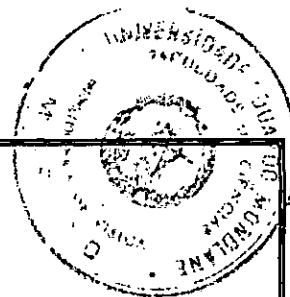


F13-05



**UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE FÍSICA**

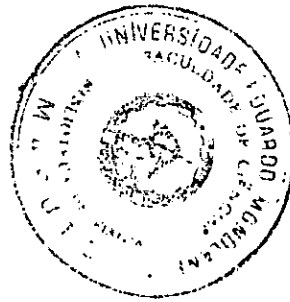
TRABALHO DE LICENCIATURA

**Caminhante Aleatório (Random Walk)
e Processos de Difusão em um Campo Externo.**

Autor: Nhalusse, Dionisio Ricardo

Supervisor: Prof. Dr. Yuri Rakov

Maputo, Agosto de 2005



DECLARAÇÃO DE HONRA

Declaro por minha honra que o presente trabalho é reflexo da minha própria investigação com o recurso ao material que ao longo do mesmo é referenciado.

Maputo, Agosto de 2005

O autor

Dionísio Ricardo Nhalusse

(Dionísio Ricardo Nhalusse)

DEDICATÓRIA

Aos meus Pais

e

meus irmãos.

AGRADECIMENTOS

- Agradeço desde já a todos aqueles que em mim depositaram esperança e confiança para que o presente trabalho se tornasse uma realidade.
- Ao meu supervisor Prof.Dr.Yuri Rakov pela disponibilidade, interesse e paciência por ele demonstrado, durante a realização deste trabalho.
- Aos Prof.Dr.Tchernych e Prof.Dr.Burdeynyi pelo convite que me fizeram para seguir o ramo da Física Teórica.
- A todos aqueles que foram meus docentes durante a minha formação, a toda comunidade do departamento e aos meus colegas nomeadamente: *Tivane, Elisio; Mungoi, Nordino; Chivambo, Dionísio; Zandamela, José; Perpétua Adélia* e tantos outros, pelo precioso apoio material e moral por eles prestado durante a vida estudantil e na realização deste trabalho.
- Aos meus pais, irmãos, tios, amigos e a todos aqueles que directa ou indirectamente contribuíram durante a minha formação.

Por isso mesmo:

Os meus agradecimentos.

RESUMO

No presente trabalho estão consideradas as seguintes questões: construção da equação de difusão livre e de difusão em um campo externo, solução analítica da equação de difusão unidimensional livre e em um campo externo, elaboração dos programas computacionais GraphD.exe e GraphG.exe para representar graficamente os processos de difusão, representação do modelo matemático do caminhante aleatório (Random Walk), elaboração do programa computacional GraphRW.exe para representar graficamente os processos do caminhante aleatório, solução da equação básica do caminhante aleatório com ajuda do método operacional, estabelecimento da relação do caminhante aleatório com os processos de difusão e simulação computacional dos processos de difusão utilizando o modelo do caminhante aleatório (Programa RWDif.exe).

LISTA DE SÍMBOLOS

σ^2 - *dispersão* de distribuição

$b = v_x$ - velocidade de deriva ao longo do eixo OX

α - comprimento do salto `a direita

β - comprimento do salto `a esquerda

p - a probabilidade do salto à direita

q - a probabilidade do salto à esquerda

τ - o intervalo do tempo entre os saltos sucessivos

n - número de saltos ou comprimento da trajectória

T_α - operador de deslocamento pela coordenada

$S(t)$ - operador de evolução

C_n^m são coeficientes binomiais

ξ - variável de integração

GraphD - programa gráfico para a difusão

GraphG – programa gráfico da função de Gauss

GraphRW – programa gráfico do Random Walk

RWDif – programa gráfico do Random Walk e difusão

LISTA DE FIGURAS

Figura	Pág's
Figura 1: Representa o volume fechado pela superfície orientada no espaço tridimensional em um meio difusivo	2
Figura 2: Representa o versor \vec{n} da superfície dS	3
Figura 3: Indicamos que $dJ > 0$ quando as partículas estão a sair do volume V (fig. 3a) e $dJ < 0$ quando as partículas entram no volume V (fig 3b).....	3
Figura 4: Representa a difusão livre do oxigénio (O_2) no ar, para $t = 1000s$	6
Figura 5: Representa a difusão livre do oxigénio (O_2) no ar, para $t = 3000s$	7
Figura 6: Representação gráfica de difusão de oxigénio no ar, usando a fórmula (54), na ausência da velocidade de deriva e para tempo variável.....	10
Figura 7: Representação gráfica de difusão de oxigénio no ar, usando a mesma fórmula (54), mas neste caso na presença da velocidade de deriva b e para tempo variável.....	11
Figura 8: Representa o modelo matemático do movimento de uma partícula browniana ao longo do eixo OX	12
Figura 9: Representa o movimento de uma partícula browniana ao longo do eixo OX	13
Figura 10: Representa a trajectória aleatória de uma partícula browniana de comprimento $n = 20$, a partir do estado $(x,n) = (0,0)$ até o estado $(x,n) = (4,20)$	13
Figura 11: Representa a distribuição de $N_p = 6002$ partículas brownianas para $n = 200$ passos.....	14
Figura 12: Representa a situação semelhante ao caso da difusão em um campo externo (Fig.7,Cap.II)...	15
Figura 13: Representa o esquema de N partículas brownianas que se encontram em movimento ao longo do eixo OX	15
Figura 14: Representa a distribuição <i>inicial de Gauss</i> (27), $\sigma = 20$	21
Figura 15: Representa o processo de difusão sem campo potencial externo ($\alpha = \beta = 6$) e sem gradiente de temperatura ($p = q = 0.5$), $n = 30$, $D = 18$ e $b = 0$	22
Figura 16: Representa a difusão num campo externo ($\alpha = 7$, $\beta = 5$) e sem gradiente de temperatura ($p = q = 0.5$), $n = 70$, $D = 18$ e $b = 1$	23
Figura 17: Representa a difusão sem campo externo potencial ($\alpha = \beta = 6$), com o gradiente de temperatura ($p = 0.6$, $q = 0.4$), $n = 100$, $D = 17.28$ e $b = 1.2$	24
Figura 18: Representa a difusão num campo externo potencial ($\alpha = 7$, $\beta = 5$), com o gradiente de temperatura ($p = 0.4$, $q = 0.6$), $n = 150$, $D = 17.28$ e $b = -0.2$	25
Figura 19: Representa um balanço entre o campo externo potencial ($\alpha = 7$, $\beta = 3$) e o gradiente de temperatura ($p = 0.3$, $q = 0.7$), $n = 50$, $D = 10.5$ e $b = 0$	26

ÍNDICE

DECLARAÇÃO DE HONRA.....	I
DEDICATÓRIA.....	II
AGRADECIMENTOS.....	III
RESUMO.....	IV
LISTA DE SÍMBOLOS	V
LISTA DE FIGURAS	VI
ÍNDICE.....	VII
Capítulo I - INTRODUÇÃO E OBJECTIVOS.....	1
1.1 Introdução.....	1
1.2 Objectivo.....	1
Capítulo II - DIFUSÃO UNIDIMENSIONAL.....	2
2.1 Equação de difusão.....	2
2.2 Difusão unidimensional em um campo externo.....	7
Capítulo III - CAMINHANTE ALEATÓRIO (RANDOM WALK)	12
3.1 Modelo matemático do movimento browniano.....	12
3.2 Método operacional da resolução da equação básica do movimento browniano.....	16
Capítulo IV - SIMULAÇÃO COMPUTACIONAL DE PROCESSOS DE DIFUSÃO.....	20
4.1 Simulação de difusão livre.....	20
4.2 Simulação de difusão em um campo externo.....	23
Capítulo V - CONCLUSÕES E RECOMENDAÇÕES.....	27
5.1 Conclusões.....	27
5.2 Recomendações.....	27
REFERENCIAIS BIBLIOGRÁFICAS.....	28
APÊNDICE.....	29

CAPÍTULO I - INTRODUÇÃO E OBJECTIVOS.

1.1 Introdução.

A existência dos computadores modernos permite efectivamente realizar a modelação computacional de vários processos físicos. Uma possibilidade importante é a modelação computacional dos processos de difusão e transporte em meios contínuos com as características diferentes, tais como os meios não homogêneos, os meios com existência do gradiente de temperatura, os meios encontrados em campos externos. Por outro lado podem ser consideradas as partículas de difusão entre as quais existem interacções. Algumas destas possibilidades são exemplos dos processos não lineares, cuja teoria está em desenvolvimento contínuo na Física Moderna.

Neste trabalho está considerada a modelação computacional do comportamento das partículas brownianas em movimento no meio, que se encontra em um campo externo potencial.

1.2 Objectivo.

O presente trabalho tem como objectivo básico estabelecer a relação entre os processos de difusão e o movimento browniano no modelo teórico do Caminhante Aleatório.

CAPITULO II - DIFUSÃO UNIDIMENSIONAL.

2.1 Equação de difusão

Difusão é o fenômeno de transferência de partículas a partir de pontos de maior concentração para pontos de menor concentração, ou seja, processo de transferência de partículas provocado por uma diferença de concentração de partículas. A transferência difusiva é um processo que se origina do movimento molecular térmico [1].

Para obter a equação de difusão vamos considerar um volume fechado pela superfície orientada no espaço tridimensional em um meio difusivo.

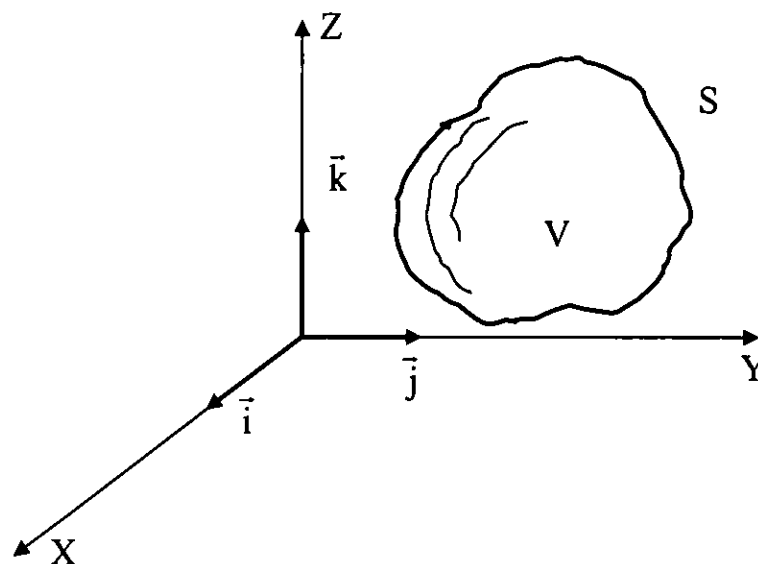


Figura 1: Representa o volume fechado pela superfície orientada no espaço tridimensional em um meio difusivo.

Sejam N o número das partículas no volume V e $n(x, y, z, t)$ a densidade volumétrica das partículas. Então:

$$N = \int_V n(x, y, z, t) dx dy dz \quad (1)$$

O número das partículas no volume V pode variar durante o tempo por causa do fluxo das partículas através da superfície S . Seja $\vec{j}(x, y, z, t)$ a densidade do fluxo das partículas.

Então:

$$dJ = \vec{j} d\vec{S} \quad (2)$$

é o fluxo das partículas através da superfície $d\vec{S}$. Aqui:

$$d\vec{S} = \vec{n} dS \quad (3)$$

onde \vec{n} é versor da superfície e dS é sua área (fig. 2).

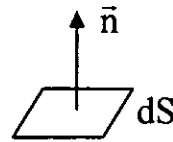


Figura 2

Reunindo fórmulas (2) e (3) temos:

$$dJ = (\vec{j}\vec{n})dS \quad (4)$$

Indicamos que $dJ > 0$ quando as partículas estão a sair do volume V (fig. 3a) e $dJ < 0$ quando as partículas entram no volume V (fig 3b).



Figura 3

O fluxo das partículas através de toda superfície S é

$$J = \oint_S \vec{j} d\vec{S} \quad (5)$$

Fórmulas (1) e (5) permitem- nos escrever a equação do balanço na forma seguinte:

$$\frac{dN}{dt} = -J \quad (6)$$

ou para o volume fixado:

$$\int_V \frac{\partial n}{\partial t} dV = -\oint_S \vec{j} d\vec{S} \quad (7)$$

Aplicando o teorema de Gauss-Ostrogradski para a integral superficial da Eq.7, temos:

$$\oint_S \vec{j} d\vec{S} = \int_V \text{div} \vec{j} dV \quad (8)$$

Substituindo (8) em (7), obteremos

$$\int_V \frac{\partial n}{\partial t} dV = -\int_V \text{div} \vec{j} dV. \quad (9)$$

O volume V é arbitrário, por isso teremos

$$\frac{\partial n}{\partial t} = -\text{div} \vec{j} \quad (10)$$

A seguir utilizando operador *nabla*:

$$\vec{\nabla} = \frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} \quad (11)$$

Em matemática demonstra-se que:

$$\text{div} \vec{A} = \vec{\nabla} \cdot \vec{A} \quad (12)$$

$$\text{rot} \vec{A} = [\vec{\nabla} \times \vec{A}] \quad (13)$$

$$\text{grad} U = \vec{\nabla} U \quad (14)$$

$$\Delta = \vec{\nabla} \cdot \vec{\nabla} \quad (15)$$

onde $\vec{A} = \vec{A}(x, y, z)$ é um campo vectorial, $U = U(x, y, z)$ é um campo escalar e Δ é o operador de Laplace:

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}. \quad (16)$$

Assim (10) tem a seguinte forma:

$$\frac{\partial n}{\partial t} = -(\vec{\nabla} \cdot \vec{j}) \quad (17)$$

Para transformar a Eq.17 à equação dada por $n(x, y, z)$, utilizamos a lei de Fick:

$$\vec{j} = -D(\vec{\nabla} n) \quad (18)$$

onde D é coeficiente de difusão.

