

# Projeto 2: Decaimento radioativo e números aleatórios

Pedro de Carvalho Braga Ilídio Silva - 9762595

Abril de 2017

## 1 Introdução

O presente projeto visa o estudo da implementação de números pseudo-aleatórios na linguagem de programação Fortran90. Serão criados e analisados geradores congruentes lineares. Em um segundo momento, será estudada uma aplicação típica: o processo de decaimento radioativo para uma determinada quantidade de amostras de núcleos instáveis. Gráficos do decaimento serão elaborados e sua natureza, discutida.

## 2 Desenvolvimento

### 2.1 Números aleatórios

Criou-se um gerador congruente linear de números aleatórios sob a forma de uma subrotina, chamada GENRDM (de “generate random number”). Seu funcionamento consistia em modificar uma variável inteira  $R$ , cujo valor inicial se nomeia “semente” ou “seed”, para um novo valor randômico a cada chamada, segundo a seguinte expressão:

$$R = MOD(aR + c, m)$$

Em que  $MOD$  é a função nativa para a operação de módulo no FORTRAN e  $a$ ,  $c$  e  $m$  são parâmetros inteiros.

Para medir o período do gerador, armazenou-se a semente também em outra variável  $A1$ , e executou-se um laço DO WHILE ( $R \neq A1$ ), que chamava GENRDM em  $R$  e incrementava uma variável contadora a cada ciclo.

É importante notar que fez-se necessário chamar a subrotina uma única vez antes do laço, para impedir que o condicional  $R \neq A1$  fosse violado já na primeira execução do loop e este, portanto, nem iniciasse. Disso resulta que a contadora deve iniciar em 1.

O programa descrito, incrementado de impressões auxiliares, foi rodado a partir de cinco valores sementes distintos, com os parâmetros  $a = 7, c = 4$  e  $m = 15$ , culminando no seguinte output:

SEED: 1		
R	1 :	11
R	2 :	6
O PERIODO VALE		3
SEED: 5		
R	1 :	9
R	2 :	7
R	3 :	8
R	4 :	0
R	5 :	4
R	6 :	2
R	7 :	3
R	8 :	10
R	9 :	14
R	10 :	12
R	11 :	13
O PERIODO VALE		12
SEED: 10		
R	1 :	14
R	2 :	12
R	3 :	13
R	4 :	5
R	5 :	9
R	6 :	7
R	7 :	8
R	8 :	0
R	9 :	4
R	10 :	2
R	11 :	3
O PERIODO VALE		12

SEED: 9

---

R	1 :	7
R	2 :	8
R	3 :	0
R	4 :	4
R	5 :	2
R	6 :	3
R	7 :	10
R	8 :	14
R	9 :	12
R	10 :	13
R	11 :	5

---

O PERIODO VALE 12

---

SEED: 12

---

R	1 :	13
R	2 :	5
R	3 :	9
R	4 :	7
R	5 :	8
R	6 :	0
R	7 :	4
R	8 :	2
R	9 :	3
R	10 :	10
R	11 :	14

---

O PERIODO VALE 12

Percebe-se que, com exceção da seed 1, o período manteve-se o mesmo para qualquer valor inicial de R, só dependendo da escolha dos parâmetros.

Repetindo o processo para  $m = 17$ , chega-se em:

---

SEED: 1

---

R	1 :	11
R	2 :	13
R	3 :	10
R	4 :	6
R	5 :	12

R	6	:	3
R	7	:	8
R	8	:	9
R	9	:	16
R	10	:	14
R	11	:	0
R	12	:	4
R	13	:	15
R	14	:	7
R	15	:	2
<hr/>			
O PERIODO VALE			16
<hr/>			
SEED:			5
<hr/>			
<hr/>			
O PERIODO VALE			1
<hr/>			
SEED:			10
<hr/>			
R	1	:	6
R	2	:	12
R	3	:	3
R	4	:	8
R	5	:	9
R	6	:	16
R	7	:	14
R	8	:	0
R	9	:	4
R	10	:	15
R	11	:	7
R	12	:	2
R	13	:	1
R	14	:	11
R	15	:	13
<hr/>			
O PERIODO VALE			16
<hr/>			
SEED:			9
<hr/>			
R	1	:	16
R	2	:	14
R	3	:	0
R	4	:	4
R	5	:	15
R	6	:	7

