7600017 - Introdução à Física Computacional - Projeto 2

Raphael Vieira Moreira da Serra

Setembro de 2025

Introdução

Primeira Tarefa

A fim de testarmos o gerador de números aleatórios, calculemos alguns momentos da distribuição aleatória gerada, isto é:

$$\langle x^n \rangle, \text{ para } n = 1, 2, 3, 4$$

Faça a média acima gerando um número grande N de números aleatórios (escolha apropriadamente N). Que resultado você esperaria? Compare com os resultados esperados e explique os obtidos.

Para implementar esse Gerador Congruente Linear, devemos usar a seguinte fórmula:

$$x_{n+1} = (ax_n + b) \ mod \ m, n = 1, 2, ...$$

Posteriormente, deveríamos testar a distribuição calculando seus momentos. Eles são calculados da seguinte forma:

$$\langle x^n \rangle = \frac{1}{N} \sum_{k=1}^{N} (X_k - c)^n$$

O código em Fortran foi da seguinte forma:

```
REAL FUNCTION N_MEDIA(soma, N, rmedia, iord, NUM_RAND)
        INTEGER NUM_RAND(1600,2)
        soma = 0
        DO i = 1, N
          soma = soma + ((NUM_RAND(i,2) - rmedia))**iord
        END DO
```

```
        N_MEDIA = soma/N
END FUNCTION

PROGRAM LCG
INTEGER NUM_RAND(1600, 2)
REAL N_MEDIA

WRITE(*,*) 'Digite A'
READ(*,*) iA
WRITE(*,*) 'Digite B'
READ(*,*) iB
WRITE(*,*) 'Digite M'
READ(*,*) M
WRITE(*,*) 'Digite a quantidade de repetições N'
READ(*,*) N
WRITE(*,*) 'Digite a seed'
READ(*,*)iseed

NUM_RAND(1, 1) = 1
NUM_RAND(1, 2) = iseed

DO i=2,N
  NUM_RAND(i, 1) = i
  NUM_RAND(i, 2) = MOD((iA*(NUM_RAND((i-1), 2)) + iB),M)
END DO

DO i=1,N
  WRITE(*,*) NUM_RAND(i,1), NUM_RAND(i,2)
END DO

!Média
soma = 0
DO i=1,N
  soma = soma + NUM_RAND(i,2)
END DO
rmedia = soma / N
WRITE(*,*) 'A média é: ', rmedia

!VariÂncia
WRITE(*,*)'A variância é: ', N_MEDIA(soma, N, rmedia, 2, NUM_RA
&ND)
!Assimetria
WRITE(*,*)'O terceiro momento é: ', N_MEDIA(soma, N, rmedia, 3,
&NUM_RAND)
!Curtose
WRITE(*,*)'O quarto momento é: ', N_MEDIA(soma, N, rmedia, 4, NU
```

```
     &M_RAND)
      END
```

Para testar o sucesso desse código, rodei uma sequência simples fácil de verificar usando papel e caneta. Segue a interação no terminal:

```
     Digite A
4
 Digite B
1
 Digite M
9
 Digite a quantidade de repetições N
10
 Digite a seed
0
           1           0
           2           1
           3           5
           4           3
           5           4
           6           8
           7           6
           8           7
           9           2
          10           0
 A média é:    3.59999990
 A variância é:    7.44000006
 O terceiro momento é:    2.59200168
 O quarto momento é:    93.3791962
```

Depois, gerei uma sequência com M igual ao maior valor que o tipo de variável $REAL*4$ consegue gerar:

```
     Digite A
35
 Digite B
6
 Digite M
32767
 Digite a quantidade de repetições N
50
 Digite a seed
1
           1           1
           2          41
           3        1441
```

4	17674
5	28790
6	24646
7	10674
8	13159
9	1833
10	31394
11	17485
12	22175
13	22490
14	748
15	26186
16	31807
17	31940
18	3828
19	2918
20	3835
21	3163
22	12410
23	8385
24	31345
25	15770
26	27684
27	18703
28	32038
29	7258
30	24667
31	11409
32	6117
33	17499
34	22665
35	6873
36	11192
37	31289
38	13810
39	24618
40	9694
41	11626
42	13712
43	21188
44	20712
45	4052
46	10758
47	16099
48	6432
49	28524

```
        50       15336
 A média é:    15561.8604
 A variância é:    99181984.0
 O terceiro momento é:    1.45348624E+11
 O quarto momento é:    1.79926381E+16
```

Segunda Tarefa

> Vamos considerar agora o problema de andarilhos aleatórios em uma dimensão. Aqui, em cada unidade de tempo, cada caminhante, independentemente onde esteja, dá um passo à direita (esquerda) com probabilidade $p(q = 1p)$. O caso $p = q = 1/2$ corresponde a um caminhante tão desnorteado que ele nã se lembra de onde veio e nem qual o rumo certo a tomar. O caso em que $p \neq q$ corresponde ao viajante aleatório em uma ladeira. A questão nesta tarefa é calcular $\langle x \rangle$ e $\langle x^2 \rangle$ após um certo número N de passos.