Com o auxílio das fórmulas (15), (17) e (18) obteremos a equação de difusão:

$$\frac{\partial n}{\partial t} = D \cdot \Delta n \quad (19)$$

Esta equação foi deduzida para $D = \text{const}$ [1,5].

Em muitos processos reais de difusão é possível transformar o problema tridimensional ao problema unidimensional. Neste caso $n = n(x, t)$ e equação (19) tem a forma:

$$\frac{\partial n}{\partial t} = D \frac{\partial^2 n}{\partial x^2} \quad (20)$$

Consideremos a difusão ao longo do eixo OX infinito. Neste caso temos o seguinte problema de Cauchy:

$$\frac{\partial n}{\partial t} = D \frac{\partial^2 n}{\partial x^2} \quad (t > 0, \quad -\infty < x < \infty) \quad (21)$$

$$n(x, 0) = \varphi(x) \quad (22)$$

onde $\varphi(x)$ é uma distribuição inicial dada.

A solução deste problema é bem conhecida [3] e tem a forma seguinte:

$$n(x, t) = \frac{1}{2\sqrt{\pi Dt}} \int_{-\infty}^{\infty} \exp\left\{-\frac{(x-\xi)^2}{4Dt}\right\} \varphi(\xi) d\xi \quad (23)$$

É chamada *integral de Poisson*.

Agora consideremos uma distribuição inicial, que se encontra muito frequentemente na Física Teórica. Isto é a δ -distribuição:

$$\varphi(x) = N\delta(x) \quad (24)$$

onde $\delta(x)$ é a função de Dirac. Realmente é uma função *generalizada* e de ponto de vista de matemática a definição correcta desta função tem a forma de integral:

$$f(x) = \int_{-\infty}^{\infty} \delta(x-\xi) f(\xi) d\xi \quad (25)$$

onde $f(x)$ é uma função usual.

O conceito de δ -função é particularmente útil para representar matematicamente tais objectos físicos abstractos como um ponto material, uma carga eléctrica pontual etc..

A distribuição (24) significa que todas as partículas estão situadas na posição $x = 0$ no instante inicial $t = 0$. A partir de fórmulas (23) - (25) achamos

$$n(x, t) = \frac{N}{2\sqrt{\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right) \quad (26)$$

Mais adiante, no lugar da densidade das partículas $n(x, t)$, vamos utilizar

$$\rho(x, t) = \frac{n(x, t)}{N} \quad (27)$$

onde $\rho(x, t)$ é a densidade de probabilidade para a coordenada da partícula difusiva.

Neste caso, a probabilidade da partícula difusiva se encontrar localizada no interior do intervalo $[x, x + dx]$ é definida por $\rho(x, t)dx$. De fórmulas (1) e (27) segue a condição de normalização:

$$\int_{-\infty}^{\infty} \rho(x, t) dx = 1 \quad (28)$$

A Eq.28 significa que a densidade de probabilidade tem como valor máximo 1 (unidade).

Então para $\rho(x, t)$ temos o seguinte problema de Cauchy:

$$\frac{\partial \rho}{\partial t} = D \frac{\partial^2 \rho}{\partial x^2} \quad (t > 0, \quad -\infty < x < \infty) \quad (29)$$

$$\rho(x, 0) = \delta(x) \quad (30)$$

De fórmulas (26) e (27) temos a solução deste problema:

$$\rho(x, t) = \frac{1}{2\sqrt{\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right) \quad (31)$$

Como exemplo de aplicação desta fórmula vamos considerar a difusão de oxigênio (O_2) no ar:

$$D = 18 \cdot 10^{-6} \text{ m}^2/\text{s} \quad (32)$$

$T = 273\text{K}$, pressão normal [1,6].

Os gráficos foram construídos com ajuda do programa **GraphD** no meio do sistema de programação Delphi.4.0, (Ver o Apêndice).

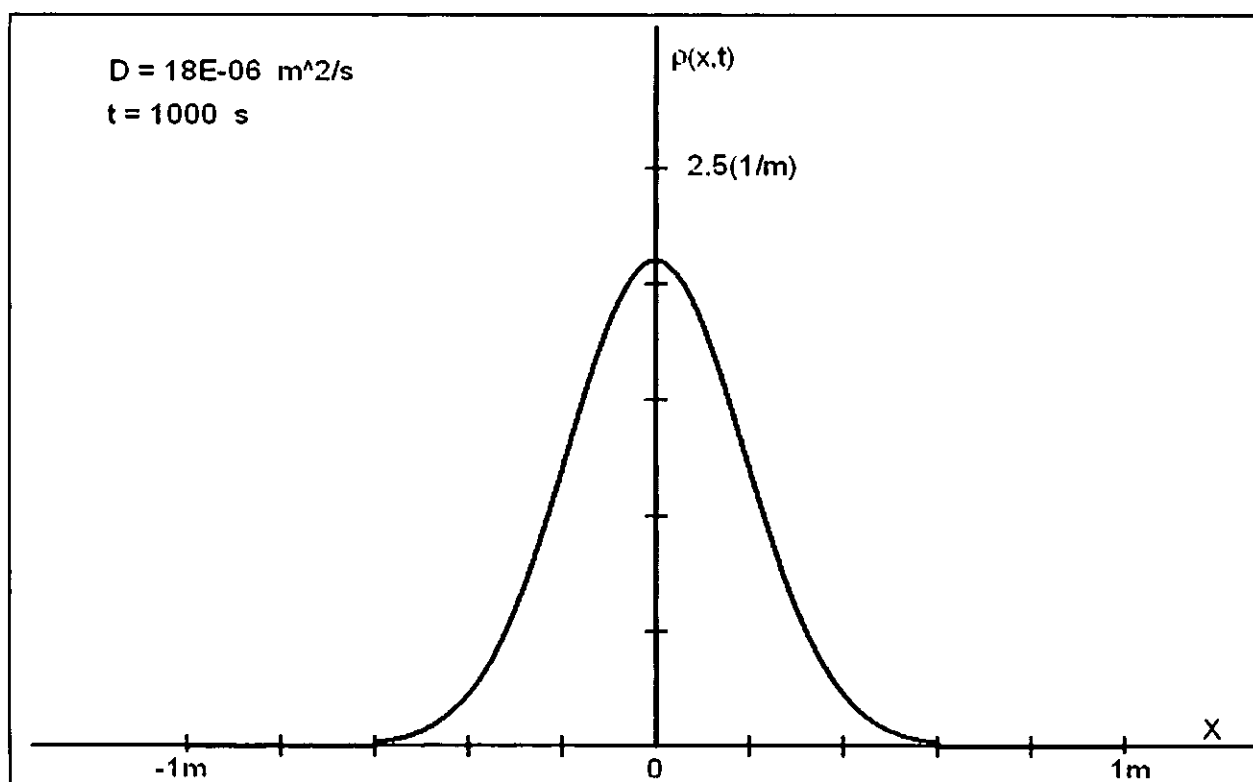


Figura 4: Representa a difusão livre do oxigênio (O_2) no ar, para $t = 1000\text{s}$.

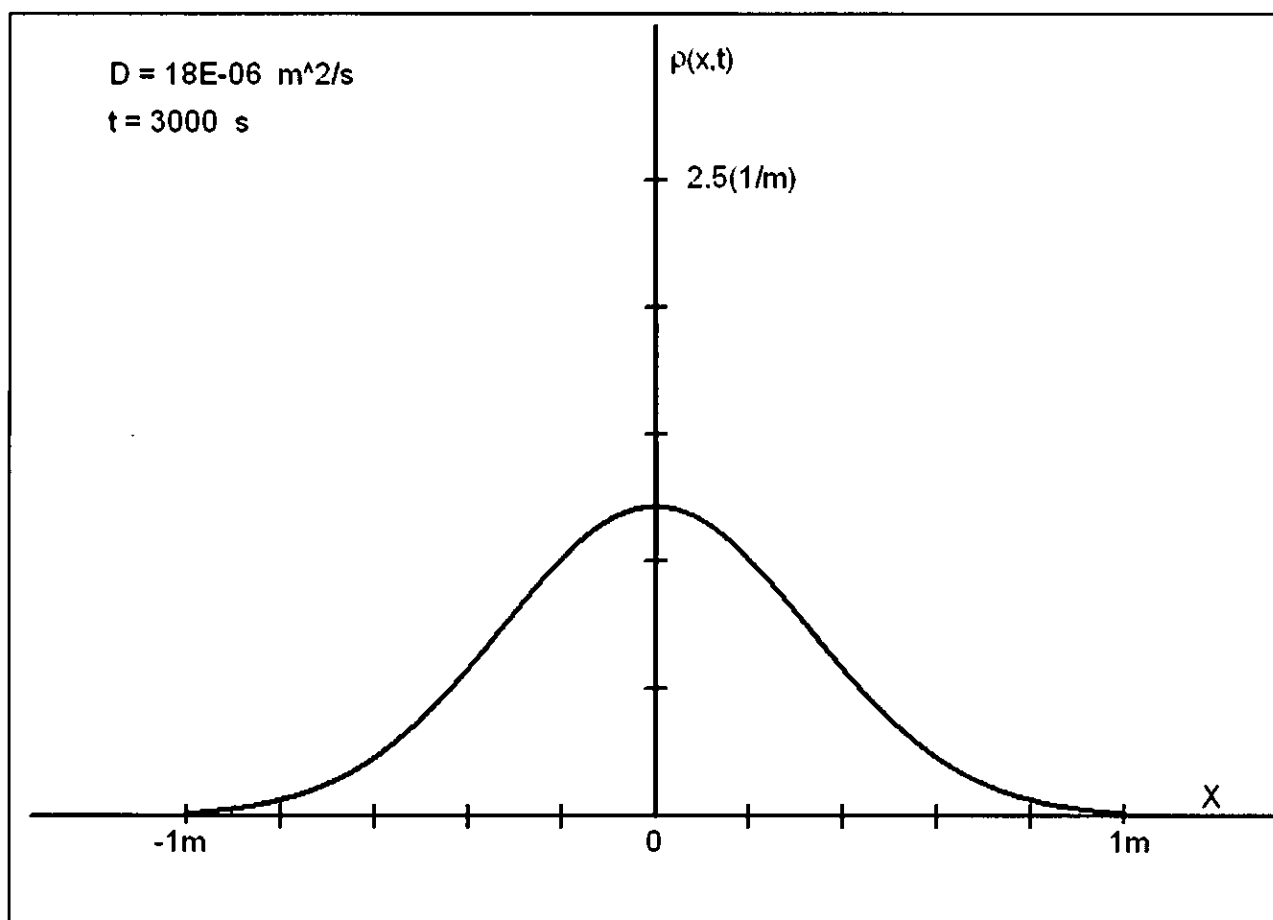


Figura 5: Representa a difusão livre do oxigênio (O_2) no ar, para $t = 3000s$.

2.2 Difusão unidimensional em um campo externo.

Seja um meio difusivo, colocado em um campo externo estacionário, por exemplo, no campo de gravidade. Então sobre cada partícula difusiva vai actuar uma força constante. Sob a acção desta força uma partícula difusiva vai ter uma velocidade constante \vec{v} que se chama *velocidade de deriva*. Assim no meio difusivo aparece o fluxo de partículas com a seguinte densidade:

$$\vec{j}_e = \vec{v}n \quad (33)$$

Em lugar da fórmula (18) vamos ter

$$\vec{j} = \vec{v}n - D(\vec{\nabla}n) \quad (34)$$

onde \vec{j} é a densidade do fluxo total. Da fórmula (17) vamos ter

$$\frac{\partial n}{\partial t} = -\vec{v}(\vec{\nabla}n) + D \cdot \Delta n \quad (35)$$

Esta é a *equação de difusão em um campo externo*. A forma unidimensional desta equação é:

$$\frac{\partial n}{\partial t} = -b \frac{\partial n}{\partial x} + D \frac{\partial^2 n}{\partial x^2} \quad (36)$$

onde $b = v_x$ é a velocidade de deriva ao longo do eixo OX [4].

