

ESTUDO DOS PROBLEMAS DE ANÁLISE DE ESTABILIDADE
DE
SISTEMAS ELÉCTRICOS DE ENERGIA

RELATÓRIO DE ESTÁGIO

MARIA CELESTE PEREIRA MARQUES

1992/93



4
621.3(247.3) [LECC 1992] MARM
25 09 89

PRODEP

Parecer Científico sobre

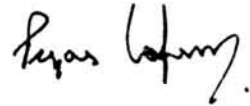
Estágio de Maria Celeste Pereira Marques

A estagiária Maria Celeste Pereira Marques realizou sob minha orientação um trabalho de estágio sobre o tema da Análise da Estabilidade de Sistemas Eléctricos de Energia, tendo sido totalmente alcançados os objectivos científicos inicialmente propostos.

A estagiária efectuou inicialmente um estudo detalhado sobre modelos matemáticos capazes de representar o comportamento dinâmico de componentes do sistema, incluindo reguladores de velocidade e tensão de alternadores, bem como dos métodos numéricos mais adequados para efectuar as análises pretendidas.

A aluna desenvolveu seguidamente programas de cálculo com base nos modelos escolhidos e nos algoritmos desenvolvidos, utilizando a linguagem C, em ambiente MS/DOS, e envolvendo um interface gráfico para apresentação de resultados ao utilizador.

Finalmente, procedeu-se ao teste e validação das ferramentas de simulação desenvolvidas, tendo os resultados obtidos demonstrado a qualidade trabalho efectuado.



Parecer Técnico

Acompanhei os trabalhos de estágio realizados no INESC pelo aluno e apreciei o relatório final apresentado, considerando que aquele relatório reflecte o trabalho efectivamente realizado.

Sou de parecer que foram atingidos os objectivos inicialmente propostos no plano de trabalhos.

O Supervisor do Estágio (INESC)



**ESTUDO DOS PROBLEMAS
DE
ANÁLISE DE ESTABILIDADE
DE
SISTEMAS ELÉCTRICOS DE ENERGIA**

RELATÓRIO DE ESTÁGIO

ano lectivo de 1992/93

Nome do estagiário: Maria Celeste Pereira Marques

Instituição de estágio: INESC

Nome do orientador: Prof. Doutor Peças Lopes

Apoio: PRODEP

Agradecimentos.

Agradeço a todos os meus colegas no INESC o apoio prestado durante o desenvolvimento do trabalho de estágio. Agradeço especialmente ao Prof. Doutor Peças Lopes e à FEUP o tempo dispendido na orientação do trabalho proposto, bem como ao Prof. Doutor Vladimiro Miranda que propôs este trabalho de estágio e ao PRODEP o apoio financeiro prestado ao longo do tempo em que fui estagiário no INESC.

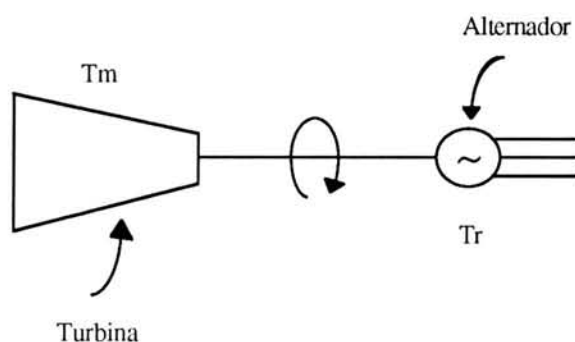
Índice.

1. Introdução aos modelos matemáticos utilizados.....	1
2. Estudo dos métodos numéricos de resolução de equações diferenciais não lineares. Avaliação da estabilidade numérica dos métodos.....	3
2.1. Métodos de passo único.....	4
2.2. Métodos de passo múltiplo.....	4
2.3. Métodos a usar.....	5
3. Definição de uma biblioteca de modelos de estados de sistemas de regulação dos grupos geradores; definição do modelo de estado global do sistema; definição de perturbações a analisar.....	9
4. Especificação dos requisitos computacionais a adoptar: hardware; linguagem de programação; bases de dados ou manipulação de ficheiros.....	13
5. Definição e especificação do algoritmo de acordo com o modelo de estado, métodos de integração numérica a adoptar e a perturbação seleccionada.....	13
6. Implementação do algoritmo de Runge-Kutta. Apresentação gráfica dos resultados da evolução temporal das grandezas de estado seleccionadas. Testes de validação. Comparação da resposta do sistema a perturbações de acordo com o tipo de reguladores e parâmetros seleccionados.....	14
7. Construção de software necessário para dotar a aplicação de uma interface amigável, e que lhe confira características didácticas.....	14
8. Implementação do algoritmo de Adams. Apresentação gráfica dos resultados da evolução temporal das grandezas de estado seleccionadas. Testes de validação. Comparação da resposta do sistema a perturbações de acordo com o tipo de reguladores e parâmetros seleccionados.....	15
9. Equações "stiff".....	15
10. Comparação dos resultados.....	16
11. Conclusões. Objectivos alcançados.....	17

12. Bibliografia.....	18
-----------------------	----

1. Introdução aos modelos matemáticos utilizados.

No sistema eléctrico de energia existe um equilíbrio entre o consumo e a produção de energia. Este equilíbrio traduz-se na igualdade de binários, o **binário motor** e o **binário resistente**. O binário motor está associado à potência que é desenvolvida pelas máquinas primárias, as chamadas **turbinas**. O binário resistente é desenvolvido pelo alternador. A fig. 1.1 mostra toda esta relação.



Tm: binário motor (binário mecânico)

Tr: binário resistente (binário electromagnético)

**Fig. 1.1 O alternador desenvolve o binário resistente.
A turbina desenvolve o binário motor.**

Em equilíbrio

$$T_m = T_r.$$

Acontece, que há frequentemente perturbações no sistema, resultantes da perda de geradores, súbitas alterações de carga ou curtos circuitos. Essas perturbações traduzem-se numa desigualdade de binários existentes, sendo necessário que os geradores sejam dotados de sistemas de regulação que venham a conseguir estabelecer novamente a igualdade de binários, i.e., o equilíbrio, a estabilidade.

Para fazer a análise da estabilidade, foi necessário encontrar modelos dinâmicos adequados, nomeadamente o **modelo da turbina** e o **modelo do alternador**. Estes modelos e seus sistemas de regulação de velocidade e de tensão são usualmente descritos através de diagramas de blocos no domínio das transformadas de Laplace (fig. 1.2 e fig. 1.3, apresentadas no apêndice A), e conduzem ao seguinte sistema de equações diferenciais não lineares:

$$\left\{ \begin{aligned}
\frac{\partial V_{F_i}}{\partial t} &= \frac{V_i - V_{F_i}}{T_{R_i}} = f_1 \left(t, V_{F_i}(t), \dots, P_i(t), V_i \left(t, V_{F_i}, \dots, P_i \right), V_{e_i}(\dots), V_{RL_i}(\dots), I_{d_i}(\dots), I_{q_i}(\dots), P_{mec_i}(\dots), P_{e_i}(\dots), P_{co_i}(\dots) \right) \\
\frac{\partial V_{R_i}}{\partial t} &= \frac{K_{A_i} V_{e_i} - V_{R_i}}{T_{A_i}} = f_2 \left(t, V_{F_i}(t), \dots, P_i(t), V_i \left(t, V_{F_i}, \dots, P_i \right), V_{e_i}(\dots), V_{RL_i}(\dots), I_{d_i}(\dots), I_{q_i}(\dots), P_{mec_i}(\dots), P_{e_i}(\dots), P_{co_i}(\dots) \right) \\
\frac{\partial E_{fd_i}}{\partial t} &= \frac{V_{RL_i} - A_{ex} e^{B_{ex} E_{fd_i}} - K_{E_i} E_{fd_i}}{T_{E_i}} = f_3(\dots) \\
\frac{\partial V_{SL_i}}{\partial t} &= \left(\frac{K_{S_i}}{T_{S_i}} E_{fd_i} - V_{SL_i} \right) \frac{1}{T_{S_i}} = f_4(\dots) \\
\frac{\partial E'_{q_i}}{\partial t} &= \frac{1}{T_{do_i}} \left(E_{fd_i} - \left(x_{d_i} - x'_{q_i} \right) I_{d_i} - E'_{q_i} \right) = f_5(\dots) \\
\frac{\partial E'_{d_i}}{\partial t} &= \frac{1}{T_{do_i}} \left(\left(x_{q_i} - x'_{d_i} \right) I_{q_i} - E'_{d_i} \right) = f_6(\dots) \\
\frac{\partial \omega_i}{\partial t} &= \frac{1}{M_i} \left(P_{mec_i} - P_{e_i} - \frac{M_i}{M_T} P_{co_i} \right) = f_7(\dots) \\
\frac{\partial \delta_i}{\partial t} &= \omega_i = f_8(\dots) \\
\frac{\partial P_{c_i}}{\partial t} &= -\frac{K_{I_i}}{2\pi f_0} \omega_i = f_9(\dots) \\
\frac{\partial P_{H_i}}{\partial t} &= \frac{1}{T_{G_i}} \left(P_{c_i} - \frac{1}{2\pi f_0 R} \omega_i - P_{H_i} \right) = f_{10}(\dots) \\
\frac{\partial P_i}{\partial t} &= \frac{1}{T_{T_i}} \left(P_{H_i} - P_i \right) = f_{11}(\dots)
\end{aligned} \right.$$

bem como às seguintes equações auxiliares:

$$V_{e_i} = V_{ref} - V_{F_i} - V_{S_i}$$

$$\text{Se } V_{R_i} > V_{R_{\max_i}} \text{ então } V_{RL_i} = V_{R_{\max_i}}$$

$$\text{Se } V_{R_i} < V_{R_{\min_i}} \text{ então } V_{RL_i} = V_{R_{\min_i}}$$

$$\text{senão } V_{RL_i} = V_{R_{\min_i}}$$

$$V_{S_i} = \frac{K_{S_i}}{T_{S_i}} E_{fd_i} - V_{SL_i}$$

$$P_{co_i} = \sum_{i=1}^m P_{mec_i} - P_{e_i} = 0$$

$$M_T = \sum_{i=1}^m M_i$$

Se $P > P_{\text{máx}}$ então $P_{\text{mec}} = P_{\text{mec}_{\text{máx}}}$

Se $P < 0$ então $P_{\text{mec}} = 0$

senão $P_{\text{mec}} = P$

onde

$i = 1, \dots, m$ (número de geradores);

t é a variável independente;

$V_{F_i}, V_{R_i}, E_{fd_i}, V_{SL_i}, E_{q_i}, E_{d_i}, \omega_i, \delta_i, P_{c_i}, P_{H_i}, P_i$ são as variáveis dependentes;

$V_i, V_{e_i}, V_{RL_i}, I_{d_i}, I_{q_i}, P_{\text{mec}_i}, P_{e_i}, P_{\text{co}_i}$ são as variáveis auxiliares;

$V_{\text{ref}}, R_0, X, R_p, X_p, R_{pd}, X_{pd}, T_{R_i}, K_{A_i}, T_{A_i}, V_{R_{\text{máx}_i}}, V_{R_{\text{mín}_i}}, A_{\text{ex}}, B_{\text{ex}}, K_{E_i}, T_{E_i}, K_{S_i}, T_{S_i}, R, K_i, T_G, T_T, P_{\text{mec}_{\text{máx}_i}}, M_i, X_{d_i}, X_{d_i}', X_{q_i}, X_{q_i}', T_{\text{do}_i}'$ são os

parâmetros dos sistemas de regulação.

O **modelo de estado** fica assim estabelecido. Foi assim definido o conjunto de equações diferenciais não lineares e algumas equações auxiliares que ligam as equações diferenciais entre si.

2. Estudo dos métodos numéricos de resolução de equações diferenciais não lineares. Avaliação da estabilidade numérica dos métodos.

Nesta etapa tive que encontrar os métodos numéricos adequados para resolver o sistema de equações diferenciais não lineares e respectivas condições iniciais, obtido no ponto anterior. Para isso usei os **métodos de variável discreta** para problemas com condições iniciais. Estes métodos tem como objectivo calcular aproximações para a solução num conjunto de pontos discretos $t_0, t_1, t_2, \dots$, da variável independente.

Dentro dos métodos de variável discreta, podemos utilizar os métodos de passo único e os métodos de passo múltiplo. Os **métodos de passo único**, em cada iteração, requerem o conhecimento dos valores das variáveis dependentes da iteração anterior, apenas, enquanto que os **métodos de passo múltiplo** requerem o conhecimento dos valores das variáveis dependentes de várias das iterações precedentes, dependendo da ordem do método escolhido.

Na avaliação de todos os métodos que estudei tive em consideração cinco aspectos fundamentais, nomeadamente a ordem do erro de truncatura global, a estimativa do erro de truncatura local, a eficiência, a possibilidade de iniciar a integração só com o conhecimento das condições iniciais e a estabilidade do método.

2.1. Métodos de passo único.

De todos os métodos explícitos de passo único que estudei, os **métodos de Runge-Kutta** são sem dúvida os mais eficientes. Estes métodos foram desenvolvidos com o objectivo de produzirem resultados com a mesma precisão que os obtidos por série de Taylor truncada, mas sem ter de calcular derivadas. Como os valores das variáveis dependentes são unicamente determinados pelas fórmulas, este método não necessita de repetidas aproximações como no **método de Euler modificado** ou integrações sucessivas como no **método de Picard**. No entanto, este método requer um grande número de operações aritméticas, mas não necessita de considerar um passo tão pequeno como no **método de Euler** e os resultados obtidos, mesmo assim, são mais precisos. O **método de Runge-Kutta-Fehlberg**, apesar de calcular uma estimativa para o erro local, permitindo assim o ajuste do passo a fim de se obterem aproximações com uma precisão previamente especificada, requer um número de operações aritméticas muito mais elevado.

2.2. Métodos de passo múltiplo.

Para desenvolver os métodos de passo múltiplo chamados **métodos de Adams**, o integral do segundo membro da equação

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t)) dt$$

é calculado usando regras de quadratura numérica baseadas na aproximação de f por um polinómio interpolador em pontos igualmente espaçados.

Nos métodos de Adams temos a considerar a **fórmula explícita de Adams-Bashforth** que tem como pontos de interpolação $t_n, t_{n-1}, t_{n-2}, t_{n-3}$ (no caso de 4ª ordem) e a **fórmula implícita de Adams-Moulton** que tem vantagens sobre a fórmula de Adams-Bashforth pois t_{n+1} é seleccionado como ponto de interpolação. Uma aproximação inicial razoavelmente próxima de y_{n+1} pode ser gerada usando a fórmula explícita de Adams-Bashforth. A fórmula é então chamada **fórmula preditora** e a fórmula implícita de Adams-Moulton **fórmula correctora**. Temos então os chamados **métodos PECE** (**P**: predict, **E**: evaluat, **C**: correct, **E**: evaluat). A diferença entre o valor corrigido e o valor previsto pode ser usada para estimar o erro local do valor corrigido.

A aplicação dum método PECE requer o conhecimento prévio dos valores das variáveis dependentes de algumas das iterações precedentes. À partida, estes valores podem ser calculados usando um método de passo único. Contudo, o erro de truncatura do método de passo único escolhido terá de ser da mesma ordem do do método de passo múltiplo. Nestes casos, a vantagem em prosseguir com um

método PECE é que estes são mais eficientes.

Além dos métodos PECE existem também os **métodos $P(EC)^kE$** . Nestes métodos são calculadas k iterações da fórmula correctora, i.e., a aproximação inicial é corrigida k vezes através da fórmula correctora. Estes métodos convergem rapidamente quando a aproximação inicial é suficientemente boa e o valor do passo suficientemente pequeno. Mas não são aconselháveis, uma vez que não se deve estimar o erro de truncatura do corrector para melhorar o valor corrigido e a correcção do corrector afecta a estabilidade deste. Logo, em vez de corrigir o corrector é mais eficiente reduzir o passo de integração.