A segunda tarefa tinha como objetivo simular andarilhos aleatórios em uma única dimensão. Na primeira parte, com probabilidades iguais e, na segunda, com probabilidadades diferentes.

A lógica do algoritmo é simples: para cada passo, gera-se um número aleatório $r \in [0, 1]$ e se esse r for maior do que 0.5, é somado 1 numa variável que computa a posição do andarilho(e se for menor, subtrai-se 1).

O código da primeira parte foi da seguinte forma:

```
REAL FUNCTION N_MEDIA(soma, N, rmedia, iord, NUM_RAND)
        INTEGER NUM_RAND(1600,2)
        soma = 0
        DO i = 1, N
          soma = soma + ((NUM_RAND(i,2) - rmedia))**iord
        END DO
        N_MEDIA = soma/N
END FUNCTION

PROGRAM ANDARILHOS
INTEGER IP(1600, 2), MN(1000,2)
INTEGER arquivo
REAL N_MEDIA

arquivo = 1

OPEN(UNIT=arquivo, FILE='dados.dat', STATUS='UNKNOWN')
```

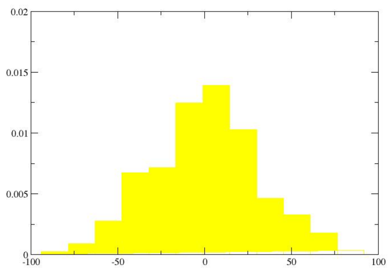
```
WRITE(*,*) 'Quantos andarilhos?'
READ(*,*) M
WRITE(*,*) 'Quantidade de passos?'
READ(*,*) N

!Organização da matriz de passos
DO i = 1, M
  IP(i,1) = i
END DO
!Fim da Organização

!Cálculo dos passos
DO i = 1, M
  k = 0
  DO j = 1, N
    r = rand()
    IF (r .GT. 0.5) THEN
            k = k + 1
    ELSE
            k = k - 1
    END IF
  END DO
  IP(i, 2) = k
END DO
!Fim cálculo

!Calculor o valor máximo e mínimo
MAXIMA = IP(1,2)
DO i = 1, M
  IF (MAXIMA .LT. IP(i,2)) THEN
          MAXIMA = IP(i,2)
  ELSE
  END IF
END DO

MINIMA = IP(1,2)
DO i = 1, M
  IF (MINIMA .GT. IP(i,2)) THEN
          MINIMA = IP(i,2)
  ELSE
  END IF
END DO
!Fim cálculo

!Criação da função N(x)
DO i = MINIMA, MAXIMA
```

```
  k = 0
  DO j = 1, M
    IF (i .EQ. IP(j,2)) THEN
            k = k + 1
    ELSE
    END IF
  END DO
  MN(i+1-MINIMA,1) = i
  MN(i+1-MINIMA,2) = k
END DO
!Fim da criação

rmedia = N_MEDIA(soma, M, 0.0, 1, IP)
WRITE(*,*) 'A média é: ', N_MEDIA(soma, M, 0.0, 1, IP)
WRITE(*,*) 'A variância é: ', N_MEDIA(soma, M, rmedia, 2, IP)

DO i=1,M
  WRITE(arquivo,*)IP(i,1), IP(i,2)
END DO
CLOSE(UNIT=arquivo)
!Fim Debugging
END PROGRAM
```

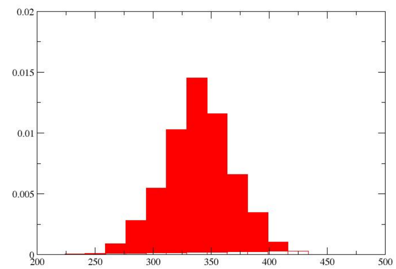
Para a segunda parte, foi somente alterado a condição do loop contido nas linhas 33 até 44. Tornando assim, o código capaz de representar diversas probabilidades.

```
DO i = 1, M
  k = 0
  DO j = 1, N
    r = rand()
    IF (r .GT. prob) THEN
            k = k + 1
    ELSE
            k = k - 1
    END IF
  END DO
  IP(i, 2) = k
END DO
```

