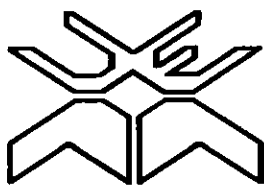
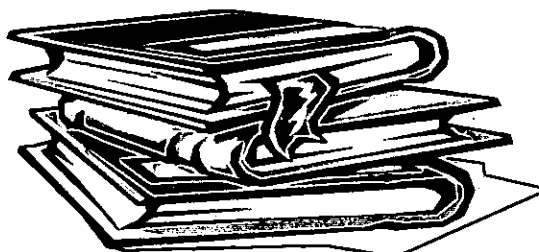


FIS-50



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

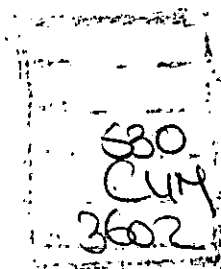
TRABALHO DE LICENCIATURA



**APLICAÇÃO DO MÉTODO DA FUNÇÃO DE GREEN EM
MODELAÇÃO COMPUTACIONAL**

Autora: Cândida Aurora Cumbe

Maputo, Março de 2000





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

TRABALHO DE LICENCIATURA



**APLICAÇÃO DO MÉTODO DA FUNÇÃO DE GREEN
EM MODELAÇÃO COMPUTACIONAL**

Supervisor: PROF. Dr. YURI RAKOV

Maputo, Março de 2000

Para ti querida mãe
Clara Enoque Cumbe
Figura ímpar da minha vida

AGRADECIMENTOS

A todos aqueles que em mim confiaram e depositaram esperança, e, a quem, de um ou de outro modo, contribuiu para que o presente trabalho de licenciatura se tornasse uma realidade, desejo profundamente expressar aqui o meu eterno reconhecimento, em especial:

- Ao Prof. Dr. Yuri Rakov, meu supervisor, pela disponibilidade e interesse em orientar e discutir os objectivos básicos deste trabalho, e pela paciência infinita que teve na discussão da metodologia e estratégias de estudar e abordar os assuntos.
- Agradeço ainda pela excelente simplicidade na transmissão de sua sabedoria durante o tratamento das questões.
- À direcção do departamento e a todos os meus docentes pela capacidade extraordinária de transmitir os seus conhecimentos ao longo do curso em especial ao Prof. Dr. Rogério Utui pela grande força moral prestada numa altura do meu curso, extremamente difícil.
- Ao dr. Genito Maure, ao dr. Víctor Saide pelas suas valiosas sugestões na abordagem de certas questões ao longo da preparação do trabalho e grande auxílio prestado no uso de software.
- Ao dr. Emílio Mosse, pela ajuda prestada na tradução de algumas bibliografias em língua russa.
- A todos os meus colegas do curso que ao longo desta caminhada sempre estiveram do meu lado prestando todo apoio moral e espiritual até que este sonho se tornasse realidade, em especial ao dr. António Osvaldo Saide, António Uaisone, Alfredo Mahoche e ao Jonas Zucule pelas maravilhosas sugestões dadas ao longo deste trabalho.
- Ao Eng^o Fumo, pela força e companheirismo prestado, criando de certo modo bases que permitiram a minha harmoniosa dedicação ao trabalho de licenciatura.
- À minha família, meus amigos e todos aqueles que directa ou indirectamente, contribuíram para que a realização do presente trabalho de licenciatura chegasse ao fim, o meu muito obrigada.

DECLARAÇÃO DE HONRA

Declaro por minha honra, que este trabalho é resultado da minha própria investigação com recurso à bibliografia e referências citadas ao longo do trabalho.

Maputo, Março de 2000

A autora

Cândida Aurora Cumbe

(Cândida Aurora Cumbe)

RESUMO

O objectivo primordial da aplicação do método da função de Green é mostrar a importância do uso das funções generalizadas no estudo de vários sistemas físicos.

O presente trabalho vem dar um contributo à concepção do valor da utilização do método da função de Green no ensino, de modo a simplificar o estudo e resolução de equações diferenciais em problemas muito complicados de física, bem como dar um conhecimento mais amplo mostrando a eficiência que o método da função de Green transporta em relação a outros métodos matemáticos de resolução de equações diferenciais.

A análise feita permite destacar a possibilidade de submeter os resultados obtidos pelo método da função de Green à modelação computacional.

ÍNDICE

I. Introdução.....	3
II. Função de Green de um Oscilador Amortecido.....	4
III. Oscilador Amortecido Sob Acção de uma Força Impulsiva.....	10
IV. Oscilador Amortecido Sob Acção de uma Força Impulsiva Periódica.....	19
V. Conclusões.....	29
VI. Recomendações.....	30
VII. Bibliografia.....	31
VIII. Apêndices	32

I. INTRODUÇÃO

O método da função de Green é um dos métodos básicos de resolução dos problemas de contorno de equações diferenciais. É um método clássico e pode ser encontrado em muitos manuais de matemática (veja, por exemplo, [1], [2]). Este método tem, também uma ampla aplicação na física teórica, em particular, em teoria quântica e física estatística [3], [4].

O desenvolvimento do método da função de Green na física teórica está ligado à utilização das funções generalizadas. Um exemplo básico da função deste tipo é a δ - função de Dirac. Ela foi introduzida na mecânica quântica pelo Dirac em 1930, no seu livro “Princípios de Mecânica Quântica” e agora utiliza-se, praticamente, em todos os manuais (veja, por exemplo, [5], [6]).

A teoria matemática fundamental das funções generalizadas foi criada 20 anos depois da introdução de δ - função na física. Esta teoria tem na matemática o nome de Teoria das Distribuições [7] – [11].

Neste trabalho é considerado o método da função de Green do ponto de vista de aplicação à modelação computacional. Para isso foi escolhido um sistema muito importante na física, “o Oscilador Amortecido forçado clássico”. Para demonstrar a possibilidade de utilização das funções generalizadas na Mecânica Clássica são consideradas as forças externas impulsivas. As fórmulas obtidas são usadas, também, para a modelação computacional no caso das forças impulsivas aleatórias.

A linguagem básica de programação usada neste trabalho é *PASCAL* no meio de Delphi 4, [14], [15].

II. FUNÇÃO DE GREEN DE UM OSCILADOR AMORTECIDO

Neste capítulo é considerada a função de Green de um oscilador amortecido que tem a seguinte equação de movimento:

$$\frac{d^2x}{dt^2} + 2\beta \frac{dx}{dt} + \omega_0^2 x = f(t) \quad (1)$$

Como se sabe, esta equação tem muitas aplicações físicas e por isso as funções $x(t)$, $f(t)$ podem ser várias grandezas físicas. Neste trabalho usam-se as seguintes definições:

β - constante de amortecimento,

ω_0 - frequência angular,

$f(t)$ - força externa generalizada.

A função incógnita $x(t)$ é uma coordenada generalizada. É possível determinar a solução da equação (1) para uma função $f(t)$ arbitrária com a ajuda da função de Green $G(t, t')$ desta equação que satisfaz à equação:

$$\frac{d^2G(t, t')}{dt^2} + 2\beta \frac{dG(t, t')}{dt} + \omega_0^2 G(t, t') = \delta(t - t'), \quad (2)$$

onde $\delta(t-t')$ é δ -função de Dirac.

Conhecendo a função de Green, a solução da equação (1) pode ser escrita como:

$$x(t) = \int_{-\infty}^{\infty} G(t, t') f(t') dt' \quad (3)$$

Realmente:

$$\frac{d^2x}{dt^2} + 2\beta \frac{dx}{dt} + \omega_0^2 x = \int_{-\infty}^{\infty} \left[\frac{d^2G(t, t')}{dt^2} + 2\beta \frac{dG(t, t')}{dt} + \omega_0^2 G(t, t') \right] f(t') dt'$$

Mas da equação (2) temos que a expressão entre parêntesis é igual à $\delta(t - t')$, isto é:

$$\frac{d^2x}{dt^2} + 2\beta \frac{dx}{dt} + \omega_0^2 x = \int_{-\infty}^{\infty} \delta(t - t') f(t') dt' = f(t),$$

Quer dizer, $x(t)$ na função (3) é solução da equação (1).

Para procurar a função de Green $G(t, t')$, isto é resolver a equação (2), vamos usar o método de transformação de Fourier.

A representação de δ -função de Dirac na transformação de Fourier é:

$$\delta(t-t') = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega(t-t')} d\omega \quad (4)$$

Por isso vamos usar:

$$G(t,t') = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{G}(\omega) e^{i\omega(t-t')} d\omega, \quad (5)$$

onde $\tilde{G}(\omega)$ é a imagem de Fourier da função $G(t,t')$.

Então a equação (2) dá-nos:

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} (-\omega^2 + 2\beta i\omega + \omega_0^2) \tilde{G}(\omega) e^{i\omega(t-t')} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega(t-t')} d\omega$$

de onde temos:

$$\begin{aligned} (-\omega^2 + 2\beta i\omega + \omega_0^2) \tilde{G}(\omega) &= 1 \\ \text{ou} \\ \tilde{G}(\omega) &= - \frac{1}{\omega^2 - 2\beta i\omega - \omega_0^2} \end{aligned} \quad (6)$$

Usando esta expressão na fórmula (5) temos:

$$G(t,t') = - \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{e^{i\omega(t-t')}}{\omega^2 - 2\beta i\omega - \omega_0^2} d\omega \quad (7)$$

A função

$$f(\omega) = \frac{e^{i\omega(t-t')}}{\omega^2 - 2\beta i\omega - \omega_0^2} d\omega \quad (8)$$

tem seguintes polos no plano complexo:

$$\begin{aligned} \omega_1 &= i\beta + \sqrt{\omega_0^2 - \beta^2} \\ \omega_2 &= i\beta - \sqrt{\omega_0^2 - \beta^2} \end{aligned} \quad (9)$$

e por causa da existência do factor $e^{i\omega(t-t')}$ pode-se usar o teorema de resíduos para calcular o integral (7).

No caso geral, existem três combinações dos parâmetros β e ω_0 . Vamos considerá-las separadamente:

a) $\omega_0 < \beta$

Neste caso, os polos da função (8) são:

$$\omega_1 = i(\beta + \xi) \quad (10)$$

$$\omega_2 = i(\beta - \xi)$$

$$\text{onde } \xi = \sqrt{\beta^2 - \omega_0^2} \quad (11)$$

A escolha do caminho de integração no plano complexo de $z = \omega + iy$ depende do sinal da diferença $t - t'$.

É necessário escolher o caminho representado na figura 1 para $t > t'$.

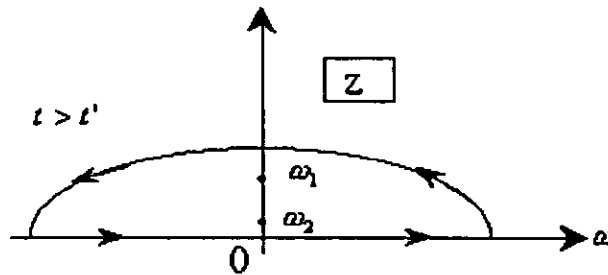


Figura 1: caminho de integração para $t > t'$ quando $\omega_0 < \beta$

Então do teorema de resíduos temos que:

$$-\frac{1}{2\pi} \int_{-\infty}^{\infty} f(\omega) d\omega = -\frac{1}{2\pi} \{2\pi i \operatorname{Res} f(\omega_1) + 2\pi i \operatorname{Res} f(\omega_2)\} \quad (12)$$

Representando

$$f(\omega) = \frac{e^{i\omega(t-t')}}{(\omega - \omega_1)(\omega - \omega_2)} \quad (13)$$

e usando a fórmula conhecida para os polos simples temos:

$$\operatorname{Res} f(\omega_1) = \lim_{\omega \rightarrow \omega_1} \left\{ \frac{(\omega - \omega_1) e^{i\omega(t-t')}}{(\omega - \omega_1)(\omega - \omega_2)} \right\} = \frac{e^{i\omega_1(t-t')}}{\omega_1 - \omega_2}$$

$$\operatorname{Res} f(\omega_2) = \lim_{\omega \rightarrow \omega_2} \left\{ \frac{(\omega - \omega_2) e^{i\omega(t-t')}}{(\omega - \omega_1)(\omega - \omega_2)} \right\} = \frac{e^{i\omega_2(t-t')}}{\omega_2 - \omega_1}$$

Usando estas expressões na fórmula (12) temos:

$$-\frac{1}{2\pi} \int_{-\infty}^{\infty} f(\omega) d\omega = -i \left\{ \frac{e^{-(\beta+\xi)(t-t')} - e^{-(\beta-\xi)(t-t')}}{2i\xi} \right\} = e^{-\beta(t-t')} \frac{e^{\xi(t-t')} - e^{-\xi(t-t')}}{2\xi}$$

De fórmulas (7) e (8) temos finalmente:

$$G(t, t') = e^{-\beta(t-t')} \frac{\text{sh} \xi(t-t')}{\xi}, \quad t > t' \quad (14)$$

onde $\text{sh} \alpha$ é o seno hiperbólico definido por:

$$\text{sh} \alpha = \frac{e^\alpha - e^{-\alpha}}{2}$$

Para $t < t'$ o caminho de integração deve ter a forma representada na figura 2.