Dos métodos de passo múltiplo, os métodos de Adams são os mais eficientes porque por um lado a fórmula explícita de Adams-Bashforth dá uma aproximação inicial muito boa, e por outro lado a fórmula implícita de Adams-Moulton tem boas características de estabilidade, o que torna este método o método preditor-corrector mais estável. Mas existem ainda outros métodos preditores-correctores, tais como o método de Milne, o método de Hamming, e o método de Hermite.

A fórmula correctora do **método de Milne** nem sempre é estável, e por isso este método não é aconselhável a não ser que o número de passos de integração seja pequeno. O **método de Hamming** é estável, mas menos estável que o de Adams. O **método de Hermite** é uma boa escolha quando a 2ª derivada é fácil de calcular, pois é estável e tem um erro de truncatura inferior ao dos métodos de Hamming e Milne. Mas como tenho um sistema, o cálculo da 2ª derivada torna-se substancialmente mais complicado e por isso este método também não é aconselhável.

2.3. Métodos a usar.

Por todas as razões que acabo de expor, optei pelo método de Adams de 4ª ordem com passo variável, e pelo método de Runge-Kutta da mesma ordem para gerar os quatro valores iniciais. Logo para controlar o passo tenho que testar o erro. Para isso, se em cada iteração o erro torna-se demasiado elevado e não permite obter a precisão desejada, reduzo o passo a metade; se o erro torna-se demasiado pequeno, mais do que é necessário, posso diminuir o tempo de execução dobrando o passo de integração. Como os métodos preditores-correctores considerados requerem um passo constante, sempre que o intervalo é alterado novos valores iniciais terão que ser gerados. Portanto optei também pelo método de Runge-Kutta para alterar o intervalo e dar os valores iniciais. Optei pela 4ª ordem porque os métodos desta ordem permitem obter resultados suficientemente precisos.

Depois de decidir quais os métodos a usar, apliquei-os a um sistema de duas equações diferenciais não lineares sem variáveis auxiliares, com o intuito de me familiarizar com os métodos. Seguidamente adaptei as fórmulas dos métodos ao meu sistema.

No método de Runge_Kutta de quarta ordem as fórmulas usadas para estimar $V_{F_i}(t_{n+1})$,

$$V_{R_i}(t_{n+1}), E_{fd_i}(t_{n+1}), V_{SL_i}(t_{n+1}), E_{q_i}'(t_{n+1}), E_{d_i}'(t_{n+1}), \omega_i(t_{n+1}), \delta_i(t_{n+1}), \\ P_{c_i}(t_{n+1}), P_{H_i}(t_{n+1}), P_i(t_{n+1}) \text{ são}$$

$$\begin{aligned} V_{F_i}^{(n+1)} &= V_{F_i}^{(n)} + \frac{1}{6}(K_{11} + 2K_{12} + 2K_{13} + K_{14}) \\ V_{R_i}^{(n+1)} &= V_{R_i}^{(n)} + \frac{1}{6}(K_{21} + 2K_{22} + 2K_{23} + K_{24}) \\ E_{fd_i}^{(n+1)} &= E_{fd_i}^{(n)} + \frac{1}{6}(K_{31} + 2K_{32} + 2K_{33} + K_{34}) \\ V_{SL_i}^{(n+1)} &= V_{SL_i}^{(n)} + \frac{1}{6}(K_{41} + 2K_{42} + 2K_{43} + K_{44}) \\ E_{q_i}'^{(n+1)} &= E_{q_i}'^{(n)} + \frac{1}{6}(K_{51} + 2K_{52} + 2K_{53} + K_{54}) \\ E_{d_i}'^{(n+1)} &= E_{d_i}'^{(n)} + \frac{1}{6}(K_{61} + 2K_{62} + 2K_{63} + K_{64}) \\ \omega_i^{(n+1)} &= \omega_i^{(n)} + \frac{1}{6}(K_{71} + 2K_{72} + 2K_{73} + K_{74}) \\ \delta_i^{(n+1)} &= \delta_i^{(n)} + \frac{1}{6}(K_{81} + 2K_{82} + 2K_{83} + K_{84}) \\ P_{c_i}^{(n+1)} &= P_{c_i}^{(n)} + \frac{1}{6}(K_{91} + 2K_{92} + 2K_{93} + K_{94}) \\ P_{H_i}^{(n+1)} &= P_{H_i}^{(n)} + \frac{1}{6}(K_{101} + 2K_{102} + 2K_{103} + K_{104}) \\ P_i^{(n+1)} &= P_i^{(n)} + \frac{1}{6}(K_{111} + 2K_{112} + 2K_{113} + K_{114}) \end{aligned}$$

onde

$$K_{1l} = h * f_l(t^{(n)}, V_{F_i}^{(n)}, V_{R_i}^{(n)}, E_{fd_i}^{(n)}, V_{SL_i}^{(n)}, E_{q_i}'^{(n)}, E_{d_i}'^{(n)}, \omega_i^{(n)}, \delta_i^{(n)}, \\ P_{c_i}^{(n)}, P_{H_i}^{(n)}, P_i^{(n)}, V_i^{(n)}, V_{e_i}^{(n)}, V_{RL_i}^{(n)}, I_{d_i}^{(n)}, I_{q_i}^{(n)}, P_{mec_i}^{(n)}, P_{e_i}^{(n)}, P_{co_i}^{(n)})$$

$$l=1, \dots, 11$$

$$\begin{aligned} K_{12} &= h * f_1(t^{(n)} + \frac{h}{2}, V_{F_i}^{(n)} + \frac{K_{11}}{2}, V_{R_i}^{(n)} + \frac{K_{21}}{2}, E_{fd_i}^{(n)} + \frac{K_{31}}{2}, V_{SL_i}^{(n)} + \frac{K_{41}}{2}, \\ &E_{q_i}'^{(n)} + \frac{K_{51}}{2}, E_{d_i}'^{(n)} + \frac{K_{61}}{2}, \omega_i^{(n)} + \frac{K_{71}}{2}, \delta_i^{(n)} + \frac{K_{81}}{2}, P_{c_i}^{(n)} + \frac{K_{91}}{2}, P_{H_i}^{(n)} + \frac{K_{101}}{2}, \\ &P_i^{(n)} + \frac{K_{111}}{2}, V_i^{(n)}, V_{e_i}^{(n)}, V_{RL_i}^{(n)}, I_{d_i}^{(n)}, I_{q_i}^{(n)}, P_{mec_i}^{(n)}, P_{e_i}^{(n)}, P_{co_i}^{(n)}) \end{aligned}$$

$$l=1, \dots, 11$$

$$\begin{aligned}
K_{13} = h * f_1 \left(t^{(n)} + \frac{h}{2}, V_{F_i}^{(n)} + \frac{K_{12}}{2}, V_{R_i}^{(n)} + \frac{K_{22}}{2}, E_{fd_i}^{(n)} + \frac{K_{32}}{2}, V_{SL_i}^{(n)} + \frac{K_{42}}{2}, \right. \\
E_{q_i}^{(n)} + \frac{K_{52}}{2}, E_{d_i}^{(n)} + \frac{K_{62}}{2}, \omega_i^{(n)} + \frac{K_{72}}{2}, \delta_i^{(n)} + \frac{K_{82}}{2}, P_{c_i}^{(n)} + \frac{K_{92}}{2}, P_{H_i}^{(n)} + \frac{K_{102}}{2}, \\
P_i^{(n)} + \frac{K_{112}}{2}, V_i^{(n)}, V_{e_i}^{(n)}, V_{RL_i}^{(n)}, I_{d_i}^{(n)}, I_{q_i}^{(n)}, P_{mec_i}^{(n)}, P_{e_i}^{(n)}, P_{co_i}^{(n)} \left. \right) \\
l=1, \dots, 11
\end{aligned}$$

$$\begin{aligned}
K_{14} = h * f_1 \left(t^{(n)} + h, V_{F_i}^{(n)} + K_{13}, V_{R_i}^{(n)} + K_{23}, E_{fd_i}^{(n)} + K_{33}, V_{SL_i}^{(n)} + K_{43}, \right. \\
E_{q_i}^{(n)} + K_{53}, E_{d_i}^{(n)} + K_{63}, \omega_i^{(n)} + K_{73}, \delta_i^{(n)} + K_{83}, P_{c_i}^{(n)} + K_{93}, P_{H_i}^{(n)} + K_{103}, \\
P_i^{(n)} + K_{113}, V_i^{(n)}, V_{e_i}^{(n)}, V_{RL_i}^{(n)}, I_{d_i}^{(n)}, I_{q_i}^{(n)}, P_{mec_i}^{(n)}, P_{e_i}^{(n)}, P_{co_i}^{(n)} \left. \right) \\
l=1, \dots, 11
\end{aligned}$$

No método de Adams de quarta ordem as fórmulas explícitas de Adams-Bashforth usadas para estimar as mesmas grandezas são

$$\begin{aligned}
V_{F_i}^{(p)(n+1)} &= V_{F_i}^{(n)} + \frac{h}{24} * [55 * f_1(A) - 59 * f_1(B) + 37 * f_1(C) - 9 * f_1(D)] \\
V_{R_i}^{(p)(n+1)} &= V_{R_i}^{(n)} + \frac{h}{24} * [55 * f_2(A) - 59 * f_2(B) + 37 * f_2(C) - 9 * f_2(D)] \\
E_{fd_i}^{(p)(n+1)} &= E_{fd_i}^{(n)} + \frac{h}{24} * [55 * f_3(A) - 59 * f_3(B) + 37 * f_3(C) - 9 * f_3(D)] \\
V_{SL_i}^{(p)(n+1)} &= V_{SL_i}^{(n)} + \frac{h}{24} * [55 * f_4(A) - 59 * f_4(B) + 37 * f_4(C) - 9 * f_4(D)] \\
E_{q_i}^{(p)(n+1)} &= E_{q_i}^{(n)} + \frac{h}{24} * [55 * f_5(A) - 59 * f_5(B) + 37 * f_5(C) - 9 * f_5(D)] \\
E_{d_i}^{(p)(n+1)} &= E_{d_i}^{(n)} + \frac{h}{24} * [55 * f_6(A) - 59 * f_6(B) + 37 * f_6(C) - 9 * f_6(D)] \\
\omega_i^{(p)(n+1)} &= \omega_i^{(n)} + \frac{h}{24} * [55 * f_7(A) - 59 * f_7(B) + 37 * f_7(C) - 9 * f_7(D)] \\
\delta_i^{(p)(n+1)} &= \delta_i^{(n)} + \frac{h}{24} * [55 * f_8(A) - 59 * f_8(B) + 37 * f_8(C) - 9 * f_8(D)] \\
P_{c_i}^{(p)(n+1)} &= P_{c_i}^{(n)} + \frac{h}{24} * [55 * f_9(A) - 59 * f_9(B) + 37 * f_9(C) - 9 * f_9(D)] \\
P_{H_i}^{(p)(n+1)} &= P_{H_i}^{(n)} + \frac{h}{24} * [55 * f_{10}(A) - 59 * f_{10}(B) + 37 * f_{10}(C) - 9 * f_{10}(D)] \\
P_i^{(p)(n+1)} &= P_i^{(n)} + \frac{h}{24} * [55 * f_{11}(A) - 59 * f_{11}(B) + 37 * f_{11}(C) - 9 * f_{11}(D)]
\end{aligned}$$

e as fórmulas implícitas de Adams-Moulton são

$$\begin{aligned}
 V_{F_i}^{(c)(n+1)} &= V_{F_i}^{(n)} + \frac{h}{24} * [9 * f_1(E) + 19 * f_1(A) - 5 * f_1(B) + f_1(C)] \\
 V_{R_i}^{(c)(n+1)} &= V_{R_i}^{(n)} + \frac{h}{24} * [9 * f_2(E) + 19 * f_2(A) - 5 * f_2(B) + f_2(C)] \\
 E_{fd_i}^{(c)(n+1)} &= E_{fd_i}^{(n)} + \frac{h}{24} * [9 * f_3(E) + 19 * f_3(A) - 5 * f_3(B) + f_3(C)] \\
 V_{SL_i}^{(c)(n+1)} &= V_{SL_i}^{(n)} + \frac{h}{24} * [9 * f_4(E) + 19 * f_4(A) - 5 * f_4(B) + f_4(C)] \\
 E'_{q_i}^{(c)(n+1)} &= E'_{q_i}^{(n)} + \frac{h}{24} * [9 * f_5(E) + 19 * f_5(A) - 5 * f_5(B) + f_5(C)] \\
 E'_{d_i}^{(c)(n+1)} &= E'_{d_i}^{(n)} + \frac{h}{24} * [9 * f_6(E) + 19 * f_6(A) - 5 * f_6(B) + f_6(C)] \\
 \omega_i^{(c)(n+1)} &= \omega_i^{(n)} + \frac{h}{24} * [9 * f_7(E) + 19 * f_7(A) - 5 * f_7(B) + f_7(C)] \\
 \delta_i^{(c)(n+1)} &= \delta_i^{(n)} + \frac{h}{24} * [9 * f_8(E) + 19 * f_8(A) - 5 * f_8(B) + f_8(C)] \\
 P_{c_i}^{(c)(n+1)} &= P_{c_i}^{(n)} + \frac{h}{24} * [9 * f_9(E) + 19 * f_9(A) - 5 * f_9(B) + f_9(C)] \\
 P_{H_i}^{(c)(n+1)} &= P_{H_i}^{(n)} + \frac{h}{24} * [9 * f_{10}(E) + 19 * f_{10}(A) - 5 * f_{10}(B) + f_{10}(C)] \\
 P_i^{(c)(n+1)} &= P_i^{(n)} + \frac{h}{24} * [9 * f_{11}(E) + 19 * f_{11}(A) - 5 * f_{11}(B) + f_{11}(C)]
 \end{aligned}$$

onde

$$\begin{aligned}
 A = & t^{(n)}, V_{F_i}^{(n)}, V_{R_i}^{(n)}, E_{fd_i}^{(n)}, V_{SL_i}^{(n)}, E'_{q_i}^{(n)}, E'_{d_i}^{(n)}, \omega_i^{(n)}, \delta_i^{(n)}, P_{c_i}^{(n)}, \\
 & P_{H_i}^{(n)}, P_i^{(n)}, V_i^{(n)}, V_{e_i}^{(n)}, V_{RL_i}^{(n)}, I_{d_i}^{(n)}, I_{q_i}^{(n)}, P_{mec_i}^{(n)}, P_{e_i}^{(n)}, P_{co_i}^{(n)}
 \end{aligned}$$

$$\begin{aligned}
 B = & t^{(n-1)}, V_{F_i}^{(n-1)}, V_{R_i}^{(n-1)}, E_{fd_i}^{(n-1)}, V_{SL_i}^{(n-1)}, E'_{q_i}^{(n-1)}, E'_{d_i}^{(n-1)}, \omega_i^{(n-1)}, \\
 & \delta_i^{(n-1)}, P_{c_i}^{(n-1)}, P_{H_i}^{(n-1)}, P_i^{(n-1)}, V_i^{(n-1)}, V_{e_i}^{(n-1)}, V_{RL_i}^{(n-1)}, I_{d_i}^{(n-1)}, I_{q_i}^{(n-1)}, \\
 & P_{mec_i}^{(n-1)}, P_{e_i}^{(n-1)}, P_{co_i}^{(n-1)}
 \end{aligned}$$

$$\begin{aligned}
 C = & t^{(n-2)}, V_{F_i}^{(n-2)}, V_{R_i}^{(n-2)}, E_{fd_i}^{(n-2)}, V_{SL_i}^{(n-2)}, E'_{q_i}^{(n-2)}, E'_{d_i}^{(n-2)}, \omega_i^{(n-2)}, \\
 & \delta_i^{(n-2)}, P_{c_i}^{(n-2)}, P_{H_i}^{(n-2)}, P_i^{(n-2)}, V_i^{(n-2)}, V_{e_i}^{(n-2)}, V_{RL_i}^{(n-2)}, I_{d_i}^{(n-2)}, I_{q_i}^{(n-2)}, \\
 & P_{mec_i}^{(n-2)}, P_{e_i}^{(n-2)}, P_{co_i}^{(n-2)}
 \end{aligned}$$

$$D = t^{n-3}, V_{F_i}^{n-3}, V_{R_i}^{n-3}, E_{fd_i}^{n-3}, V_{SL_i}^{n-3}, E_{q_i}'^{n-3}, E_{d_i}'^{n-3}, \omega_i^{n-3}, \delta_i^{n-3}, P_{c_i}^{n-3}, P_{H_i}^{n-3}, P_i^{n-3}, V_i^{n-3}, V_{e_i}^{n-3}, V_{RL_i}^{n-3}, I_{d_i}^{n-3}, I_{q_i}^{n-3}, P_{mec_i}^{n-3}, P_{e_i}^{n-3}, P_{co_i}^{n-3}$$

$$E = t^{n+1}, V_{F_i}^{(p)(n+1)}, V_{R_i}^{(p)(n+1)}, E_{fd_i}^{(p)(n+1)}, V_{SL_i}^{(p)(n+1)}, E_{q_i}'^{(p)(n+1)}, E_{d_i}'^{(p)(n+1)}, \omega_i^{(p)(n+1)}, \delta_i^{(p)(n+1)}, P_{c_i}^{(p)(n+1)}, P_{H_i}^{(p)(n+1)}, P_i^{(p)(n+1)}, V_i^{(n+1)}, V_{e_i}^{(n+1)}, V_{RL_i}^{(n+1)}, I_{d_i}^{(n+1)}, I_{q_i}^{(n+1)}, P_{mec_i}^{(n+1)}, P_{e_i}^{(n+1)}, P_{co_i}^{(n+1)}$$