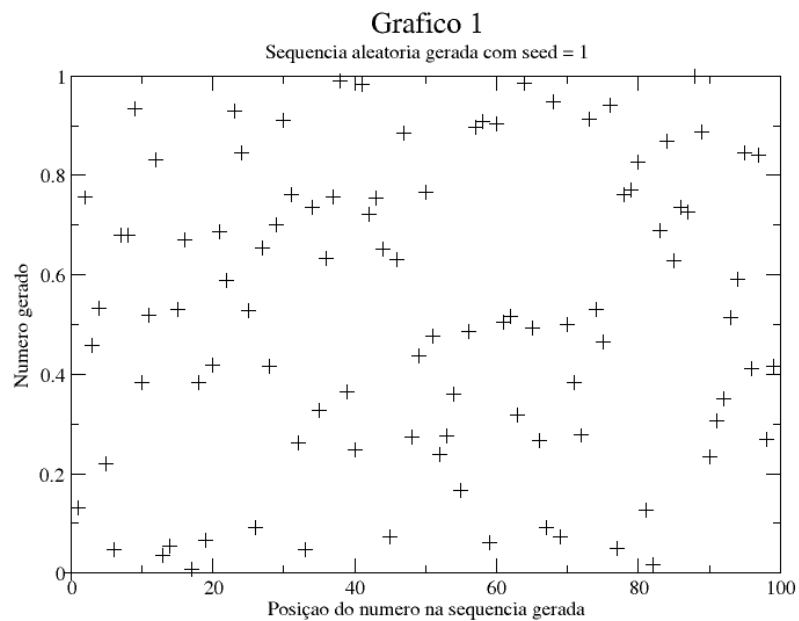
R	7 :	2
R	8 :	1
R	9 :	11
R	10 :	13
R	11 :	10
R	12 :	6
R	13 :	12
R	14 :	3
R	15 :	8
<hr/>		
O PERIODO VALE		16
<hr/>		
SEED:		12
<hr/>		
R	1 :	3
R	2 :	8
R	3 :	9
R	4 :	16
R	5 :	14
R	6 :	0
R	7 :	4
R	8 :	15
R	9 :	7
R	10 :	2
R	11 :	1
R	12 :	11
R	13 :	13
R	14 :	10
R	15 :	6
<hr/>		
O PERIODO VALE		16

Dessa vez, o caso da semente 5 mostrou-se *sui generis*. É perceptível que o laço não foi executado sequer uma vez, ou seja, R continua igual a A1 mesmo após GENRDM ser chamada, e a subrotina então forma uma sequência constante.

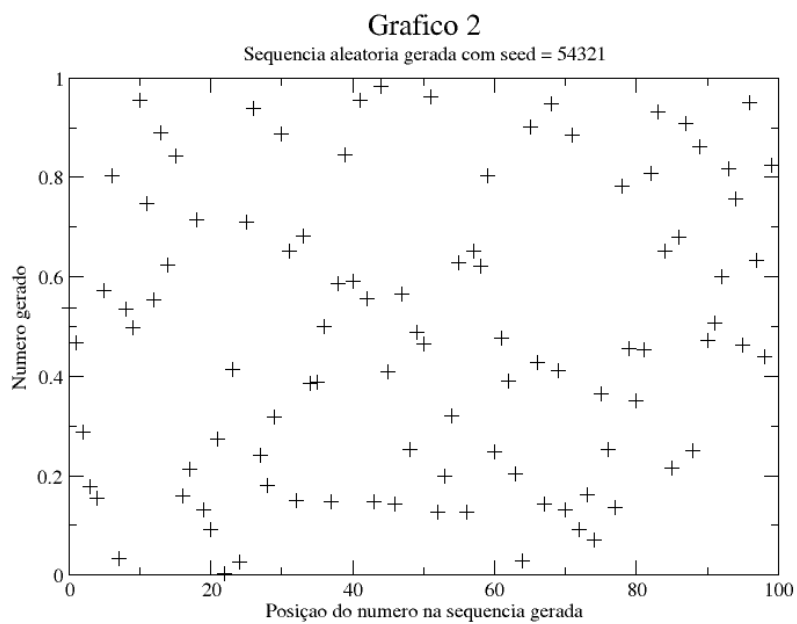
Em seguida, foi implementado o gerador "Padrão Mínimo" de Park e Miller adotando-se os parâmetros  $a = 7^5 = 16807$ ,  $c = 0$  e  $m = 2^{31} - 1 = 2147483647$ . Para evitar overflow de inteiros e o surgimento de números negativos inesperados, usou-se inteiros do tipo 2, que ocupam 8 bytes de memória. Para limitar os números gerados ao intervalo de 0 a 1, dividiu-se o resultado por m (somente ao imprimir, não modificando o valor de R).