Vamos considerar a difusão ao longo do eixo infinito e resolvemos o seguinte problema de Cauchy:

$$\frac{\partial n}{\partial t} = -b \frac{\partial n}{\partial x} + D \frac{\partial^2 n}{\partial x^2} \quad (t > 0, \quad -\infty < x < \infty) \quad (37)$$

$$n(x, 0) = \varphi(x) \quad (38)$$

Vamos resolver este problema utilizando a transformação de Fourier:

$$n(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \tilde{n}(\alpha, t) e^{-i\alpha x} d\alpha \quad (39)$$

$$\tilde{n}(\alpha, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} n(x, t) e^{i\alpha x} dx \quad (40)$$

Então

$$\frac{\partial n}{\partial t} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{\partial \tilde{n}(\alpha, t)}{\partial t} e^{-i\alpha x} d\alpha \quad (41)$$

$$\frac{\partial n}{\partial x} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (-i\alpha) \tilde{n}(\alpha, t) e^{-i\alpha x} d\alpha \quad (42)$$

$$\frac{\partial^2 n}{\partial x^2} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (-\alpha^2) \tilde{n}(\alpha, t) e^{-i\alpha x} d\alpha \quad (43)$$

Utilizando estas fórmulas e equação (37) vamos ter

$$\frac{\partial \tilde{n}}{\partial t} = (i\alpha b - \alpha^2 D) \tilde{n}(\alpha, t) \quad (44)$$

Depois de integração trivial temos

$$\tilde{n}(\alpha, t) = A(\alpha) \exp(i\alpha b t - \alpha^2 D t) \quad (45)$$

Substituindo esta expressão na fórmula (39) obteremos

$$n(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} A(\alpha) \exp(-\alpha^2 D t) e^{-i\alpha(x-bt)} d\alpha \quad (46)$$

A função $A(\alpha)$ pode-se procurar, usando condição inicial (38)

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} A(\alpha) e^{-i\alpha x} d\alpha \quad (47)$$

de onde, utilizando a transformação inversa de Fourier (40), temos

$$A(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \varphi(\xi) e^{i\alpha\xi} d\xi \quad (48)$$

Com o auxílio das fórmulas (46) e (48) obteremos a solução do problema considerado:

$$n(x, t) = \frac{1}{2\sqrt{\pi Dt}} \int_{-\infty}^{\infty} \exp\left\{-\frac{(x - bt - \xi)^2}{4Dt}\right\} \varphi(\xi) d\xi \quad (49)$$

Seja $\varphi(x)$ a distribuição de Gauss

$$\varphi(x) = \frac{N}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (50)$$

Então

$$n(x, t) = \frac{N}{\sqrt{2\pi(\sigma^2 + 2Dt)}} \exp\left\{-\frac{(x - bt)^2}{2\sigma^2 + 4Dt}\right\} \quad (51)$$

Para a densidade de probabilidade $\rho(x, t)$ que é definida pela fórmula (27), temos o seguinte problema de Cauchy:

$$\frac{\partial \rho}{\partial t} = -b \frac{\partial \rho}{\partial x} + D \frac{\partial^2 \rho}{\partial x^2} \quad (t > 0, \quad -\infty < x < \infty) \quad (52)$$

$$\rho(x, 0) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (53)$$

Aqui σ^2 é a *dispersão* de distribuição. De fórmulas (51) e (27) obteremos a solução deste problema

$$\rho(x, t) = \frac{1}{\sqrt{2\pi(\sigma^2 + 2Dt)}} \exp\left\{-\frac{(x - bt)^2}{2\sigma^2 + 4Dt}\right\} \quad (54)$$

Como um caso particular, temos a fórmula (31) se $b = 0$ e $\sigma = 0$.

Para a difusão do oxigênio no ar (32) obteremos os resultados gráficos representados pelas fig.6 e fig. 7, com ajuda do programa **GraphG**. Resultados deste capítulo vão ser usados mais adiante no Cap.III, para calibragem do modelo do Caminhante Aleatório.

A difusão sem deriva:

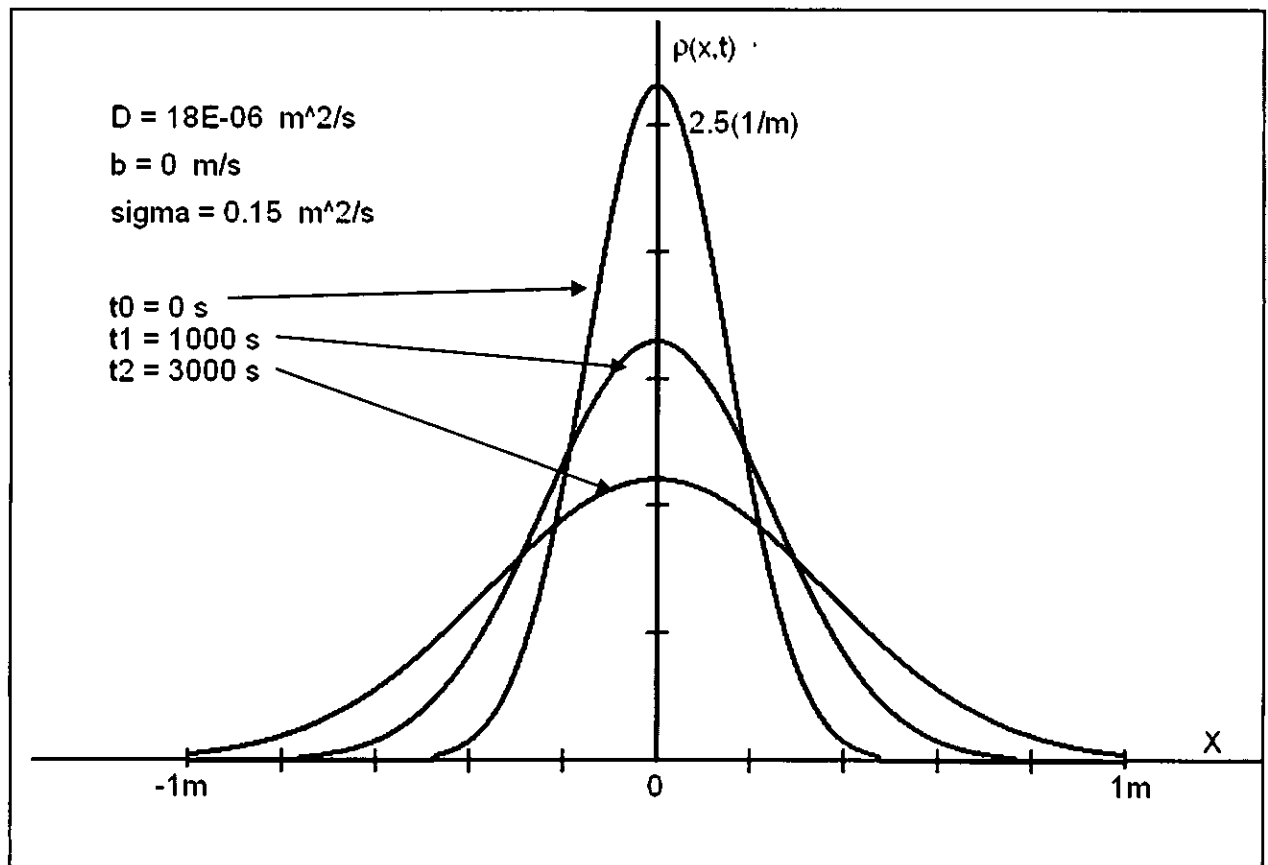


Figura 6: Representação gráfica de difusão de oxigênio no ar, usando a fórmula (54), na ausência da velocidade de deriva e para tempo variável.

A difusão em um campo externo:

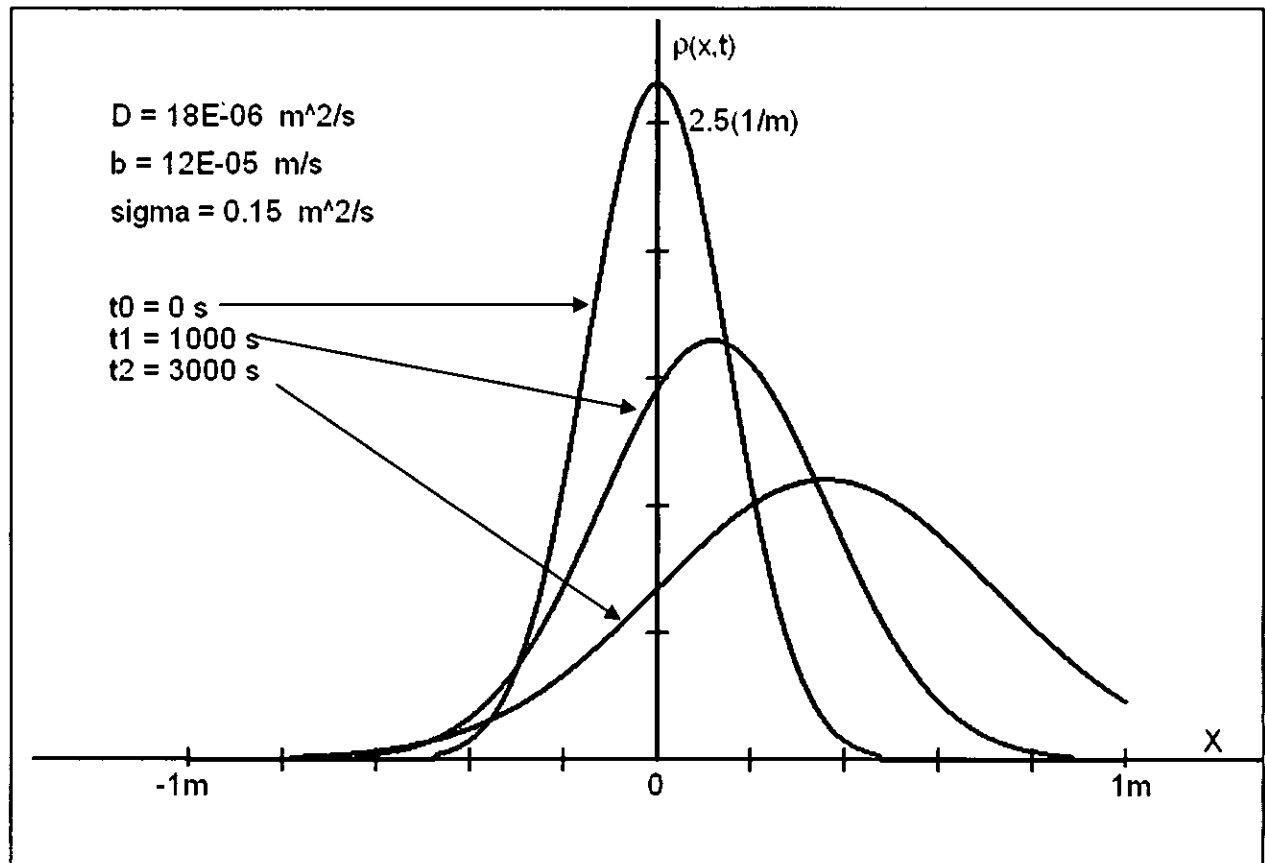


Figura 7: Representação gráfica de difusão de oxigênio no ar, usando a mesma fórmula (54), mas neste caso na presença da velocidade de deriva b e para tempo variável.

CAPITULO III - CAMINHANTE ALEATÓRIO (RANDOM WALK).

3.1 Modelo matemático do movimento browniano.

Diz-se *browniano* o movimento de carácter caótico e contínuo de partículas microscópicas em suspensão num gás ou num líquido. O movimento browniano deve-se às flutuações da pressão que as moléculas do gás ou do líquido exercem sobre as partículas em suspensão.

Em consequência das flutuações da pressão as partículas brownianas sofrem, de todas as partes, uma acção das forças desequilibradas que originam o movimento complexo e visível dessas partículas.

Como modelo matemático do movimento browniano utiliza-se o modelo de *Caminhante Aleatório* ou o modelo de *Random Walk* [8,11]. Consideremos o movimento de uma partícula browniana ao longo do eixo OX. Seja no instante do tempo t , a partícula situada no ponto x , então o estado dela se pode representar pelo par (x, t) . No instante do tempo $t + \tau$ a partícula pode-se encontrar na posição $x + \alpha$, com a probabilidade p ou na posição $x - \beta$, com a probabilidade q . É claro que $p + q = 1$. Este processo está representado na figura 1.

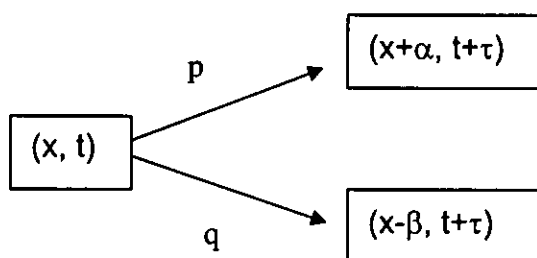


Figura 8: Representa o modelo matemático do movimento de uma partícula browniana ao longo do eixo OX.

Então, os parâmetros básicos do modelo considerado são: α, β, p, q e τ .

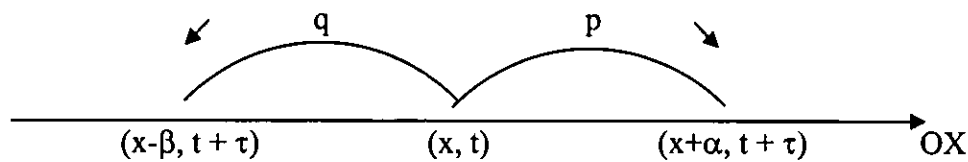


Figura 9: Representa o movimento de uma partícula browniana ao longo do eixo OX.

