Universidade de São Paulo

Introdução à Física Computacional - Projeto 2

João Victor Dell Agli Floriano - 10799783

1. Tarefa A - Cálculo da Média de x

Este programa tem como objetivo testar o gerador de números aleatórios do FORTRAN, para o qual calcularemos a média de valores gerados por ele da seguinte maneira:

$$< x^i >, i = 1, 2, 3, 4 \tag{1}$$

No caso:

$$< x^i >= \frac{1}{N}\sum_{n=1}^{N} x^i, 0 \leq x \leq 1 \tag{2}$$

A seguir, o programa que implementa o cálculo da média dos números aleatórios gerados pela função **rand()** do FORTRAN, que gera $N = 10^7$ números aleatórios:

```
      program main


        N = 10000000
        r = 0.0


        call srand(1001)


        do i = 1, 4
                 r = 0.0
                 do j = 0, N
                    r = r + rand()**i
                 end do
                 write(*,*) "i = ", i, "<x**i> = ", r/N
        end do

        stop
      end program main
```

Compilando e rodando o programa, temos:

```
joao@joao-Inspiron-15-7000-Gaming: ~/Área de Trabalho/Intro a Fiscomp/Segundo...
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Segu
do Projeto/tarefaA$ f77 med.f -o med
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Segu
do Projeto/tarefaA$ ./med
 i =           1 <x**i> =   0.500177681
 i =           2 <x**i> =   0.330046237
 i =           3 <x**i> =   0.246917993
 i =           4 <x**i> =   0.197750300
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Segu
do Projeto/tarefaA$
```

Obtendo-se assim resultados curiosos para i variando de 1 a 4. Verifica-se, por exemplo, que para i = 1, ou seja, x, a média dos números aleatórios gerados gira em torno de 0.5, assim como para i = 2, x^2, a média gira em torno de 0.33. Analisando o valor das médias crescendo o i, percebe-se algo importante: os valores os quais as médias se aproximam são iguais aos valores ao se integrar x^i de 0 a 1:

$$\int_0^1 x dx = \frac{x^2}{2}\Big|_0^1 = 0.5 \tag{3}$$

$$\int_0^1 x^2 dx = \frac{x^3}{3}\Big|_0^1 = 0.33...$$

$$\int_0^1 x^3 dx = \frac{x^4}{4}\Big|_0^1 = 0.25$$

$$\int_0^1 x^4 dx = \frac{x^5}{5}\Big|_0^1 = 0.20$$

Resultado este esperado, pois a equação (2) que define a média se assemelha à uma das definições da integral[1]:

$$\int_0^1 x^i dx = \lim_{N\to\infty} \sum_{i=0}^{N} x^i \frac{(1-0)}{N} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N} x^i \tag{4}$$

Que, tirando o limite:

$$\frac{1}{N} \sum_{i=0}^{N} x^i = < x^i > \tag{5}$$

[1] Definição disponível em: https://pt.wikipedia.org/wiki/Integral

2. Tarefa B - Andarilhos Aleatórios em 1D

B1)

Nesta tarefa, consideramos o caso de um número M de andarilhos que, em uma dimensão, tem a probabilidade p de dar um passo de comprimento idx para a direita, e q = 1 - p de dar um passo de mesmo comprimento para a esquerda. Com isso, por (2) podemos calcular a posição média final de todos os andarilhos nesse sistema, que será a nossa <x>, e a média das posições quadráticas, que será a nossa <x²>.