No método de Adams a estimativa para o erro é dada por

$$\text{erro} \approx \frac{19}{270} \left| y_{n+1}^{(c)} - y_{n+1}^{(p)} \right|$$

onde

$$y_{n+1}^{(c)} = V_{F_i}^{(c)(n+1)}, V_{R_i}^{(c)(n+1)}, E_{fd_i}^{(c)(n+1)}, V_{SL_i}^{(c)(n+1)}, E_{q_i}'^{(c)(n+1)}, E_{d_i}'^{(c)(n+1)}, \omega_i^{(c)(n+1)}, \delta_i^{(c)(n+1)}, P_{c_i}^{(c)(n+1)}, P_{H_i}^{(c)(n+1)}, P_i^{(c)(n+1)}$$

e

$$y_{n+1}^{(p)} = V_{F_i}^{(p)(n+1)}, V_{R_i}^{(p)(n+1)}, E_{fd_i}^{(p)(n+1)}, V_{SL_i}^{(p)(n+1)}, E_{q_i}'^{(p)(n+1)}, E_{d_i}'^{(p)(n+1)}, \omega_i^{(p)(n+1)}, \delta_i^{(p)(n+1)}, P_{c_i}^{(p)(n+1)}, P_{H_i}^{(p)(n+1)}, P_i^{(p)(n+1)}, \text{respectivamente.}$$

3. Definição de uma biblioteca de modelos de estados de sistemas de regulação dos grupos geradores; definição do modelo de estado global do sistema; definição de perturbações a analisar.

No tópico 1, apresentei o modelo de estado do sistema mas faltava ainda calcular as condições iniciais e as restantes variáveis auxiliares. Grande parte das grandezas do sistema são expressas no campo dos complexos sendo expressas num referencial que não coincide com o referencial do alternador, sistema de eixos q e d, sendo para isso necessário fazer projecções da parte real e parte imaginária das grandezas complexas do sistema para o sistema de eixos da máquina, fig. 3.1. Da análise deste

$$\begin{aligned}
I_i &= \sqrt{I_q^2 + I_d^2} \\
V_q &= E_q' - X_d' * I_d \\
V_d &= E_d' - X_q' * I_q \\
V &= \sqrt{V_q^2 + V_d^2} \\
P_{e_i} &= E_d' * I_d + E_q' * I_q
\end{aligned} \tag{3.2}$$

Para calcular as condições iniciais são necessários determinados valores que à partida se admitem fixos, nomeadamente a tensão no barramento do gerador (V_i), potência eléctrica inicial (P_{e_i}), tensão no barramento de referência (V_{ref}), resistência equivalente (R) e reactância equivalente (X). Dados estes valores e simplificando a equação física que define a Potência eléctrica, (3.3), obtenho uma equação não linear em ordem ao ângulo Φ , (3.4). Para determinar este ângulo, tive que recorrer mais uma vez a métodos numéricos, agora para resolução de uma equação não linear. Optei pelo **método de Newton**, porque é um método muito eficaz, sendo sabido que a solução exacta se encontra no intervalo $[0, \pi/2]$, muito próxima do zero, sendo também as condições de aplicabilidade deste método satisfeitas no intervalo considerado.

$$P_{e_i} = \text{Re}\{V_i I_i^*\} \tag{3.3}$$

$$P_{e_i} = \frac{V_i^2 * R + V_i * V_{ref} * X * \sin \phi - V_i * V_{ref} * \cos \phi * R}{R^2 + X^2} \tag{3.4}$$

Determinado o ângulo, e analisando novamente o referencial, pude então efectuar operações com complexos determinar todas as condições iniciais, que são as seguintes:

$$P_{mec_i}^{(0)} = P_{e_i}^{(0)}$$

$$P_i^{(0)} = P_{mec_i}^{(0)}$$

$$P_{H_i}^{(0)} = P_i^{(0)}$$

$$P_{c_i}^{(0)} = P_{H_i}^{(0)}$$

$$\omega_i^{(0)} = 0$$

$$I_i^{(0)} = \sqrt{\left(\frac{V_i^{(0)} * \cos \phi_i^{(0)} * R + V_i^{(0)} * \sin \phi_i^{(0)} * X - R * V_{ref}}{R^2 + X^2} \right)^2 + \left(\frac{V_i^{(0)} * \sin \phi_i^{(0)} * R - V_i^{(0)} * \cos \phi_i^{(0)} * X + X * V_{ref}}{R^2 + X^2} \right)^2}$$

$$\gamma_i^{(0)} = \arccos \left(\frac{V_i^{(0)} * \cos \phi_i^{(0)} * R + V_i^{(0)} * \sin \phi_i^{(0)} * X - R * V_{ref}}{I_i^{(0)} (R^2 + X^2)} \right)$$

$$E_i'^{(0)} = \sqrt{\left(-X_{d_i}' * I_i^{(0)} * \sin \gamma_i^{(0)} + V_i^{(0)} * \cos \phi_i^{(0)} \right)^2 + \left(X_{d_i}' * I_i^{(0)} * \cos \gamma_i^{(0)} + V_i^{(0)} * \sin \phi_i^{(0)} \right)^2}$$

$$\theta_i^{(0)} = \arccos \left(-X_{d_i}' * I_i^{(0)} * \sin \gamma_i^{(0)} + V_i^{(0)} * \cos \phi_i^{(0)} \right)$$

$$E_{qa}^{(0)} = \sqrt{\left(V_i^{(0)} * \cos \phi_i^{(0)} - X_{q_i}' * I_i^{(0)} * \sin \gamma_i^{(0)} \right)^2 + \left(V_i^{(0)} * \sin \phi_i^{(0)} + X_{q_i}' * I_i^{(0)} * \cos \gamma_i^{(0)} \right)^2}$$

$$\delta_i^{(0)} = \arccos \left(\frac{V_i^{(0)} * \cos \phi_i^{(0)} - X_{q_i}' * I_i^{(0)} * \sin \gamma_i^{(0)}}{E_{qa}^{(0)}} \right)$$

$$\alpha_i^{(0)} = \delta_i^{(0)} - \theta_i^{(0)}$$

$$E_{q_i}'^{(0)} = E_i'^{(0)} * \cos \alpha_i^{(0)}$$

$$E_{d_i}'^{(0)} = E_i'^{(0)} * \sin \alpha_i^{(0)}$$

$$V_{F_i}^{(0)} = V_i^{(0)}$$

$$\beta_i'^{(0)} = \delta_i^{(0)} - \gamma_i^{(0)}$$

$$I_{d_i}^{(0)} = I_i^{(0)} * \sin \beta_i'^{(0)}$$

$$E_{fd_i}^{(0)} = \left(X_{d_i} - X_{d_i}' \right) * I_{d_i}^{(0)} + E_{q_i}'^{(0)}$$

$$V_{RL_i}^{(0)} = A_{ex} * e^{B_{ex} * E_{fd_i}^{(0)}} + K_{E_i} * E_{fd_i}^{(0)}$$

$$V_{R_i}^{(0)} = V_{RL_i}^{(0)}$$

$$V_{SL_i}^{(0)} = \frac{K_{S_i}}{T_{S_i}} E_{fd_i}^{(0)}$$

4. Especificação dos requisitos computacionais a adoptar; hardware; linguagem de programação; bases de dados ou manipulação de ficheiros.

Em termos de hardware usei um PC 486, para obter resultados em tempo útil.

Dado que o número de parâmetros do sistema é reduzido e à especificidade da aplicação, optei por um pequeno programa de gestão de ficheiros e não uma base de dados.

5. Definição e especificação do algoritmo de acordo com o modelo de estado, métodos de integração numérica a adoptar e a perturbação seleccionada.

Nesta fase implementei os algoritmos para ambos os métodos adoptados. No método de Runge-Kutta usei o passo fixo (i.e., o método clássico), uma vez que este método não dá uma estimativa do erro local. No método de Adams, faço o teste do erro para controlar o passo. E representei os fluxogramas para todo o programa, incluindo os algoritmos de cálculo e a parte dos menus (ver apêndice B).

6. Implementação do algoritmo de Runge-Kutta. Apresentação gráfica dos resultados da evolução temporal das grandezas de estado seleccionadas. Testes de validação. Comparação da resposta do sistema a perturbações de acordo com o tipo de reguladores e parâmetros seleccionados.

Nesta fase, 1º implementei o método de Newton, porque o ângulo obtido por este método é que permite calcular todas as condições iniciais necessárias para o arranque do método de Runge-Kutta. Depois implementei o método de Runge-Kutta. Dado que este trabalho tem um carácter pedagógico, implementei o método de Runge-kutta como alternativa para posteriormente comparar os resultados e a estabilidade de ambos os métodos.

Foram efectuadas várias simulações, 1º para um modelo reduzido, i.e., sem reguladores, depois com um dos reguladores apenas, e finalmente com todos os reguladores em funcionamento. Pelo método de Newton, obtive sempre um ângulo muito próximo de zero, o que já seria de esperar, e não são efectuadas mais de três iterações. Para testar o método de Runge-kutta, eu implementei o método de Euler que como já disse quando seleccionado para o passo um valor demasiado pequeno, 1ms, este método dá uma boa estabilidade numérica, mas apenas para 0.5 segundos, porque depois há que considerar os erros acumulados.

Quanto à parte gráfica destinada à apresentação das evoluções temporais das grandezas de estado, aproveitei um procedimento já desenvolvido no INESC.

7. Construção de software necessário para dotar a aplicação de uma interface amigável, e que lhe confira características didácticas.

Para definição e gestão das janelas, para desenho de texto, para desenho e controlo dos botões de diálogo, para controlo do rato e para desenho dos gráficos, usei o sistema Atega. Este sistema oferece um conjunto de rotinas gráficas destinadas ao modo gráfico de alta resolução do controlador EGA (Enhanced Graphics Adapter).

Usando este sistema, desenvolvi uma interface amigável, muito fácil de utilizar, uma vez que este trabalho tem um carácter pedagógico, onde se procurou dar toda a opção de escolha ao utilizador. Além disso, foram definidos alguns ficheiros de dados, com situações típicas. O utilizador poderá inserir outros ficheiros, ou ler ou alterar os já existentes. Todos estes ficheiros de dados são guardados no directório corrente com uma extensão previamente especificada, para poder efectuar a procura.

8. Implementação do algoritmo de Adams. Apresentação gráfica dos resultados da evolução temporal das grandezas de estado seleccionadas. Testes de validação. Comparação da resposta do sistema a perturbações de acordo com o tipo de reguladores e parametros seleccionados.

Para implementar o método de Adams-Bashforth, tive que fazer várias alterações no de Runge-Kutta para poder usar algumas das funções já definidas. Depois, implementei o método de Adams com passo constante, e finalmente o método de Adams com passo variável.

Para testar a aplicabilidade e eficiência do método de integração de Adams usei o de Runge-Kutta com passo 10 ms, pois sabe-se da literatura da especialidade que o método de Runge-Kutta com este passo de integração permite obter resultados muito precisos.

9. Equações "stiff".

Equações diferenciais "stiff" surgem frequentemente em equações físicas que representam fenómenos com constantes de tempo muito diferentes. A essência de "stiffness" é que a solução a ser calculada varia lentamente mas existem perturbações que são rapidamente enfraquecidas. A presença de tais perturbações complica o cálculo numérico da solução que varia lentamente. Normalmente a precisão exigida mantém o tamanho do passo num limite onde instabilidade não ocorre, mas nos problemas "stiff", estabilidade limita o tamanho do passo. Um procedimento baseado nos métodos de Adams é ineficiente por esta razão.

Uma família de métodos propostos por **Gear** permite resolver com sucesso as equações diferenciais "stiff" num grande número de casos. Estes métodos consistem em fórmulas implícitas baseadas em diferenciação numérica. As suas propriedades são tais que estes métodos são estáveis para todos os valores do passo e são exactos. É importante salientar que nos sistemas "stiff", estes métodos com passo 0.1 segundos permitem obter resultados com a mesma precisão que os obtidos pelo método de Adams com passo 1 ms. Portanto, estes métodos são muito eficientes em termos de tempo de execução.

Neste trabalho o problema de "stiff" só foi abordado teoricamente, porque segundo os especialistas, este problema raramente surge num sistema com uma máquina (como é o caso estudado) apesar do modelo ser muito completo, mas sim em sistemas multi-máquina em que as várias máquinas sentem a perturbação de maneira diferente, situação para a qual a formulação e os programas desenvolvidos foram preparadas para tratar.

10. Comparação dos resultados.

Comparei os resultados obtidos pelo método de Runge-Kutta com passo 10 ms com o de Adams com passo 20 ms, 30 ms e 40 ms e com passo variável, para várias simulações, e cheguei às seguintes conclusões:

- Os resultados obtidos pelo método de Adams com passo 20 ms tem a mesma precisão que os resultados obtidos pelo método de Runge-Kutta passo 10 ms. Mas em cada iteração o método de Adams envolve mais cálculo que o de Runge-Kutta.

- Os resultados obtidos pelo método de Adams com passo 40 ms são quase tão precisos quanto os de Runge-Kutta, para tempos globais de estudo inferiores a 10 segundos. Apesar do tempo de execução ser inferior ao de Runge-Kutta, para intervalos superiores a 10 segundos, os erros começam a acumular-se e as soluções de Adams começam a desviar-se das de Runge-Kutta.

- Os resultados obtidos pelo método de Adams passo 30 ms são tão precisos quanto os de Runge-Kutta, os erros acumulados não são muito significativos, e os tempos de execução são inferiores aos de Runge-Kutta. Para o Runge-Kutta foram efectuadas 1000 iterações em aproximadamente 6.5 segundos de execução; para o Adams foram efectuadas apenas 334 iterações em cerca de 5 segundos de execução (para um estudo de duração 10 segundos).

- Para o método de Adams com passo variável, obtiveram-se resultados com a mesma precisão que os de Runge-Kutta, mas com tempos de execução muito inferiores. No método de Adams foram efectuadas somente 172 iterações, e em aproximadamente 1 segundo de CPU apenas.

Nota: Ver no apêndice C uma listagem dos gráficos obtidos para estas simulações.