Durante o tempo a partícula browniana vai passar uma trajetória aleatória, por exemplo, a que se segue:

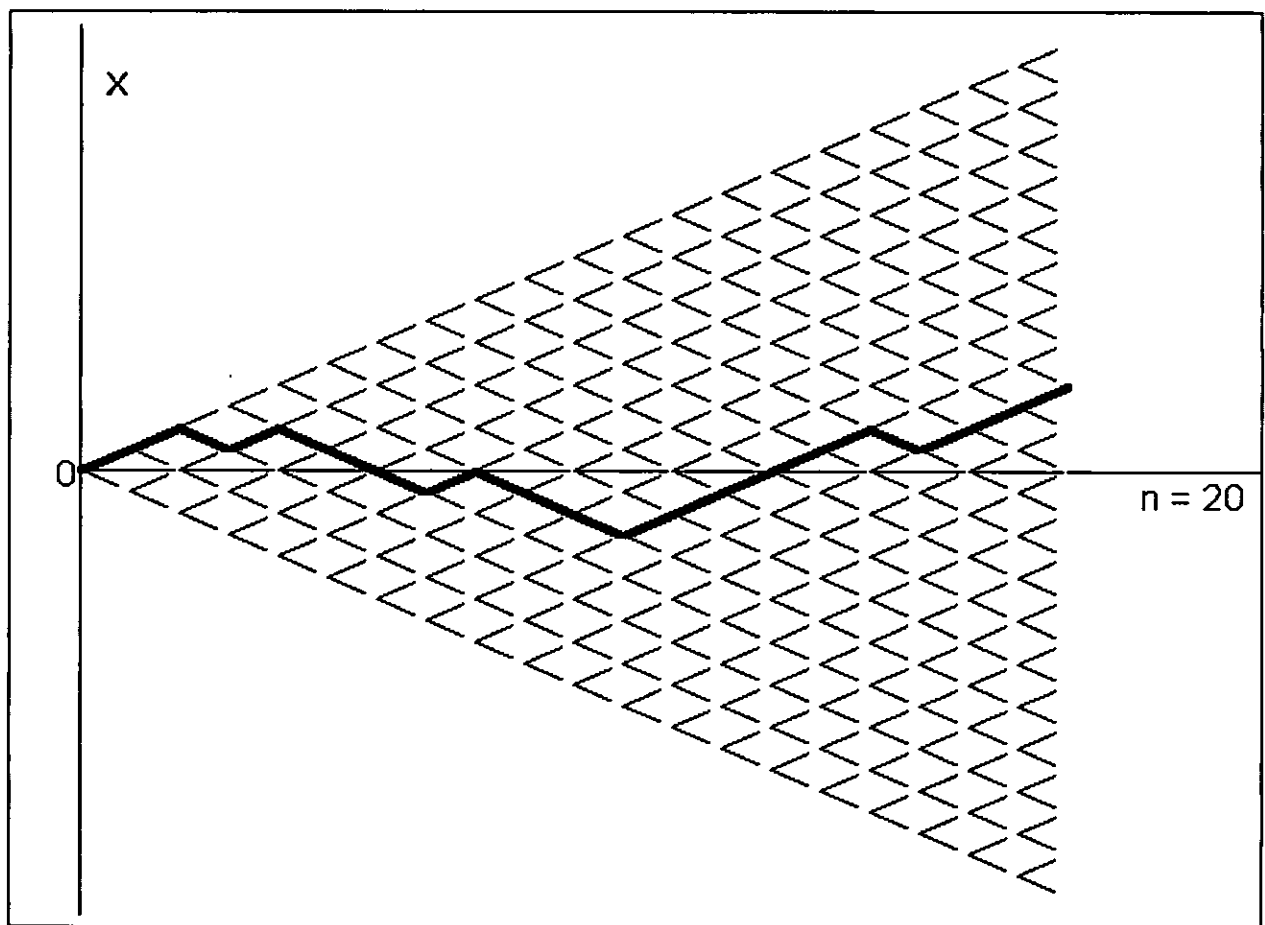


Figura 10: Representa a trajetória aleatória de uma partícula browniana de comprimento $n = 20$, a partir do estado $(x,n) = (0,0)$ até o estado $(x,n) = (4,20)$. Aqui n é o número de saltos e o tempo total do movimento é $t = n\tau$.

A relação entre difusão, movimento macroscópico irreversível, e o movimento browniano foi estabelecido por A. Einstein [7], numa teoria que inaugurou o moderno tratamento dos fenômenos irreversíveis através do estudo das flutuações no equilíbrio.

A simulação computacional do processo de Random Walk pode demonstrar a relação entre este modelo e difusão. Para isso foi elaborado o programa computacional **GraphRW**.

Neste programa, a simulação do processo de Random Walk foi realizada em linguagem de programação *Pascal* com ajuda da operação **Random** e esquematicamente tem a seguinte forma:

```
p:= 0.5;
x:= 0;
.....
if (Random < p) then x:= x+1 else x:= x-1;
.....
```

O resultado da execução deste programa é representado na figura que se segue:

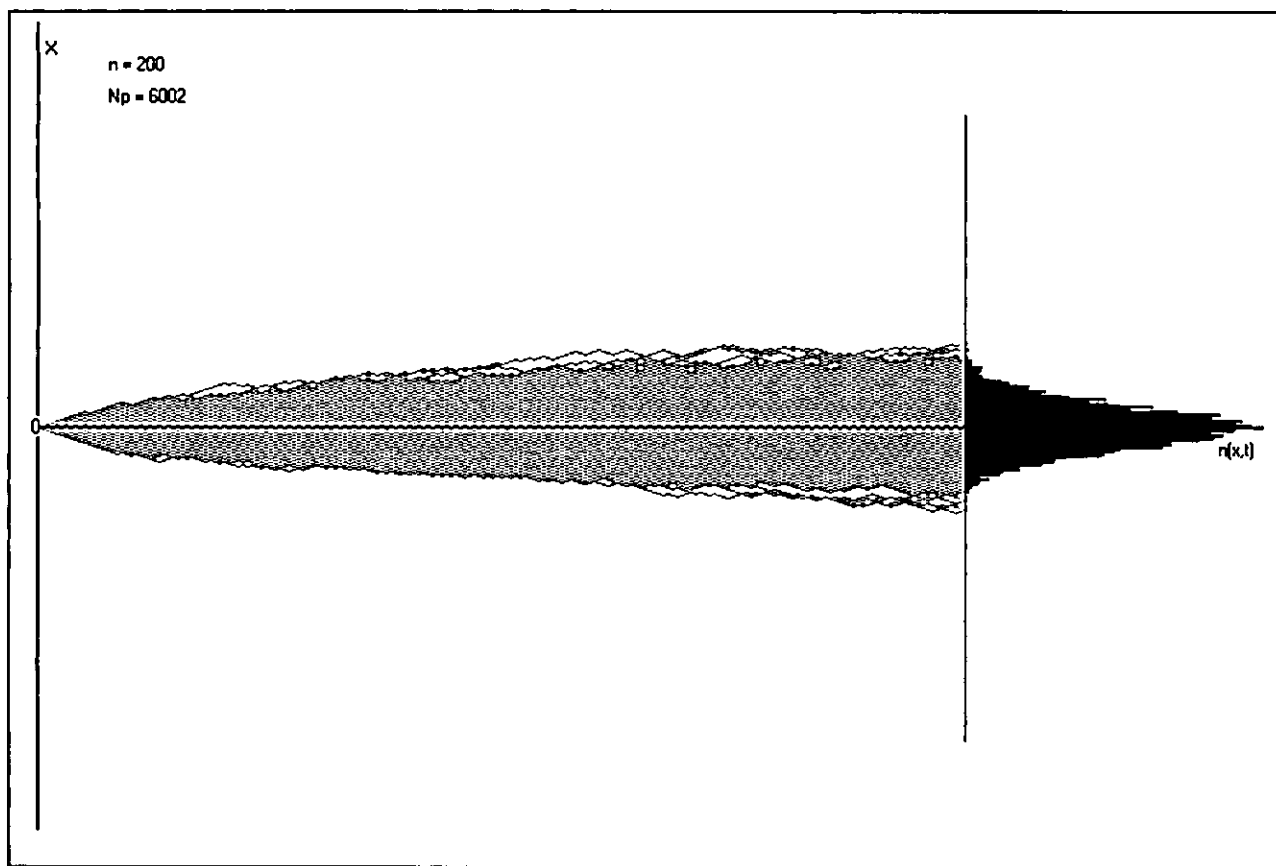


Figura 11: Representa a distribuição de $N_p = 6002$ partículas brownianas para $n = 200$ passos.

Este resultado é semelhante ao caso da difusão sem deriva (Fig.6, Cap.II).

É o processo de movimento aleatório de N_p partículas brownianas. Cada partícula iniciou o movimento a partir do estado inicial $(x,t) = (0,0)$ e depois de n passos atinge uma posição sobre o eixo OX . Finalmente vamos ter uma distribuição $n(x,t)$ destas partículas brownianas.

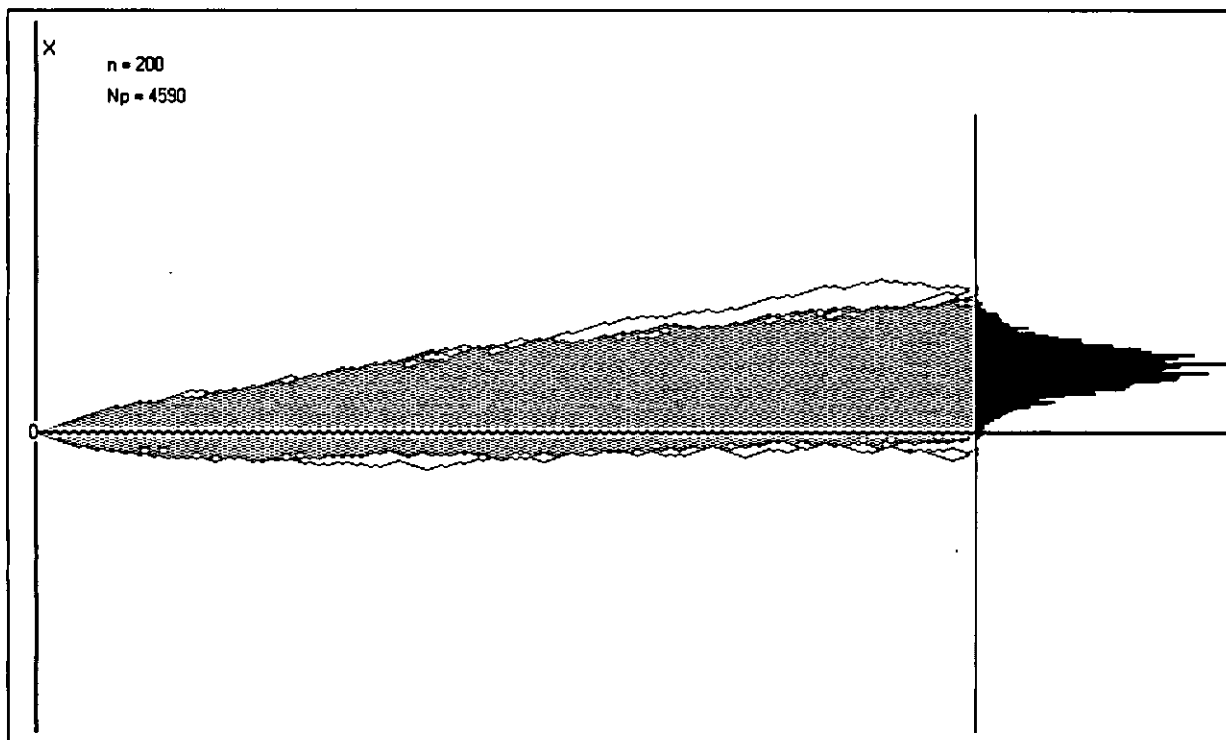


Figura 12: Representa a situação semelhante ao caso da difusão em um campo externo (Fig.7, Cap.II).

A fig.9 permite-nos determinar a ligação dos parâmetros básicos com as características físicas dos processos de difusão.

Para tal consideremos um sistema de N partículas brownianas que se encontram em movimento ao longo do eixo OX. Seja $n(x,t)$ a densidade do número de partículas, então:

$$N = \int_{-\infty}^{\infty} n(x,t) dx \quad (1)$$

é a condição de normalização e $n(x,t)dx$ é o número de partículas que se encontram no intervalo $(x, x+dx)$ no instante do tempo t. Então temos seguinte esquema do movimento:

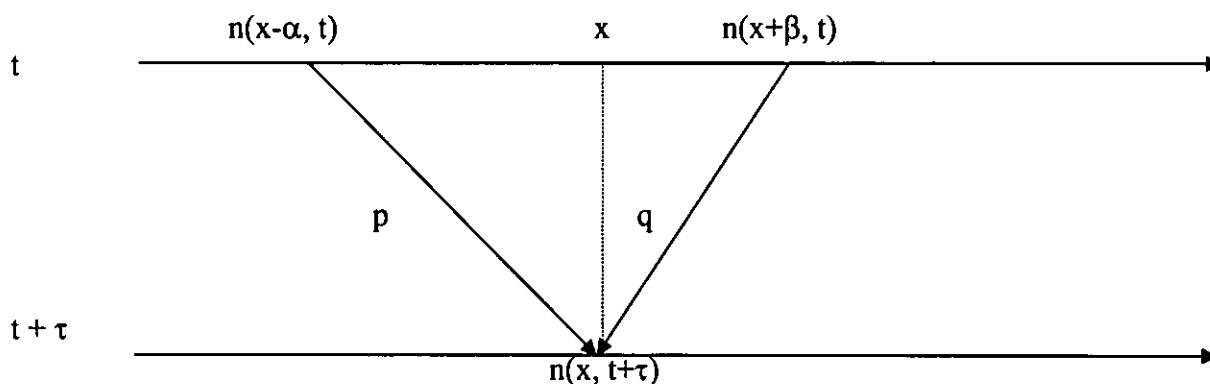


Figura 13: Representa o esquema de N partículas brownianas que se encontram em movimento ao longo do eixo OX.