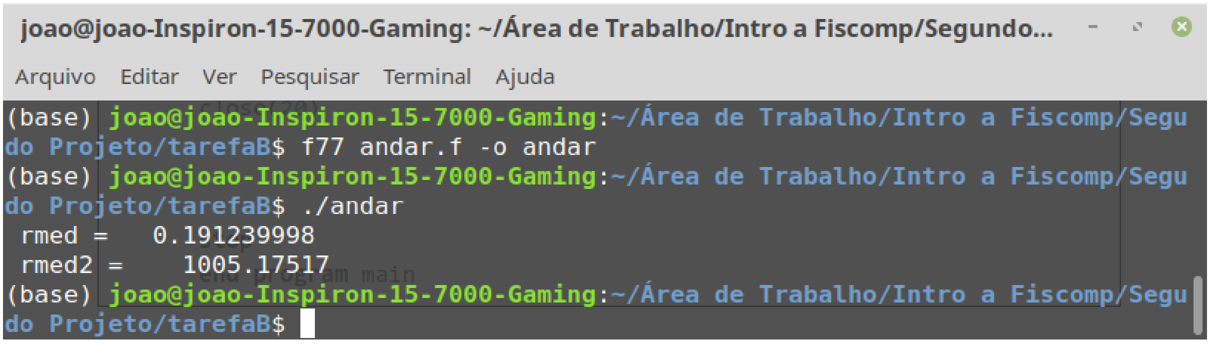
No caso de p = q = 0.5, para N = 1000 passos e M = 10^5 andarilhos, o seguinte programa implementa o que foi discutido:

```
      program main




      parameter (jseg = 100) !numero de divisoes a serem feitas espaco
para o plot do histograma
      parameter (M = 100000)
      integer iandarilho(M)
      integer iespaco(jseg)
      real xmed(M)
      real x2med(M)



      r = 0.0 !numero
      p = 0.50 !probabilidade p
      rmed = 0.0 !média das posições medias
      rmed2 = 0.0 !média das posições quadráticas médias
      N = 1000 !número de passos, definindo o tamanho máximo do domínio
(de -1000 a 1000)
      idx = 1 !step que consiga acessar o domínio inteiro de -1000 a 1000





      !inicializaremos todos os andarilhos em 0
      do k = 1, M
            iandarilho(k) = 0
            xmed(k) = 0
            x2med(k) = 0
      end do


      !Andarilhos andando
      call srand(1001)
      do j = 1, M
```