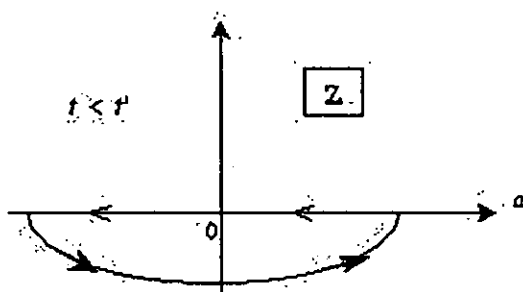


Figura 2: caminho de integração para $t < t'$ quando $\omega_0 < \beta$

O teorema de resíduos dá-nos neste caso:

$$G(t, t') = 0, \quad t < t' \quad (15)$$

As fórmulas (14) e (15) podem ser representadas como uma fórmula, usando a θ -função de Heviside (veja o Apêndice I):

$$G(t, t') = \theta(t-t') e^{-\beta(t-t')} \frac{\text{sh} \xi(t-t')}{\xi} \quad (16)$$

b) $\omega_0 = \beta$

Neste caso, temos um polo duplo:

$$\omega_1 = \omega_2 = i\beta$$

Para $t > t'$ o caminho de integração está representado na figura 3.

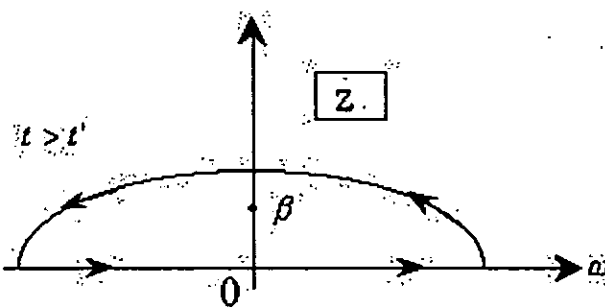


Figura 3: caminho de integração para $t > t'$ quando $\omega_0 = \beta$

para calcular o resíduo no polo duplo podemos usar a fórmula geral para o resíduo em um polo de m-ésima ordem z_i :

$$\text{Res}f(z_i) = \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} [f(z)(z-z_i)^m]_{z=z_i}$$

Então:

$$-\frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{e^{i\omega(t-t')}}{(\omega-i\beta)^2} d\omega = -\frac{1}{2\pi} \{2\pi i [i(t-t')e^{-\beta(t-t')}] \}$$

Finalmente, temos que,

$$G(t,t') = (t-t')e^{-\beta(t-t')}, \quad t > t'. \tag{17}$$

No caso de $t < t'$ o caminho de integração é representado na figura 2 e similarmente temos:

$$G(t,t') = 0, \quad t < t' \tag{18}$$

E, com a ajuda de θ -função encontramos a fórmula final:

$$G(t,t') = \theta(t-t')(t-t')e^{-\beta(t-t')} \tag{19}$$

c) $\omega_0 > \beta$

Neste caso temos:

$$\omega_1 = i\beta + \Omega \tag{20}$$

$$\omega_2 = i\beta - \Omega$$

$$\text{onde } \Omega = \sqrt{\omega_0^2 - \beta^2} \tag{21}$$

Para $t > t'$ o caminho de integração está representado na figura 4.

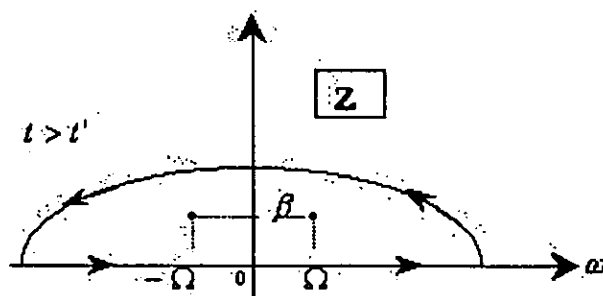


Figura 4: caminho de integração para $t > t'$ quando $\omega_0 > \beta$

Das fórmulas (12) e (13) temos que:

$$-\frac{1}{2\pi} \int_{-\infty}^{\infty} f(\omega) d\omega = e^{-\beta(t-t')} \frac{e^{\Omega(t-t')} - e^{-\Omega(t-t')}}{2i\Omega}$$

ou

$$G(t, t') = e^{-\beta(t-t')} \frac{\text{sen}\Omega t}{\Omega}, \quad t > t'. \tag{22}$$

Da mesma forma como nos casos a) e b) encontramos para $t < t'$:

$$G(t, t') = 0, \quad t < t' \tag{23}$$

finalmente:

$$G(t, t') = \theta(t - t') e^{-\beta(t-t')} \frac{\text{sen}\Omega(t - t')}{\Omega} \tag{24}$$

Então as fórmulas (16), (19) e (24) dão-nos as soluções da equação (2) para as diferentes escolhas dos parâmetros β e ω_0 .

III. OSCILADOR AMORTECIDO SOB ACÇÃO DE UMA FORÇA IMPULSIVA: $f(t) = f_0 \delta(t)$

Neste capítulo vamos considerar o comportamento do oscilador amortecido sob a acção de uma força impulsiva.

$$f(t) = f_0 \delta(t) \quad (25)$$

Esta expressão é a forma simbólica de uma força que se imprime no instante $t = 0$ e actua durante o intervalo τ , tendo $\tau \rightarrow 0$, (figura 5).

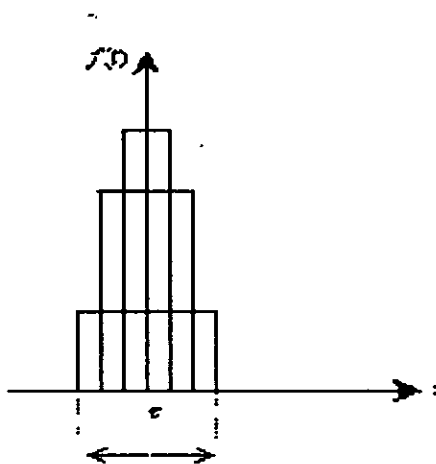


Figura 5: força actuando durante um instante de tempo τ com $\tau \rightarrow 0$

Este tipo de força chama-se força impulsiva. Destacamos que sem usar δ -função é preciso cada vez realizar o limite $\tau \rightarrow 0$ em cálculos concretos com esta força. Mas com a ajuda de δ -função as integrações necessárias são triviais. Realmente, usando a expressão (25) e fórmula (3) temos para solução particular da equação (1) a seguinte expressão:

$$x(t) = f_0 \int_{-\infty}^{\infty} G(t, t') \delta(t') dt' \quad (26)$$

O resultado da integração é:

$$x(t) = f_0 G(t, 0) \quad (27)$$

A teoria geral de equações diferenciais lineares diz que a solução geral de qualquer equação heterogênea é a soma da solução particular da equação heterogênea e da solução geral da equação homogênea. No nosso caso a equação homogênea é:

$$\frac{d^2 x_0}{dt^2} + 2\beta \frac{dx_0}{dt} + \omega_0^2 x_0 = 0 \quad (28)$$

A solução desta equação é bem conhecida e por isso vamos excluir esta solução da nossa análise, escolhendo:

$$x_0 = 0, \quad \frac{dx_0}{dt} = 0 \quad \text{para } t = 0. \quad (29)$$

Para solução particular (27), temos três possibilidades em dependência dos parâmetros β e ω_0 . Nós vamos investigar cada caso concreto para determinar os possíveis tipos de movimento do sistema. Além disso, vamos usar este exemplo simples para escrever um programa básico computacional, que será possível usar para os casos mais complicados.

As fórmulas (27) e (16) dão-nos a seguinte solução particular da equação (1):

$$x(t) = \frac{f_0}{\xi} e^{-\beta t} \operatorname{sh} \xi t, \quad \omega_0 < \beta, \quad t > 0. \quad (30)$$

Na figura (6), são representados os gráficos de movimento para duas escolhas do parâmetro β .

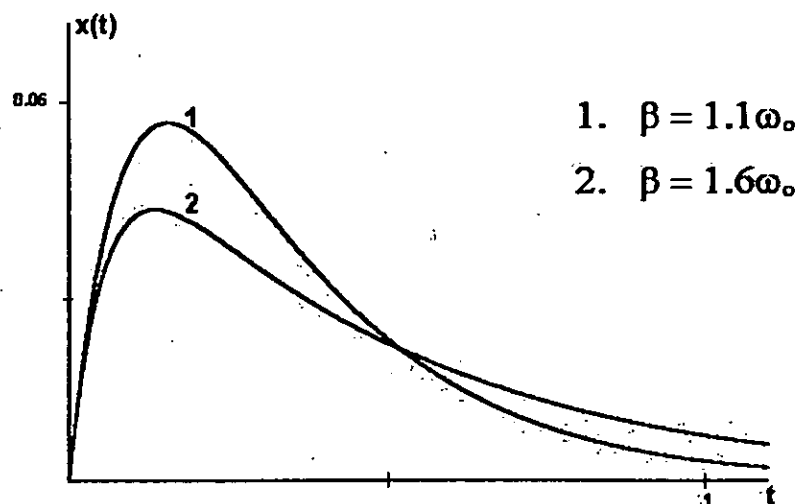


Figura 6: O gráfico mostra o movimento aperiódico para duas escolhas do parâmetro β

Para os cálculos numéricos são usados os seguintes valores:

$$T_0 = 1, \quad f_0 = 1, \quad \omega_0 = \frac{2\pi}{T_0}, \quad \xi = \sqrt{\beta^2 - \omega_0^2} \quad (31)$$

Esta escolha de parâmetros permite escrever um programa universal, que pode ser usado para vários sistemas físicos. No caso concreto é suficiente multiplicar

T_0 e f_0 por grandezas reais físicas.

O gráfico da figura (6), representa um movimento que se chama aperiódico.

As fórmulas (27) e (19) dão-nos:

$$x(t) = f_0 t e^{-\beta t}, \quad \omega_0 = \beta, \quad t > 0. \quad (32)$$

com o gráfico representado na figura 7.

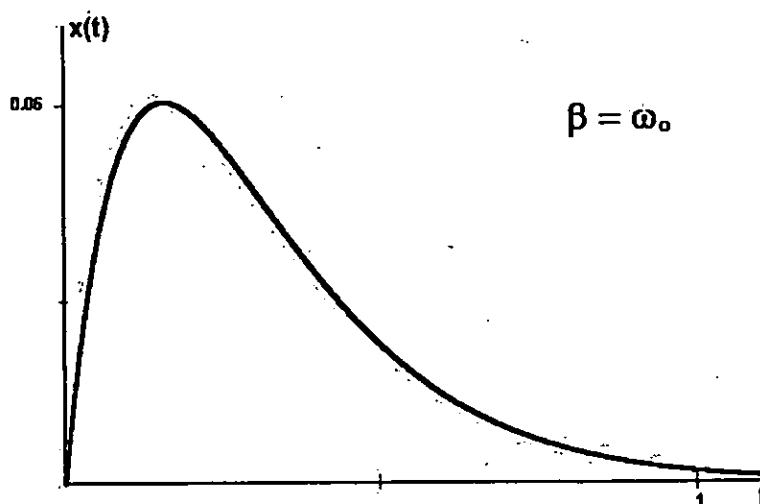


Figura 7: caso particular do movimento aperiódico $\beta = \omega_0$

É um caso particular do movimento aperiódico.

Finalmente, as fórmulas (27) e (24) dão-nos:

$$x(t) = \frac{f_0}{\Omega} e^{-\beta t} \text{sen} \Omega t, \quad \omega_0 > \beta, \quad t > 0. \quad (33)$$

Na figura 8, são representados os gráficos de movimento para duas escolhas do parâmetro β .

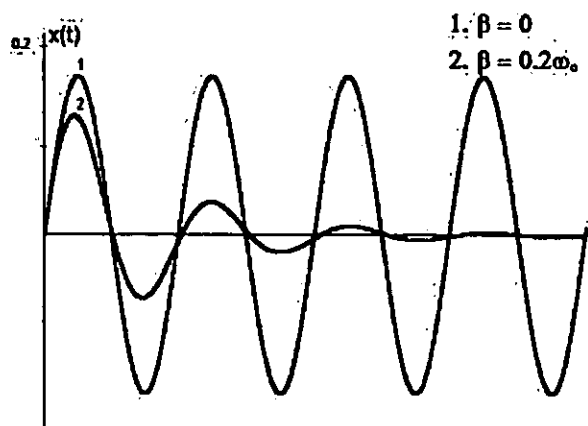


Figura 8:
 gráfico 1: Oscilações harmónicas livres
 gráfico 2: Oscilações amortecidas

O caso $\beta < \omega_0$, dá-nos um movimento do tipo das oscilações amortecidas, enquanto que, para $\beta = 0$, temos oscilações harmónicas livres.

Mais adiante nós vamos considerar os casos de várias forças externas no regime de $0 \leq \beta < \omega_0$.