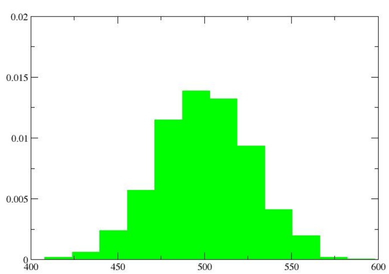
Os histogramas mapeando quantos andarilhos existem por cada posição foram os seguintes:



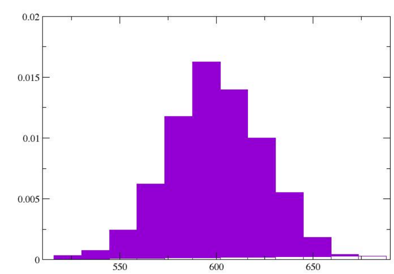
Histograma para $p = \frac{1}{2}$. $\langle x \rangle = 0.9$ e $\langle x^2 \rangle = 959.7$.



Histograma para $p = \frac{1}{3}$. $\langle x \rangle = 339$ e $\langle x^2 \rangle = 888$.



Histograma para $p = \frac{1}{4}$. $\langle x \rangle = 500$ e $\langle x^2 \rangle = 758$.



Histograma para $p = \frac{1}{5}$. $\langle x \rangle = 600$ e $\langle x^2 \rangle = 662$.

As curvas tem comportamento Gaussiano como seria esperado pelo fato de que variáveis aleatórias tem Distribuição Normal.

Terceira Tarefa

Considere agora o caso do andarilho bidimensional não enviesado, i.e., com iguais chances($\frac{1}{4}$) ele dá um passo em qualquer direção dos pontos cardeais: norte, sul, leste e oeste. Calcule $\langle r \rangle$ e Δ^2. Repare que estes andarilhos perfazem o mesmo tipo de movimento que moléculas no processo de difusão, como por exemplo a difusão de um pingo de leite numa xícara de café. Façaa um diagrama das posições das moléculas após um número N de passos ($N = 10, 10^2, 10^3, 10^4, 10^5, 106$)

A fim de resolver o problema de andarilhos bidimensionalmente, eu generalizei a lógica da segunda tarefa usando números complexos. Assim:

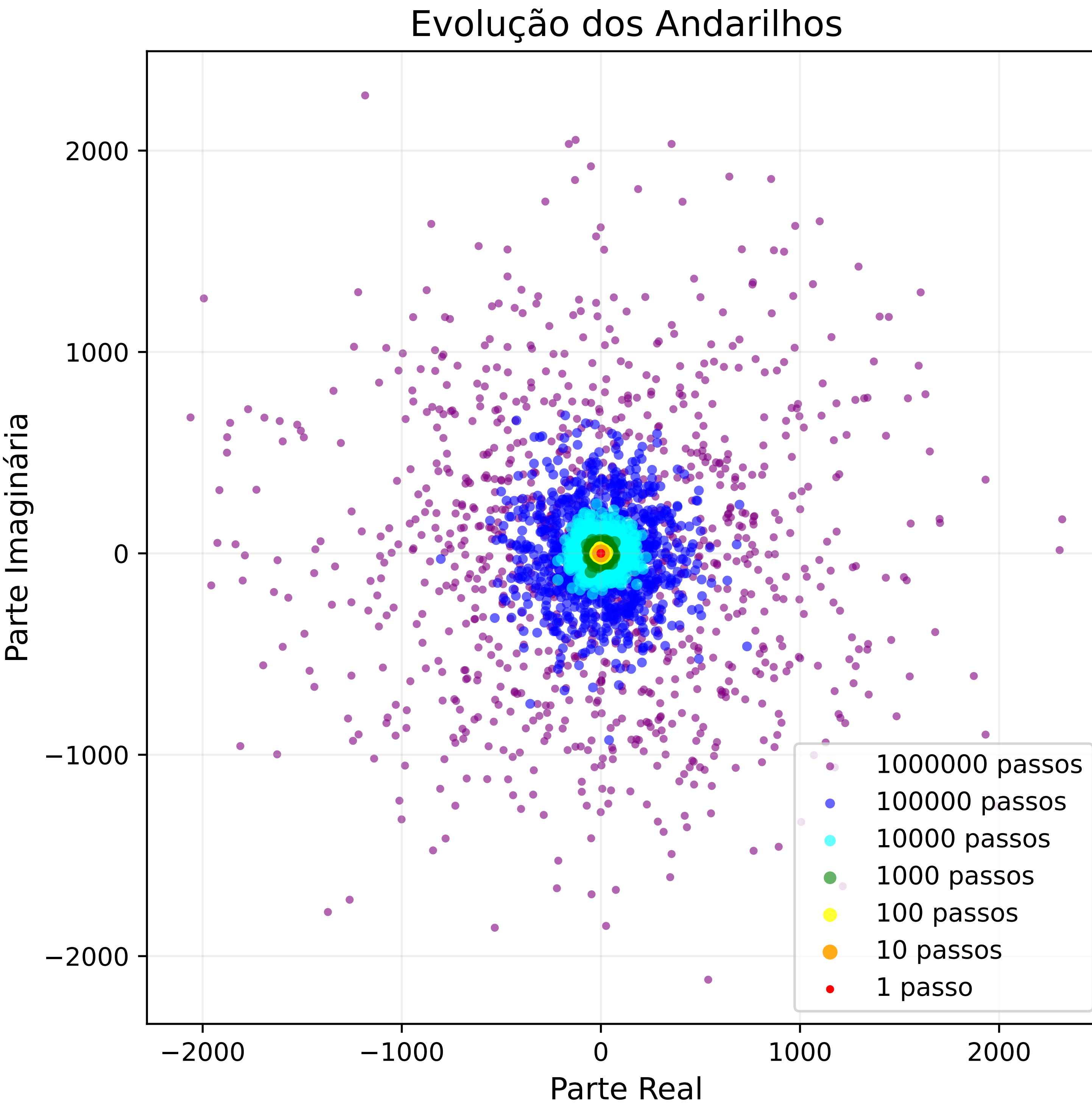
Resultado de rand()	Soma
r <0.25	+1
0.25 <r <0.5	+i
0.5 <r <0.75	-1
0.75 <r	-i