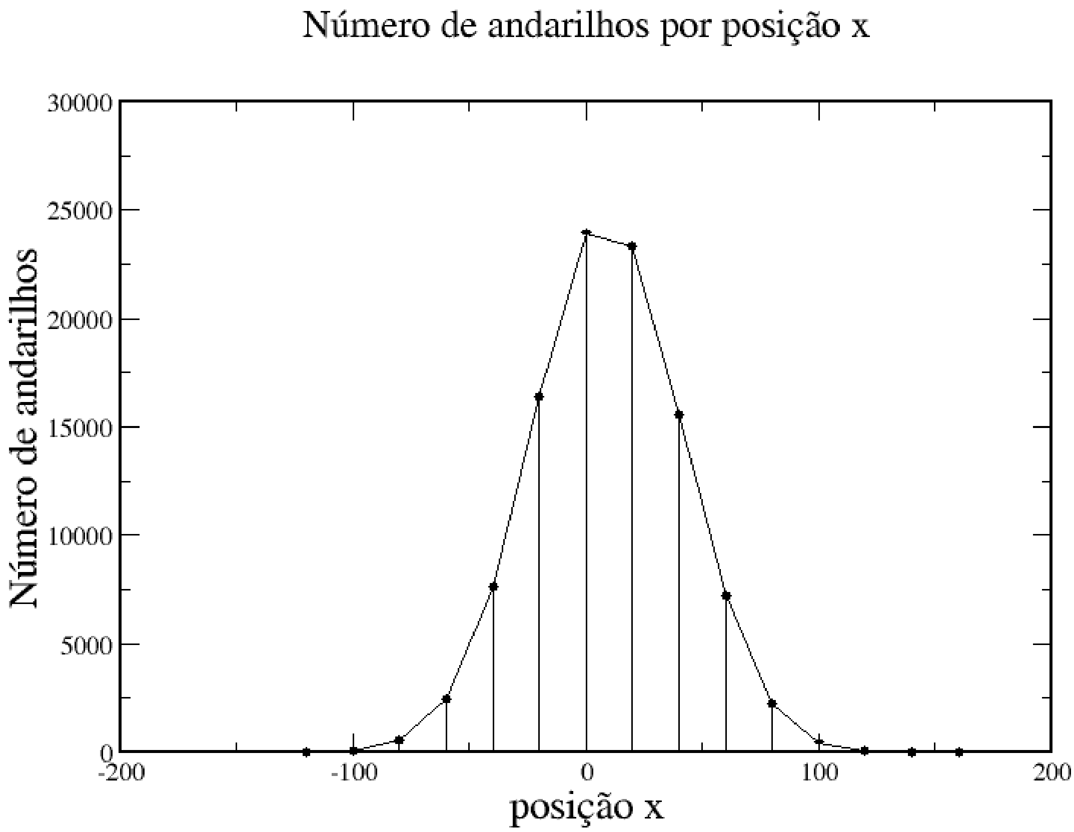
```fortran
		do i = 1, N
			r = rand()
			if(r .LT. p) then
			iandarilho(j) = iandarilho(j) + idx
			else
			iandarilho(j) = iandarilho(j) - idx
			end if
		end do
		xmed(j) = xmed(j) + iandarilho(j)
		x2med(j) = x2med(j) + iandarilho(j)**2
	end do



	!do para calcular as posicoes medias, posicoes quadraticas medias
	!e a media das posicoes medias e quadraticas medias
	do i = 1, M
		rmed = rmed + xmed(i)
		rmed2 = rmed2 + x2med(i)
	end do


	rmed = rmed/M
	rmed2 = rmed2/M

	write(*,*) "rmed = ", rmed
	write(*,*) "rmed2 = ", rmed2




	istart = -N !extremo negativo do dominio
	iseg = -1*2*istart/jseg !tamanho do reticulado em função do número
de passos (istart = N)

	open(20, file = "saida-B-1-10799783.dat")

	do i = 1, jseg

		istart = istart + iseg
		iespaco(i) = 0

		do j = 1, M

			if(iandarilho(j) .LE. istart) then
			if(iandarilho(j) .GT. (istart -iseg)) then
				iespaco(i) = iespaco(i) + 1
			end if
			end if

		end do

		if(iespaco(i) .gt. 0) then
			write(20,*) istart, iespaco(i)
```

```
            end if

      end do

      close(20)



      stop
      end program main
```

Rodando este programa, teremos:



Além de $<x>$ e $<x^2>$, é possível também montar um histograma das posições de cada andarilho. Isto é feito definindo um número máximo de segmentações desejadas a serem feitas no domínio, e a partir de uma variável auxiliar iterativa, checar quantos andarilhos se encontram dentro de um intervalo [x, x + segmentação]. Feito isso, o histograma é montado:



Rendendo uma curva do tipo gaussiana, que mostra uma distribuição de um maior número de andarilhos ao redor da posição 0, valor esperado visto que a probabilidade de andar para a direita é igual a de andar para a esquerda.

B2)

Repetindo o código agora para p = 0.33..., 0.25 e 0.2 respectivamente, temos:

Para verificar se estes resultados são condizentes, é possível comparar com as formas analíticas[2] derivadas para <x> e <x²>:

Sendo <nd> a média de passos à direita e <ne> a média de passos à esquerda:

$$< x >=< n_d > - < n_e >= N(p-q) \tag{6}$$

$$< x^2 >= 4Npq + N^2 - 4N^2pq \tag{7}$$

Que, para p = 0.33…, 0.25 e 0.20:

$$\begin{gathered} p = \frac{1}{3} \\ < x >= 1000 \cdot (0.333 - 0.666) = 1000 \cdot (-0.333) = -333.333... \\ < x^2 >= 4 \cdot 1000 \cdot 0.33... \cdot 0.66... + 1000^2(1 - 4 \cdot 0.33... \cdot 0.66...) = 113775 \end{gathered} \tag{8}$$