No fim deste capítulo consideraremos a estrutura dum programa básico computacional. Este programa é de linguagem **Pascal 6.0** e se chama **OSCIL. PAS**. O algoritmo deste programa tem duas partes principais.

A primeira parte é o cálculo de valores de uma função dada $x(t)$ para várias grandezas do argumento t .

A segunda parte é a representação gráfica dos resultados numéricos no écran do monitor.

A realização da primeira parte é muito simples para as funções representadas pelas fórmulas (30), (32), (33), porque em **Pascal** existem procedimentos e funções matemáticas padronizadas. Então, ao escolher valores concretos para parâmetros, como, por exemplo, no (31), é muito simples calcular $x(t)$ para $t_{\min} \leq t \leq t_{\max}$.

Na segunda parte do algoritmo é preciso dar a relação entre as variáveis t e $x(t)$ e as posições delas no écran do monitor.

Pode-se descrever a posição de qualquer “pixel” no écran com a ajuda de números N_x e N_y , que dão as distâncias em “pixels” da esquerda e de cima do écran (figura 9).

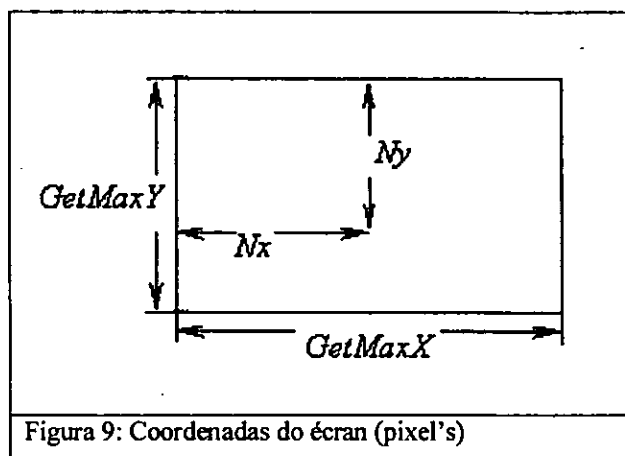


Figura 9: Coordenadas do ecrã (pixel's)

Em *Pascal* 6.0 o número total de "pixel's" na direcção horizontal dá a função *GetMaxX*, e na direcção vertical dá a função *GetMaxY*.

Para cálculos numéricos de qualquer função $x(t)$ é necessário indicar os limites de variação do argumento, no nosso caso estes limites são t_{min} e t_{max} . Conhecendo isso, é possível calcular o valor máximo e o valor mínimo para a função dada, isto é x_{max} e x_{min} . Graficamente esta situação está representada na figura 10.

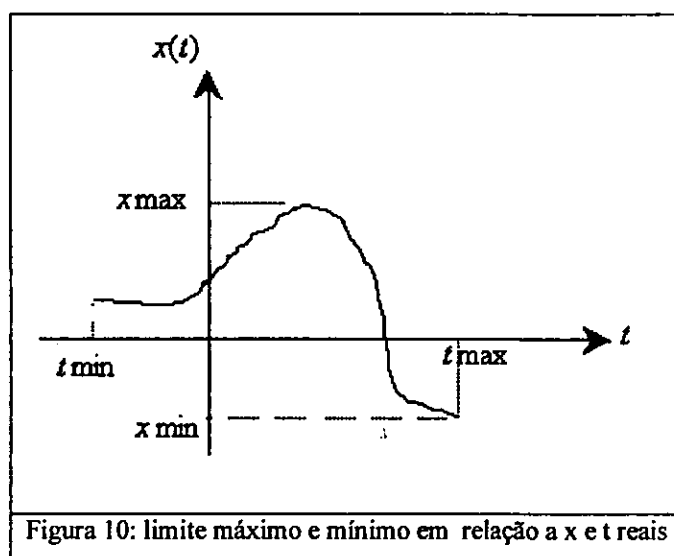
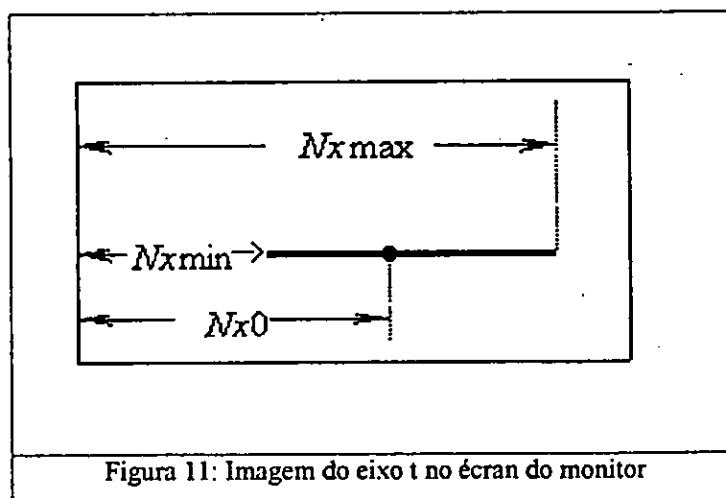


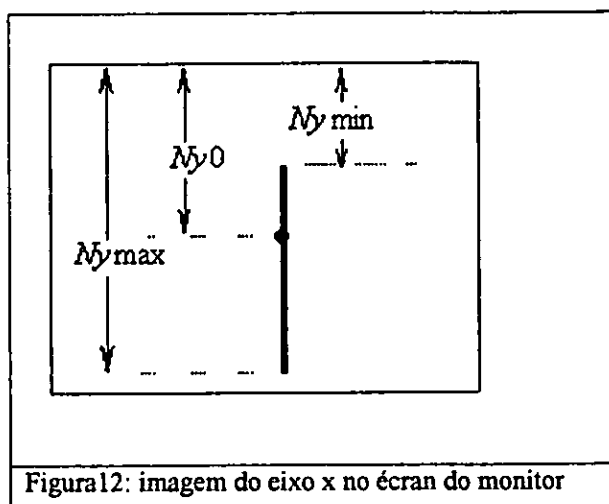
Figura 10: limite máximo e mínimo em relação a x e t reais

Nota-se que a origem de coordenadas coincide com o ponto $t = 0, x = 0$.

A imagem do eixo t no ecrã do monitor tem a forma representada na figura 11.



A imagem do eixo x no écran do monitor tem a forma representada na figura 12.



Então, para uma representação gráfica de qualquer função é preciso determinar os seguintes parâmetros:

$$t_{min}, \quad t_{max}, \quad x_{min}, \quad x_{max}$$

(34)

$$N_{xmin}, \quad N_{xmax}, \quad N_{ymin}, \quad N_{ymax}$$

Conhecendo estes parâmetros e a posição do centro do sistema de coordenadas ($t = 0, x = 0$), temos:

$$Nx0 = \frac{Nx \text{ min} * t \text{ max} - Nx \text{ max} * t \text{ min}}{t \text{ max} - t \text{ min}} \quad (35)$$

$$Ny0 = \frac{Ny \text{ max} * x \text{ max} - Ny \text{ min} * x \text{ min}}{x \text{ max} - x \text{ min}}$$

As grandezas $Nx0$, $Ny0$ dão-nos a posição do centro do sistema de coordenadas no écran do monitor. Na realização do algoritmo concreto, podem ter lugar duas possibilidades. A primeira é o cálculo de coordenadas de écran Nx e Ny de qualquer ponto determinado pelas coordenadas t e x (figura 13)

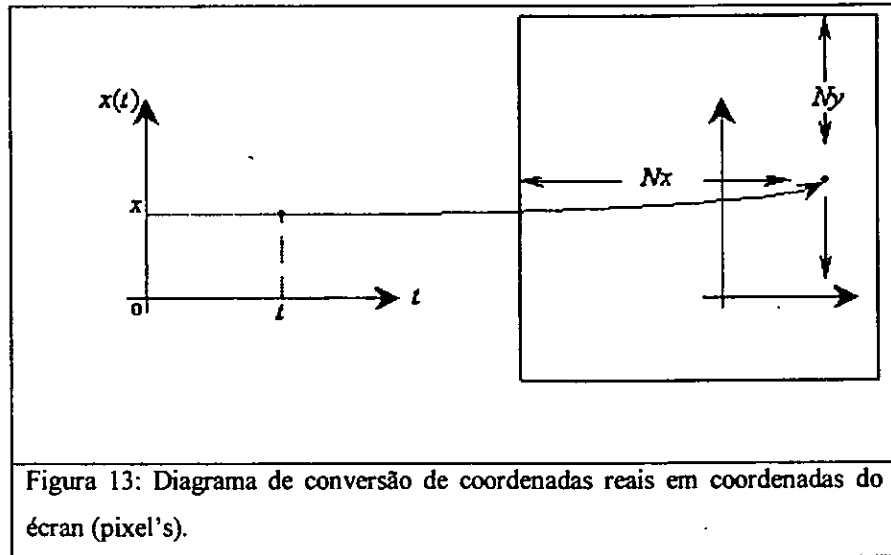


Figura 13: Diagrama de conversão de coordenadas reais em coordenadas do écran (pixel's).

Neste caso é preciso usar as seguintes fórmulas:

$$Nx = Nx0 + \frac{(Nx \text{ max} - Nx \text{ min})}{(t \text{ max} - t \text{ min})} * t \quad (36)$$

$$Ny = Ny0 - \frac{(Ny \text{ max} - Ny \text{ min})}{(x \text{ max} - x \text{ min})} * x$$

A segunda possibilidade é a transformação inversa. Conhecendo a posição de qualquer ponto no écran, isto é sabendo Nx , Ny , podemos determinar as variáveis t e x , (figura 14).

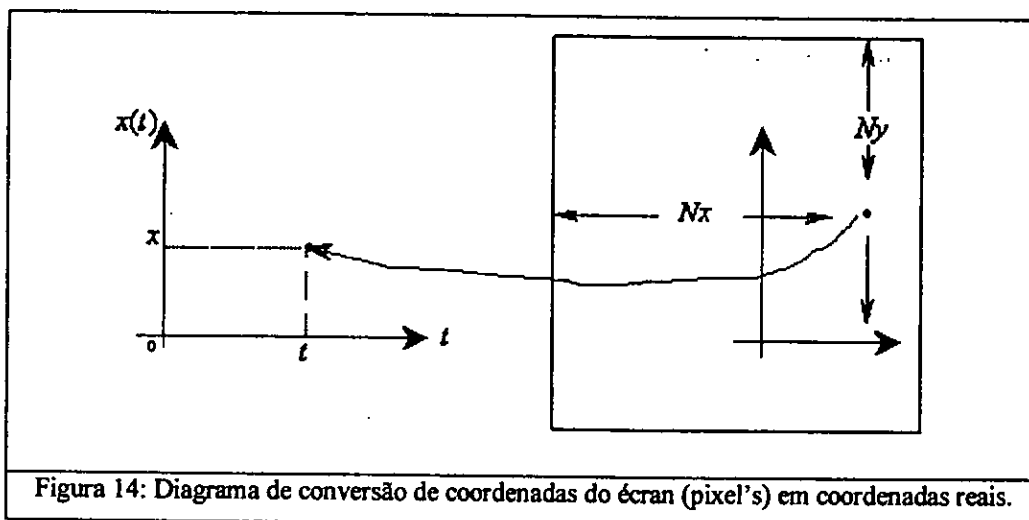


Figura 14: Diagrama de conversão de coordenadas do écran (pixel's) em coordenadas reais.

As fórmulas necessárias são:

$$t = \frac{(t \text{ max} - t \text{ min})}{(Nx \text{ max} - Nx \text{ min})} * (Nx - Nx0) \quad (37)$$

$$x = \frac{(x \text{ max} - x \text{ min})}{(Ny \text{ max} - Ny \text{ min})} * (Ny0 - Ny)$$

Quanto às fórmulas (35), (36) notamos que não estão indicados os procedimentos necessários em *Pascal* para transformar grandezas do tipo “real” em grandezas do tipo “integer”.

No fim deste capítulo consideramos como exemplo concreto, a função (32):

$$x(t) = f_0 t e^{-\beta t}, \quad \beta = \omega_0$$

Usando a linguagem *Pascal* 6.0 podemos escolher:

$$T_0 := 1;$$

$$f_0 := 1;$$

$$\omega_0 := \frac{2\pi}{T_0};$$

$$\beta := \omega_0;$$

$$x \text{ max} := 0.4;$$

$$x \text{ min} := 0;$$

$$t \text{ max} := T_0;$$

$$t \text{ min} := 0;$$

$$\begin{aligned} N_{x \min} &= \text{GetMaxX} \div 10, \\ N_{x \max} &= \text{GetMaxX} - N_{x \min}, \\ N_{y \min} &= \text{GetMaxY} \div 10, \\ N_{y \max} &= \text{GetMaxY} - N_{y \min}, \end{aligned} \quad (38)$$

As designações ω_0 , β são simbólicas.

Todo o programa *OSCIL.PAS* está representado no Apêndice II.

IV. OSCILADOR AMORTECIDO SOB ACÇÃO DE UMA FORÇA IMPULSIVA PERIÓDICA

Neste capítulo consideramos o caso da força impulsiva periódica:

$$f(t) = f_0 \sum_{k=0}^{\infty} \delta(t - t_k) \quad (39)$$

onde $t_k = k\tau$ para $k = 0, 1, 2, 3, \dots$

isso significa que dentro de cada intervalo de tempo τ , actua uma força impulsiva sobre o sistema o que é representado na figura 15.