11. Conclusões. Objectivos alcançados.

Foi estudado teoricamente o problema de análise de estabilidade de sistemas eléctricos de energia tendo sido desenvolvido com sucesso um programa de simulação correspondente.

As vantagens e desvantagens dos métodos adoptados podem ser expressas:

- Desvantagem do método de Adams com passo variável:

O método de Adams requer o conhecimento de quatro valores para iniciar a integração e sempre que o tamanho do passo de integração é alterado. Estes podem ser calculados usando o método de Runge-Kutta.

- Vantagem do método de Adams com passo variável:

O método explícito de Runge-Kutta, precisa de um passo de integração pequeno para assegurar estabilidade numérica. Portanto maior tempo de execução. Enquanto que no de Adams, como a fórmula implícita de Adams-Moulton tem boas características de estabilidade, é possível tomar um passo de integração grande sem instabilidade numérica, o que diminui o tempo de execução, e os resultados obtidos são de igual precisão.

Desenvolvi assim uma ferramenta com fins didácticos, que poderá ser incorporada num sistema multi-máquina, e que tem como objectivo determinar a sensibilidade de certos parâmetros relativamente à estabilidade do sistema, podendo calcular os tempos críticos de eliminação do defeito para poder regular o tempo de actuação dos sistemas de protecção das centrais.

12.Bibliografia.

- Anderson and Foward, "Power System Control and Stability", Iowa State University Press.
- Yao-Nan Yu, "Electric Power System Dynamics".
- P. Henrici, "Discrete Variable Methods in Ordinary Differential Equations", Wiley, 1964.
- Ralston and Rabinowitz, "A First Course in Numerical Analysis", McGraw-Hill, 1978.
- S. D. Conte & C. de Boor, "Elementary Numerical Analysis", McGraw-Hill, 1980.
- M. R. Valença, "Métodos Numéricos", INIC, 1989.
- Gear, G. W., "Numerical Inicial Value Problems in Ordinary Differential Equations", Prentice-Hall, Englewood Cliffs, N. J., 1971.
- Shampine, L. F. e Gordon, M. K., "Computer Solution of Ordinary Differential Equations", W. H. Freeman, San Francisco, 1975.

Apêndice A

Sistemas de Regulação

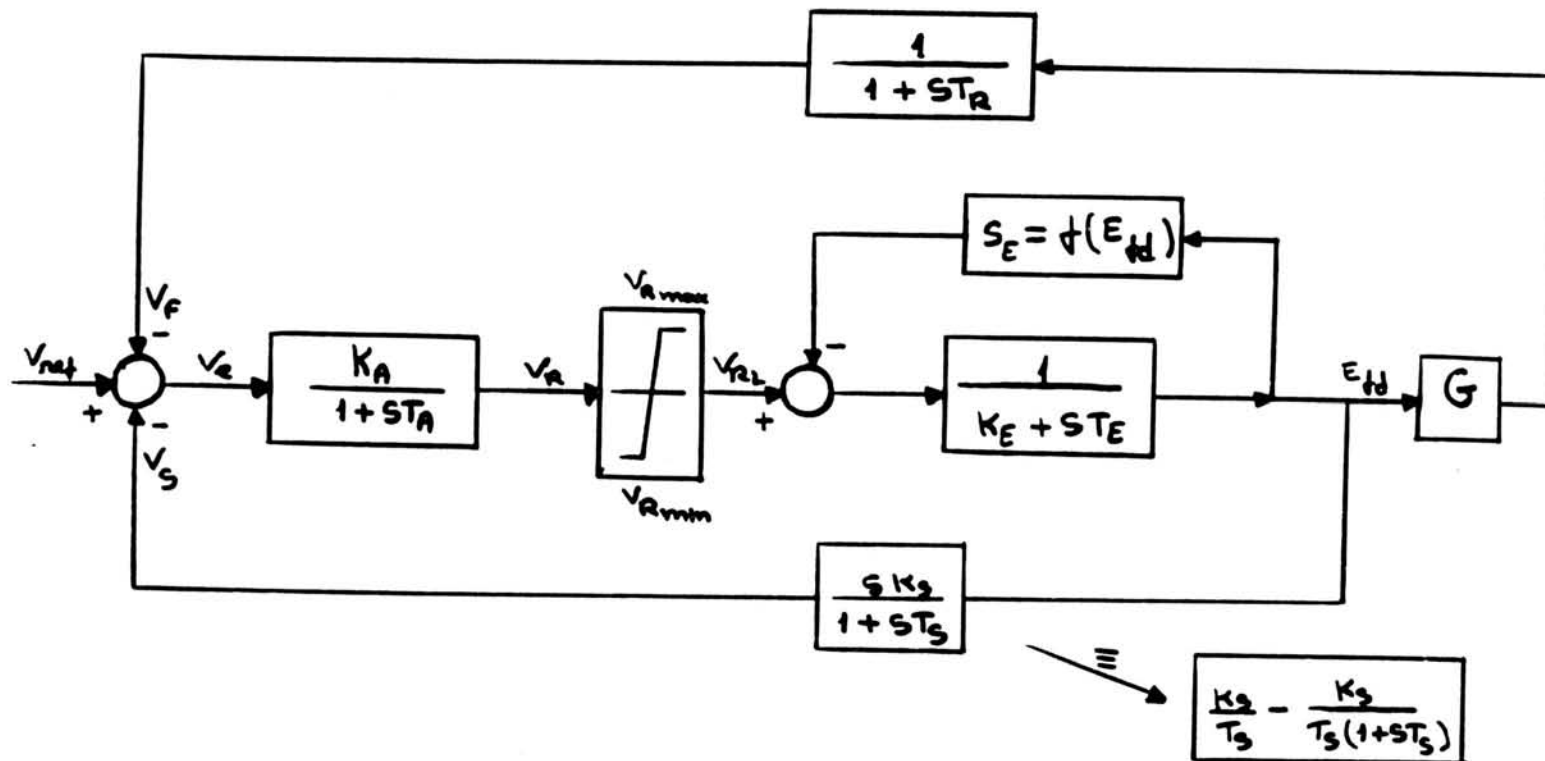


Fig. 1.2 Regulador de Tensão

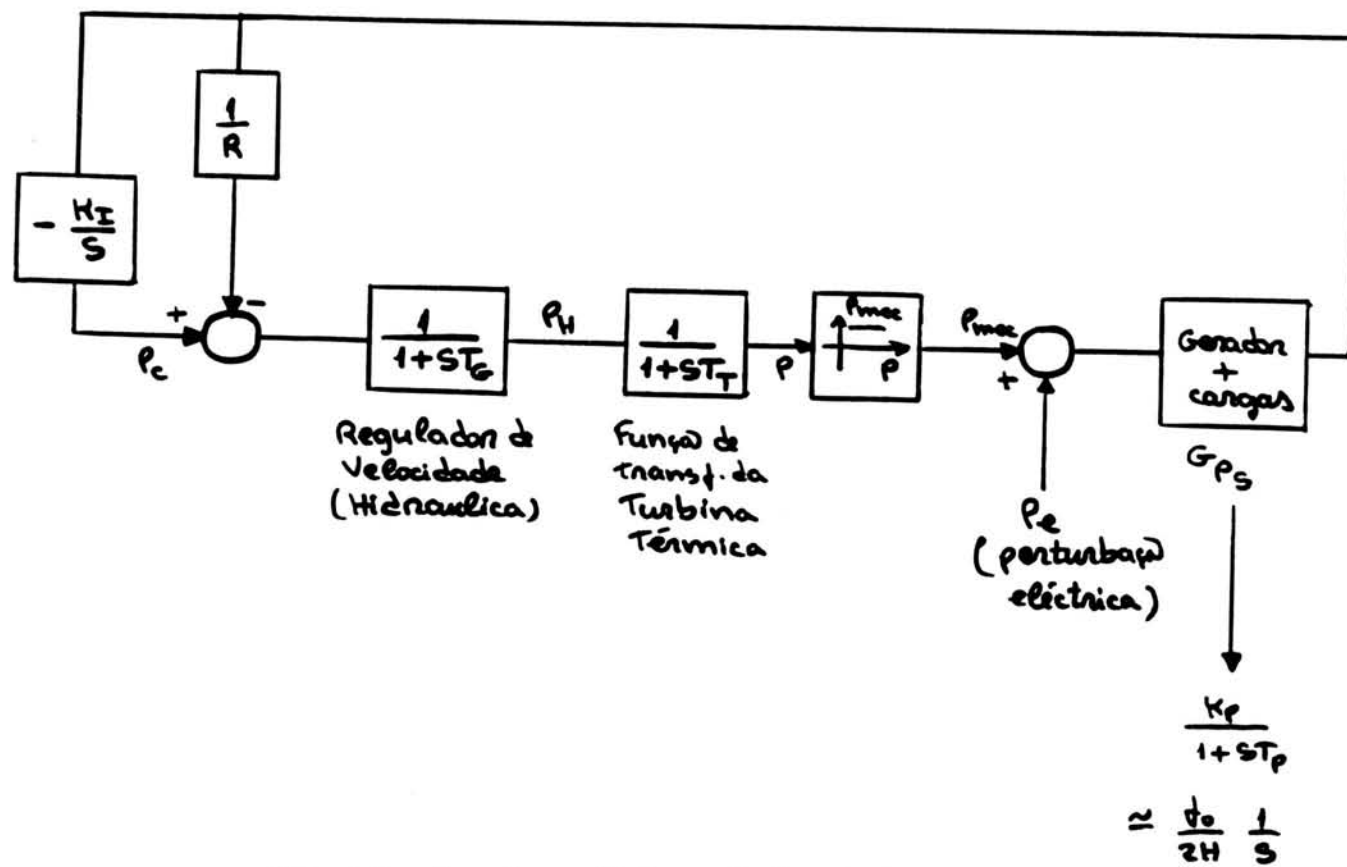
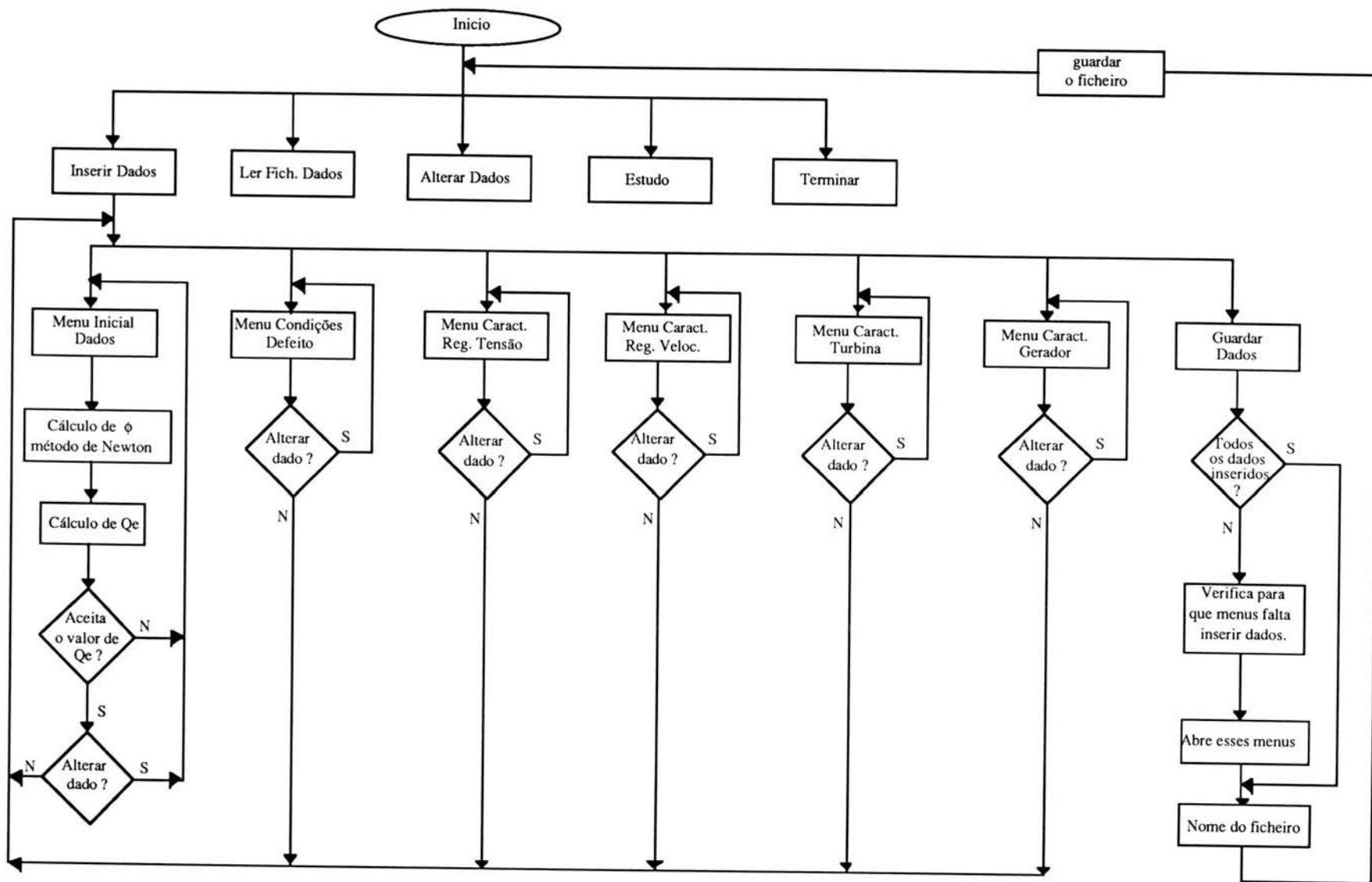
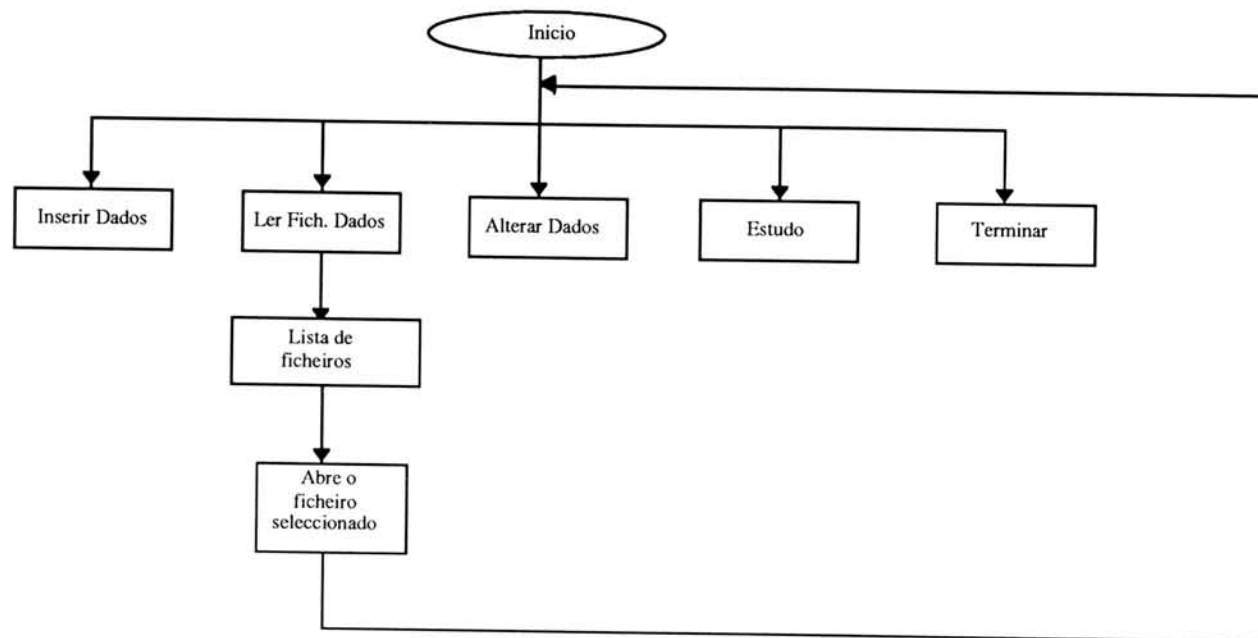


