. 1 forneça valores somente no intervalo $]0, 1[$, ele NÃO possui a propriedade de **aleatoriedade** descrita acima pois a sequência de números gerada não será distribuída uniformemente. Isto pode ser confirmado através do cálculo do histograma `H(0:M)`, onde as entradas `H(m)` definem a quantidade de números de uma sequência com N números que se encontram no subintervalo $[x_{\mathtt{m}}, x_{\mathtt{m}} + \Delta x[$, com $x_m = \mathtt{m}\Delta x$ e o número de subintervalos `M` sendo definido de acordo com o valor da discretização $\Delta x = x_{\mathtt{m+1}} - x_{\mathtt{m}}$. A partir das sequências de números x_n com N números, obtenha histogramas normalizados $\hat{\mathtt{H}}(\mathtt{m}) = C\mathtt{H}(\mathtt{m})$, onde $C = 1/(\sum_{\mathtt{m}=0}^{\mathtt{M}} \mathtt{H}(\mathtt{m})\Delta x)$, com $\Delta x = 0.1$ para $N = 10^4$, 10^5 e 10^6. Quanto maior o N melhor será a aproximação entre os histogramas e a distribuição de probabilidades $\rho(x)$ utilizada para amostrar o número x. Discuta se os histogramas convergem para uma distribuição uniforme ou não.

Exercício 3. Considere agora o gerador de números pseudo-aleatórios disponível no arquivo `marsaglia.f90`, o qual é uma adaptação do gerador desenvolvido por Marsaglia [2] e que possui todas as características descritas acima. Tal gerador consiste de três subrotinas:

`ranmar(xr)`: subrotina principal que fornece um número aleatório `xr` entre 0 e 1 a cada vez que é chamada;

`rmaget()`: armezena o **estado atual** do gerador no arquivo `seedou.rng`;

`rmaset()`: inicializa o gerador dependendo das variáveis lógicas `seed_list` e `get_seed`. Com isso, três tipos de inicialização são possíves:

- ◇ **valores default:** `seed_list=.FALSE.` e `get_list=.FALSE.`, nesse caso não é necessário nenhum arquivo de entrada para inicializar o gerador;
- ◇ **utilizando lista:** `seed_list=.TRUE.` e `get_list=.FALSE.`, aqui é preciso tanto do arquivo `seeds.txt` (que contém uma lista de sementes) quanto do arquivo `input.dat` (o qual define qual semente da lista será utilizada);
- ◇ **continuando sequência:** `seed_list=.FALSE.` e `get_list=.TRUE.`, nesse caso é necessário renomear o arquivo `seedou.rng` para `seedin.rng`, o qual será lido para inicializar o estado do gerador exatamente do ponto onde foi chamado pela última vez.

a) Assim como feito anteriormente, mostre através de um gráfico que uma sequência de números pseudo-aleatórios x_k com $k = 1, \ldots, 100$ pode ser também obtida em duas partes, isto é, obtendo primeiro x_k com $k = 1, \ldots, 50$ e depois utilizando a inicialização **continuando sequência** para obter x_k com $k = 51, \ldots, 100$.
b) Refaça o item (b) do Exercício 1 mas agora utilizando o **gerador** do arquivo `marsaglia.f90`.

Referências:
[1] H. Gould, J. Tobochnik, W. Christian. An introduction to computer simulation methods: Applications to physical system (agosto de 2016).
[2] B. A. Berg. Markov Chain Monte Carlo Simulations and Their Statistical Analysis (World Scientific, 2004).