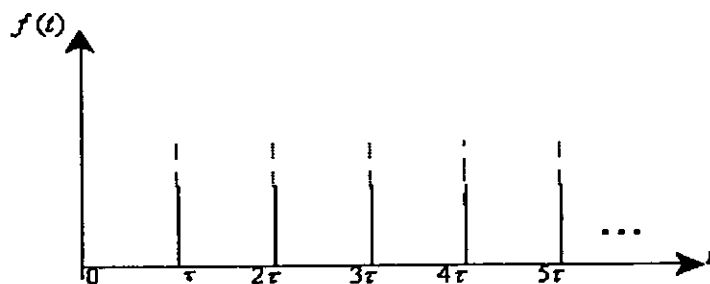


Figura 15: Acção da força impulsiva periódica sobre o sistema

Destacamos, mais uma vez, que a utilização da função generalizada (δ -função) dá uma expressão muito simples para a força, o contrário da matemática clássica, onde é necessário usar, para este caso, a operação do limite. (veja a figura (5)).

Consideremos o regime oscilante ($0 \leq \beta < \omega_0$) e, usando as fórmulas (3) e (24) temos:

$$x(t) = \frac{f_0}{\Omega} \int_{-\infty}^{\infty} \theta(t - t') e^{-\beta(t-t')} \text{sen} \Omega(t - t') \sum_{k=0}^{\infty} \delta(t' - k\tau) dt' \quad (40)$$

o que dá:

$$x(t) = \frac{f_0}{\Omega} \sum_{k=0}^{\infty} \theta(t - k\tau) e^{-\beta(t-k\tau)} \text{sen} \Omega(t - k\tau) \quad (41)$$

Do ponto de vista de programação, esta expressão é muito simples e é interessante compará-la com a expressão obtida com a ajuda de outros métodos matemáticos.

Um dos métodos muito forte é o método operacional na solução de equações diferenciais lineares que usa as transformações de Laplace.

O caso da força (39), foi considerado no livro [12] para um oscilador sem atrito ($\beta=0$).

O resultado é seguinte:

$$x(t) = \frac{f_0}{\omega_0} \left\{ \frac{1}{\omega_0 \tau} - \frac{\cos \omega_0 \left(t + \frac{\tau}{2} \right)}{2 \operatorname{sen} \frac{\omega_0 \tau}{2}} + 2 \omega_0 \tau \sum_{k=1}^{\infty} \frac{\cos \left(\frac{2\pi k t}{\tau} \right)}{(\omega_0 \tau)^2 - (2\pi k)^2} \right\} \quad (42)$$

A nossa expressão (41), para $\beta=0$, tem a forma seguinte:

$$x(t) = \frac{f_0}{\omega_0} \sum_{k=0}^{\infty} \theta(t - k\tau) \operatorname{sen} \omega_0(t - k\tau) \quad (43)$$

É claro que a expressão (43), é muito mais simples que (42) do ponto de vista de programação. Então podemos concluir que o método da função de Green é mais eficaz que o método operacional no caso considerado.

No fim desta comparação vamos demonstrar como é possível obter a expressão (42) a partir da (43). Isto é um exemplo da utilização de funções generalizadas nos cálculos concretos. Então é preciso calcular a soma:

$$S(t) = \sum_{k=0}^{\infty} \theta(t - k\tau) \operatorname{sen}(\omega_0 t - k\omega_0 \tau) \quad (44)$$

Para $t > 0$.

Usando a fórmula $\theta(x) = 1 - \theta(-x)$ (veja Apêndice I), vamos escrever (44) na forma:

$$S(t) = \sum_{k=0}^{\infty} \operatorname{sen}(\omega_0 t - k\omega_0 \tau) - \sum_{k=0}^{\infty} \theta(k\tau - t) \operatorname{sen}(\omega_0 t - k\omega_0 \tau) \quad (45)$$

Depois:

$$\begin{aligned} \sum_{k=0}^{\infty} \operatorname{sen}(\omega_0 t - k\omega_0 \tau) &= \operatorname{sen} \omega_0 t + \sum_{k=1}^{\infty} \operatorname{sen}(\omega_0 t - k\omega_0 \tau) \\ \sum_{k=0}^{\infty} \operatorname{sen}(\omega_0 t - k\omega_0 \tau) &= \operatorname{sen} \omega_0 t + \left(\sum_{k=1}^{\infty} \cos k\omega_0 \tau \right) \operatorname{sen} \omega_0 t - \left(\sum_{k=1}^{\infty} \operatorname{sen} k\omega_0 \tau \right) \cos \omega_0 t \end{aligned} \quad (46)$$

Para calcular as somas obtidas vamos usar as fórmulas conhecidas (veja Apêndice I).

$$\begin{aligned} \sum_{k=1}^{\infty} \operatorname{sen} kx &= \frac{1}{2} \operatorname{ctg} \frac{x}{2} \\ \sum_{k=1}^{\infty} \cos kx &= -\frac{1}{2} + \pi \sum_{n=-\infty}^{\infty} \delta(x - 2\pi n) \end{aligned} \quad (47)$$

Então:

$$\sum_{k=1}^{\infty} \text{sen} k\omega_0\tau = \frac{1}{2} \text{ctg} \frac{\omega_0\tau}{2} \quad (48)$$

$$\sum_{k=1}^{\infty} \text{cos} k\omega_0\tau = -\frac{1}{2} \quad \text{para } \tau \neq \frac{2\pi}{\omega_0} n$$

Finalmente, podemos representar (46) na forma:

$$\sum_{k=0}^{\infty} \text{sen}(\omega_0 t - k\omega_0\tau) = \frac{\text{sen}\omega_0 t}{2} - \frac{1}{2} \text{ctg} \frac{\omega_0\tau}{2} \text{cos}\omega_0 t \quad (49)$$

Para calcular a segunda soma na fórmula (45) vamos usar o método de adição de Poisson (veja Apêndice I). As fórmulas básicas são as seguintes:

$$\sum_{n=-\infty}^{\infty} \varphi(n) = \sum_{k=-\infty}^{\infty} \tilde{\varphi}(2\pi k) \quad (50)$$

$$\text{onde } \tilde{\varphi}(2\pi k) = \int_{-\infty}^{\infty} \varphi(x) e^{2\pi i k x} dx,$$

para qualquer função $\varphi(x)$.

Para usar esta fórmula notamos que $\theta(n\tau - t) = 0$ para $t > 0$ $n < 0$ e por isso temos:

$$\sum_{n=0}^{\infty} \theta(n\tau - t) \text{sen}(\omega_0 t - n\omega_0\tau) = \sum_{n=-\infty}^{\infty} \theta(n\tau - t) \text{sen}(\omega_0 t - n\omega_0\tau) \quad (51)$$

Neste caso podemos escolher na fórmula (50),

$$\varphi(n) = \theta(n\tau - t) \text{sen}(\omega_0 t - n\omega_0\tau) \quad (52)$$

e

$$\tilde{\varphi}(2\pi k) = \int_{-\infty}^{\infty} \theta(x\tau - t) \text{sen}(\omega_0 t - x\omega_0\tau) e^{2\pi i k x} dx \quad (53)$$

A existência de $\theta(x\tau - t)$ dá:

$$\tilde{\varphi}(2\pi k) = \int_{t/\tau}^{\infty} \text{sen}(\omega_0 t - x\omega_0\tau) e^{2\pi i k x} dx \quad (54)$$

Usando $x = y + t/\tau$ temos:

$$\tilde{\varphi}(2\pi k) = -e^{2\pi i k t/\tau} \int_0^{\infty} \text{sen}(\omega_0 \tau y) e^{2\pi i k y} dy \quad (55)$$

Para determinar o integral obtido no limite superior de integração vamos representá-lo na forma seguinte:

$$\int_0^{\infty} \text{sen}(\omega_0 \tau y) e^{2\pi k y} dy = \lim_{\epsilon \rightarrow 0} \int_0^{\infty} \text{sen}(\omega_0 \tau y) e^{-\epsilon y} e^{2\pi k y} dy \quad (56)$$

Depois disso, a integração é trivial e temos:

$$\tilde{\varphi}(2\pi k) = \frac{\omega_0 \tau}{(2\pi k)^2 - (\omega_0 \tau)^2} e^{2\pi k t/\tau} \quad (57)$$

Finalmente as fórmulas (50), (52) e (57) dão-nos:

$$\sum_{n=-\infty}^{\infty} \theta(n\tau - t) \text{sen}(\omega_0 t - n\omega_0 \tau) = \omega_0 \tau \sum_{k=-\infty}^{\infty} \frac{e^{2\pi k t/\tau}}{(2\pi k)^2 - (\omega_0 \tau)^2} \quad (58)$$

$$\sum_{n=0}^{\infty} \theta(n\tau - t) \text{sen}(\omega_0 t - n\omega_0 \tau) = -\frac{1}{\omega_0 \tau} + 2\omega_0 \tau \sum_{k=1}^{\infty} \frac{\cos(2\pi k t/\tau)}{(2\pi k)^2 - (\omega_0 \tau)^2} \quad (59)$$

Usando (45), (49) e (59) temos para S(t):

$$S(t) = \frac{1}{\omega_0 \tau} + \frac{\text{sen} \omega_0 t}{2} - \frac{1}{2} \text{ctg} \frac{\omega_0 \tau}{2} \cos \omega_0 t - 2\omega_0 \tau \sum_{k=1}^{\infty} \frac{\cos(2\pi k t/\tau)}{(2\pi k)^2 - (\omega_0 \tau)^2} \quad (60)$$

É muito simples transformar esta fórmula na forma seguinte:

$$S(t) = \frac{1}{\omega_0 \tau} - \frac{\cos \omega_0 (t + \tau/2)}{2 \text{sen} \frac{\omega_0 \tau}{2}} + 2\omega_0 \tau \sum_{k=1}^{\infty} \frac{\cos(2\pi k t/\tau)}{(\omega_0 \tau)^2 - (2\pi k)^2} \quad (61)$$

Sabendo que $x(t) = \frac{f_0}{\omega_0} S(t)$ temos para x(t) a expressão (42).

O método operacional também torna possível a obtenção duma fórmula para x(t) no caso de $\beta \neq 0$. Os cálculos, neste caso, são bastante complicados e, por isso, escrevemos só a fórmula final:

$$x(t) = f_0 \left\{ \frac{1}{\omega_0^2 \tau} + \frac{e^{-\beta t}}{2\Omega} \left[\text{sen} \Omega t - \frac{\text{sh} \beta \tau * \text{sen} \Omega t + \text{sen} \Omega \tau * \cos \Omega t}{\text{ch} \beta \tau - \cos \Omega \tau} \right] + \right. \\ \left. + \frac{2}{\tau} \sum_{k=1}^{\infty} \frac{(\omega_0^2 - \lambda_k^2) \cos \lambda_k t + 2\beta \lambda_k \text{sen} \lambda_k t}{(\omega_0^2 - \lambda_k^2)^2 + 4\beta^2 \lambda_k^2} \right\} \quad (62)$$

$$\text{onde } \lambda_k = \frac{2\pi k}{\tau}, \quad \Omega = \sqrt{\omega_0^2 - \beta^2}$$

Destacamos que da aplicação do método da função de Green resulta a expressão muito mais simples:

$$x(t) = \frac{f_0}{\Omega} \sum_{k=0}^{\infty} \theta(t - k\tau) \text{sen}\Omega(t - k\tau) e^{-\beta(t-k\tau)}$$

O que é a fórmula (41).

Esta fórmula é básica para investigar o comportamento do oscilador amortecido em vários casos particulares da escolha dos parâmetros. Para isso foi usado o programa **DIPLOMA** (veja o Apêndice II).

Este programa foi escrito na **Linguagem Delphi 4**, porque, por um lado, esta tem como base a linguagem **Pascal**. Por outro lado, Delphi 4 possibilita a utilização de todos os métodos dos sistemas Windows 95 e Windows 98. Os resultados do funcionamento deste programa são representados nas figuras 16-19 onde, como exemplos, são consideradas as variações dos parâmetros τ e β .