Daqui segue

$$n(x, t + \tau) = pn(x - \alpha, t) + qn(x + \beta, t) \quad (2)$$

Introduzindo a função

$$\rho(x, t) = \frac{n(x, t)}{N} \quad (3)$$

que é a densidade de probabilidade com a condição de normalização

$$\int_{-\infty}^{\infty} \rho(x, t) dx = 1 \quad (4)$$

temos a equação básica do movimento de um sistema das partículas brownianas:

$$\rho(x, t + \tau) = p\rho(x - \alpha, t) + q\rho(x + \beta, t) \quad (5)$$

Esta equação é bem conhecida e tem várias modificações na física e matemática.

Vamos considerar o método operacional da solução de seguinte problema de Cauchy:

$$\begin{aligned} \rho(x, t + \tau) &= p\rho(x - \alpha, t) + q\rho(x + \beta, t) & (-\infty < x < \infty, t > 0) \\ \rho(x, 0) &= \varphi(x) \end{aligned} \quad (6)$$

onde $\varphi(x)$ é a distribuição inicial.

3.2 Método operacional da resolução da equação básica do movimento browniano.

O cálculo operacional (ou simbólico), é um dos domínios importantes na análise matemática. Em mecânica quântica, em electrónica e noutros ramos da ciência utiliza-se os métodos do cálculo operacional para a resolução de diferentes problemas.

Método operacional da resolução do problema de Cauchy proposto na Eq.6:

Seja T_α o operador de deslocamento pela coordenada:

$$T_\alpha \rho(x, t) = \rho(x + \alpha, t) \quad (7)$$

com as seguintes propriedades:

$$\begin{aligned} T_\alpha T_\beta &= T_\beta T_\alpha = T_{\alpha+\beta} \\ T_{n\alpha} &= T_\alpha^n \quad (n = 1, 2, 3, \dots) \\ T_{-\alpha} &= T_\alpha^{-1} \end{aligned} \quad (8)$$

Então a equação (5) podemos representar na forma:

$$\rho(x, t + \tau) = [p T_{-\alpha} + q T_{\beta}] \rho(x, t) \quad (9)$$

Introduzimos o operador de evolução $S(t)$:

$$\rho(x, t) = S(t)\varphi(x) \quad (10)$$

$$S(0) = 1 \quad (11)$$

e escrevemos (9) na forma:

$$S(t + \tau)\varphi(x) = [p T_{-\alpha} + q T_{\beta}] S(t)\varphi(x) \quad (12)$$

A função $\varphi(x)$ é a função arbitrária, então temos a igualdade operacional:

$$S(t + \tau) = [p T_{-\alpha} + q T_{\beta}] S(t) \quad (13)$$

e para $t = 0$, usando (11) temos:

$$S(\tau) = [p T_{-\alpha} + q T_{\beta}] \quad (14)$$

A seguir:

$$S(\tau) = \underbrace{S(\tau)S(\tau)S(\tau)S(\tau)S(\tau)\dots\dots\dots S(\tau)S(\tau)}_n \quad (15)$$

Então temos:

$$S(n\tau) = [p T_{-\alpha} + q T_{\beta}]^n \quad (16)$$

Segundo a fórmula de binómio de Newton temos:

$$S(n\tau) = \sum_{m=0}^n C_n^m p^m T_{-\alpha}^m q^{n-m} T_{\beta}^{n-m} \quad (17)$$

onde C_n^m são coeficientes binomiais. Usando propriedades (8) podemos escrever:

$$S(n\tau) = \sum_{m=0}^n C_n^m p^m q^{n-m} T_{(n-m)\beta - m\alpha} \quad (18)$$

Este é o operador de evolução do problema (6). Segundo a fórmula (10) a solução do problema (6) tem a seguinte representação:

$$\rho(x, n\tau) = \sum_{m=0}^n C_n^m p^m q^{n-m} \varphi[x + (n - m)\beta - m\alpha] \quad (19)$$

Esta fórmula pode ser usada para os cálculos computacionais.

E agora podemos determinar a ligação dos parâmetros α , β , τ , p , q com as características físicas dos processos de difusão. Para isso vamos considerar o comportamento assintótico da distribuição binomial (ou distribuição de Bernoulli):

$$P_{m,n} = C_n^m p^m q^{n-m} \quad (20)$$

Para $n \rightarrow \infty$ temos [9]:

$$P(m,n) \approx \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

$$\sigma = \sqrt{npq} \quad (21)$$

$$x = \frac{m - np}{\sigma}$$

Então a representação (19) tem a forma assintótica seguinte:

$$\rho(x, n\tau) \approx \frac{1}{\sigma\sqrt{2\pi}} \sum_{m=0}^n \exp\left\{-\frac{(m - np)^2}{2\sigma^2}\right\} \varphi[x + (n - m)\beta - m\alpha] \quad (22)$$

Introduzindo $\xi = x + (n - m)\beta - m\alpha$, $t = n\tau$ e usando as condições: $n\alpha \gg x$ e $n\beta \gg x$, podemos passar à integração:

$$\rho(x, t) = \frac{1}{2\sqrt{\pi Dt}} \int_{-\infty}^{\infty} \exp\left\{-\frac{(x - bt - \xi)^2}{4Dt}\right\} \varphi(\xi) d\xi \quad (23)$$

Aqui:

$$D = \frac{pq(\alpha + \beta)^2}{2\tau} \quad (24)$$

$$b = \frac{p\alpha - q\beta}{\tau}$$

Comparando a fórmula (23) com a Eq.II.49, conclui-se que é solução da equação de difusão no problema de Cauchy:

$$\frac{\partial \rho}{\partial t} = D \frac{\partial^2 \rho}{\partial x^2} - b \frac{\partial \rho}{\partial x} \quad (-\infty < x < \infty, t > 0) \quad (25)$$

$$\rho(x, 0) = \varphi(x)$$

Para $\varphi(x) = \delta(x)$ temos a solução fundamental deste problema:

$$\rho(x, t) = \frac{e^{-\frac{(x-bt)^2}{4Dt}}}{2\sqrt{\pi Dt}} \quad (26)$$

e para a distribuição de Gauss:

$$\varphi(x) = \frac{\exp\left\{-\frac{x^2}{2\sigma^2}\right\}}{\sigma\sqrt{2\pi}} \quad (27)$$

temos a seguinte solução:

$$\rho(x, t) = \frac{\exp\left\{-\frac{(x-bt)^2}{2(\sigma^2 + 2Dt)}\right\}}{\sqrt{2\pi(\sigma^2 + 2Dt)}} \quad (28)$$

A fórmula (28) é igual ao resultado encontrado na Eq.II.54.

Em processos de difusão os coeficientes D e b são chamados coeficiente de difusão e velocidade de deriva respectivamente.

Então, as expressões (24) determinam a ligação de parâmetros básicos do modelo matemático considerado com as características físicas dos processos de difusão. Esta relação permite elaborar o programa computacional de simulação dos processos de difusão na base do modelo do Caminhante Aleatório. Os resultados dessa simulação são representados no Capítulo que se segue.

CAPITULO IV - SIMULAÇÃO COMPUTACIONAL DE PROCESSOS DE DIFUSÃO.

4.1 Simulação de difusão livre.

Vamos verificar os resultados teóricos obtidos por simulação em computador. Para isso foi elaborado o programa nomeado *RWDif.exe* [12] no meio do sistema de programação Delphi 4.0, que realiza a simulação computacional do comportamento das partículas brownianas em movimento no meio com o gradiente de temperatura, que se encontra num campo externo potencial. O programa *RWDif.exe* tem a possibilidade de simular várias combinações das condições físicas dos processos de difusão.

A simulação dos saltos das partículas brownianas pode ser realizada com ajuda da função *Random* da linguagem da programação *Pascal*. Cada chamada da função *Random* realiza um número pseudoaleatório no intervalo $0 \leq \text{Random} < 1$, então

$$\text{If Random} < q \text{ then } s := s - \beta \text{ else } s := s + \alpha; \quad (29)$$

é o procedimento da formação do deslocamento s de uma partícula browniana.

Depois de n chamadas da expressão (29), a variável s vai ser igual à coordenada da partícula sobre o eixo OX no instante do tempo $t = n\tau$, se $s = 0$ para $t = 0$.

Os resultados da simulação computacional dos processos de difusão das partículas brownianas são representados em seguintes figuras ($N = 10^6$, $\tau = 1$, $t = n$).

Todos gráficos foram construídos com ajuda do programa **RWDif** (Ver o Apêndice).

Distribuição de Gauss:

$p = 0.500$
 $q = 0.500$
 $\alpha = 6$
 $\beta = 6$
 $D = 18.000$
 $b = 0.000$
 $n = 0$
 $N_p = 1000000$
 $L_d = 55$
 $\sigma = 20$
 $N_l = 600$
 $N_h = 400$
 $K_l = 1$
 $K_h = 50$

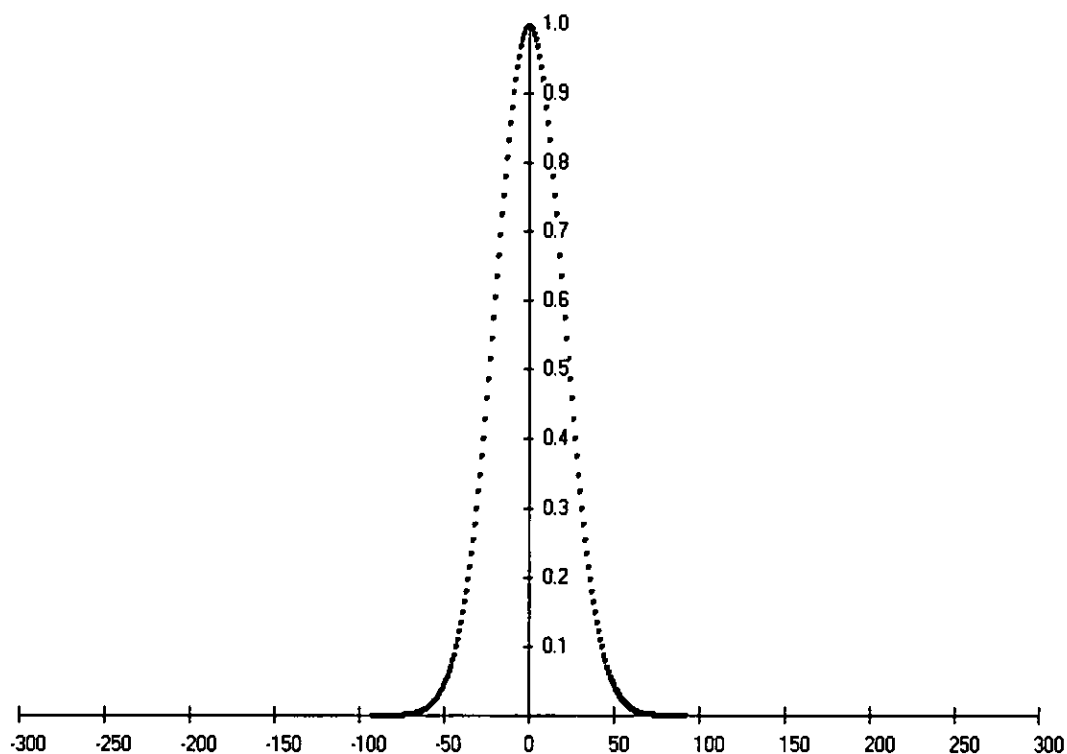


Figura 14: Representa a distribuição *inicial de Gauss* (27), $\sigma = 20$.

Aqui e mais adiante o eixo vertical é eixo da função $\rho(x,t)$ e o eixo horizontal é eixo OX.

Nós podemos ligar os parâmetros α e β com a existência do campo externo potencial quando $\alpha \neq \beta$ e parâmetros p e q com a existência do gradiente de temperatura ($p \neq q$).

Em todas figuras que se seguem, por cima dos gráficos pintados temos pontos pretos que representam as curvas teóricas calculadas pela fórmula (28).