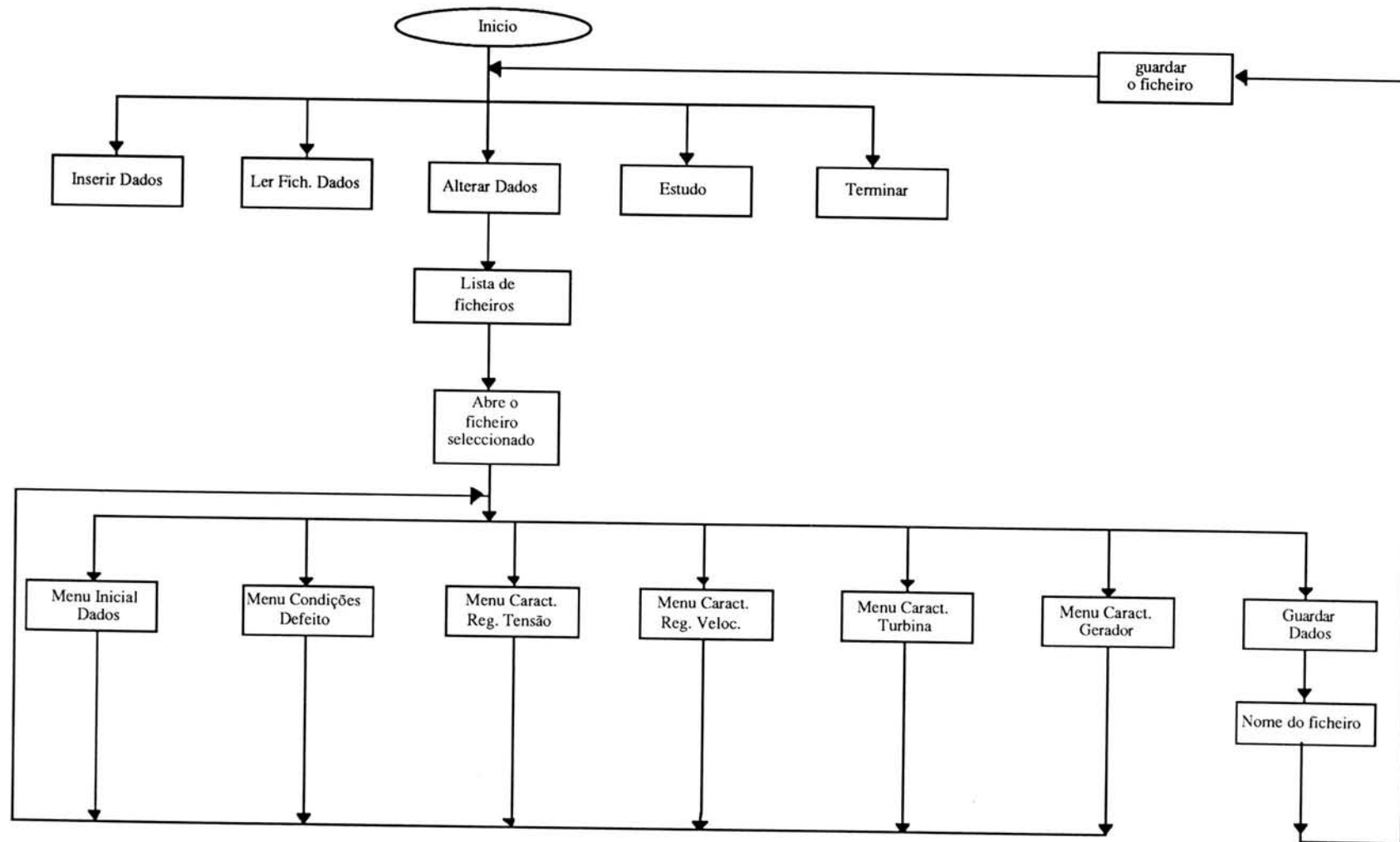
Fig. 1.3 Gerador + Sistema de Regulação de Velocidade