Na figura 16 é representado o gráfico do movimento para:

$$\tau = 5 * T, \quad \beta = 0,05 * \omega_0 \tag{63}$$

$$\text{onde } T = \frac{2\pi}{\Omega}, \quad \Omega = \sqrt{\omega_0^2 - \beta^2}$$

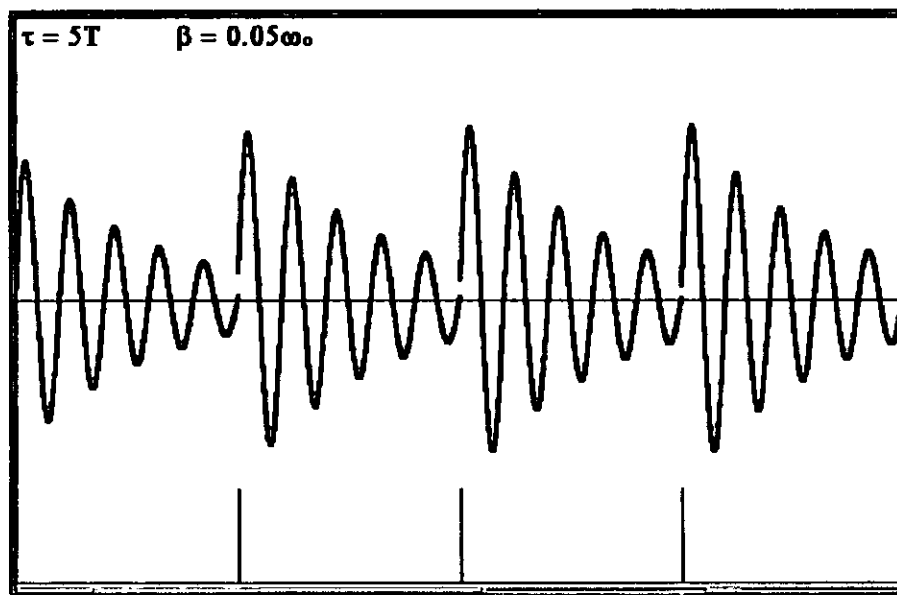


Figura 16

As rectas verticais representam os momentos de acção da força impulsiva periódica. Obviamente, depois de cada acção da força impulsiva temos oscilações livres amortecidas.

Na figura 17 temos o comportamento do sistema no caso de:

$$\tau = 1,17T, \quad \beta = 0 \tag{64}$$

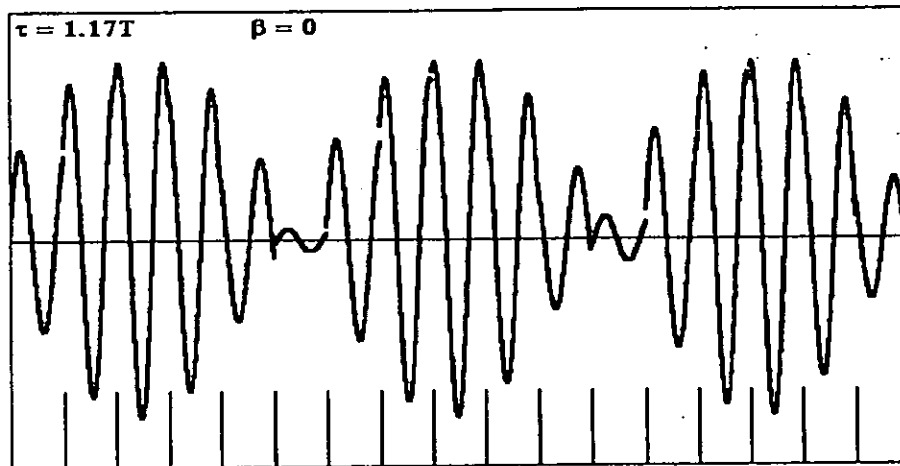


Figura 17

São batimentos porque o valor de τ se aproxima ao período T . Na próxima figura 18 temos:

$$\tau = T, \quad \beta = 0,015\omega_0 \quad (65)$$

e dá-se a ressonância:

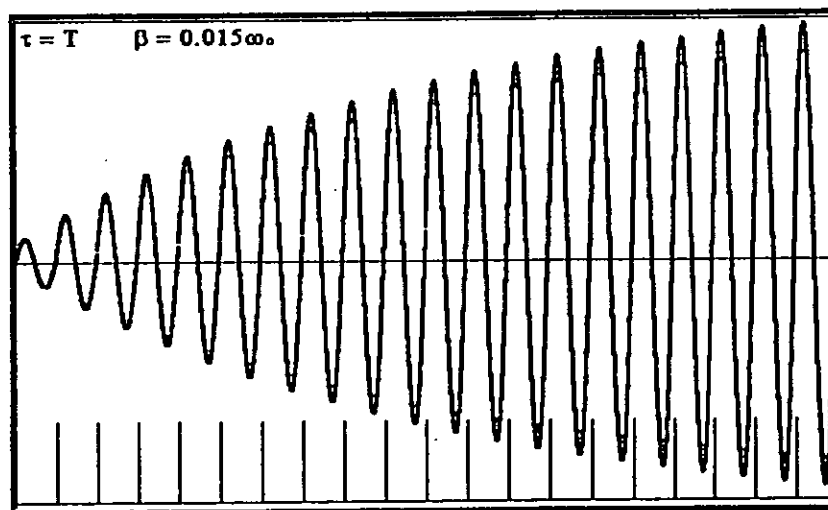


Figura 18

E na última figura 19 podemos ver o comportamento do sistema para

$$\tau = 0,83T, \quad \beta = 0,02\omega_0 \quad (66)$$

o que, também, dá os batimentos, mas com a variação menor da amplitude.

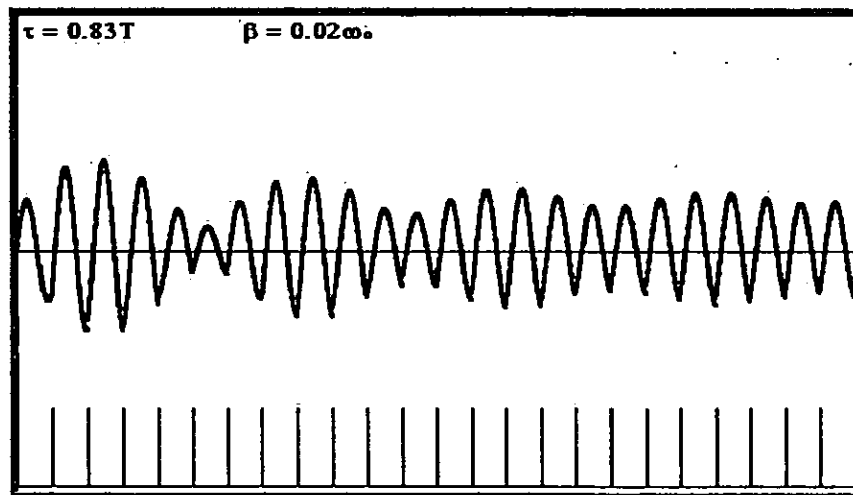


Figura 19.

Figuras 16,17,18 e 19, mostram o comportamento do oscilador sob acção da força periódica para 4 escolhas dos parâmetros τ e β

Exemplos considerados demonstram que o comportamento do oscilador amortecido sob a acção duma força impulsiva periódica é muito semelhante ao comportamento deste oscilador sob a acção duma força harmónica.

São exemplos de acção duma força determinada, mas é possível usar os resultados obtidos pelo método da função de Green para a modelação computacional do comportamento do oscilador amortecido sob a acção das forças aleatórias. Neste caso existem várias possibilidades, por exemplo, na expressão (39):

$$f(t) = f_0 \sum_{k=0}^{\infty} \delta(t - t_k)$$

t_k é igual a $k\tau$, mas f_0 pode ser uma grandeza aleatória. A figura 20 mostra um exemplo da força deste tipo.

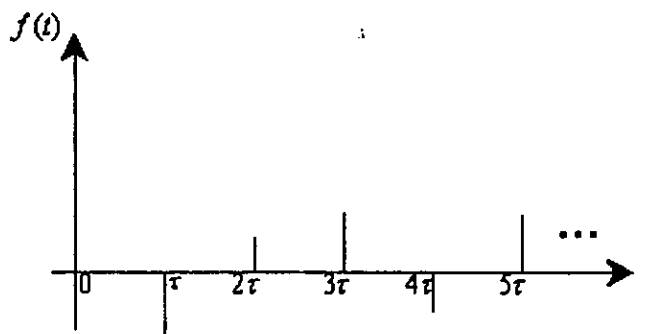


Figura 20: Acção da força impulsiva periódica com f_0 aleatória em que $t_k = k\tau$

Denominamos esta força como força impulsiva periódica aleatória.

Outra possibilidade é a seguinte: $f_0 = \text{const}$, mas os instantes t_k são grandezas aleatórias. É possível também, considerar uma combinação destes casos.

No programa **DIPLOMA** (veja Apêndice II), é realizado o caso de t_k aleatórios mais detalhadamente, nas fórmulas (39) e (41) como se pode ver a seguir:

$$f(t) = f_0 \sum_{k=0}^{\infty} \delta(t - t_k)$$

$$x(t) = \frac{f_0}{\Omega} \sum_{k=0}^{\infty} \theta(t - t_k) \text{sen} \Omega(t - t_k) e^{-\beta(t - t_k)}$$

t_k são grandezas aleatórias, que estão situadas dentro dos intervalos do tempo

$$k\tau < t_k < (k+1)\tau, \quad k = 0, 1, 2, 3, \dots \quad (67)$$

A figura 21 mostra um exemplo desta força.

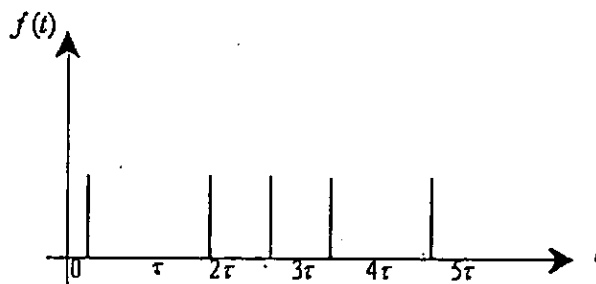


Figura 21: Acção da força impulsiva periódica com t_k aleatórias para $f_0 = \text{const}$ dentro dum intervalo $k\tau < t_k < (k+1)\tau$ para $k=1,2,3,\dots$

Denominamos esta força como a força impulsiva estocástica quasi-periódica.

A modelação deste processo é uma operação muito simples em *Pascal*. Para isso é preciso usar o procedimento **RANDOM** que liga o gerador de números quasi-aleatórios. Os resultados da modelação computacional são representados nas figuras 22 a 25.