Simulação de difusão sem deriva (livre):

$p = 0.500$
 $q = 0.500$
 $\alpha = 6$
 $\beta = 6$
 $D = 18.000$
 $b = 0.000$
 $n = 30$
 $N_p = 1000000$
 $L_d = 55$
 $\sigma = 20$
 $N_l = 600$
 $N_h = 400$
 $K_j = 1$
 $K_h = 50$

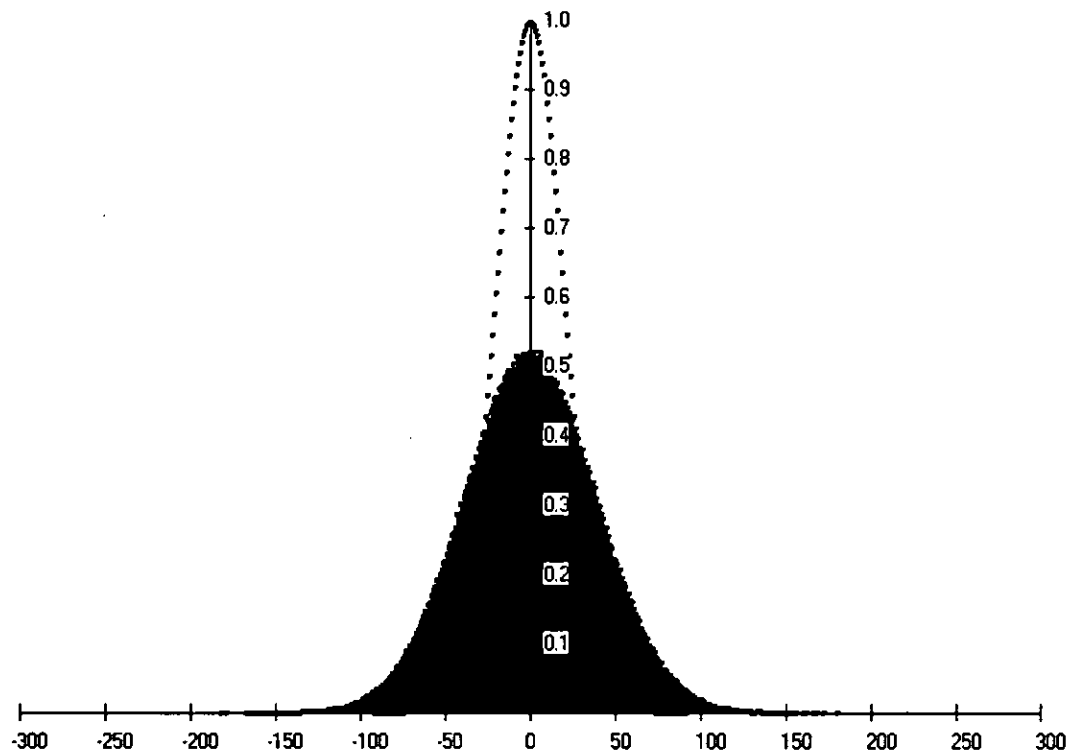


Figura 15: Representa o processo de difusão sem campo potencial externo ($\alpha = \beta = 6$) e sem gradiente de temperatura ($p = q = 0.5$), $n = 30$, $D = 18$ e $b = 0$.

4.2 Simulação de difusão em um campo externo.

Aqui temos a representação gráfica dos processos de simulação de difusão na presença de campos externos e gradiente de temperatura.

$p = 0.500$
 $q = 0.500$
 $\alpha = 7$
 $\beta = 5$
 $D = 18.000$
 $b = 1.000$
 $n = 70$
 $N_p = 1000000$
 $L_d = 55$
 $\sigma = 20$
 $N_l = 600$
 $N_h = 400$
 $K_l = 1$
 $K_h = 50$

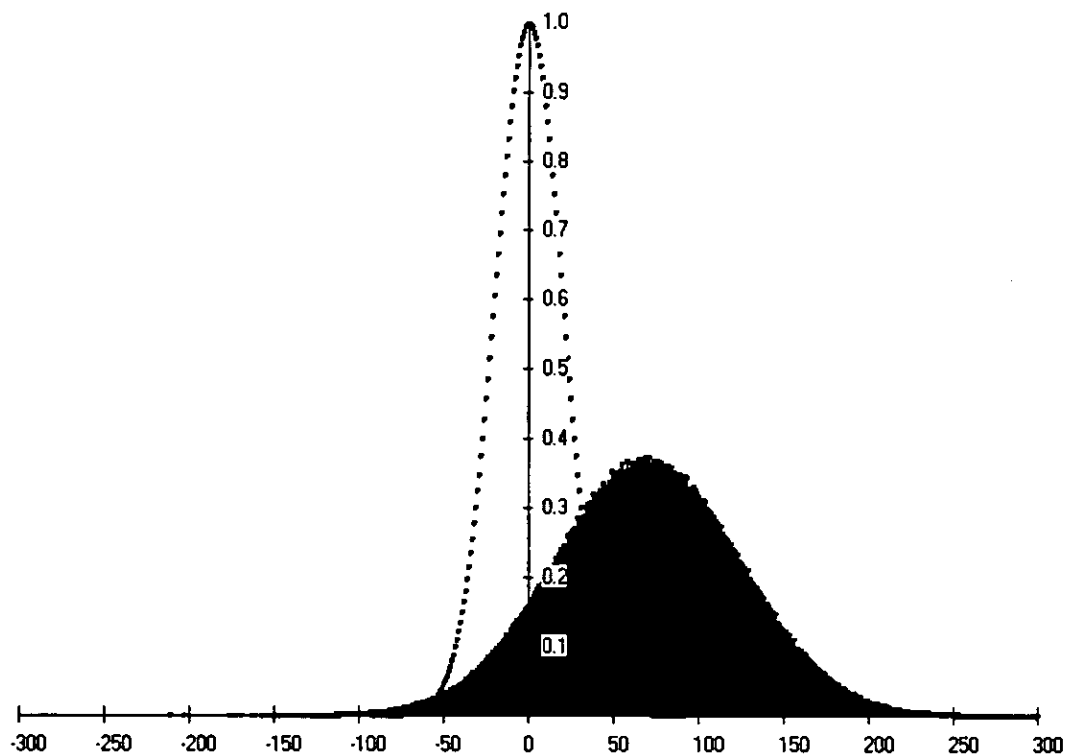


Figura 16: Representa a difusão num campo externo ($\alpha = 7$, $\beta = 5$) e sem gradiente de temperatura ($p = q = 0.5$), $n = 70$, $D = 18$ e $b = 1$.

$p = 0.600$
 $q = 0.400$
 $\alpha = 6$
 $\beta = 6$
 $D = 17.280$
 $b = 1.200$
 $n = 100$
 $N_p = 1000000$
 $L_d = 55$
 $\sigma = 20$
 $N_I = 600$
 $N_h = 400$
 $K_I = 1$
 $K_h = 50$

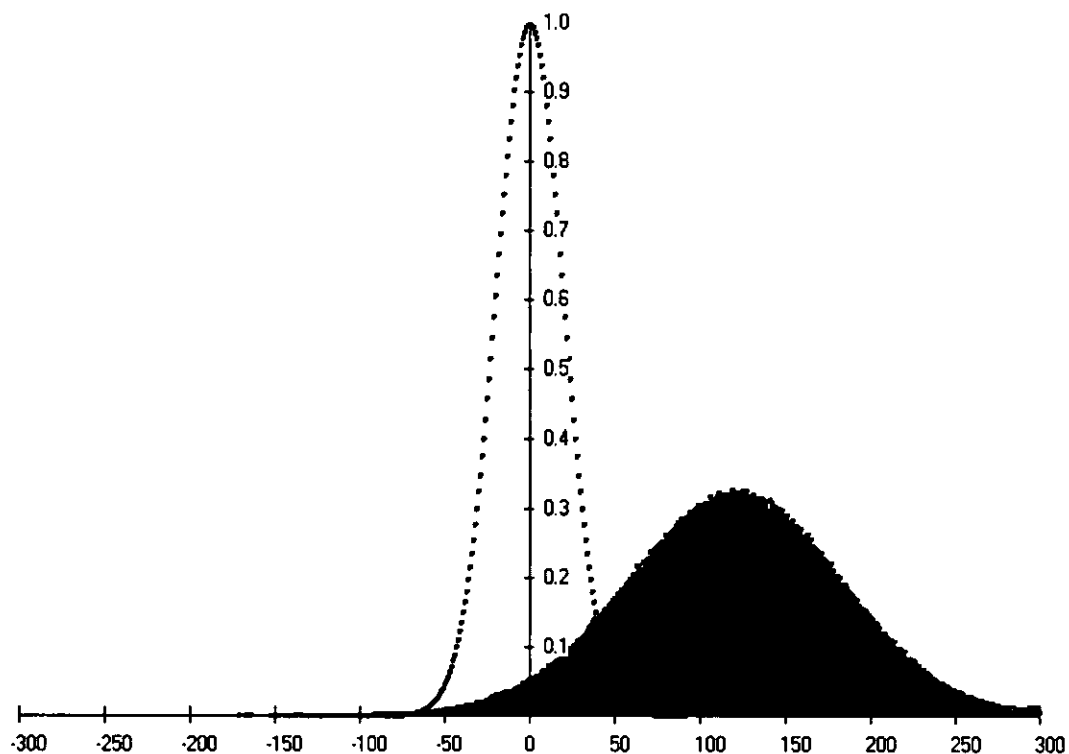


Figura 17: Representa a difusão sem campo externo potencial ($\alpha = \beta = 6$), com o gradiente de temperatura ($p = 0.6$, $q = 0.4$), $n = 100$, $D = 17.28$ e $b = 1.2$.

$p = 0.400$
 $q = 0.600$
 $\alpha = 7$
 $\beta = 5$
 $D = 17.280$
 $b = -0.200$
 $n = 150$
 $N_p = 1000000$
 $L_d = 55$
 $\sigma = 20$
 $N_l = 600$
 $N_h = 400$
 $K_l = 1$
 $K_h = 50$

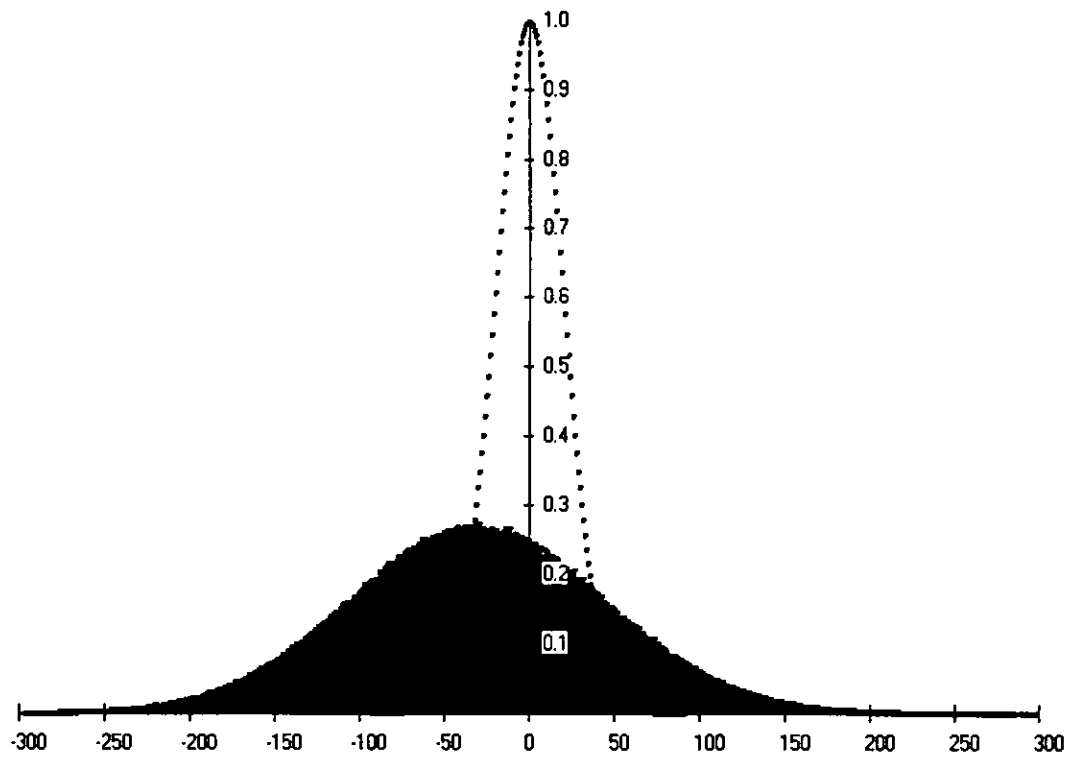


Figura 18: Representa a difusão num campo externo potencial ($\alpha = 7$, $\beta = 5$), com o gradiente de temperatura ($p = 0.4$, $q = 0,6$), $n = 150$, $D = 17.28$ e $b = -0.2$.