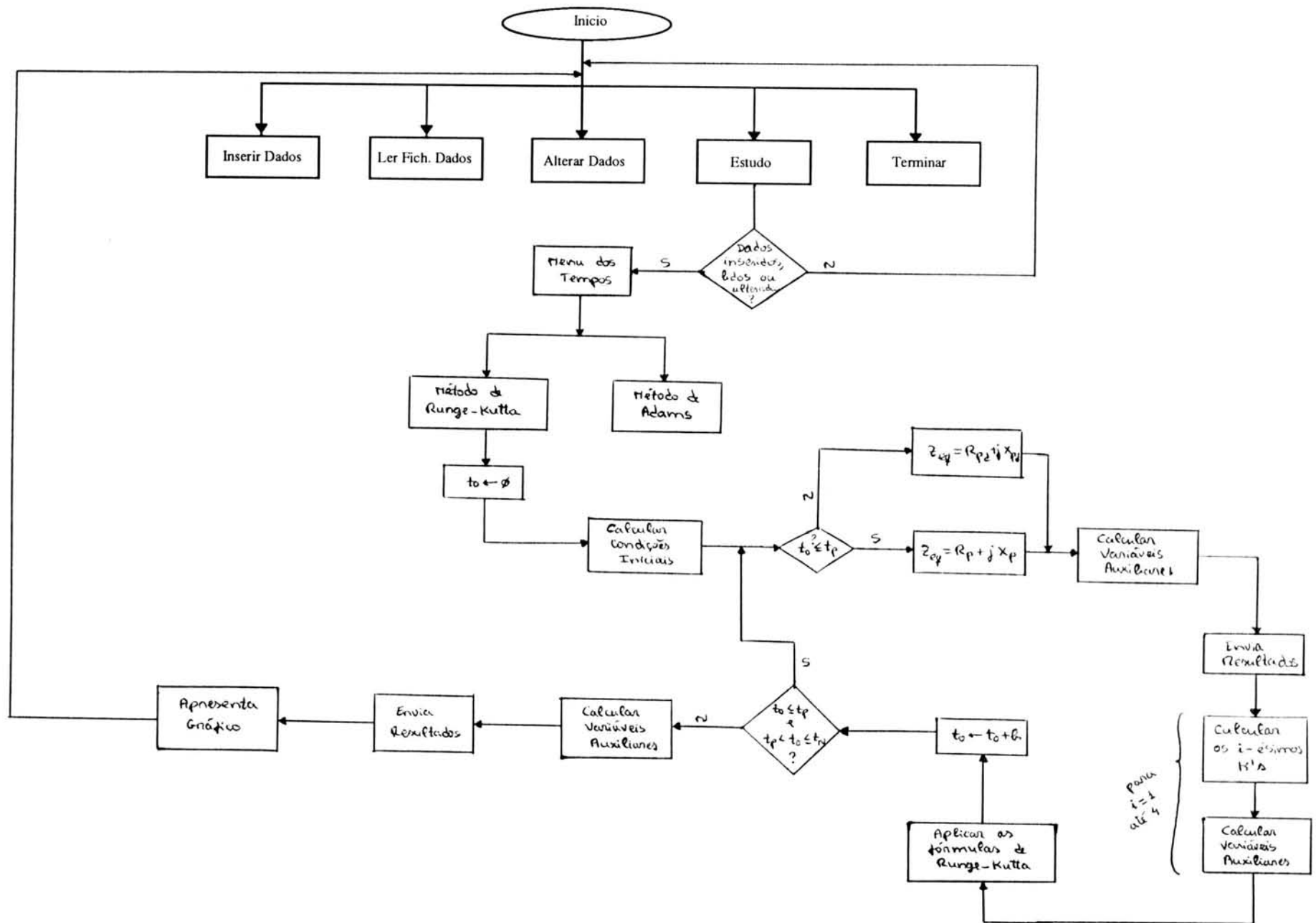
Apêndice B

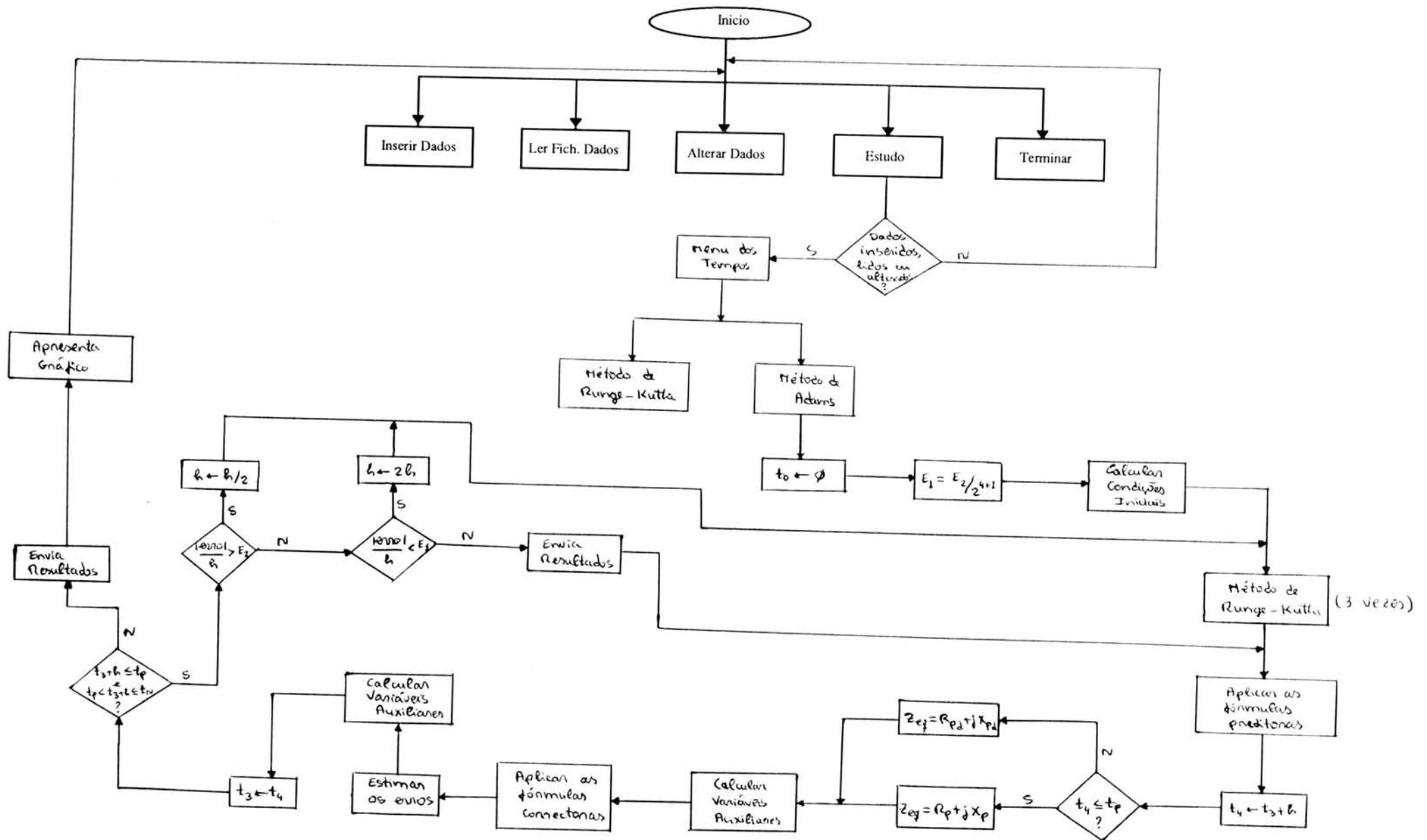
Fluxogramas





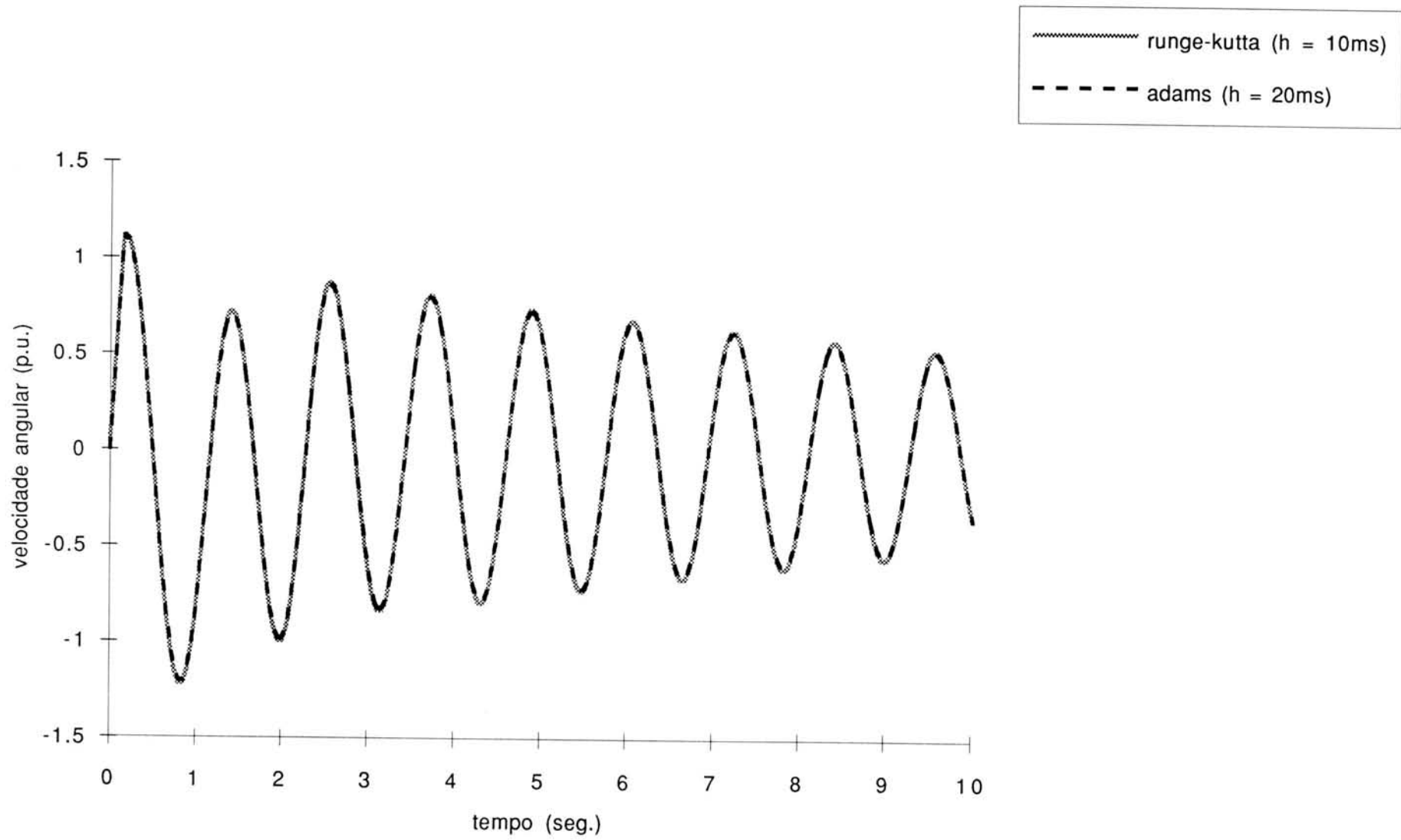


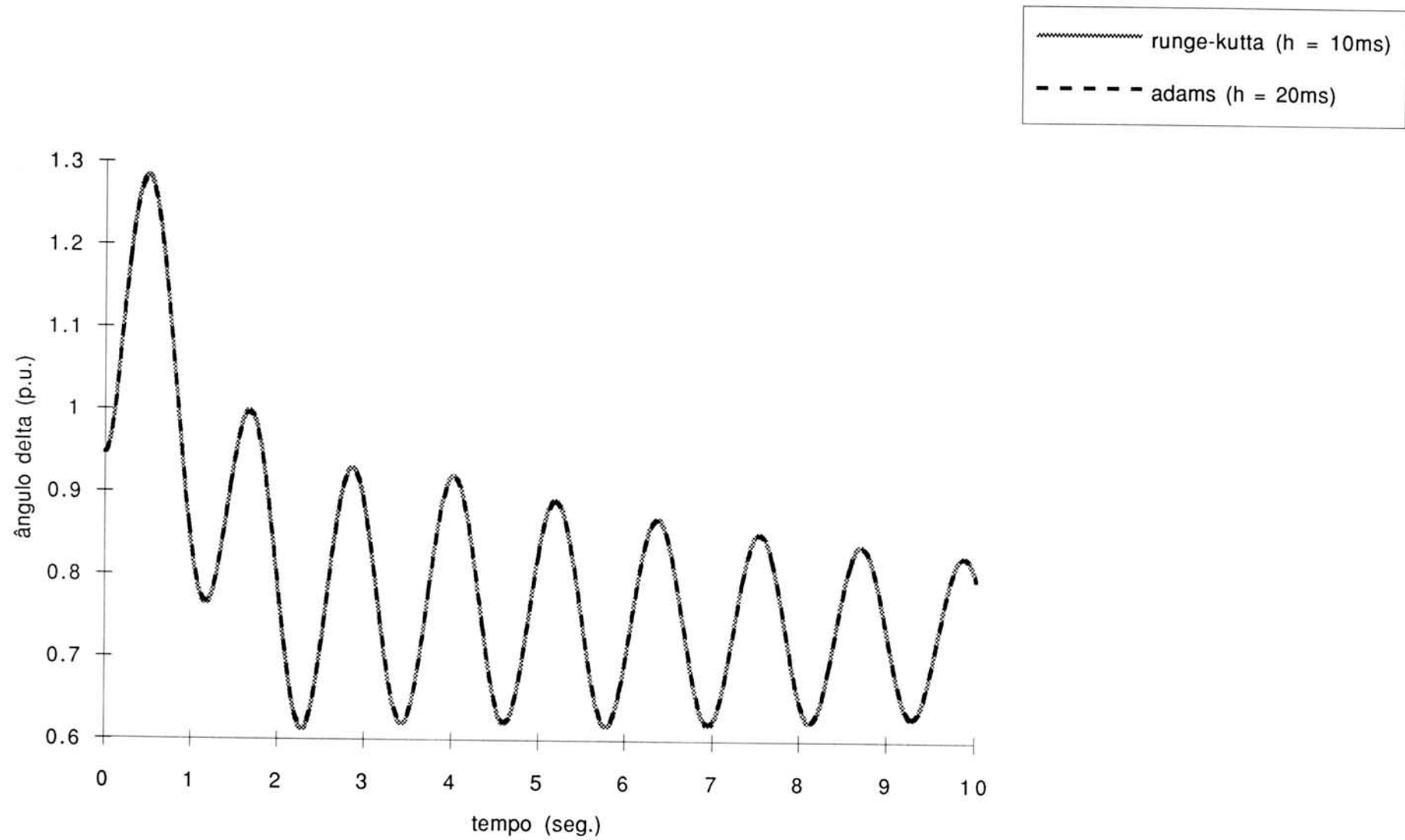


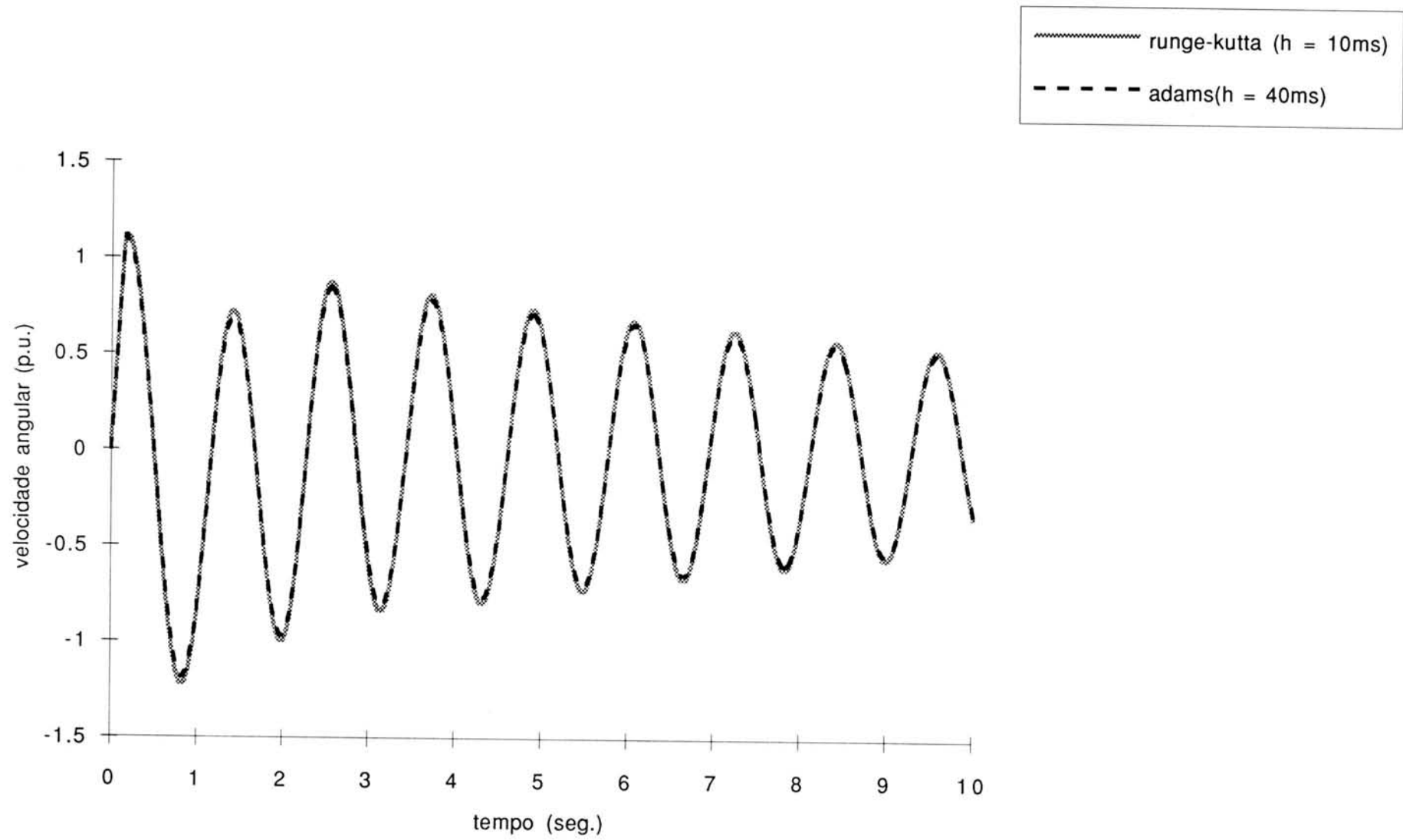


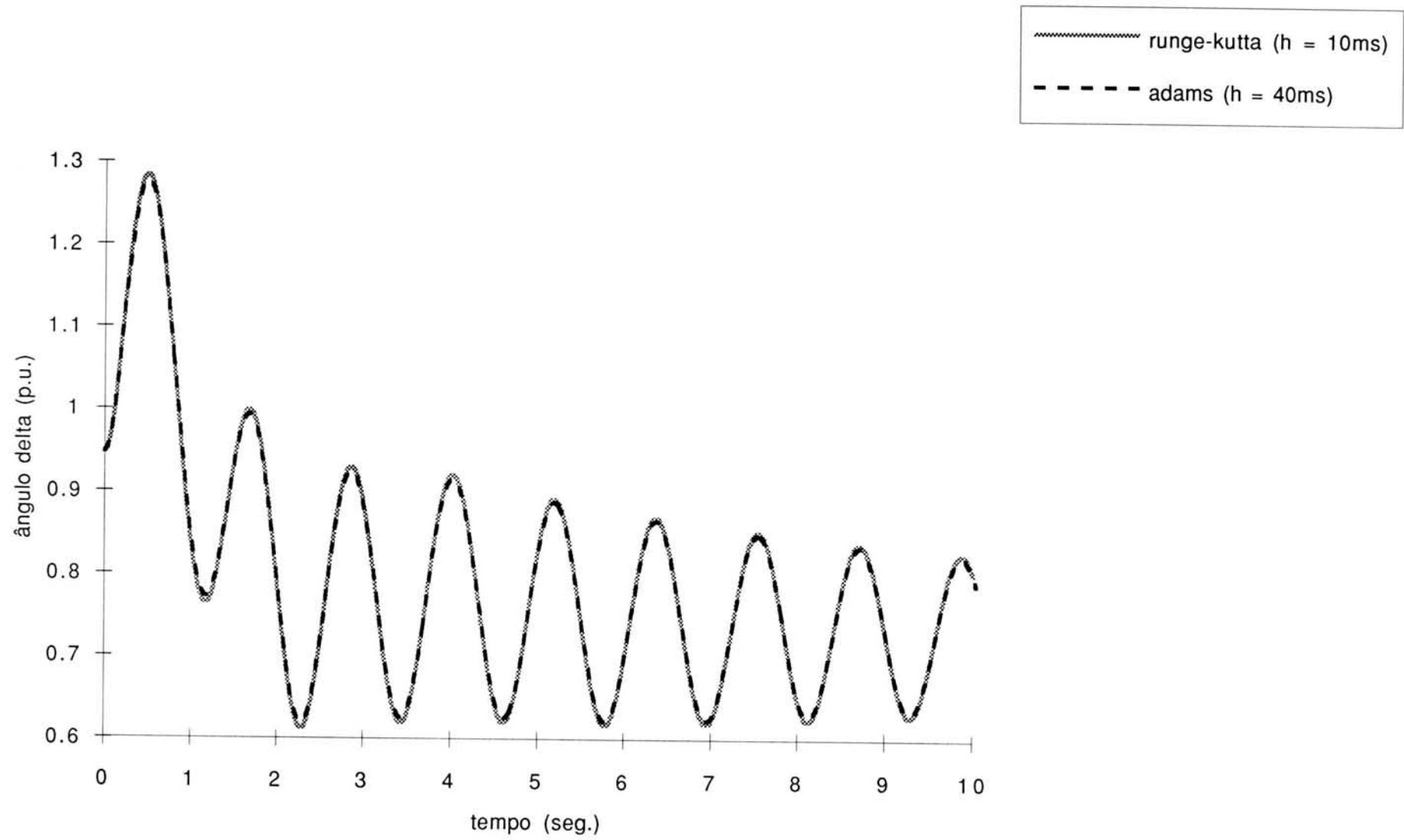
Apêndice C

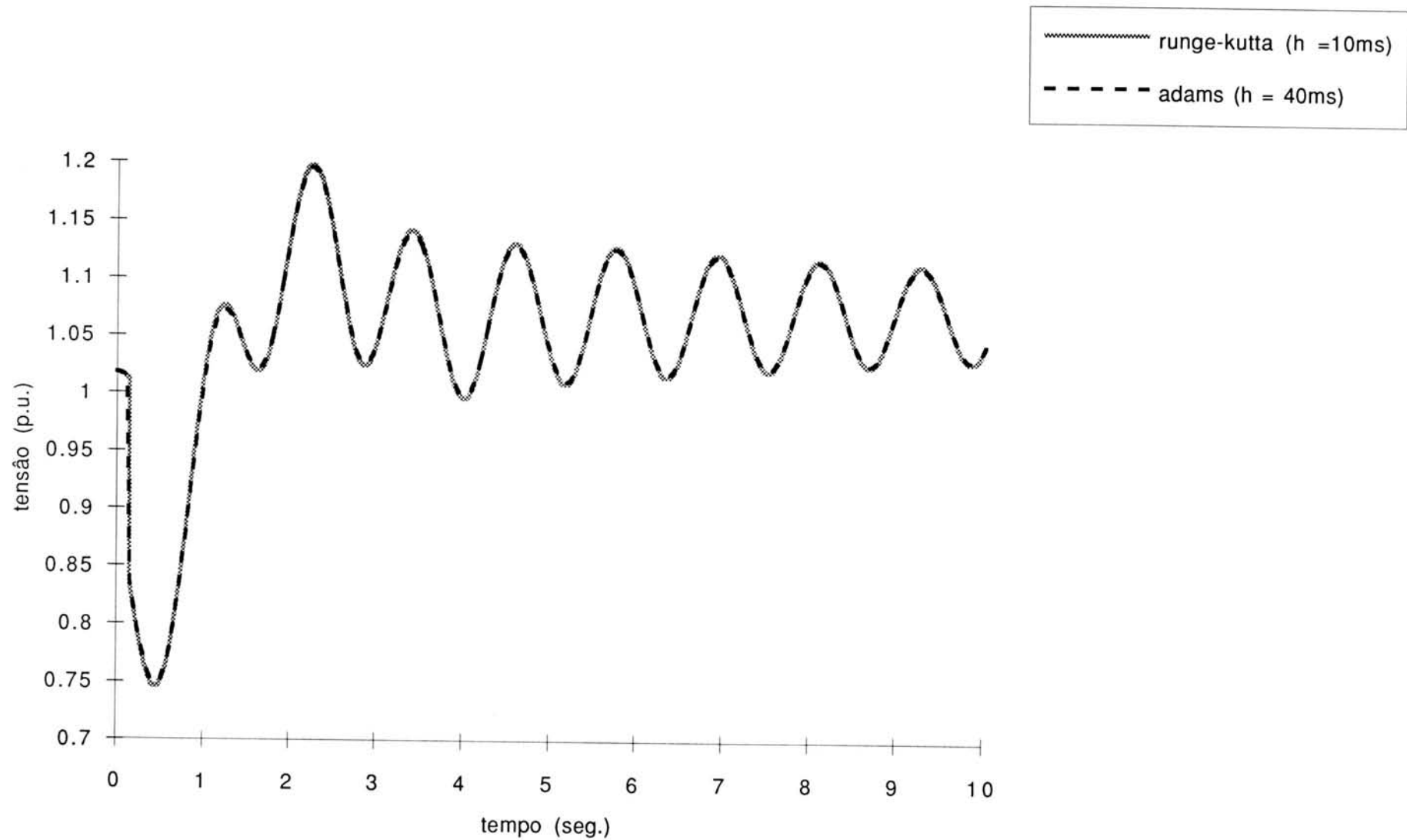
Alguns Resultados Obtidos

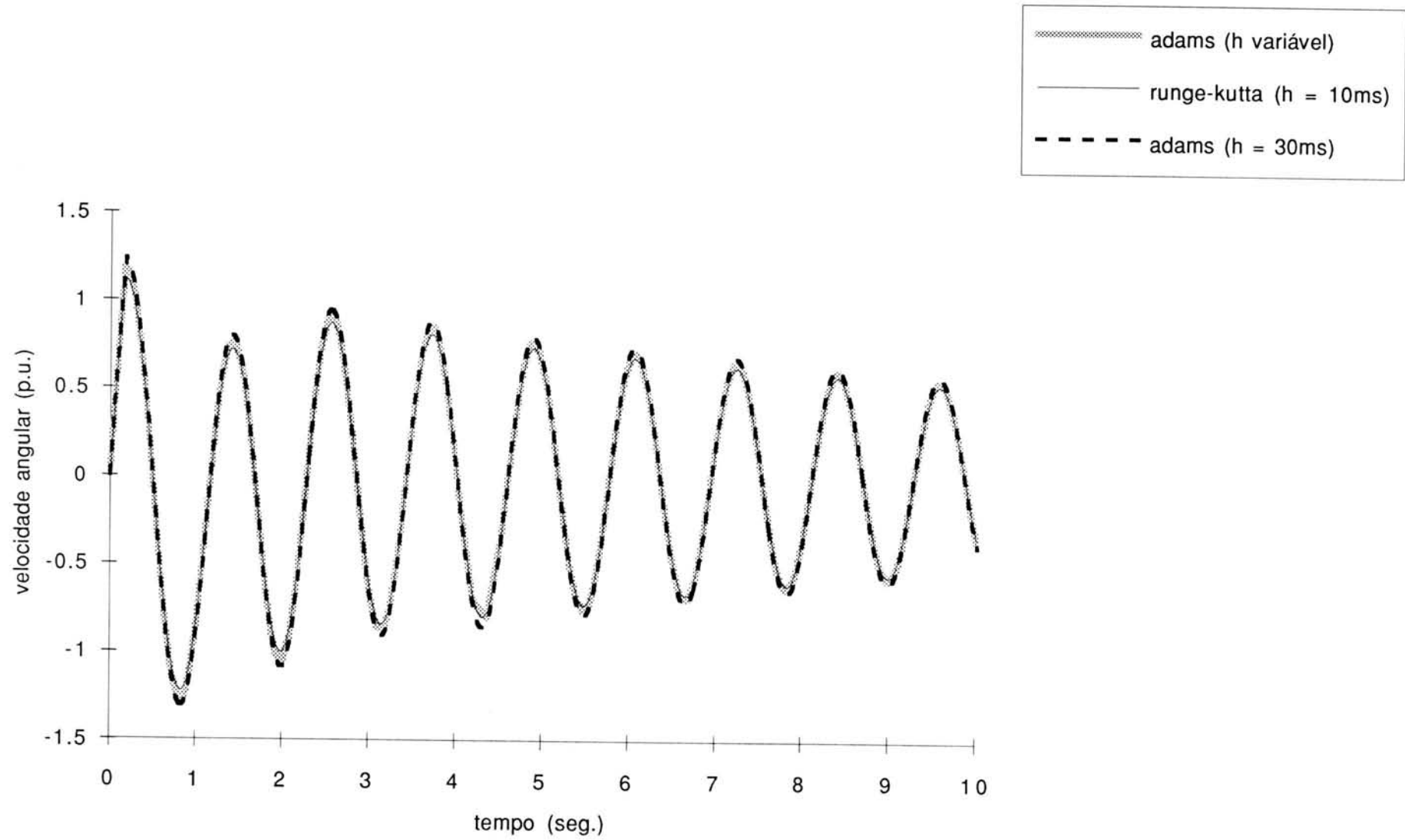


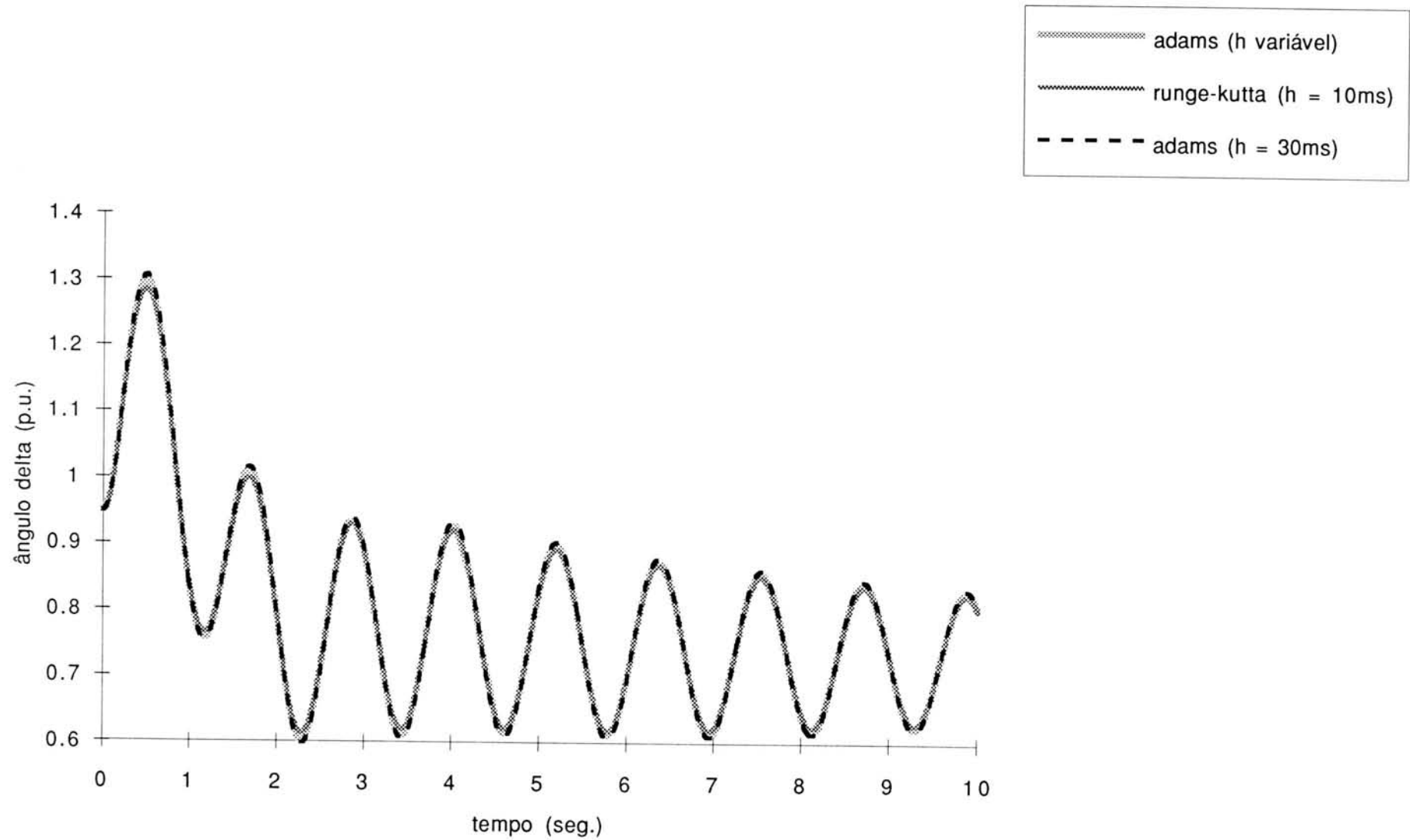


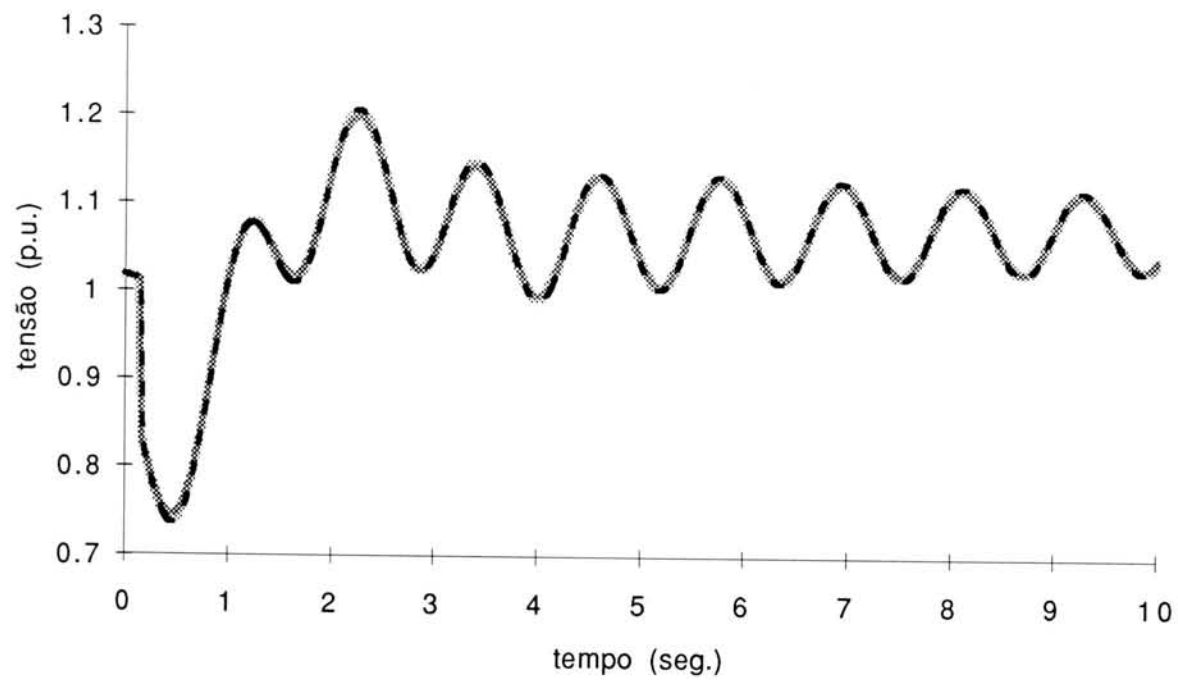












Apêndice D

Listagem do Programa ANESE

```

/***** protot.h *****/
#include <stdio.h>
#include <math.h>
#include <errno.h>

#include "vga.h"
/* #include <ega.h> */
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

#define NUM_MAQUINAS 1
#define NUM_FUNCOES 11
#define ORDEM 4
#define NUM_PASSOS 1000
#define PI 3.1415927
#define FO 50
#define KMAX 20

#define ALLSCREEN 0,0,639,349
#define INSCREEN 10,10,539,339

#define NUM_BOTOES 6

#define NAO -1
#define SIM 1
#define NUM_MAX_CHARACTER 9

#define NUM_VAR 6

double pow(double,double);
double acos(double);
double sin(double);
double cos(double);
double exp(double);
double fabs(double);
double sqrt(double);

double funct1(int);
double funct2(int);
double funct3(int);
double funct4(int);
double funct5(int);
double funct6(int);
double funct7(int);
double funct8(int);
double funct9(int);

double funct10(int);
double funct11(int);

double newton();

float LerFloat(int,int,int,int,int);

FILE *fopen();

```

```
/****** structs.h *****/

typedef struct menu_inicial{
    double pe0;
    double vref;
    double v0;
    double ro;
    double x;
} m_ini;

typedef struct menu_defeito{
    double rp;
    double xp;
    double rpd;
    double xpd;
} m_def;

typedef struct menu_reg_tensao{
    double tr;
    double ka;
    double ta;
    double vrmax;
    double vrmin;
    double aex;
    double bex;
    double ke;
    double te;
    double ks;
    double ts;
} m_ten;

typedef struct menu_reg_veloc{
    double r;
    double ki;
    double tg;
} m_vel;

typedef struct menu_turbina{
    double tt;
    double pmecmax;
    double m;
} m_tur;
```

```
/****** externs.h *****/

#include "protot.h"
#include "structs.h"

extern double h, tp, tn;

extern double t0;
extern double t[NUM_PASSOS];

extern int i;

extern m_ini dad_ini[];
extern m_def dad_def[];
extern m_ten dad_ten[];
extern m_vel dad_vel[];
extern m_tur dad_tur[];
extern m_ger dad_ger[];

extern double mt;

extern double fi0;