A implementação foi da seguinte forma:

```
REAL FUNCTION N_MEDIA(N, rmedia, iord, NUM_RAND)
        COMPLEX NUM_RAND(1600,8)
        soma = 0
        DO i = 1, N
          soma = soma + ((NUM_RAND(i,8) - rmedia))**iord
        END DO
        N_MEDIA = soma/N
END FUNCTION

COMPLEX FUNCTION WALK(inicio)
  COMPLEX inicio
  r = rand()
  IF (r .LT. 0.25) THEN
          WALK = inicio + (1,0)
  ELSE IF (r .LT. 0.5) THEN
          WALK = inicio + (0,1)
  ELSE IF (r. LT. 0.75) THEN
          WALK = inicio + (-1,0)
  ELSE
          WALK = inicio + (0, -1)
  END IF
END FUNCTION

PROGRAM ANDARILHOS_2D
INTEGER arquivo
REAL N_MEDIA
COMPLEX IP(1600, 8)
COMPLEX passo, media, WALK
PARAMETER (N = 1E6)

arquivo = 1

OPEN(UNIT=arquivo, FILE='dados.txt', STATUS='UNKNOWN')

WRITE(*,*) 'Quantos andarilhos?'
READ(*,*) M

DO i = 1, M
```

```fortran
    IP(i,1) = i
  END DO

  DO i = 1, M
    IP(i,2) = WALK((0,0))
      DO j = 0, 5
      passo = IP(i,2+j)
        DO k = 1, (9*10**j)
          passo = WALK(passo)
        END DO
      IP(i,3+j) = passo
      END DO
  END DO

  !Média
  media = N_MEDIA(M, 0.0, 1, IP)
  WRITE(*,*) 'A média é: ', media
  !Fim Média

  !Debugging
  DO i = 1, M
    WRITE(arquivo,*)IP(i,1), IP(i,2), IP(i,3), IP(i,4), IP(i,5),  IP
 &(i,6),IP(i,7), IP(i,8)
  END DO
  !Fim Debugging

  CLOSE(UNIT=arquivo)
  END PROGRAM
```

Posteriormente, montei um diagrama dos passos em cada geração:


Evolução dos Andarilhos
Parte Imaginária
Parte Real
2000
1000
0
−1000
−2000
−2000
−1000
0
1000
2000
1000000 passos
100000 passos
10000 passos
1000 passos
100 passos
10 passos
1 passo

Quarta Geração

Vamos verificar o aumento da entropia e a flecha do tempo no exercício anterior. Calcule a entropia como função do número de passos N das moléculas (que é proporcional ao tempo $t = N\Delta t$, onde Δt é o intervalo de tempo médio entre passos). A entropia é dada por:

$$S = -\sum_i P_i \ln P_i,$$

onde P_i é a probabilidade de se encontrar o sistema em um certo micro-estado i. Para se definir o micro-estado i definimos um reticulado (muito maior que o tamanho de um passo) e vemos quantas moléculas encontramos em cada célula do reticulado.