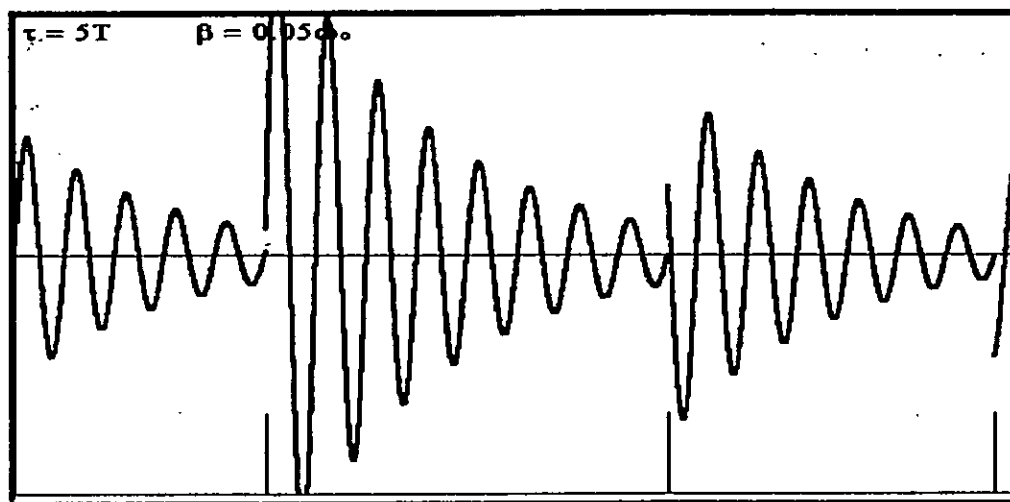


Figura 22

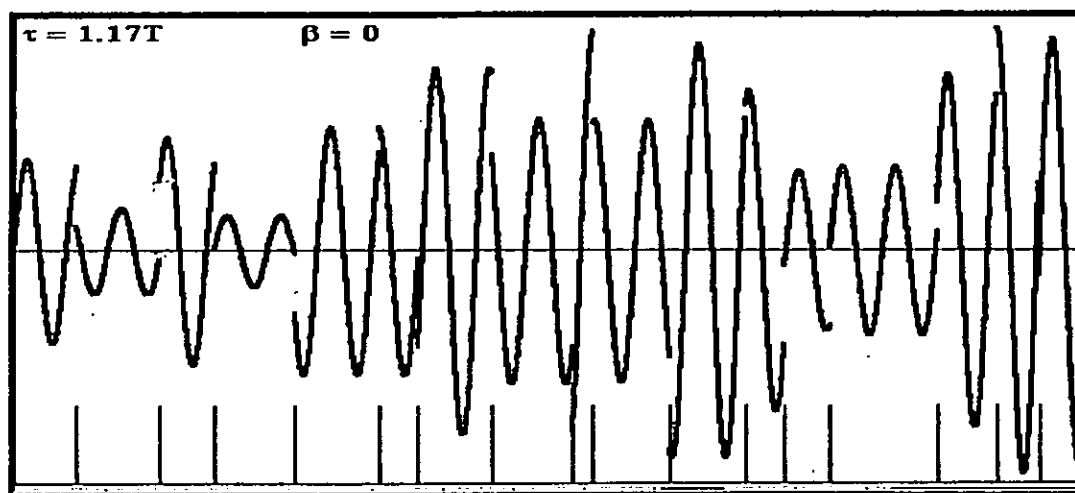


Figura 23

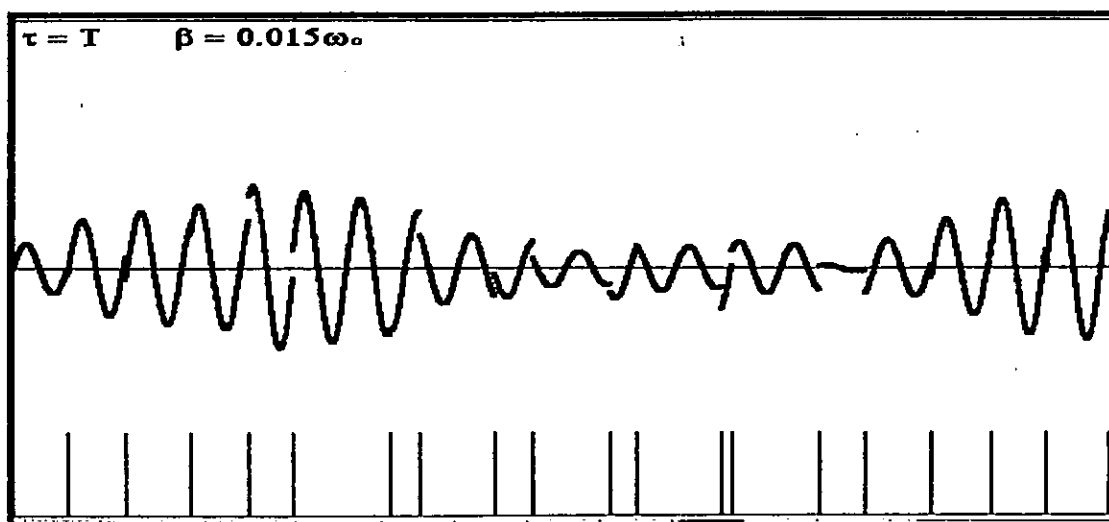


Figura 24

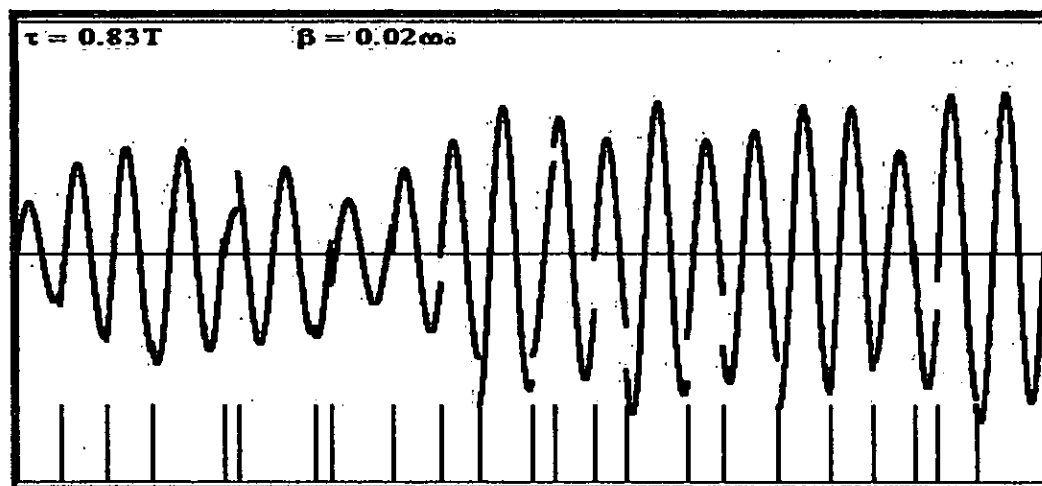


Figura 25

Figuras 22,23,24 e 25, mostram os resultados da modelação computacional no caso de forças aleatórias.

Foram escolhidos os mesmos valores dos parâmetros τ e β : (63), (64), (65), (66), que consideramos no caso da força impulsiva periódica. Isso possibilita a comparação do comportamento do oscilador naqueles casos.

Destacamos que não é o objectivo deste trabalho fazer uma análise detalhada dos resultados da modelação computacional do comportamento do oscilador, por isso indicamos, somente, que sob a acção da força impulsiva estocástica as oscilações podem ter grande aumento das amplitudes, em comparação com o caso da força impulsiva periódica.

V. CONCLUSÕES

O objectivo deste trabalho foi demonstrar uma aplicação possível do método da função de Green à modelação computacional.

Como exemplo foi escolhido o sistema muito importante em física, o oscilador amortecido forçado.

O trabalho considerou uma força impulsiva representada através de δ -função de Dirac, para demonstrar a importância da utilização das funções generalizadas.

A solução analítica (41) deste problema tem a forma muito mais simples que a solução obtida pelo método operacional (62).

Usando a fórmula (41), foram escritos programas computacionais (veja Apêndice II).

Finalmente, a modelação computacional foi aplicada à análise do caso da força impulsiva estocástica quasi- periódica.

Destacamos que a fórmula (41) obtida pelo método da função de Green, torna a modelação computacional com grandezas aleatórias num problema muito simples.

Concluimos que o método da função de Green a par das funções generalizadas, pode ser aplicado à modelação computacional de vários sistemas físicos para os quais é possível procurar a expressão analítica usando a função de Green.

VI. RECOMENDAÇÕES

1. Realizar a modelação computacional do oscilador amortecido forçado (41), com grandezas f_0 , ω_0 , β aleatórios, que por exemplo, pode ter uma aplicação na análise do comportamento dos circuitos eléctricos oscilantes.
2. Realizar a modelação computacional do oscilador amortecido sob acção da força:

$$f(t) = f_0 \sum_{k=0}^{\infty} \theta(t - t_k) \theta(t_k + b - t) \quad (68)$$

onde $t_k = k * \tau$ e $b < \tau$ (veja a figura 26), e investigar também, o caso de f_0, t_k, b aleatórios.

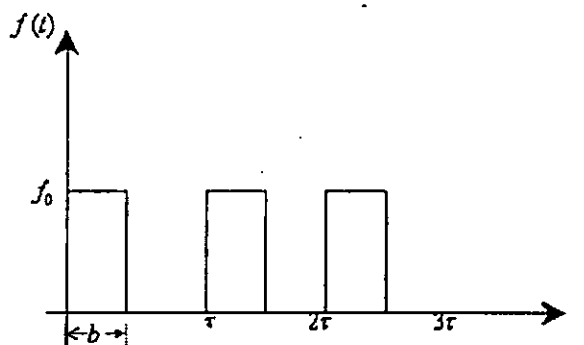


Figura 26: Acção da força $f(t) = f_0 \sum_{k=0}^{\infty} \theta(t - t_k) \theta(t_k + b - t)$
no caso de f_0, t_k e b aleatórios.

3. Explorar a possibilidade de usar, no ensino, as noções básicas das funções generalizadas e o método da função de Green, para solução de vários problemas de Mecânica Geral, Electricidade e Magnetismo, Mecânica Analítica e Electrodinâmica Clássica.

VII. BIBLIOGRAFIA

1. Tjonov A.N., Samarsky A.A. Ecuaciones de la Física Matemática, 2ª edición, Mir, 1980.
2. Pinsky M.A. Partial Differential Equations and Boundary- Value Problems with Applications, Second Edition, McGraw-Hill, 1991.
3. Fradkin E.S. O Método da Função de Green na Teoria Quântica do Campo, Tr. Fian, 1965, 29.
4. Bonch – Bruevich V.L., Tiablikov S. V. O Método da Função de Green na Mecânica Estatística, M., Fismatgiz, 1961.
5. Landau L., Lifshitz E. Mecânica Quântica, T1,2. mir, Moscovo, 1985.
6. Fock V.A. Princípios de Mecânica Quântica, Mir, Moscovo, 1986.
7. Shvarz L. Métodos Matemáticos para Ciências Físicas, Mir, M. 1965.
8. Gelfand I. M., Shilov G. E. Funções Generalizadas e Operações com elas, Fismatgiz, M., 1959.
9. Bremerman G. Distribuições, Variáveis complexas e Transformações de Fourier, Mir, M., 1968.
10. Vladimirov V.S. Funções Generalizadas na Física Matemática, Nauka, M., 1976.
11. Mukusinski A., Sikorski R. Teoria Elementar de Funções Generalizadas, 1,2. M., IL, 1965.
12. Lavrentiev M. A., Shabat B.V. Métodos da Teoria de Funções de Variável Complexa, M., Nauka, 1965.
13. Landau L., Lifshitz E. Física Estatística, M., Nauka, 1976.
14. Marchenko A. I. Object *Pascal* 2.0, Kiev, Junior, 1998.
15. Boas R., Verwij M., Gunter H. Das Delphi Buch, Sybex, Dusseldorf, 1999.

VIII. APÊNDICES

APÊNDICE I

Neste apêndice consideramos as definições e propriedades de algumas funções generalizadas e, também, obtemos as fórmulas usadas nos capítulos II-IV.

I. a. θ função de Heviside

A função

$$\theta(x) = \begin{cases} 0 & \text{para } x < 0 \\ 1 & \text{para } x > 0 \end{cases} \quad (\text{A1})$$

chama-se θ -função de Heviside.

Para um deslocamento do argumento temos:

$$\theta(x - x_0) = \begin{cases} 0 & \text{para } x < x_0 \\ 1 & \text{para } x > x_0 \end{cases} \quad (\text{A2})$$

uma consequência óbvia desta fórmula é:

$$\theta(x - x_0) = 1 - \theta(x_0 - x) \quad (\text{A3})$$

I. b. δ -função de Dirac

A definição intuitiva de δ -função é:

$$\delta(x) = \begin{cases} 0 & \text{para } x \neq 0 \\ \infty & \text{para } x = 0 \end{cases} \quad (\text{A4})$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

Para um deslocamento do argumento temos:

$$\delta(x - x_0) = \begin{cases} 0 & \text{para } x \neq x_0 \\ \infty & \text{para } x = x_0 \end{cases} \quad (\text{A5})$$

$$\int_{\alpha}^{\beta} \delta(x - x_0) dx = \begin{cases} 1 & \text{para } x_0 \in [\alpha, \beta] \\ 0 & \text{para } x_0 \notin [\alpha, \beta] \end{cases}$$

A definição rigorosa da δ -função é:

$$\int_{-\infty}^{\infty} f(x) \delta(x - x_0) dx = f(x_0) \quad (\text{A6})$$

onde $f(x)$ é qualquer função usual. Por causa das propriedades (A4), (A5), (A6), a δ -função chama-se a função generalizada.

É simples provar as seguintes fórmulas úteis:

$$\delta(ax) = \frac{\delta(x)}{|a|} \quad (\text{A7})$$

onde $a = \text{const}$,

$$\delta(\varphi(x)) = \sum_i \frac{\delta(x-x_i)}{|\varphi'(x_i)|}, \quad \varphi(x_i) = 0 \quad (\text{A8})$$

onde $\varphi(x)$ é uma função usual. A definição da derivada da δ -função é:

$$\int_{-\infty}^{\infty} f(x)\delta'(x)dx = -\int_{-\infty}^{\infty} f'(x)\delta(x)dx \quad (\text{A9})$$

esta definição serve, também, para qualquer função generalizada, por exemplo, consideramos,

$$\int_{-\infty}^{\infty} f(x)\theta'(x)dx = -\int_{-\infty}^{\infty} f'(x)\theta(x)dx = -\int_0^{\infty} f'(x)dx = -f(x)\Big|_0^{\infty} = f(0),$$

$$\text{onde } f(\infty) = f(-\infty) = 0$$

mas $f(0) = \int_{-\infty}^{\infty} f(x)\delta(x)dx$, de onde temos:

$$\int_{-\infty}^{\infty} f(x)\theta'(x)dx = \int_{-\infty}^{\infty} f(x)\delta(x)dx \quad (\text{A10})$$

para $f(x)$ arbitrária. Isso significa que,

$$\delta(x) = \frac{d\theta(x)}{dx} \quad (\text{A11})$$

Existem muitas representações de δ -função. Nós consideramos dois exemplos básicos. O primeiro exemplo é a representação integral

$$\delta(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} dk \quad (\text{A12})$$

O segundo exemplo é a representação através da tal chamada, δ -sucessão: $\delta(x, \varepsilon)$.

$$\delta(x) = \lim_{\varepsilon \rightarrow 0} \delta(x, \varepsilon) \quad (\text{A13})$$

Existem muitas fórmulas concretas para $\delta(x, \varepsilon)$, uma delas é:

$$\delta(x, \varepsilon) = \frac{1}{\pi} \frac{\varepsilon}{x^2 + \varepsilon^2} \quad (\text{A14})$$

Nota-se que a passagem ao limite $\varepsilon \rightarrow 0$ deve ser realizada depois da integração da $\delta(x, \varepsilon)$, com qualquer função $f(x)$ para ter,

$$\lim_{\varepsilon \rightarrow 0} \int_{-\infty}^{\infty} f(x) \delta(x, \varepsilon) dx = f(0) \quad (A15)$$

I.c. O Método do Factor de Corte

Consideramos a expressão (A12):

$$\delta(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} dk$$

No sentido usual este integral não existe, mas é possível calculá-lo com a ajuda do método do factor de corte. Para isso faz-se a seguinte transformação:

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} dk = \frac{1}{2\pi} \int_0^{\infty} (e^{ikx} + e^{-ikx}) dk \quad (A16)$$

Agora introduz-se o factor de corte $e^{-\varepsilon k}$:

$$\frac{1}{2\pi} \int_0^{\infty} (e^{ikx} + e^{-ikx}) dk = \lim_{\varepsilon \rightarrow 0} \int_0^{\infty} (e^{ikx} + e^{-ikx}) e^{-\varepsilon k} dk. \quad (A17)$$

Neste caso a integração é trivial e temos:

$$\lim_{\varepsilon \rightarrow 0} \int_0^{\infty} (e^{ikx} + e^{-ikx}) e^{-\varepsilon k} dk = \lim_{\varepsilon \rightarrow 0} \frac{1}{2\pi} \frac{\varepsilon}{x^2 + \varepsilon^2} \quad (A18)$$

Então com a ajuda do método do factor de corte temos:

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} dk = \lim_{\varepsilon \rightarrow 0} \frac{1}{2\pi} \frac{\varepsilon}{x^2 + \varepsilon^2} \quad (A19)$$

Realmente esta expressão é (A13),

$$\delta(x) = \lim_{\varepsilon \rightarrow 0} \delta(x, \varepsilon)$$

onde $\delta(x, \varepsilon)$ está representada em (A14).