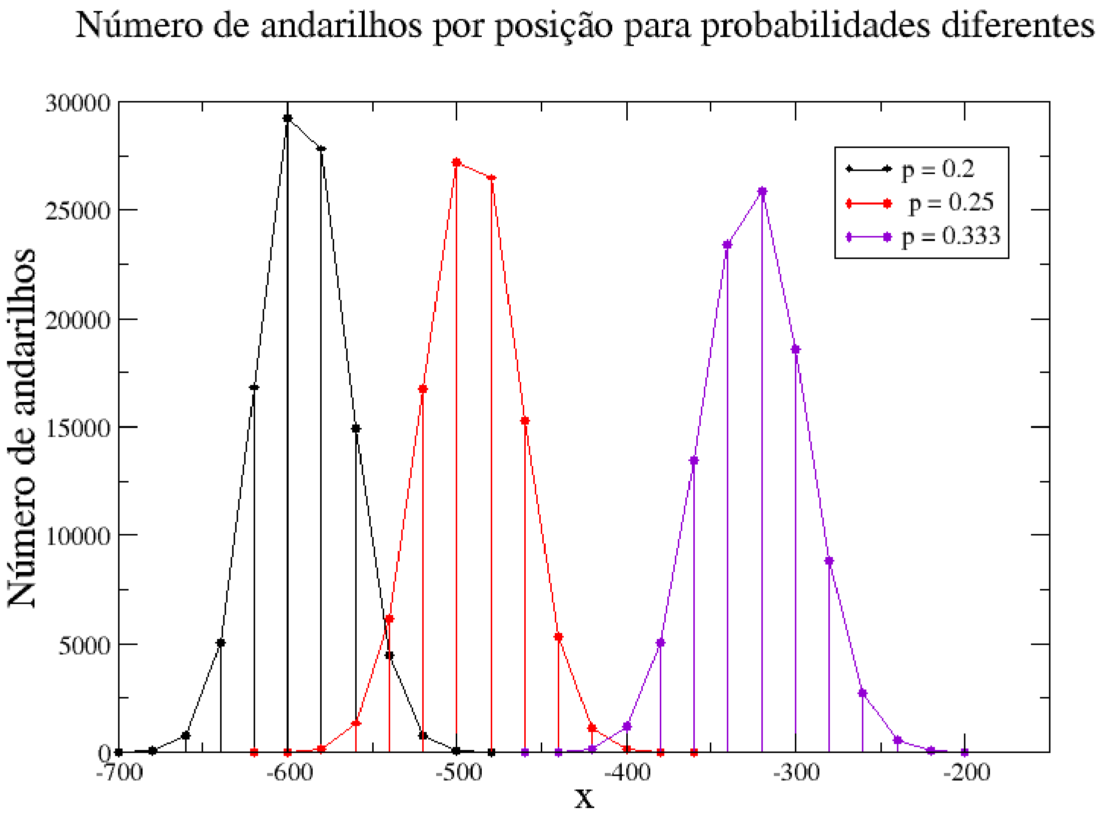
$$\begin{gathered} p = \frac{1}{4} \\ < x >= 1000 \cdot (0.25 - 0.75) = 1000 \cdot (-0.5) = -500 \\ < x^2 >= 4 \cdot 1000 \cdot 0.25 \cdot 0.75 + 1000^2(1 - 4 \cdot 0.25 \cdot 0.75) = 250750 \end{gathered} \tag{9}$$

$$\begin{gathered} p = \frac{1}{5} \\ < x >= 1000 \cdot (0.2 - 0.8) = 1000 \cdot (-0.6) = -600 \\ < x^2 >= 4 \cdot 1000 \cdot 0.20 \cdot 0.80 + 1000^2(1 - 4 \cdot 0.20 \cdot 0.80) = 360640 \end{gathered} \tag{10}$$

Nos permitindo verificar, assim, que os valores calculados pelo programa se aproximam muito dos valores esperados pela forma analítica.

Novamente, é possível fazer um histograma das posições dos andarilhos para cada caso:

[2] Resultados retirados das notas de aula suplementares.