Foram criadas cinco séries aleatórias de 100 termos, cada uma com uma seed diferente, e suas médias e desvio padrão foram determinados com um dos programas criados no último projeto:

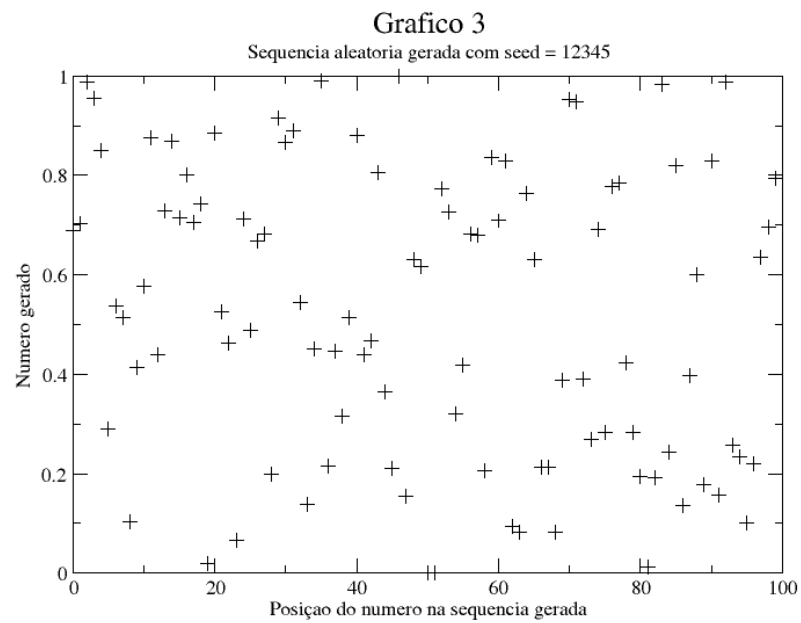




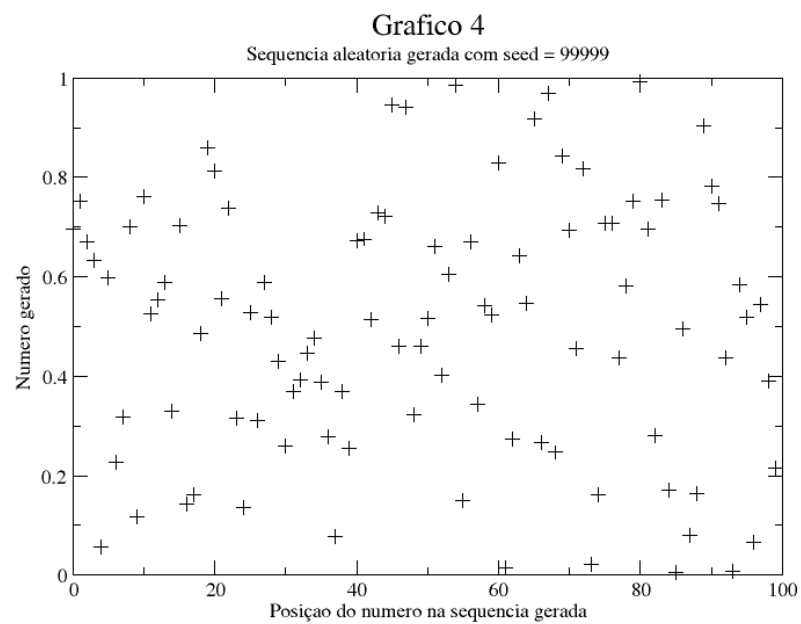
Mini-App: 3.14.2017.02.2017



Mini-App: 3.14.2017.02.2017

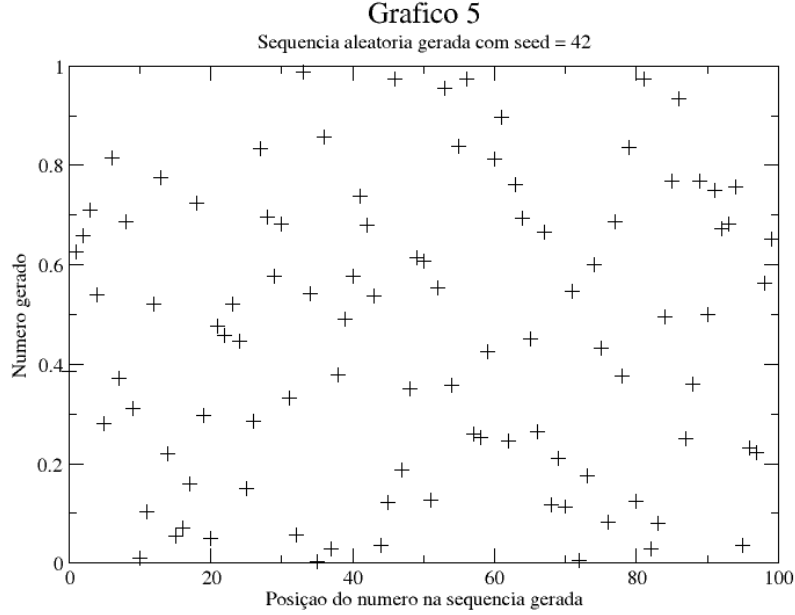


Ilhas Apr. 3 14:36:40 2017



Ilhas Apr. 3 14:30:29 2017





Idade Apr. 3 12:05:40 2017

Como esperado, a análise visual não revela nenhum padrão consistente nas representações gráficas. Tal desordem mostra-se útil em aplicações simples, menos formais, nas quais o rigor na aleatoriedade não se faz necessária.

## 2.2 Decaimento radioativo

O objetivo do vigente procedimento foi simular um processo de decaimento radioativo, por meio da produção de números pseudo-aleatórios.

Para esse programa, utilizou-se a função `ran2`, que gera números pseudo-aleatórios entre 0 e 1 com sementes diferentes a cada vez. Pode-se encontrá-la no site do Model, Analysis, and Prediction Program, da NASA ([https://map.nasa.gov/GEOSgcm\\_f90toHTML/ht](https://map.nasa.gov/GEOSgcm_f90toHTML/ht)), Acesso em : 6abr2017).

A probabilidade  $P$  de que um átomo decaia em um intervalo de tempo  $dt$  se relaciona com o tempo de meia vida  $\tau$  da seguinte forma:

$$dP = -\frac{1}{\tau}dt \quad (1)$$

Definiu-se o tempo de meia vida como 1, de forma que a probabilidade de decaimento equivale ao intervalo de tempo considerado, que, por sua vez, foi definido como 0.1. A quantidade inicial de átomos foi estabelecida como 1000, e o tempo máximo de simulação  $t_{max}$ , em quanto tempo esta chegará ao fim, será tomado como 8 (decidido a partir da observação dos resultados).

Elaborou-se laço que vai de  $t=1$  até  $t=t_{\text{max}}/dt$ , e que gera  $R$  números aleatórios com a função `ran2` e testa se foi obtido valor menor que  $P$ , incrementando em uma unidade uma variável de contagem caso a condicional se aplique. Tal processo corresponde à contar quantos átomos decaíram em um intervalo de tempo.