I.d. Cálculo da soma: $\sum_{k=1}^{\infty} \text{sen}(kx)$

Para a obtenção desta soma, pode-se decompôr em série de Fourier a função,

$$f(x) = \text{ctg}(\pi x), \quad (A20)$$

com o período igual a 1. A representação na série de Fourier é:

$$\operatorname{ctg}(\pi x) = \sum_{k=-\infty}^{\infty} C_k e^{i2\pi kx}, \quad (\text{A21})$$

onde,

$$C_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{\cos(\pi x)}{\operatorname{sen}(\pi x)} e^{-i2\pi kx} dx \quad (\text{A22})$$

É possível calcular este integral usando o teorema de resíduos, escolhendo para $k < 0$ o contorno representado na figura 27.

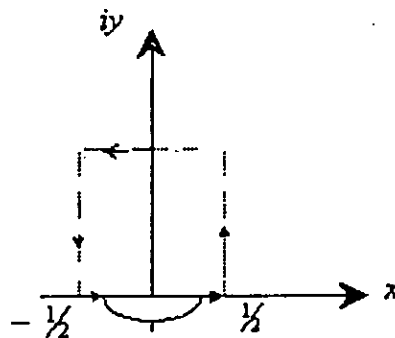


Figura 27: contorno usado no caso $k < 0$

Obtemos

$$C_k = i \quad \text{para} \quad k < 0 \quad (\text{A23})$$

Escolhendo para $k > 0$ o contorno representado na figura 28

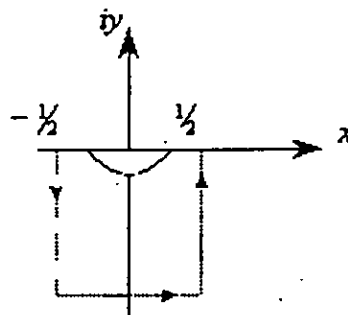


Figura 28: contorno usado no caso $k > 0$

Obtemos,

$$C_k = -i \quad \text{para} \quad k > 0 \quad (\text{A24})$$

Para $k = 0$ temos $C_0 = 0$ porque neste caso a função de integração é ímpar. É possível representar as fórmulas (A23) e (A24) como uma única fórmula

$$C_k = -i \operatorname{sign}(k) \quad (\text{A25})$$

onde,

$$\text{sign}(k) = \begin{cases} 1 & \text{para } k > 0 \\ 0 & \text{para } k = 0 \\ -1 & \text{para } k < 0 \end{cases} \quad (\text{A26})$$

Notamos que

$$\text{Sign}(k) = \theta(k) - \theta(-k) \quad (\text{A27})$$

Usando (A21) e (A25) temos:

$$\text{ctg}(\pi x) = -i \sum_{k=-\infty}^{\infty} \text{sign}(k) e^{i2\pi kx} = i \sum_{k=1}^{\infty} (e^{-i2\pi kx} - e^{i2\pi kx}) = 2 \sum_{k=1}^{\infty} \text{sen}(2\pi kx) \quad (\text{A28})$$

finalmente temos:

$$\sum_{k=1}^{\infty} \text{sen}(2\pi kx) = \frac{1}{2} \text{ctg}(\pi x) \quad (\text{A29})$$

podemos reescrever esta fórmula de tal modo:

$$\sum_{k=1}^{\infty} \text{sen}(kx) = \frac{1}{2} \text{ctg}\left(\frac{x}{2}\right) \quad (\text{A30})$$

I. e. Cálculo da Soma: $\sum_{k=1}^{\infty} \cos(kx)$

Existem vários métodos de obtenção desta soma. Nós optamos pelo método usado no livro [13].

Considera-se a função,

$$f(x) = \sum_{n=-\infty}^{\infty} \delta(x-n) \quad (\text{A31})$$

A decomposição na série de Fourier é:

$$\sum_{n=-\infty}^{\infty} \delta(x-n) = \sum_{k=-\infty}^{\infty} C_k e^{i2\pi kx} \quad (\text{A32})$$

A função $f(x)$ tem o período igual a 1, por isso,

$$C_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_{n=-\infty}^{\infty} \delta(x-n) e^{-i2\pi kx} dx \quad (\text{A33})$$

por causa da propriedade (A5) temos só um membro não igual a zero, quando $n=0$:

$$C_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} \delta(x) e^{-i2\pi kx} dx \quad (\text{A34})$$

de onde temos

$$C_k = 1 \quad (\text{A35})$$

usando este valor para C_k encontramos da fórmula (A32) a seguinte relação:

$$\sum_{n=-\infty}^{\infty} \delta(x-n) = \sum_{k=-\infty}^{\infty} e^{i2\pi kx} \quad (\text{A36})$$

ou

$$\sum_{n=-\infty}^{\infty} \delta\left(\frac{x}{2\pi} - n\right) = \sum_{k=-\infty}^{\infty} e^{ikx} \quad (\text{A37})$$

Usando (A7) pode-se transformar esta fórmula de seguinte modo:

$$2\pi \sum_{n=-\infty}^{\infty} \delta(x-2\pi n) = 1 + \sum_{k=1}^{\infty} e^{ikx} + \sum_{k=-1}^{-\infty} e^{ikx} \quad (\text{A38})$$

Isto é:

$$2\pi \sum_{n=-\infty}^{\infty} \delta(x-2\pi n) = 1 + 2 \sum_{k=1}^{\infty} \cos(kx) \quad (\text{A39})$$

Finalmente temos a fórmula necessária:

$$\sum_{k=1}^{\infty} \cos kx = -\frac{1}{2} + \pi \sum_{n=-\infty}^{\infty} \delta(x-2\pi n) \quad (\text{A40})$$

I.f. Fórmula de Adição de Poisson

Seja $\varphi(x)$ qualquer função dada e é necessário calcular a soma $\sum_{n=-\infty}^{\infty} \varphi(n)$.

Usando (A36).

$$\sum_{n=-\infty}^{\infty} \delta(x-n) = \sum_{k=-\infty}^{\infty} e^{i2\pi kx}$$

podemos escrever

$$\int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x-n) \varphi(x) dx = \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} e^{i2\pi kx} \varphi(x) dx \quad (\text{A41})$$

depois da integração temos:

$$\sum_{n=-\infty}^{\infty} \varphi(n) = \sum_{k=-\infty}^{\infty} \tilde{\varphi}(2\pi k), \quad (\text{A42})$$

onde

$$\tilde{\varphi}(2\pi k) = \int_{-\infty}^{\infty} \varphi(x) e^{i2\pi kx} dx \quad (\text{A43})$$

a fórmula (A42), é a fórmula de Adição de Poisson.

Usando a expressão:

$$2\pi \sum_{n=-\infty}^{\infty} \delta(x-2\pi n) = \sum_{k=-\infty}^{\infty} e^{ikx} \quad (\text{A44})$$

em lugar da (A36) pode-se obter a outra forma da fórmula de Adição de Poisson:

$$2\pi \sum_{n=-\infty}^{\infty} \varphi(2\pi n) = \sum_{k=-\infty}^{\infty} \tilde{\varphi}(k), \quad (\text{A45})$$

onde

$$\tilde{\varphi}(k) = \int_{-\infty}^{\infty} \varphi(x) e^{ikx} dx \quad (\text{A46})$$

APÊNDICE II

Neste Apêndice, estão representados os textos dos programas *OSCIL-PAS* e *DIPLOMA*.

PROGRAM OSCIL-PAS;

USES Graph,CRT;

const

T0 =1;

f0 =1;

var

Gd,Gm,i : integer;

Nx,Nx0,Nxmin,Nxmax : integer;

Ny,Ny0,Nymin,Nymax : integer;

x,xmin,xmax : real;

t,tmin,tmax : real;

omega0,beta : real;

BEGIN

Gd := Detect;

Initgraph(Gd,Gm,"");

if Graphresult <> grok then Halt(1);

SetgraphMode(2);

DirectVideo := False;

ClearDevice;

omega0:= 2*pi/T0;

beta:= omega0;

xmax:= 0.07;

xmin:= 0;

tmax:= T0;

tmin:= 0;

Nxmin := GetMaxX div 10;

Nxmax := GetMaxX - Nxmin;

Nymin := GetMaxY div 10;

```
Nymax := GetMaxY - Nymin;  
Nx0 := Round((Nxmin*tmax - Nxmax*tmin)/(tmax - tmin));  
Ny0 := Round((Nymax*xmax - Nymin*xmin)/(xmax - xmin));  
Line (Nxmin,Ny0,Nxmax,Ny0);  
Line (Nx0,Nymin,Nx0,Nymax);  
for Nx:= Nxmin to Nxmax do  
  begin  
    t := ((tmax-tmin)/(Nxmax-Nxmin))*(Nx-Nx0);  
    x := f0*t*exp(-beta*t);  
    Ny := Ny0-Round(((Nymax-Nymin)/(xmax-xmin))*x);  
    PutPixel(Nx,Ny,15);  
  end;  
Readln;  
Closegraph;  
END.
```

```

program Diploma;
uses
  Forms,
  SetDip in 'SetDip.pas' (Form1),
  Forcel in 'Forcel.pas' (Form2),
  Force2 in 'Force2.pas' (Form3);
($R *.RES)
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.Run;
end.

```

```

unit SetDip;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, Menus, ExtCtrls;
type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    Forcell: TMenuItem;
    Imagem1: TImage;
    Timer1: TTimer;
    Force2: TMenuItem;
  procedure FormCreate(Sender: TObject);
  procedure ForcellClick(Sender: TObject);
  procedure FormPaint(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
  procedure Force2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1 : TForm1;
  Ne, Nr, Nv, Nx : integer;
implementation
uses Forcel, Force2;
($R *.DFM)
procedure TForm1.FormCreate(Sender: TObject);
begin
  WindowState:= wsMaximized;
  Timer1.Interval:= 1;
end;
procedure TForm1.FormPaint(Sender: TObject);
var
  st1: string;
begin
  Canvas.Brush.Color:= clBtnFace;
  Canvas.Font.Color:= clGreen;
  Canvas.Font.Height:= ClientHeight div 10;
  Canvas.Font.Style:= [fsBold];
  Canvas.Font.Name:= 'Arial';
  st1:= 'Função de Green';
  Canvas.TextOut((ClientWidth-Canvas.TextWidth(st1)) div 2,
    ClientHeight div 8,st1);

```

```

unit Forcel;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  jpeg, ExtCtrls, Menus, Printers, Math;
type
  TForm2 = class(TForm)
    MainMenu: TMainMenu;
    File1: TMenuItem;
    Print1: TMenuItem;
    Exit1: TMenuItem;
    Forcell: TMenuItem;
    Variant1: TMenuItem;
    Variant2: TMenuItem;
    Variant3: TMenuItem;
    Image1: TImage;
    procedure Axis;
    procedure FormKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormCreate(Sender: TObject);
    procedure Print1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Variant1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Variant2Click(Sender: TObject);
    procedure Variant3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
const
  T0 = 1;
  f0 = 1;
var
  Form2: TForm2;
  Nxmin, Nxmax, Nx0 : integer;
  Nymin, Nymax, Ny0 : integer;
  x, xmin, xmax : real;
  t, tmin, tmax : real;
  omega0, Omega, ksi : real;
  beta, Tbeta : real;
  Knt, Kxn : real;
implementation
uses SetDip;
{$R *.DFM}
procedure TForm2.Axis;
begin
  Canvas.Pen.Color:= clBlack;
  Canvas.Pen.Width:= 2;
  Canvas.MoveTo(Nx0, Ny0);
  Canvas.LineTo(Nxmax, Ny0);
  Canvas.MoveTo(Nx0, Nymin);
  Canvas.LineTo(Nx0, Nymax);
end;
procedure TForm2.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if Key = vk_escape then Close;
  if Key = vk_f5 then Variant1Click(Sender);
  if Key = vk_f6 then Variant2Click(Sender);
  if Key = vk_f7 then Variant3Click(Sender);
end;