A fim de calcular o aumento de entropia, usei-me de células no formato de círculos concêntricos de raio igual a $20n, n \in \mathbb{N}, 1 < n < 100$ porque os últimos pontos estavam perto do número 2000 como pode ser observado no diagrama.

Assim, para descobrir quais pontos estavam em quais células somente seria necessário calcular o módulo do número complexo, dividir ele por 20, pegar a parte inteira da divisão e somar 1. Facilitando a computação da entropia.

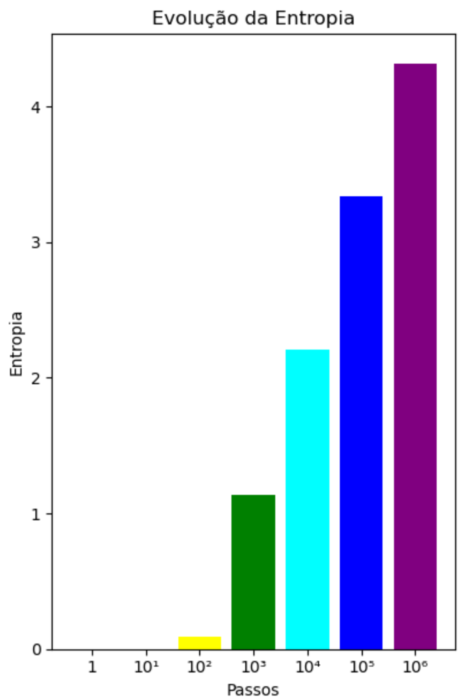
A implementação foi:

```
REAL FUNCTION N_MEDIA(N, rmedia, iord, NUM_RAND)
        COMPLEX NUM_RAND(1600,8)
        soma = 0
        DO i = 1, N
          soma = soma + ((NUM_RAND(i,8) - rmedia))**iord
        END DO
        N_MEDIA = soma/N
END FUNCTION

COMPLEX FUNCTION WALK(inicio)
  COMPLEX inicio
  r = rand()
  IF (r .LT. 0.25) THEN
          WALK = inicio + (1,0)
  ELSE IF (r .LT. 0.5) THEN
          WALK = inicio + (0,1)
  ELSE IF (r. LT. 0.75) THEN
          WALK = inicio + (-1,0)
  ELSE
          WALK = inicio + (0, -1)
  END IF
END FUNCTION
```

```fortran
PROGRAM ANDARILHOS_2D
INTEGER arquivo
REAL N_MEDIA
REAL ABSOLUTES(1600, 7), CIRCULOS(1600,7), ENTROPIA(1,7)
COMPLEX IP(1600, 8)
COMPLEX passo, media, WALK

WRITE(*,*) 'Quantos andarilhos?'
READ(*,*) M

DO i = 1, M
  IP(i,1) = i
END DO

DO i = 1, M
  IP(i,2) = WALK((0,0))
    DO j = 0, 5
    passo = IP(i,2+j)
      DO k = 1, (9*10**j)
        passo = WALK(passo)
      END DO
    IP(i,3+j) = passo
    END DO
END DO

!Tomar uma lista de valores absolutos/passo
DO i = 1, M
  DO j = 1, 7
    ABSOLUTES(i, j) = CABS(IP(i, j+1))
  END DO
END DO

!Tomar uma lista de qual círculo contem cada andarilho
DO i = 1, M
  DO j = 1,7
    CIRCULOS(i,j) = INT(ABSOLUTES(i,j)/20)+1
  END DO
END DO

!Cálculo de entropia
DO i = 1,7
vez = 0
  DO j = 1, 100
  counter = 0
    DO k = 1, M
      IF (j .EQ. CIRCULOS(k,i)) THEN
```

```
                counter = counter + 1
          ELSE
          END IF
        END DO
      P = counter/M
      IF (P .GT. 0) THEN
        entrop = P * LOG(P)
      ELSE
        entrop = 0
      END IF
      vez = vez + entrop
      END DO
    ENTROPIA(1,i) = -vez
    END DO

    !Debugging
    DO i = 1,M
    WRITE(*,*) 'Andarilho número', i
      DO j = 1,7
        WRITE(*,*)ABSOLUTES(i,j)
        WRITE(*,*)CIRCULOS(i,j)
      END DO
    END DO

    DO i = 1, 7
      WRITE(*,*)ENTROPIA(1,i)
    END DO
    END PROGRAM
```

Depois, montei um gráfico dos resultados usando o mesmo esquema de cores do diagram a fim de facilitar a compreensão.


Evolução da Entropia
Entropia
Passos
0
1
2
3
4
1
10¹
10²
10³
10⁴
10⁵
10⁶