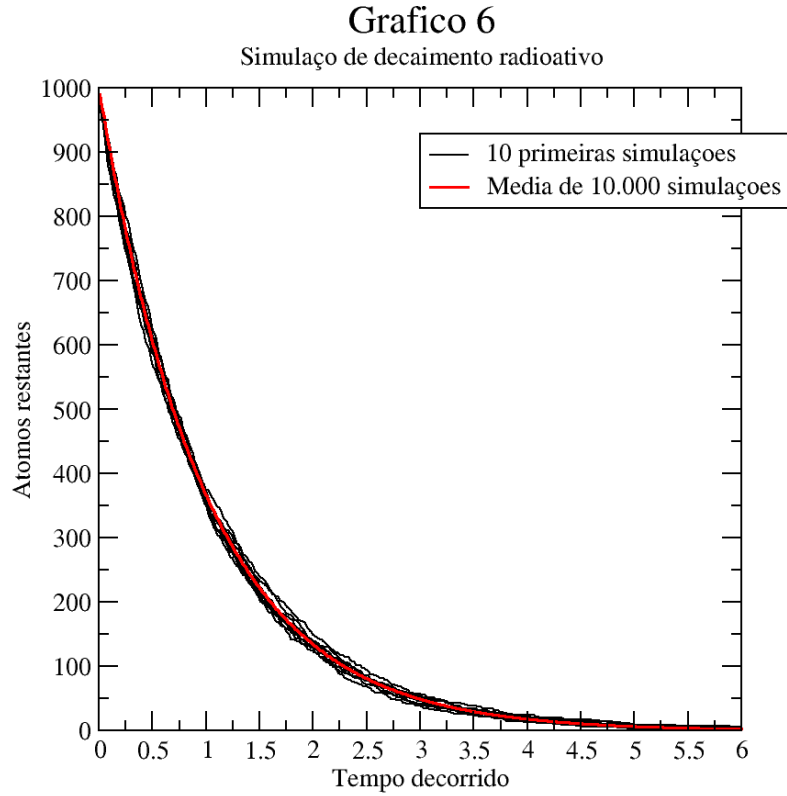
Na fase seguinte, o número de átomos decaídos era subtraído da quantidade inicial de átomos, a variável de contagem era zerada, e o loop rodava novamente, para contar os que decairiam no intervalo de tempo seguinte. Observou-se que, na maioria das vezes, não restavam mais do que dois átomos em  $t=8$ , motivando a escolha  $t_{\text{max}}=8$ .

O procedimento foi repetido  $R = 10^4$  vezes (tratam-se, portanto, de dois laços DO encadeados), e foram calculadas a média ( $\bar{x}$ ), desvio padrão:

$$\sigma = \sqrt{\frac{1}{R} \sum_{i=1}^R (x_i - \bar{x})^2}$$

e variância normalizada pela média, para cada intervalo de tempo. A variância normalizada é definida como o quadrado do desvio padrão sobre a média dos valores.

As 10 primeiras simulações e as médias de cada intervalo são compiladas no gráfico 6.



A natureza exponencial do decaimento se torna clara pela visualização do gráfico 6, o que se explica pela relação mostrada pela equação 1. Aproximando-se  $dP$  para o número  $dN$  de átomos decaídos no intervalo  $dt$  sobre a quantidade total  $N$  de átomos restantes e integrando-se a igualdade obtém-se:

$$dP = -\frac{1}{\tau} dt$$

$$\frac{dN}{N} = -\frac{1}{\tau} dt$$

$$\ln N = -\frac{t}{\tau} + t_0$$

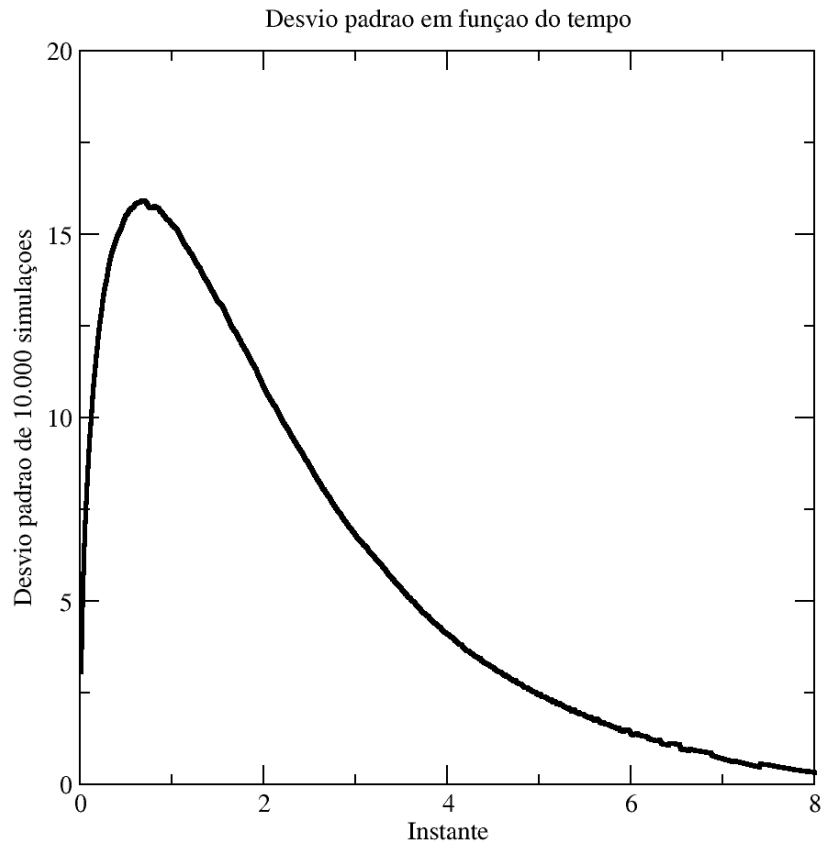
Em que  $t_0$  é o tempo inicial, que culmina, se definido como nulo, em

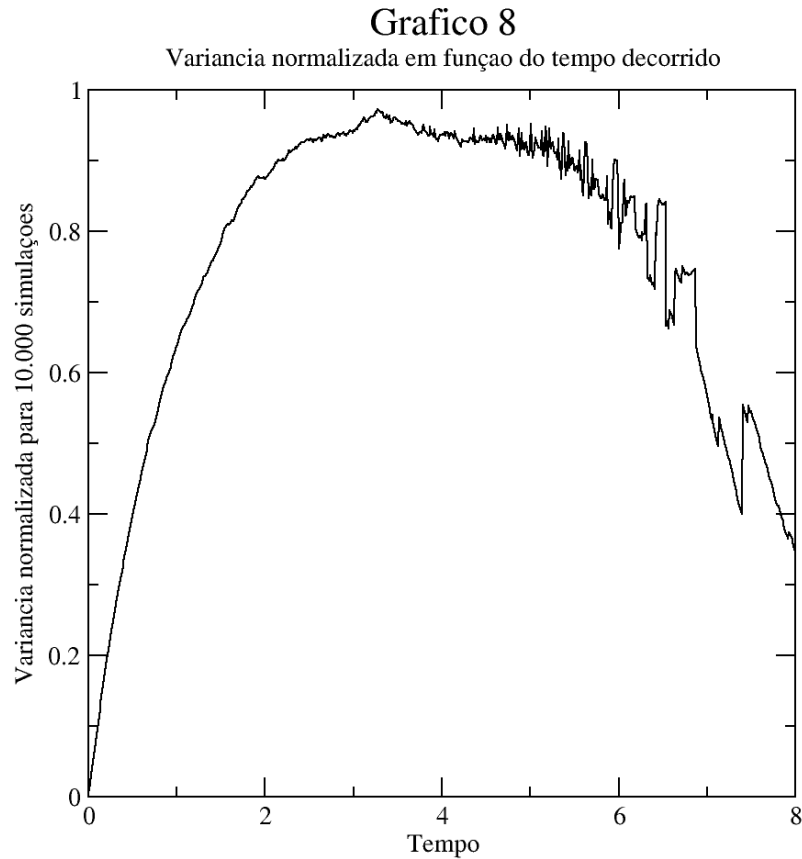
$$N = e^{-\frac{t}{\tau}}$$

Resultado condizente com o comportamento do gráfico 6.

O desvio padrão e a variância normalizada para cada intervalo de tempo nos  $10^4$  processos simulados são apresentados, respectivamente, pelos gráficos 7 e 8.