extern double vref0;

extern double vf0;
extern double vr0;
extern double efd0;
extern double vs10;
extern double eqlinha0;
extern double edlinha0;
extern double w0;
extern double delta0;
extern double pc0;
extern double ph0;
extern double p0;

extern double vf1;
extern double vr1;
extern double efd1;
extern double vs11;
extern double eqlinha1;
extern double edlinha1;
extern double w1;
extern double delta1;
extern double pc1;
extern double ph1;

extern double p1;

extern double vf[];
extern double vr[];
extern double efd[];
extern double vs1[];
extern double eqlinha[];
extern double edlinha[];
extern double w[];
extern double delta[];
extern double pc[];
extern double ph[];
extern double p[];
extern double t1;

extern double v[];
extern double ve[];
extern double vrl[];
extern double id[];
extern double iq[];
extern double pmec[];
extern double pe[];
extern double pco[];

extern double i0;
extern double fil;
extern double elinha0;
extern double betalinha0;

extern double k[NUM_FUNCÕES][ORDEM];

extern int dados;
extern int posicao;

extern char filename[NUM_MAX_CARACT];

extern int cont;
extern int adam;
extern int alterah;
extern int pontos_adams;
```

```

/***** externs.c *****/
/

#include "protot.h"
#include "structs.h"

double h,tp,tn;

double t0=0.0;
double t[NUM_PASSOS];

int i=0;

m_ini dad_ini[NUM_MAQUINAS];
m_def dad_def[NUM_MAQUINAS];
m_ten dad_ten[NUM_MAQUINAS];
m_vel dad_vel[NUM_MAQUINAS];
m_tur dad_tur[NUM_MAQUINAS];
m_ger dad_ger[NUM_MAQUINAS];

double mt=0.0;

double fi0;

double vref0;

double vf0;
double vr0;
double efd0;
double vsl0;
double eqlinha0;
double edlinha0;
double w0;
double delta0;
double pc0;
double ph0;
double p0;

double vfl;
double vrl;
double efdl;
double vsl1;
double eqlinha1;
double edlinha1;
double wl;
double deltal;
double pcl;

double phl;
double pl;

double vf[NUM_VAR];
double vr[NUM_VAR];
double efd[NUM_VAR];
double vsl[NUM_VAR];
double eqlinha[NUM_VAR];
double edlinha[NUM_VAR];
double w[NUM_VAR];
double delta[NUM_VAR];
double pc[NUM_VAR];
double ph[NUM_VAR];
double p[NUM_VAR];
double tl;

double v[NUM_VAR];
double ve[NUM_VAR];
double vrl[NUM_VAR];
double id[NUM_VAR];
double iq[NUM_VAR];
double pmec[NUM_VAR];
double pe[NUM_VAR];
double pco[NUM_VAR];

double i0;
double fil;
double elinha0;
double betalinha0;

double k[NUM_FUNCOES][ORDEM];

int dados;
int posicao;

char filename[NUM_MAX_CARACT];

int cont=0;
int adam=0;
int alterah=0;
int pontos_adams;