3. Tarefa C - Andarilhos Aleatórios em 2D

Para este programa, generalizamos o caso de andarilhos para duas dimensões, agora com probabilidade p = 0.25 igual para todas as quatro direções (norte, sul, leste e oeste). Assim, podemos calcular a posição média em duas dimensões, além do delta quadrado, definido como:

$$\Delta^2 = <\vec{r}^2> - <\vec{r}> \cdot <\vec{r}> \tag{11}$$

O seguinte programa implementa o andarilho em duas dimensões, assim como o cálculo das médias e do delta quadrado:

```
program main


parameter (M = 100) !numero de andarilhos
dimension iandarilho(M,2) !posicoes dos andarilhos
dimension rmed(M,2)
dimension r2med(M,2)
dimension omed(2)
dimension omed2(2)


p = 0.25 !probabilidade de ir para cima, baixo, esquerda ou direita
r = 0.0  !número aleatório
```

```fortran
idx = 1  !step

N = 1000 !número de passos

omed(1)  = 0.0
omed(2)  = 0.0
omed2(1) = 0.0
omed2(2) = 0.0
delta2 = 0.0 !variavel para calcular a delta2

do i = 1, M
      do j = 1, 2
            iandarilho(i, j) = 0.0
            rmed(i, j)    = 0.0
            r2med(i, j)  = 0.0
      end do
end do


call srand(1001)
do i = 1, M
      do j = 1, N
            r = rand()
            if(r .LE. p) then

            iandarilho(i, 1) = iandarilho(i, 1) + idx

            else if (r .LE. p + p) then

            iandarilho(i, 2) = iandarilho(i, 2) + idx

            else if (r .LE. p + p + p) then

            iandarilho(i, 1) = iandarilho(i, 1) - idx

            else

            iandarilho(i, 2) = iandarilho(i, 2) - idx

            end if


      end do
      rmed(i, 1) = rmed(i, 1) + iandarilho(i, 1)
      rmed(i, 2) = rmed(i, 2) + iandarilho(i, 2)

      r2med(i, 1) = r2med(i, 1) + iandarilho(i, 1)**2
      r2med(i, 2) = r2med(i, 2) + iandarilho(i, 2)**2
end do



do i = 1, M
      omed(1) = omed(1) + rmed(i, 1)
```