### Grafico 7





Ambos tem valores iniciais nulos, visto que no primeiro instante,  $N$  era igual em todas as simulações. Em um primeiro momento, observa-se rápido crescimento nos dois gráficos visto que a quantidade de átomos decaídos em um intervalo depende da mesma quantidade no intervalo anterior, e tratando-se de um processo aleatório, o desvio padrão dos valores tende a aumentar com o tempo.

Contudo como a quantidade de átomos decaídos tende a ser (dada as características estocásticas do conjunto) proporcional a  $N$ , ela tende a valores cada vez menores com o passar do tempo, já que  $N$  diminui. Isto é, a quantidade restante de átomos varia cada vez menos, levando o desvio padrão de  $N$  a convergir novamente, visto que o desvio padrão também depende de  $N$ .

A normalização com relação à média procura eliminar essa dependência de  $N$  na divergência dos valores, de forma que esperaríamos ver uma assíntota horizontal no gráfico 8.

Tanto o gráfico 7 como o 8 apresentam ruído, oscilações imprevisíveis dos valores, à medida que  $t$  aumenta. Isto se deve à perda de acuidade do desvio padrão conforme a razão entre a diferença entre os valores e os próprios valores

aumenta ( $\frac{dN}{N}$ ), ou seja, quando  $dN$  torna-se significativa com relação a  $N$ , perde sua natureza diferencial. A normalização, originadora dos valores plotados em 8, amplifica esse fenômeno, fazendo com que o ruído ofusque completamente o caráter assintótico previsto para gráfico 8.

É possível relacional intuitivamente o espaçamento vertical entre as séries do gráfico 6 ao comportamento do desvio padrão mostrado no gráfico 7: ambos tem variação qualitativamente sincronizada.

Foi também elaborado o programa abaixo, que calcula uma matriz hist de dimensões 2 por R, que armazena, para um único intervalo de tempo, quantas vezes cada quantidade de átomos aparece entre as R amostras, sendo que a quantidade ocupa uma coluna e o número de átomos a outra, e dados relacionados ocupam a mesma linha. Na grande maioria das vezes, não serão usadas todas as R linhas da matriz, e um condicional foi utilizado para imprimir somente os dados relevantes (descartar os itens nulos de hist).

```

===== HISTOGRAMA =====

allocate ( hist (2,r) )

do tesc= 100, 800, 100 ! loop mosntruoso

!----- limpa lixo -----!
do i=1, r
    hist (1,i) = 0
    hist (2,i) = 0
end do
!-----!

! POS      a pr xima posi   o que ainda n o foi
! ocupada em hist , necessario para nao sobrescrever

POS = 1

inedito = .true.

! tesc      a posi   o do tempo escolhido
! (ex. se tesc = 5, teremos os dados
! referentes ao quinto instante:
!          (5 * dt) = 0.5)

do ri = 1, r
    do i =1, r
        if (mat(tesc , ri) .eq. hist(1,i)) then
            hist(2,i) = hist(2,i)+1
            inedito = .false.
        end if
    end do
end do

```

```

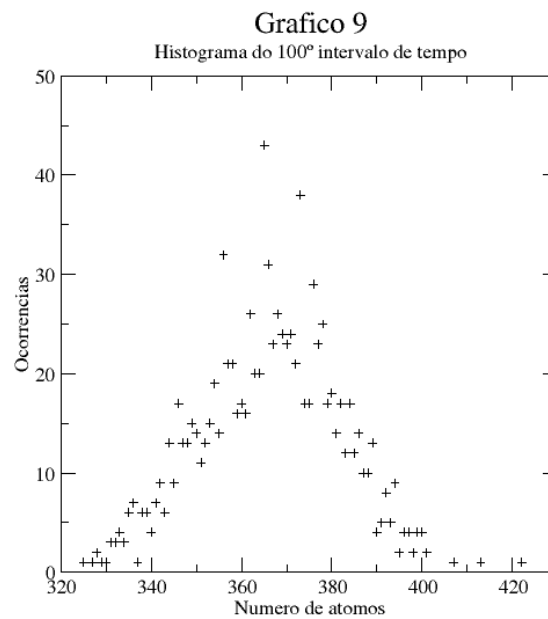
        exit
    end if
end do

if (inedito) then
    hist(1, pos) = mat(tesc, ri)
    hist(2, pos) = 1
    pos = pos+1
end if

inedito=.true.
end do