Simulação de difusão sem deriva (livre):

$p = 0.300$
 $q = 0.700$
 $\alpha = 7$
 $\beta = 3$
 $D = 10.500$
 $b = 0.000$
 $n = 50$
 $N_p = 1000000$
 $L_d = 55$
 $\sigma = 20$
 $N_l = 600$
 $N_h = 400$
 $K_l = 1$
 $K_h = 50$

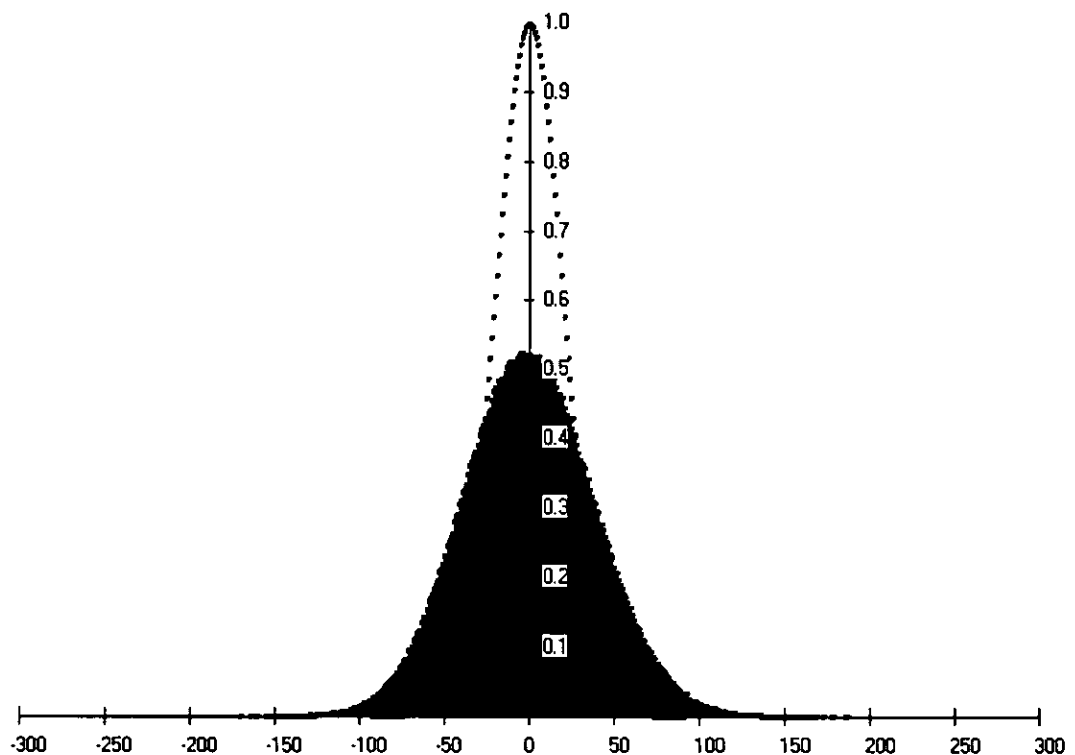


Figura 19: Representa um balanço entre o campo externo potencial ($\alpha = 7, \beta = 3$) e o gradiente de temperatura ($p = 0.3, q = 0.7$), $n = 50, D = 10.5$ e $b = 0$.

A condição geral para a difusão representada na Fig.19 é $b = 0$. Da expressão (24) para b , temos a seguinte condição: $p\alpha - q\beta = 0$.

Daqui existem duas possibilidades:

a) $\alpha = \beta$ e $p = q$, que é o caso do caminhante aleatório simétrico e corresponde a uma difusão sem existência do gradiente de temperatura e os campos externos (Fig.15).

b) $\alpha \neq \beta$ e $p \neq q$, mas

$$\frac{\alpha}{\beta} = \frac{q}{p} \tag{30}$$

Neste caso podem existir o gradiente de temperatura e um campo externo, mas a relação (30) corresponde a uma "compensação" da ação destes factores e pode ser usada para ligar os parâmetros do modelo com as características físicas da difusão.

CAPITULO V - CONCLUSÕES E RECOMENDAÇÕES.

5.1 Conclusões.

Uma vez estabelecida a relação entre os processos de difusão e o movimento browniano no modelo do caminhante Aleatório e tendo em conta os resultados das simulações realizadas na base dessa relação, conclui-se que os programas usados nesses processos de simulação representam muitas possibilidades diferentes de modelar os processos de difusão. Além disso, estes programas podem ser usados, também, para simular em computador processos de difusão limitados e permitem encontrar, nestes casos, respostas para problemas matemáticos muito complicados.

5.2 Recomendações.

Realizado o trabalho, recomenda-se que mais estudos sejam realizados para a modelação computacional dos processos de difusão mais complicados, por exemplo: processos de difusão limitados e considerar ainda problemas de contorno (quando temos condições de fronteira), problemas não lineares (quando existe interação entre as partículas), para contribuir deste modo no desenvolvimento da própria ciência e para o processo de ensino e aprendizagem.

Como sugestão, começar os estudos a partir dos resultados alcançados neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS.

1. Incropera, F. D., De Witt, D.P. (1992), *Fundamentos de Transferência de Calor e de Massa*. 3ª Edição, Rio de Janeiro.
2. Calouste, J.M., Richardson, J.F. (1977), *Fluxo de Fluidos, Transferência de Calor e Transferência de Massa*, Tecnologia Química. v. I, 3ª Edição, Lisboa, Fundação Calouste Gulbenkian.
3. Tijonov, A.N., Samarski, A.A. (1980), *Ecuaciones de la fisica matemática*. Mir, Moscú.
4. Budak, B.M., Samarski, A.A., Tijonov, A.N. (1984), *Problemas de la fisica matemática*. v. I, II, Mir, Moscú.
5. Pisonou, N. (1997), *Cálculo Diferencial e Integral*. Editora Lopes da Silva, v. II.
6. Kochkin N.I., Chirkévitch, M.G. (1986), *Prontuário de fisica elementar*. Mir, Moscovo.
7. Lage, E.J.S. (1995), *Fisica Estatística*. Lisboa. Fundação Calouste Gulbenkian.
- 8 Einstein, A. (1956), *Investigations on the theory of the Brownian movement*. Dover Publications..
9. Feller, W. (1970, 1971), *An introduction to probability theory and its applications*. v.1, v.2. New York. John Wiley & Sons.
10. Spitzer, F. (1964), *Principles of Random Walk*. New York. Princeton, New Jersey.
11. Borodin, A.N., Salminen, P. (1996), *Handbook of Brownian Motion. Facts and Formulae*. Berlin. Birkhauser Verlag.
12. Rakov, Y.A. (2003), *Materiais da Primeira AMostra de Ciências e Tecnologias* MESCT.

APÊNDICE.

Códigos dos programas usados na simulação dos processos descritos no trabalho.

▪ Programa GraphD:

```

uses
  Forms,
  MainGraD in 'MainGraD.pas' (Form1);
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.

unit MainGraD;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  jpeg, ExtCtrls, Menus, StdCtrls, Clipbrd;

type
  TForm1 = class(TForm)
    Image1: TImage;
    MainMenu1: TMainMenu;
    Options1: TMenuItem;
    Parameters1: TMenuItem;
    Clipboard1: TMenuItem;
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    Edit2: TEdit;
    Label5: TLabel;
    Button1: TButton;
    Image2: TImage;
    Button2: TButton;
    Image3: TImage;
    procedure ShowGrph;
    procedure GrphE;
    procedure Axes;
    procedure FormCreate(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Parameters1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Clipboard1Click(Sender: TObject);
    procedure FormKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  xmax: real = 2; {m}
  ymax: real = 2.5; {1/m}
  Nxmax: integer = 600;
  Nymax: integer = 400;

var
  Form1: TForm1;

```

```

Nx,Ny,dw,dh,dwl,dhl: integer;
ArsE: array of integer;
D,t: real;

implementation

{$R *.DFM}

procedure TForm1.ShowGrph;
const
  d: integer = 1;
var
  i: integer;
begin
  Image2.Canvas.Pen.Width:= 3;
  Image2.Canvas.Pen.Style:= psSolid;
  Image2.Canvas.Pen.Color:= clBlue;

  Image2.Canvas.MoveTo(2*dw,0);
  Image2.Canvas.LineTo(2*dw,4*dh-50);

  for i:=1 to Nxmax-1 do
  begin
    Image2.Canvas.MoveTo(2*dw-3*dwl+i,4*dh-50-ArsE[i]);
    Image2.Canvas.LineTo(2*dw-3*dwl+i+d,4*dh-50-ArsE[i+d]);
  end;
end;

procedure TForm1.GrphE;
var
  i: integer;
  x,y: real;
  rNx,rNy,rNxmax,rNymax: real;
begin
  rNxmax:= Nxmax;
  rNymax:= Nymax;
  for i:= 1 to Nxmax do
  begin
    rNx:= i-(Nxmax div 2);
    x:= (xmax/rNxmax)*rNx;
    y:= 0;
    if sqr(x)/(4*D*t) < 5 then
      y:= exp(-sqr(x)/(4*D*t))/(2*sqrt(pi*D*t));
    rNy:= (rNymax/ymax)*y;
    Ny:= Round(rNy);
    ArsE[i]:= Ny;
  end;
end;

procedure TForm1.Axes;
var
  i: integer;
begin
  Image2.Canvas.Pen.Width:= 2;
  Image2.Canvas.Pen.Style:= psSolid;
  Image2.Canvas.Pen.Color:= clBlack;
  Image2.Canvas.MoveTo(0,4*dh-50);
  Image2.Canvas.LineTo(4*dw,4*dh-50);
  Image2.Canvas.MoveTo(2*dw,0);
  Image2.Canvas.LineTo(2*dw,4*dh-50);

  for i:= 1 to 5 do
  begin
    Image2.Canvas.MoveTo(2*dw-7,4*dh-50-i*80);
    Image2.Canvas.LineTo(2*dw+7,4*dh-50-i*80);
  end;
  Image2.Canvas.Font.Name:= 'Arial';

```

```

Image2.Canvas.Font.Size:= 14;
Image2.Canvas.Font.Color:= clBlack;
Image2.Canvas.TextOut (2*dw+20, 4*dh-63-Nymax, '2.5 (1/m)');
Image2.Canvas.TextOut (2*dw-6, 4*dh-45, '0');
Image2.Canvas.Font.Name:= 'Symbol';
Image2.Canvas.Font.Size:= 16;
Image2.Canvas.Font.Color:= clBlack;
Image2.Canvas.TextOut (2*dw+10, 5, 'r');
Image2.Canvas.Font.Name:= 'Arial';
Image2.Canvas.Font.Size:= 14;
Image2.Canvas.TextOut (2*dw+20, 10, '(x, t)');
for i:= 1 to 5 do
begin
    Image2.Canvas.MoveTo (2*dw+i*60, 4*dh-50-7);
    Image2.Canvas.LineTo (2*dw+i*60, 4*dh-50+7);
end;
Image2.Canvas.TextOut (2*dw+3*dwl-10, 4*dh-45, '1m');

for i:= 1 to 5 do
begin
    Image2.Canvas.MoveTo (2*dw-i*60, 4*dh-50-7);
    Image2.Canvas.LineTo (2*dw-i*60, 4*dh-50+7);
end;
Image2.Canvas.TextOut (2*dw-3*dwl-20, 4*dh-45, '-1m');
Image2.Canvas.TextOut (2*dw+3*dwl+50, 4*dh-73, 'X');

Image2.Canvas.TextOut (50, 20, 'D = '+ Edit1.Text+' m^2/s ');
Image2.Canvas.TextOut (50, 50, 't = '+ Edit2.Text+' s ');
Image2.Visible:= True;

end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    WindowState:= wsMaximized;
    KeyPreview:= True;
end;

procedure TForm1.FormActivate(Sender: TObject);
var
    i: integer;
begin
    dh:= ClientHeight div 4;
    dw:= ClientWidth div 4;
    dhl:= Nymax div 4;
    dwl:= Nxmax div 6;
    SetLength(Arse, Nxmax+1);
    for i:=0 to Nxmax do Arse[i]:= 0;
    Image1.AutoSize:= True;
    Image1.Top:= dh;
    Image1.Left:= (ClientWidth - Image1.Width) div 2;
    Image1.Visible:= True;
    Panell.Visible:= False;
    Image2.Left:= 0;
    Image2.Top:= 0;
    Image2.Width:= ClientWidth;
    Image2.Height:= ClientHeight;
    Image2.Canvas.Brush.Color:= clWhite;
    Image2.Canvas.FillRect(ClientRect);
    Image2.Visible:= False;
    Image3.AutoSize:= True;
    Image3.Top:= dh div 4;
    Image3.Left:= (ClientWidth - Image3.Width) div 2;
    Image3.Visible:= True;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin

```

```
D:= StrToFloat(Edit1.Text);
t:= StrToFloat(Edit2.Text);
Image1.Visible:= False;
Panell.Visible:= False;
Axes;
GrphE;
ShowGrph;
end;

procedure TForm1.Parameters1Click(Sender: TObject);
begin
  Panell.Visible:= True;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Image1.Visible:= False;
  Image3.Visible:= False;
  Button2.Visible:= False;
  Panell.Visible:= True;
end;

procedure TForm1.ClipBoard1Click(Sender: TObject);
begin
  Clipboard.Assign(Image2.Picture);
  ShowMessage('Save into Clipboard');
end;

procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if Key = vk_escape then
  begin
    Image2.Canvas.Brush.Color:= clWhite;
    Image2.Canvas.FillRect(ClientRect);
  end;
end;
end.
```


▪ Programa GraphG:

```

uses
  Forms,
  MainGraG in 'MainGraG.pas' {Form1};
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.

unit MainGraG;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  jpeg, ExtCtrls, Menus, StdCtrls, Clipbrd;

type
  TForm1 = class(TForm)
    Image1: TImage;
    MainMenu1: TMainMenu;
    Options1: TMenuItem;
    Parameters1: TMenuItem;
    Clipboard1: TMenuItem;
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    Edit2: TEdit;
    Label5: TLabel;
    Button1: TButton;
    Image2: TImage;
    Button2: TButton;
    Image3: TImage;
    Label6: TLabel;
    Edit3: TEdit;
    Label7: TLabel;
    Label8: TLabel;
    Edit4: TEdit;
    Label9: TLabel;
    procedure ShowGrph;
    procedure GrphE;
    procedure Axes;
    procedure FormCreate(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Parameters1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Clipboard1Click(Sender: TObject);
    procedure FormKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  xmax: real = 2; {m}
  ymax: real = 2.5; {1/m}
  Nxmax: integer = 600;
  Nymax: integer = 400;

var
  Form1: TForm1;