```

```

Canvas.Font.Height:= 16;
Canvas.Font.Style:= [fsBold];
Nx:= Nx0 +Round(0.5/Knt);
Canvas.TextOut(Nx, Ny0-9, '|');
Nx:= Nx0 +Round(1/Knt);
Canvas.TextOut(Nx, Ny0-9, '|');
Canvas.TextOut(Nx, Ny0+7, '|');
Ny:= Ny0 - Round(Kxn*0.06);
Canvas.TextOut(Nx0-41, Ny, '0.06  --');
Ny:= Ny0 - Round(Kxn*0.03);
Canvas.TextOut(Nx0-5, Ny, '--');
Canvas.Font.Height:= 24;
Canvas.Font.Style:= [fsBold];
Nx:= Nx0 +Round(0.18/Knt);
Ny:= Ny0 - Round(Kxn*0.058);
Canvas.TextOut(Nx, Ny, '1');
Nx:= Nx0 +Round(0.18/Knt);
Ny:= Ny0 - Round(Kxn*0.045);
Canvas.TextOut(Nx, Ny, '2');
Canvas.Font.Name:= 'Symbol';
Canvas.Font.Height:= 36;
Nx:= Nx0 +Round(0.8/Knt);
Ny:= Ny0 - Round(Kxn*0.06);
Canvas.TextOut(Nx, Ny, '1.  b = 1.1w');
Nx:= Nx + Canvas.TextWidth('1.  b = 1.1w');
Canvas.Font.Height:= 18;
Canvas.TextOut(Nx, Ny+18, 'o');
Canvas.Font.Height:= 36;
Nx:= Nx0 +Round(0.8/Knt);
Ny:= Ny0 - Round(Kxn*0.06);
Canvas.TextOut(Nx, Ny+50, '2.  b = 1.6w');
Nx:= Nx + Canvas.TextWidth('2.  b = 1.6w');
Canvas.Font.Height:= 18;
Canvas.TextOut(Nx, Ny+68, 'o');
Canvas.Font.Name:= 'Arial';
Axis;
Canvas.Pen.Color:= clBlack;
beta:= 1.1*omega0;
ksi:= sqrt(sqr(beta) - sqr(omega0));
for Nx:= Nxmin to Nxmax do
begin
  t:= Knt*(Nx-Nx0);
  x:= f0*exp(-beta*t)*sinh(ksi*t)/ksi;
  Ny:= Ny0 - Round(Kxn*x);
  Canvas.MoveTo(Nx, Ny);
  t:= Knt*(Nx+1-Nx0);
  x:= f0*exp(-beta*t)*sinh(ksi*t)/ksi;
  Ny:= Ny0 - Round(Kxn*x);
  Canvas.LineTo(Nx+1, Ny);
end;
beta:= 1.6*omega0;
ksi:= sqrt(sqr(beta) - sqr(omega0));
for Nx:= Nxmin to Nxmax do
begin
  t:= Knt*(Nx-Nx0);
  x:= f0*exp(-beta*t)*sinh(ksi*t)/ksi;
  Ny:= Ny0 - Round(Kxn*x);
  Canvas.MoveTo(Nx, Ny);
  t:= Knt*(Nx+1-Nx0);
  x:= f0*exp(-beta*t)*sinh(ksi*t)/ksi;
  Ny:= Ny0 - Round(Kxn*x);
  Canvas.LineTo(Nx+1, Ny);
end;
end;
end;

```



```

procedure TForm2.Variant3Click(Sender: TObject);
var
  Nx, Ny: integer;
  st1: string;
begin
  xmax:= 0.2;
  xmin:= -xmax;
  tmin:= 0;
  tmax:= 4*T0;
  Nx0:= Round((Nxmin*tmax-Nxmax*tmin)/(tmax-tmin));
  Ny0:= Round((Nymin*xmax-Nymin*xmin)/(xmax-xmin));
  Knt:=(tmax-tmin)/(Nxmax-Nxmin);
  Kxn:=(Nymin-Nymin)/(xmax-xmin);
  Canvas.Brush.Color:= clWhite;
  Canvas.FillRect(ClientRect);
  Canvas.Font.Color:= clBlack;
  Canvas.Font.Height:= 24;
  Canvas.Font.Style:= [fsBold];
  Canvas.TextOut(Nxmin+5, Nymin-12, 'x(t)');
  Canvas.TextOut(Nxmax+5, Ny0, 't');
  st1:= 'Figura 8';
  Canvas.TextOut((ClientWidth-Canvas.TextWidth(st1)) div 2,
    ClientHeight-24, st1);
  Canvas.Font.Height:= 16;
  Ny:= Ny0 - Round(Kxn*0.2);
  Canvas.TextOut(Nx0-31, Ny, '0.2 --');
  Ny:= Ny0 - Round(Kxn*0.1);
  Canvas.TextOut(Nx0-5, Ny, '--');
  Canvas.Font.Height:= 16;
  Canvas.Font.Style:= [fsBold];
  Nx:= Nx0 +Round(0.25/Knt);
  Ny:= Ny0 - Round(Kxn*0.18);
  Canvas.TextOut(Nx, Ny, '1');
  Nx:= Nx0 +Round(0.25/Knt);
  Ny:= Ny0 - Round(Kxn*0.14);
  Canvas.TextOut(Nx, Ny, '2');
  Canvas.Font.Name:= 'Symbol';
  Canvas.Font.Height:= 28;
  Nx:= Nx0 +Round(3/Knt);
  Ny:= Ny0 - Round(Kxn*0.23);
  Canvas.TextOut(Nx, Ny, '1. b = 0');
  Canvas.Font.Height:= 28;
  Nx:= Nx0 +Round(3/Knt);
  Ny:= Ny0 - Round(Kxn*0.25);
  Canvas.TextOut(Nx, Ny+50, '2. b = 0.2w');
  Nx:= Nx + Canvas.TextWidth('2. b = 0.2w');
  Canvas.Font.Height:= 14;
  Canvas.TextOut(Nx, Ny+64, 'o');
  Canvas.Font.Name:= 'Arial';
  Axis;
  Canvas.Pen.Color:= clBlack;
  Canvas.Pen.Width:= 3;
  beta:= 0;
  Omega:= sqrt(sqrt(omega0)-sqrt(beta));
  for Nx:= Nxmin to Nxmax do
  begin
    t:= Knt*(Nx-Nx0);
    x:= f0*exp(-beta*t)*sin(Omega*t)/Omega;
    Ny:= Ny0 - Round(Kxn*x);
    Canvas.MoveTo(Nx, Ny);
  end
end

```

```

unit Force2;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, Menus, Printers;
type
  TForm3 = class(TForm)
    Image1: TImage;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Print1: TMenuItem;
    Exit1: TMenuItem;
    Force21: TMenuItem;
    Variant01: TMenuItem;
    Variant11: TMenuItem;
    Variant21: TMenuItem;
    Variant31: TMenuItem;
    Timer1: TTimer;
    function Xs(kmf:integer):real;
    procedure FormCreate(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Print1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure FormKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure Variant01Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Variant11Click(Sender: TObject);
    procedure Variant21Click(Sender: TObject);
    procedure Variant31Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
const
  T0 = 1;
  f0 = 1;
var
  Form3: TForm3;
  Nxmin, Nxmax, Nx0 : integer;
  Nymn, Nymax, Ny0 : integer;
  Nx, Ny, km, kf, rp, fr : integer;
  x, xmin, xmax : real;
  t, tmin, tmax, tk : real;
  artk : array[0..500] of real;
  omega0, Omega, Tom : real;
  beta, tau : real;
  Knt, Kxn : real;
  st1, st2, st3, stm : string;
implementation
uses SetDip;
{$R *.DFM}
function TForm3.Xs(kmf:integer):real;
var
  k : integer;
  xm : real;
begin
  xm:= 0;
  for k:= 0 to kmf do
    xm:= xm + (f0*exp(-beta*(t-artk[k]))*sin(Omega*(t-artk[k])))/Omega;
  Xs:= xm;
end;

```

```

if Key = vk_up then
begin
  if rp > 10 then rp:= rp - 10;
  stm:= IntToStr(rp);
  ShowMessage('Interval convention '+stm);
end;
if Key = vk_down then
begin
  if rp < 71 then rp:= rp + 10;
  stm:= IntToStr(rp);
  ShowMessage('Interval convention '+stm);
end;
if Key = vk_prior then
begin
  fr:= 1;
  ShowMessage('RANDOM');
end;
if Key = vk_next then
begin
  fr:= 0;
  ShowMessage('PERIODICA');
end;
end;
procedure TForm3.Variant01Click(Sender: TObject);
begin
  xmax:= 0.3;
  xmin:= -xmax;
  tmin:= 0;
  tmax:= 20*Tom;
  Nx0:= Round((Nxmin*tmax-Nxmax*tmin)/(tmax-tmin));
  Ny0:= Round((Nymax*xmax-Nymin*xmin)/(xmax-xmin));
  Knt:=(tmax-tmin)/(Nxmax-Nxmin);
  Kxn:=(Nymax-Nymin)/(xmax-xmin);
  Nx:= Nxmin;
  Ny:= 0;
  beta:= 0.05*omega0;
  Omega:= sqrt(sqrt(omega0)-sqrt(beta));
  Tom:= 2*pi/Omega;
  st1:= 't = 5T';
  st2:= 'b = 0.05w';
  st3:= 'o';
  tau:= 5*Tom;
  tk:= tau;
  artk[0]:= 0;
  km:= 0;
  Timer1.Enabled:= True;
end;
procedure TForm3.Variant11Click(Sender: TObject);
begin
  xmax:= 0.4;
  xmin:= -xmax;
  tmin:= 0;
  tmax:= 20*Tom;
  Nx0:= Round((Nxmin*tmax-Nxmax*tmin)/(tmax-tmin));
  Ny0:= Round((Nymax*xmax-Nymin*xmin)/(xmax-xmin));
  Knt:=(tmax-tmin)/(Nxmax-Nxmin);
  Kxn:=(Nymax-Nymin)/(xmax-xmin);
  Nx:= Nxmin;
  Ny:= 0;
  beta:= 0;
  Omega:= sqrt(sqrt(omega0)-sqrt(beta));
  Tom:= 2*pi/Omega;

```

```

procedure TForm3.Timer1Timer(Sender: TObject);
var
  i,j,kmr : integer;
begin
  if Nx > 10*Nxmax then Timer1.Enabled:= False;
  if km > 499 then Timer1.Enabled:= False;
  Canvas.Brush.Color:= clWhite;
  Canvas.FillRect(ClientRect);
  Canvas.Pen.Color:= clBlack;
  Canvas.Pen.Width:= 1;
  Canvas.Rectangle(Nxmin, Nymin, Nxmax, Nymax);
  Canvas.MoveTo(Nxmin, Ny0);
  Canvas.LineTo(Nxmax, Ny0);
  Canvas.Pen.Color:= clBlack;
  Canvas.Pen.Width:= 3;
  Canvas.Font.Color:= clBlack;
  Canvas.Font.Name:= 'Symbol';
  Canvas.Font.Height:= 24;
  Canvas.TextOut(Nxmin+5, 1, st1);
  Canvas.TextOut(Nxmin+2*Canvas.TextWidth(st1), 1, st2);
  i:= 2*Canvas.TextWidth(st1) + Canvas.TextWidth(st2);
  Canvas.Font.Height:= 14;
  Canvas.TextOut(Nxmin+i, 10, st3);
  Canvas.Font.Name:= 'Arial';
  kmr:= 1;
  for i:= Nxmin to Nxmax do
  begin
    Nx:= Nx + 1;
    for j:= 1 to rp do
    begin
      t:= (Nx-1-Nx0)*Knt;
      kf:= Round(Int(t/tk));
      if kf = 1 then
      begin
        km:= km + 1;
        tk:= tau*(km + 1 + fr*Random);
        artk[km]:= tk - tau;
      end;
      if km <> kmr then
      begin
        kmr:= km;
        Canvas.Pen.Color:= clBlack;
        Canvas.Pen.Width:= 2;
        Canvas.MoveTo(i, (5*Nymax div 6));
        Canvas.LineTo(i, Nymax);
        Canvas.Pen.Color:= clBlack;
        Canvas.Pen.Width:= 3;
      end;
      x:= Xs(km);
      Ny:= Ny0 - Round(Kxn*x);
      Canvas.MoveTo(i-1, Ny);
      t:= (Nx-Nx0)*Knt;
      x:= Xs(km);
      Ny:= Ny0 - Round(Kxn*x);
      Canvas.LineTo(i, Ny);
    end;
  end;
  Timer1.Enabled:= False;
end;
end.

```