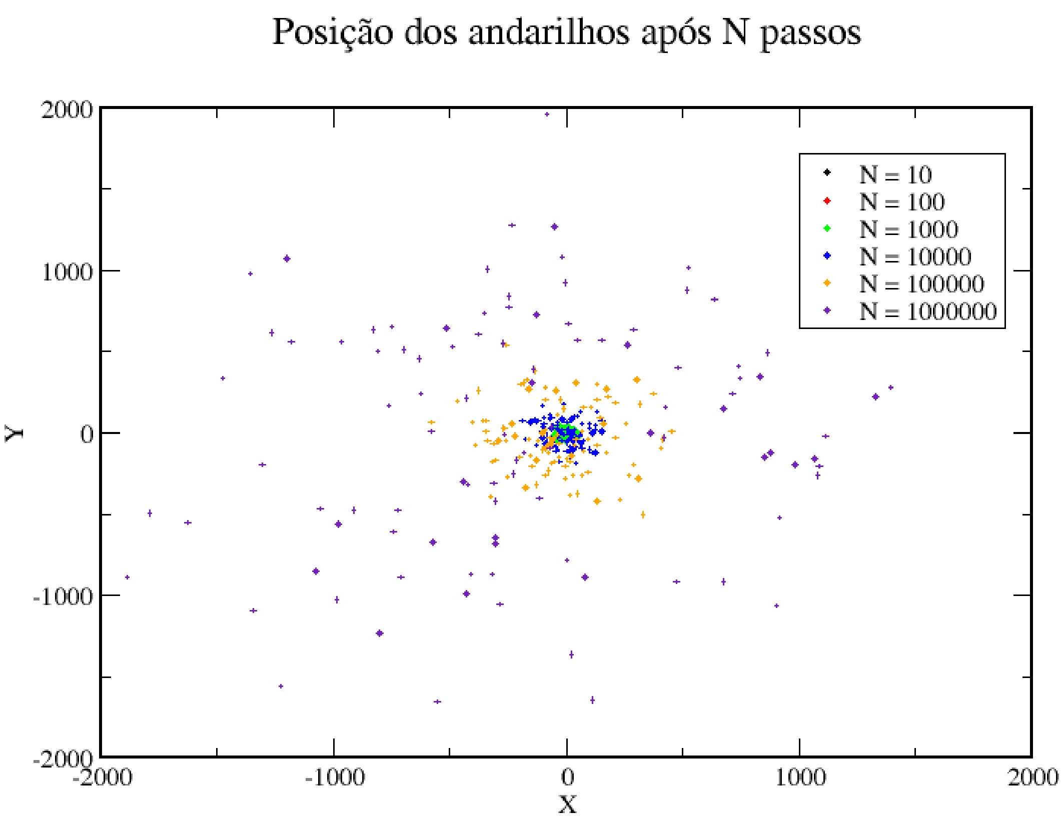
```
            omed(2) = omed(2) + rmed(i, 2)

            omed2(1) = omed2(1) + r2med(i, 1)
            omed2(2) = omed2(2) + r2med(i, 2)


      end do

      omed = omed/M
      omed2 = omed2/M

      write(*,*) "<x> = ", omed(1), "<y> = ", omed(2)
      write(*,*) "<x**2> = ", omed2(1), "<y**2> = ", omed2(2)

      delta2 = omed2(1) + omed2(2) - omed(1)**2 - omed(2)**2
      write(*,*) "delta**2 = ",delta2

      open(20, file = "saída-C-1-10799783.dat")
      do i = 1, M
            write(20, *) iandarilho(i, 1), iandarilho(i, 2)
      end do
      close(20)



      stop
      end program main
```

Executando o programa, temos:

```
joao@joao-Inspiron-15-7000-Gaming: ~/Área de Trabalho/Intro a Fiscomp/Segundo...
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Segu
do Projeto/tarefaC$ f77 bidim.f -o bidim
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Segu
do Projeto/tarefaC$ ./bidim
 <x> =   -2.93000007     <y> =   -0.469999999
 <x**2> =    544.090027     <y**2> =    445.329987
 delta**2 =    980.614258
(base) joao@joao-Inspiron-15-7000-Gaming:~/Área de Trabalho/Intro a Fiscomp/Segu
do Projeto/tarefaC$
```

O que, novamente, confere com os valores esperados, pois $\langle x \rangle$ e $\langle y \rangle$ são muito próximos a zero para probabilidades iguais em todas as direções, $\langle x^2 \rangle$ e $\langle y^2 \rangle$ também é próximo ao resultado esperado em (7).

Esse código também nos permite fazer um diagrama da posição de M = 100 andarilhos após N passos, com N = 10, 100, 1000, 10000, 100000, 1000000:

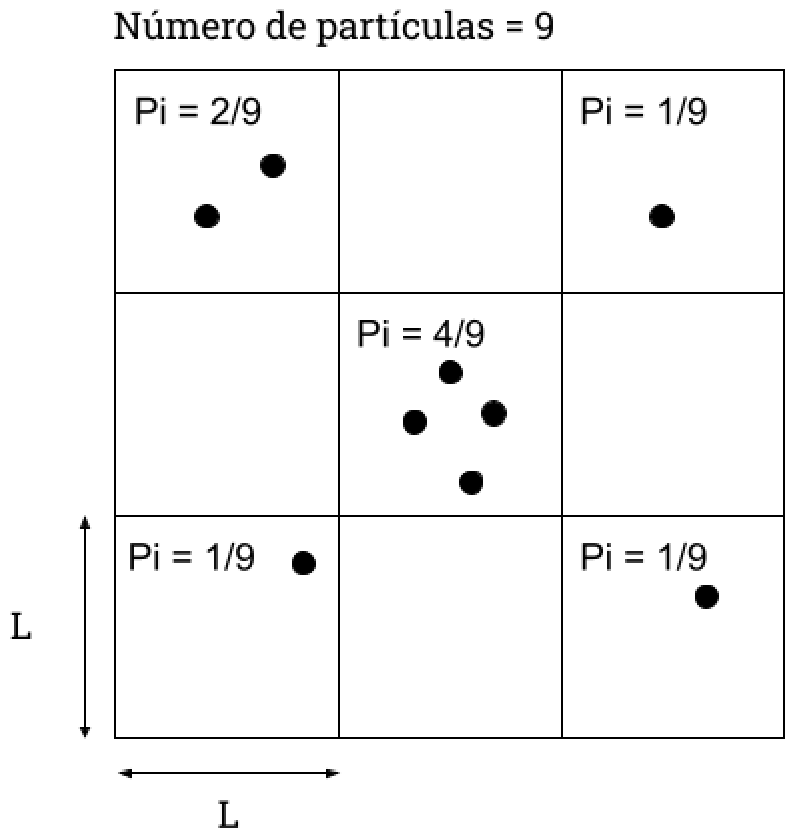


4. Tarefa D - Cálculo da Entropia em 2D

Este programa tem como objetivo calcular a entropia do sistema de moléculas simulado na tarefa anterior. A entropia nesse sistema é definida como:

$$S = -\sum_{i=1} P_i ln(P_i) \tag{12}$$

Sendo Pi a probabilidade de se encontrar uma molécula em um micro-estado i, sendo a mesma definida como a quantidade de moléculas encontrada em um reticulado, um segmento do espaço bidimensional de lado L (maior que um passo), dividida pelo número total de moléculas existentes no sistema. A seguir, um exemplo de como o espaço é subdividido:



A seguir, o código que implementa esse cálculo:

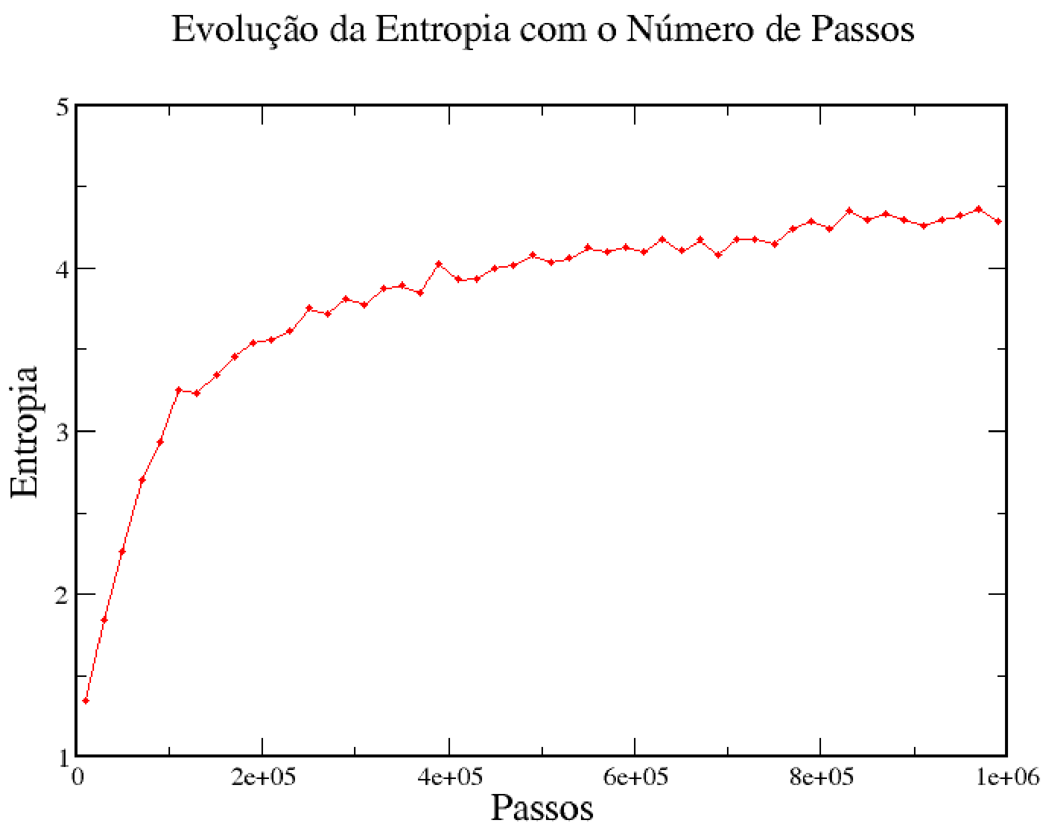
```
program main

parameter (jseg = 100) !segmentações a serem feitas no espaço
parameter (M = 100) !número de andarilhos
dimension iandarilho(M,2)
dimension rmed(M,2)
dimension r2med(M,2)
dimension omed(2)
dimension omed2(2)
dimension iespaco(jseg,jseg)




p = 0.25
r = 0.0
idx = 1
N = 10000 !número de passos por ciclo
S = 0.0
ibound = N
ixstart = -ibound
iystart = -ibound


omed(1)  = 0.0
omed(2)  = 0.0
```

```
      omed2(1) = 0.0
      omed2(2) = 0.0

      do i = 1, M
            do j = 1, 2
                  iandarilho(i, j) = 0.0
                  rmed(i, j)    = 0.0
                  r2med(i, j)   = 0.0
            end do
      end do


      call srand(1003)
      open(21, file = "saida-D-1-10799783.dat")
      do l = 1, 100 !ciclo para aumentar o número de passos e calcular a
entropia

            do i = 1, M
                  do j = 1, N
                  r = rand()
                  if(r .LE. p) then

                        iandarilho(i, 1) = iandarilho(i, 1) + idx

                  else if (r .LE. p + p) then

                        iandarilho(i, 2) = iandarilho(i, 2) + idx

                  else if (r .LE. p + p + p) then

                        iandarilho(i, 1) = iandarilho(i, 1) - idx

                  else

                        iandarilho(i, 2) = iandarilho(i, 2) - idx

                  end if


                  end do

                  rmed(i, 1) = rmed(i, 1) + iandarilho(i, 1)
                  rmed(i, 2) = rmed(i, 2) + iandarilho(i, 2)

                  r2med(i, 1) = r2med(i, 1) + iandarilho(i, 1)**2
                  r2med(i, 2) = r2med(i, 2) + iandarilho(i, 2)**2

            end do




            iseg = -1*2*ixstart/jseg
```

```fortran
        do j = 1, jseg
              ixstart = ixstart + iseg
              iystart = -ibound
              do i = 1, jseg

              iespaco(j,i) = 0
              iystart = iystart + iseg

              do k = 1, M
              if(iandarilho(k,1) .LE. ixstart .AND.
     &iandarilho(k,1) .GT. (ixstart -iseg) .AND. iandarilho(k,2) .LE.
     &iystart .AND.  iandarilho(k,2) .GT. (iystart -iseg)) then
                    iespaco(j,i) = iespaco(j,i) + 1
                    end if
              end do


              end do
        end do




        !calculo da entropia
        do i = 1, jseg
              do j = 1, jseg
              e = iespaco(j,i)
              resp = e/M
              if(resp .GT. 0.0) then
                    S = S - (resp)*log(resp)
              end if
              end do
        end do


        !escrita dos valores das entropias num documento externo
para o plot
        if(S .NE. 0.0) then
              write(*,*) l*N, ", S = ",S
              write(21,*) l*N, S
        end if
        S = 0.0

      end do
      close(21)



      stop
      end program main
```

Executando o código, temos:

Com os valores de evolução da entropia, é possível elaborar um gráfico:



Mostrando que, conforme as partículas andam mais, a entropia cresce, porém na forma da curva do logaritmo natural, chegando, depois de um tempo, a um limite de crescimento.