```

Fez-se necessário igualar todos os elementos da matriz a 0, inicialmente, pois o lixo de memória se constituiu um empecilho. O programa foi rodado para diferentes intervalos, e os resultados são exibidos nos gráficos a seguir:



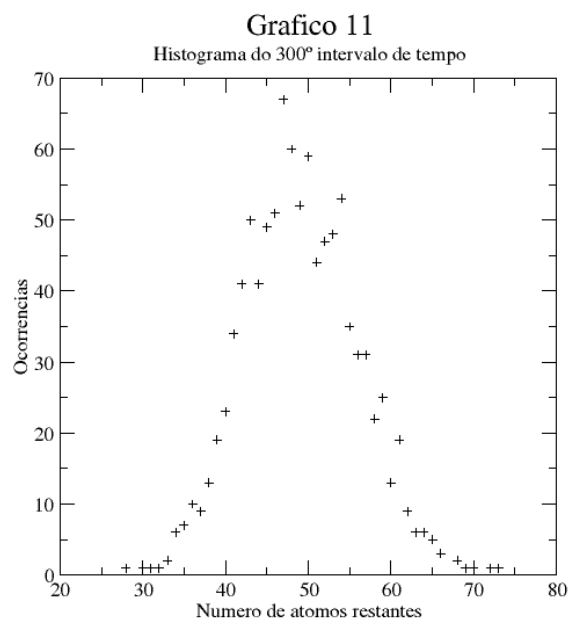
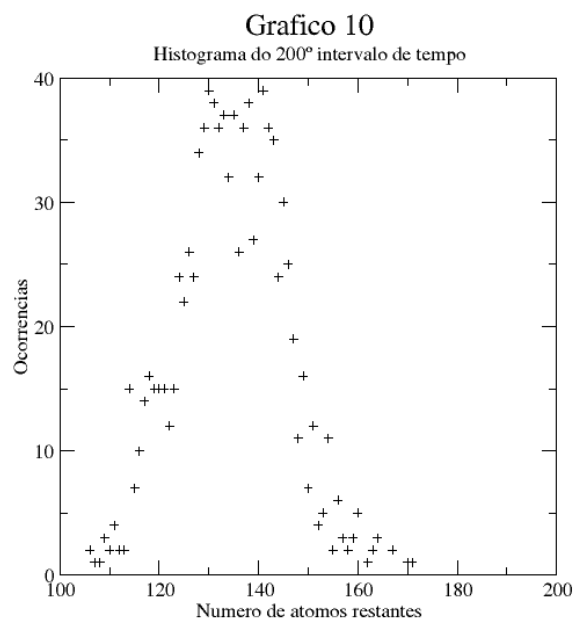




Grafico 12

Histograma do 400º intervalo de tempo

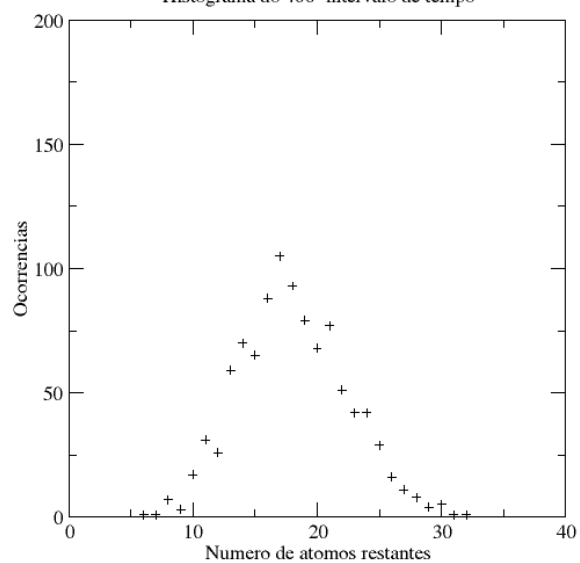


Grafico 13

Histograma do 500º intervalo de tempo