```

```

Nx, Ny, dw, dh, dwl, dhl, dt: integer;
ArsE: array of integer;
D, b, sigma, t: real;

implementation

{$R *.DFM}

procedure TForm1.ShowGrph;
const
  d: integer = 1;
var
  i: integer;
begin
  Image2.Canvas.Pen.Width:= 3;
  Image2.Canvas.Pen.Style:= psSolid;
  Image2.Canvas.Pen.Color:= clBlue;

  Image2.Canvas.MoveTo(2*dw, 0);
  Image2.Canvas.LineTo(2*dw, 4*dh-50);

  for i:=1 to Nxmax-1 do
  begin
    Image2.Canvas.MoveTo(2*dw-3*dwl+i, 4*dh-50-ArsE[i]);
    Image2.Canvas.LineTo(2*dw-3*dwl+i+d, 4*dh-50-ArsE[i+d]);
  end;
end;

procedure TForm1.GrphE;
var
  i: integer;
  x, y: real;
  rNx, rNy, rNxmax, rNymax: real;
begin
  rNxmax:= Nxmax;
  rNymax:= Nymax;
  for i:= 1 to Nxmax do
  begin
    rNx:= i-(Nxmax div 2);
    x:= (xmax/rNxmax)*rNx;
    y:= 0;
    if sqrt(x-b*t)/(2*sqrt(sigma)+4*D*t) < 5 then
      y:= exp(-sqrt(x-b*t)/(2*sqrt(sigma)+4*D*t))/sqrt(2*pi*(sqrt(sigma)+2*D*t));
    rNy:= (rNymax/y)*y;
    Ny:= Round(rNy);
    ArsE[i]:= Ny;
  end;
end;

procedure TForm1.Axes;
var
  i: integer;
begin
  Image2.Canvas.Pen.Width:= 2;
  Image2.Canvas.Pen.Style:= psSolid;
  Image2.Canvas.Pen.Color:= clBlack;
  Image2.Canvas.MoveTo(0, 4*dh-50);
  Image2.Canvas.LineTo(4*dw, 4*dh-50);
  Image2.Canvas.MoveTo(2*dw, 0);
  Image2.Canvas.LineTo(2*dw, 4*dh-50);

  for i:= 1 to 5 do
  begin
    Image2.Canvas.MoveTo(2*dw-7, 4*dh-50-i*80);
    Image2.Canvas.LineTo(2*dw+7, 4*dh-50-i*80);
  end;
  Image2.Canvas.Font.Name:= 'Arial';
  Image2.Canvas.Font.Size:= 14;

```

```

Image2.Canvas.Font.Color:= clBlack;
Image2.Canvas.TextOut(2*dw+20,4*dh-63-Nymax,'2.5(1/m)');
Image2.Canvas.TextOut(2*dw-6,4*dh-45,'0');
Image2.Canvas.Font.Name:= 'Symbol';
Image2.Canvas.Font.Size:= 16;
Image2.Canvas.Font.Color:= clBlack;
Image2.Canvas.TextOut(2*dw+10,0,'r');
Image2.Canvas.Font.Name:= 'Arial';
Image2.Canvas.Font.Size:= 14;
Image2.Canvas.TextOut(2*dw+20,5,'(x,t)');

for i:= 1 to 5 do
begin
  Image2.Canvas.MoveTo(2*dw+i*60,4*dh-50-7);
  Image2.Canvas.LineTo(2*dw+i*60,4*dh-50+7);
end;
Image2.Canvas.TextOut(2*dw+3*dwl-10,4*dh-45,'1m');

for i:= 1 to 5 do
begin
  Image2.Canvas.MoveTo(2*dw-i*60,4*dh-50-7);
  Image2.Canvas.LineTo(2*dw-i*60,4*dh-50+7);
end;
Image2.Canvas.TextOut(2*dw-3*dwl-20,4*dh-45,'-1m');
Image2.Canvas.TextOut(2*dw+3*dwl+50,4*dh-73,'X');

Image2.Canvas.TextOut(50,50,'D = '+ Edit1.Text+' m^2/s ');
Image2.Canvas.TextOut(50,80,'b = '+ Edit3.Text+' m/s ');
Image2.Canvas.TextOut(50,110,'sigma = '+ Edit4.Text+' m^2/s ');

Image2.Canvas.TextOut(50,170+20*dt,'t'+IntToStr(dt)+' = '+Edit2.Text+' s');
dt:= dt + 1;
Image2.Visible:= True;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  WindowState:= wsMaximized;
  KeyPreview:= True;
end;

procedure TForm1.FormActivate(Sender: TObject);
var
  i: integer;
begin
  dt:= 0;
  dh:= ClientHeight div 4;
  dw:= ClientWidth div 4;
  dh1:= Nymax div 4;
  dwl:= Nxmax div 6;
  SetLength(ArsE,Nxmax+1);
  for i:=0 to Nxmax do ArsE[i]:= 0;
  Image1.AutoSize:= True;
  Image1.Top:= dh div 2;
  Image1.Left:= (ClientWidth - Image1.Width) div 2;
  Image1.Visible:= True;
  Panell.Visible:= False;
  Image2.Left:= 0;
  Image2.Top:= 0;
  Image2.Width:= ClientWidth;
  Image2.Height:= ClientHeight;
  Image2.Canvas.Brush.Color:= clWhite;
  Image2.Canvas.FillRect(ClientRect);
  Image2.Visible:= False;
  Image3.AutoSize:= True;
  Image3.Top:= 0;
  Image3.Left:= (ClientWidth - Image3.Width) div 2;
  Image3.Visible:= True;

```

```
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  D:= StrToFloat(Edit1.Text);
  b:= StrToFloat(Edit3.Text);
  sigma:= StrToFloat(Edit4.Text);
  t:= StrToFloat(Edit2.Text);
  Image1.Visible:= False;
  Panell.Visible:= False;
  Axes;
  GrphE;
  ShowGrph;
end;

procedure TForm1.Parameters1Click(Sender: TObject);
begin
  Panell.Visible:= True;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Image1.Visible:= False;
  Image3.Visible:= False;
  Button2.Visible:= False;
  Panell.Visible:= True;
end;

procedure TForm1.ClipBoard1Click(Sender: TObject);
begin
  Clipboard.Assign(Image2.Picture);
  ShowMessage('Save into Clipboard');
end;

procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if Key = vk_escape then
  begin
    dt:= 0;
    Image2.Canvas.Brush.Color:= clWhite;
    Image2.Canvas.FillRect(ClientRect);
  end;
end;
end.
```

```

    • Programa GraphRW:

    uses
    Forms,
    MainGrRW in 'MainGrRW.pas' {Form1};
    {$R *.RES}
begin
    Application.Initialize;
    Application.CreateForm(TForm1, Form1);
    Application.Run;
end.

unit MainGrRW;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, ExtCtrls, Menus, jpeg, ClipBrd;

type
    TForm1 = class(TForm)
        Image1: TImage;
        Image2: TImage;
        MainMenu1: TMainMenu;
        Options1: TMenuItem;
        Parameters1: TMenuItem;
        Clipboard1: TMenuItem;
        Panell: TPanel;
        Label1: TLabel;
        Label2: TLabel;
        Edit1: TEdit;
        Edit2: TEdit;
        Button1: TButton;
        Button2: TButton;
        Timer1: TTimer;
        procedure ShowGrph;
        procedure Axes;
        procedure FormCreate(Sender: TObject);
        procedure FormActivate(Sender: TObject);
        procedure Parameters1Click(Sender: TObject);
        procedure Clipboard1Click(Sender: TObject);
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
        procedure Timer1Timer(Sender: TObject);
        procedure FormKeyDown(Sender: TObject; var Key: Word;
            Shift: TShiftState);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

const
    Nxmax: integer = 600;
    Nymax: integer = 400;

var
    Form1: TForm1;
    Nx, Ny, dw, dh, dwl, dhl: integer;
    p: real;
    alfa, n, dn, Np: integer;
    ArsRW, ArsTr: array of integer;

implementation

```

```

{$R *.DFM}

procedure TForm1.ShowGrph;
var
  i,s: integer;
begin
  Image2.Canvas.Pen.Width:= 1;
  Image2.Canvas.Pen.Style:= psSolid;
  Image2.Canvas.Pen.Color:= clGreen;
  Np:= Np +1;
  s:= 0;
  ArsTr[0]:= 0;
  for i:= 1 to n do
  begin
    if Random < p then s:= s + 1 else s:= s - 1;
    ArsTr[i]:= s;
  end;
  ArsRW[n-s]:= ArsRW[n-s] + 1;
  for i:= 1 to n do
  begin
    Image2.Canvas.MoveTo(5+(i-1)*dn,2*dh-alfa*ArsTr[i-1]);
    Image2.Canvas.LineTo(5+i*dn,2*dh-alfa*ArsTr[i]);
  end;
  Np:= Np +1;
  s:= 0;
  ArsTr[0]:= 0;
  for i:= 1 to n-1 do
  begin
    if Random < p then s:= s + 1 else s:= s - 1;
    ArsTr[i]:= s;
  end;
  ArsRW[n-s]:= ArsRW[n-s] + 1;
  for i:= 1 to n-1 do
  begin
    Image2.Canvas.MoveTo(5+(i-1)*dn,2*dh-alfa*ArsTr[i-1]);
    Image2.Canvas.LineTo(5+i*dn,2*dh-alfa*ArsTr[i]);
  end;

  Image2.Canvas.Pen.Width:= 2;
  Image2.Canvas.Pen.Color:= clBlue;
  for i:= 0 to 2*n do
  begin
    Image2.Canvas.MoveTo(5+Nxmax,2*dh-(n-i));
    Image2.Canvas.LineTo(5+Nxmax+ArsRW[i],2*dh-(n-i));
  end;
  Image2.Canvas.TextOut(50,20,'n = '+ IntToStr(n));
  Image2.Canvas.TextOut(50,40,'Np = '+ IntToStr(Np));
end;

procedure TForm1.Axes;
begin
  Image2.Canvas.Pen.Width:= 2;
  Image2.Canvas.Pen.Style:= psSolid;
  Image2.Canvas.Pen.Color:= clBlack;
  Image2.Canvas.MoveTo(5,0);
  Image2.Canvas.LineTo(5,4*dh);
  Image2.Canvas.MoveTo(5,2*dh);
  Image2.Canvas.LineTo(4*dw,2*dh);
  Image2.Canvas.TextOut(10,10,'X');
  Image2.Canvas.TextOut(0,2*dh-7,'0');
  Image2.Canvas.TextOut(4*dw-30,2*dh+7,'n(x,t)');
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  WindowState:= wsMaximized;

```

```

    KeyPreview:= True;
end;

procedure TForm1.FormActivate(Sender: TObject);
var
    i: integer;
begin
    dh:= ClientHeight div 4;
    dw:= ClientWidth div 4;
    dh1:= Nymax div 4;
    dw1:= Nxmax div 4;
    Np:= 0;
    p:= 0.5;
    n:= 200;
    dn:= Nxmax div n;
    alfa:= Nymax div n;
    alfa:= alfa div 2;
    SetLength(ArsRW,2*n+1);
    for i:=0 to 2*n do ArsRW[i]:= 0;
    SetLength(ArsTr,n+1);
    for i:=0 to n do ArsTr[i]:= 0;

    Image1.AutoSize:= True;
    Image1.Top:= dh;
    Image1.Left:= (ClientWidth - Image1.Width) div 2;
    Image1.Visible:= True;
    Panell.Visible:= False;
    Image2.Left:= 0;
    Image2.Top:= 0;
    Image2.Width:= ClientWidth;
    Image2.Height:= ClientHeight;
    Image2.Canvas.Brush.Color:= clWhite;
    Image2.Canvas.FillRect(ClientRect);
    Image2.Visible:= False;
    Timer1.Interval:= 1;
    Timer1.Enabled:= False;
end;

procedure TForm1.Parameters1Click(Sender: TObject);
begin
    Panell.Top:= 0;
    Panell.Left:= 0;
    Panell.Visible:= True;
    Timer1.Enabled:= False;
end;

procedure TForm1.ClipBoard1Click(Sender: TObject);
begin
    Clipboard.Assign(Image2.Picture);
    ShowMessage('Save into Clipboard');
end;

procedure TForm1.Button1Click(Sender: TObject);
var
    i: integer;
begin
    Np:= 0;
    p:= StrToFloat(Edit1.Text);
    n:= StrToInt(Edit2.Text);

    dn:= Nxmax div n;
    alfa:= Nymax div n;
    alfa:= alfa div 2;
    SetLength(ArsRW,2*n+1);
    for i:=0 to 2*n do ArsRW[i]:= 0;
    SetLength(ArsTr,n+1);
    for i:=0 to n do ArsTr[i]:= 0;

```

```
    Image1.Visible:= False;
    Panel1.Visible:= False;
    Button2.Visible:= False;
    Image2.Visible:= True;
    Axes;
    Randomize;
    Timer1.Enabled:= True;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    Image1.Visible:= False;
    Panel1.Visible:= False;
    Button2.Visible:= False;
    Image2.Visible:= True;
    Axes;
    Randomize;
    Timer1.Enabled:= True;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
    ShowGrph;
end;

procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if Key = vk_escape then
        begin
            Image2.Canvas.Brush.Color:= clWhite;
            Image2.Canvas.FillRect(ClientRect);
        end;
    if Key = vk_F1 then Timer1.Enabled:= not Timer1.Enabled;
end;
end.
```