```

```
/****** anese.c *****/
```

```
#include "externs.h"
```

```
main()
{
    InitAtega(EGA); /* InitAtega(); */
    InitTexega();
    LoadFont(0,"system");
    SetGraphicsClip(ALLSCREEN);
    MouseSpan(1,1,639,349);
    ActivateFont(0);
    SetPenSize(1,1);
    DefineCursor(283,BLACK,284,WHITE+INTS);
    dados=0;
    posicao=0;
    arranca();
    ExitAtega();
}
```

```
/****** newton.c *****/
```

```
#include "externs.h"
```

```
double newton()
{
    double e;
    double erro;
    double f0, flinha0;
    double fil=0.0;
    double aux1, aux2, aux3, aux4;
    int k=0;
    e=pow(10.,-3.);
    aux1=pow(dad_ini[i].v0,2.)*dad_ini[i].ro;
    aux4=(pow(dad_ini[i].ro,2.)+pow(dad_ini[i].x,2.));
    do {
        fi0=fil;
        aux2=dad_ini[i].v0*dad_ini[i].vref*dad_ini[i].x*sin(fi0);
        aux3=dad_ini[i].v0*dad_ini[i].vref*cos(fi0)*dad_ini[i].ro;
        f0=aux1+aux2-aux3-dad_ini[i].pe0*aux4;
        flinha0=dad_ini[i].v0*dad_ini[i].vref*dad_ini[i].x*cos(fi0)+dad_ini[i].v0*dad_ini[i].vref*dad_ini[i].ro*sin(fi0);
        fil=f0/flinha0;
        k=k+1;
        erro=fabs(fil-fi0)/fabs(fil);
    } while (fabs(fil-fi0)/fabs(fil)>=e && k<=KMAX); /* enddo */
    if (fabs(fil-fi0)/fabs(fil)<e){
        aux1=pow(dad_ini[i].v0,2.)*dad_ini[i].x;
        aux2=dad_ini[i].v0*dad_ini[i].vref*cos(fil)*dad_ini[i].x;
        aux3=dad_ini[i].v0*dad_ini[i].vref*sin(fil)*dad_ini[i].ro;
        fi0=fil;
        return{(aux1-aux2-aux3)/aux4};
    } /* endif */
}
```

```

/***** modelo.c *****/
#include "externs.h"

double funct1(ind)
int ind;
{
    double aux;
    aux=(v[ind]-vf[ind])/dad_ten[i].tr;
    return(aux);
}

double funct2(ind)
int ind;
{
    double aux;
    aux=(dad_ten[i].ka*ve[ind]-vr[ind])/dad_ten[i].ta;
    return(aux);
}

double funct3(ind)
int ind;
{
    double aux,aux2;
    aux=dad_ten[i].bex*efd[ind];
    aux2=(vr1[ind]-dad_ten[i].aex*exp(aux)-dad_ten[i].ke*efd[ind])/dad_ten[i].te;
    return(aux2);
}

double funct4(ind)
int ind;
{
    double aux;
    aux=((dad_ten[i].ks/dad_ten[i].ts)*efd[ind]-vsl[ind])/dad_ten[i].ts;
    return(aux);
}

double funct5(ind)
int ind;
{
    double aux;

    aux=(1/dad_ger[i].tdolinha)*(efd[ind]-(dad_ger[i].xd-dad_ger[i].xdlinha)*id[ind]-eqlinha[ind]);
    return(aux);
}

double funct6(ind)
int ind;
{
    double aux;
    aux=(1/dad_ger[i].tdolinha)*((dad_ger[i].xq-dad_ger[i].xqlinha)*iq[ind]-edlinha[ind]);
    return(aux);
}

double funct7(ind)
int ind;
{
    double aux;
    aux=(pmec[ind]-pe[ind]-(dad_tur[i].m/mt)*pco[ind])/dad_tur[i].m;
    return(aux);
}

double funct8(ind)
int ind;
{
    return(w[ind]);
}

double funct9(ind)
int ind;
{
    double aux;
    aux=(-dad_vel[i].ki/(2*PI*FO))*w[ind];
    return(aux);
}

double funct10(ind)
int ind;
{
    double aux;
    aux=(pc[ind]-(1/(2*PI*FO*dad_vel[i].r))*w[ind]-ph[ind])/dad_vel[i].tg;
    return(aux);
}

```

```
double funct11(ind)
int ind;
{
    double aux;
    aux=(ph[ind]-p[ind])/dad_tur[i].tt;
    return(aux);
}
```

```

/***** rkutta.c *****/

#include "externs.h"

FILE *ap;

void cond_inic()
{
    double aux1,aux2,aux0;
    double gama0;
    double eqa0;
    double teta0;
    double alfa0;
    pmec[0]=dad_ini[i].pe0;
    p0=pmec[0];
    ph0=p0;
    pc0=ph0;
    w0=0.;
    aux0=pow(dad_ini[i].ro,2.)+pow(dad_ini[i].x,2.);
    aux1=(dad_ini[i].v0*cos(fi0)*dad_ini[i].ro+dad_ini[i].v0*sin(fi0)*dad_ini[i].x-dad_ini[i].ro*dad_ini[i].vref)/aux0;
    aux2=(dad_ini[i].v0*sin(fi0)*dad_ini[i].ro-dad_ini[i].v0*cos(fi0)*dad_ini[i].x+dad_ini[i].x*dad_ini[i].vref)/aux0;
    i0=sqrt(pow(aux1,2.)+pow(aux2,2.));
    gama0=acos(aux1/i0);
    aux1=-dad_ger[i].xdlinha*i0*sin(gama0)+dad_ini[i].v0*cos(fi0);
    aux2=dad_ger[i].xdlinha*i0*cos(gama0)+dad_ini[i].v0*sin(fi0);
    elinha0=sqrt(pow(aux1,2.)+pow(aux2,2.));
    teta0=acos(aux1/elinha0);
    aux1=dad_ini[i].v0*cos(fi0)-dad_ger[i].xq*i0*sin(gama0);
    aux2=dad_ini[i].v0*sin(fi0)+dad_ger[i].xq*i0*cos(gama0);
    eqa0=sqrt(pow(aux1,2.)+pow(aux2,2.));
    delta0=acos(aux1/eqa0);
    alfa0=delta0-teta0;
    eqlinha0=elinha0*cos(alfa0);
    edlinha0=elinha0*sin(alfa0);
    vf0=dad_ini[i].v0;
    betalinha0=delta0-gama0;
    id[0]=i0*sin(betalinha0);
    efd0=(dad_ger[i].xd-dad_ger[i].xdlinha)*id[0]+eqlinha0;
    vrl[0]=dad_ten[i].aex*exp(dad_ten[i].bex*efd0)+dad_ten[i].ke*efd0;
    vref0=vrl[0]/dad_ten[i].ka+dad_ini[i].v0;
    vr0=vrl[0];
    vsl0=(dad_ten[i].ks/dad_ten[i].ts)*efd0;

    double vq0;
    double vd0;
    double vs0;
    if (t0<=tp){
        raux=dad_def[i].rp;
        xaux=dad_def[i].xp;
    } else {
        raux=dad_def[i].rpd;
        xaux=dad_def[i].xpd;
    } /* endif */
    aux1=xaux+dad_ger[i].xdlinha;
    aux2=pow(raux,2.)+pow(xaux,2.);
    aux3=cos(delta[ind]);
    aux4=sin(delta[ind]);
    iq[ind]=(-aux1*edlinha[ind]+raux*eqlinha[ind]-(raux*aux3-aux1*aux4)*dad_ini[i].vref)/aux2;
    id[ind]=(raux*edlinha[ind]+aux1*eqlinha[ind]-(raux*aux4+aux1*aux3)*dad_ini[i].vref)/aux2;
    vq0=eqlinha[ind]-dad_ger[i].xdlinha*id[ind];
    vd0=edlinha[ind]-dad_ger[i].xqlinha*iq[ind];
    v[ind]=sqrt(pow(vq0,2.)+pow(vd0,2.));
    i0=sqrt(pow(iq[ind],2.)+pow(id[ind],2.));
    pe[ind]=edlinha[ind]*id[ind]+(eqlinha[ind]+(dad_ger[i].xqlinha-dad_ger[i].xdlinha)*id[ind])*iq[ind];
    vs0=(dad_ten[i].ks/dad_ten[i].ts)*efd[ind]-vsl[ind];
    ve[ind]=vref0-vf[ind]-vs0;
    if (vr[ind]>dad_ten[i].vrmax){
        vrl[ind]=dad_ten[i].vrmax;
    } else {
        if (vr[ind]<dad_ten[i].vrmin){
            vrl[ind]=dad_ten[i].vrmin;
        } else {
            vrl[ind]=vr[ind];
        } /* endif */
    } /* endif */
    if (p[ind]>dad_ten[i].pmecmax){
        pmec[ind]=dad_ten[i].pmecmax;
    } else {
        if (p[ind]<0){
            pmec[ind]=0.;
        } else {
            pmec[ind]=p[ind];
        } /* endif */
    } /* endif */
    pco[ind]=0.;
}

void envia_result(ind)
int ind;
{
    fprintf(ap,"%3.3lf ",t0);
    fprintf(ap,"%t%3.6lf\t%3.6lf\t%3.6lf\t%3.6lf\t%3.6lf\t%3.6lf\t%3.6lf\t%3.6lf",efd[ind],v[ind],i0,pe[ind],delta[ind],w[ind],pmec[ind],elinha0);
    fprintf(ap,"\n");
}

void novas_var_aux(ind)
int ind;
{
    double aux1,aux2,aux3,aux4;
    double raux,xaux;

```

```

void guarda_val_rkutta()
{
    t[cont]=t0;
    if (cont==0)
        cont=5;
    vf[cont]=vf[0];
    vr[cont]=vr[0];
    efd[cont]=efd[0];
    vs1[cont]=vs1[0];
    eqlinha[cont]=eqlinha[0];
    edlinha[cont]=edlinha[0];
    w[cont]=w[0];
    delta[cont]=delta[0];
    pc[cont]=pc[0];
    ph[cont]=ph[0];
    p[cont]=p[0];
    v[cont]=v[0];
    ve[cont]=ve[0];
    vrl[cont]=vrl[0];
    id[cont]=iq[0];
    iq[cont]=iq[0];
    pe[cont]=pe[0];
    pmec[cont]=pmec[0];
    pco[cont]=pco[0];

    if (cont==5)
        cont=1;
    else
        cont++;
}

void prep_calculo()
{
    vf[0]=vf0;
    vr[0]=vr0;
    efd[0]=efd0;
    vs1[0]=vs10;
    eqlinha[0]=eqlinha0;
    edlinha[0]=edlinha0;
    w[0]=w0;
    delta[0]=delta0;
    pc[0]=pc0;
    ph[0]=ph0;
    p[0]=p0;
}

void formulas1()
{
    int l;
    double f[NUM_FUNCÕES];
    prep_calculo();
    t1=t0;
    if (alterah==0){

```

```

        novas_var_aux(0);
        envia_result(0);
    }
    else
        alterah=0;
    if (adam==1){
        guarda_val_rkutta();
        pontos_adams+=1;
    }
    f[0]=funct1(0);
    f[1]=funct2(0);
    f[2]=funct3(0);
    f[3]=funct4(0);
    f[4]=funct5(0);
    f[5]=funct6(0);
    f[6]=funct7(0);
    f[7]=funct8(0);
    f[8]=funct9(0);
    f[9]=funct10(0);
    f[10]=funct11(0);
    for (l=0;l<NUM_FUNCÕES;l++){
        k[l][0]=h*f[l];
    } /* endfor */
}

void formulas2()
{
    int l;
    double f[NUM_FUNCÕES];
    vf[0]=vf0+(k[0][0]/2.);
    vr[0]=vr0+(k[1][0]/2.);
    efd[0]=efd0+(k[2][0]/2.);
    vs1[0]=vs10+(k[3][0]/2.);
    eqlinha[0]=eqlinha0+(k[4][0]/2.);
    edlinha[0]=edlinha0+(k[5][0]/2.);
    w[0]=w0+(k[6][0]/2.);
    delta[0]=delta0+(k[7][0]/2.);
    pc[0]=pc0+(k[8][0]/2.);
    ph[0]=ph0+(k[9][0]/2.);
    p[0]=p0+(k[10][0]/2.);
    t1=t0+(h/2.);
    novas_var_aux(0);
    f[0]=funct1(0);
    f[1]=funct2(0);
    f[2]=funct3(0);
    f[3]=funct4(0);
    f[4]=funct5(0);
    f[5]=funct6(0);
    f[6]=funct7(0);
    f[7]=funct8(0);
    f[8]=funct9(0);
    f[9]=funct10(0);
    f[10]=funct11(0);
    for (l=0;l<NUM_FUNCÕES;l++){
        k[l][1]=h*f[l];
    } /* endfor */
}

```

```

void formulas3()
{
    int i;
    double f[NUM_FUNCIONES];
    vf[0]=vf0+(k[0][1]/2.);
    vr[0]=vr0+(k[1][1]/2.);
    efd[0]=efd0+(k[2][1]/2.);
    vs1[0]=vs10+(k[3][1]/2.);
    eqlinha[0]=eqlinha0+(k[4][1]/2.);
    edlinha[0]=edlinha0+(k[5][1]/2.);
    w[0]=w0+(k[6][1]/2.);
    delta[0]=delta0+(k[7][1]/2.);
    pc[0]=pc0+(k[8][1]/2.);
    ph[0]=ph0+(k[9][1]/2.);
    p[0]=p0+(k[10][1]/2.);
    t1=t0+(h/2.);
    novas_var_aux(0);
    f[0]=funct1(0);
    f[1]=funct2(0);
    f[2]=funct3(0);
    f[3]=funct4(0);
    f[4]=funct5(0);
    f[5]=funct6(0);
    f[6]=funct7(0);
    f[7]=funct8(0);
    f[8]=funct9(0);
    f[9]=funct10(0);
    f[10]=funct11(0);
    for (i=0;i<NUM_FUNCIONES;i++){
        k[i][2]=h*f[i];
    } /* endfor */
}

```

```

void formulas4()
{
    int i;
    double f[NUM_FUNCIONES];
    vf[0]=vf0+k[0][2];
    vr[0]=vr0+k[1][2];
    efd[0]=efd0+k[2][2];
    vs1[0]=vs10+k[3][2];
    eqlinha[0]=eqlinha0+k[4][2];
    edlinha[0]=edlinha0+k[5][2];
    w[0]=w0+k[6][2];
    delta[0]=delta0+k[7][2];
    pc[0]=pc0+k[8][2];
    ph[0]=ph0+k[9][2];
    p[0]=p0+k[10][2];
    t1=t0+h;
    novas_var_aux(0);
    f[0]=funct1(0);
    f[1]=funct2(0);
    f[2]=funct3(0);
    f[3]=funct4(0);
    f[4]=funct5(0);

```

```

    f[5]=funct6(0);
    f[6]=funct7(0);
    f[7]=funct8(0);
    f[8]=funct9(0);
    f[9]=funct10(0);
    f[10]=funct11(0);
    for (i=0;i<NUM_FUNCIONES;i++){
        k[i][3]=h*f[i];
    } /* endfor */
}

```

```

void formulas_rkutta()
{
    int i;
    formulas1();
    formulas2();
    formulas3();
    formulas4();
    vf1=vf0+(1./6.)*(k[0][0]+2.*k[0][1]+2.*k[0][2]+k[0][3]);
    vr1=vr0+(1./6.)*(k[1][0]+2.*k[1][1]+2.*k[1][2]+k[1][3]);
    efd1=efd0+(1./6.)*(k[2][0]+2.*k[2][1]+2.*k[2][2]+k[2][3]);
    vs11=vs10+(1./6.)*(k[3][0]+2.*k[3][1]+2.*k[3][2]+k[3][3]);
    eqlinha1=eqlinha0+(1./6.)*(k[4][0]+2.*k[4][1]+2.*k[4][2]+k[4][3]);
    edlinha1=edlinha0+(1./6.)*(k[5][0]+2.*k[5][1]+2.*k[5][2]+k[5][3]);
    w1=w0+(1./6.)*(k[6][0]+2.*k[6][1]+2.*k[6][2]+k[6][3]);
    delta1=delta0+(1./6.)*(k[7][0]+2.*k[7][1]+2.*k[7][2]+k[7][3]);
    pc1=pc0+(1./6.)*(k[8][0]+2.*k[8][1]+2.*k[8][2]+k[8][3]);
    ph1=ph0+(1./6.)*(k[9][0]+2.*k[9][1]+2.*k[9][2]+k[9][3]);
    p1=p0+(1./6.)*(k[10][0]+2.*k[10][1]+2.*k[10][2]+k[10][3]);
}

```

```

void actual_var()
{
    t0=t0+h;
    vf0=vf1;
    vr0=vr1;
    efd0=efd1;
    vs10=vs11;
    eqlinha0=eqlinha1;
    edlinha0=edlinha1;
    w0=w1;
    delta0=delta1;
    pc0=pc1;
    ph0=ph1;
    p0=p1;
}

```

```

void runge_kutta2()
{
    do {
        formulas_rkutta();
        actual_var();
    }
}

```

```
    } while ( t0<tn );
    prep_calculo();
    novas_var_aux(0);
    envia_result(0);
    if (adam==1){
        guarda_val_rkutta();
    }
}

void runge_kutta()
{
    int i;
    for (i=0;i<NUM_MAQUINAS;i++){
        mt=mt+dad_tur[i].m;
    }
    cond_inic();
    ap=fopen("output","w+");
    if (ap==(FILE *)NULL){
        Limpa(INSCREEN,CYAN);
        Janela(80,140,469,240,MAGENTA+XOR);
        mensagem(70,140,389,100,"N o    possivel abrir o ficheiro output.");
        GetKey();
        ExitAtega();
        exit(1);
    }
    runge_kutta2();
    fclose(ap);
}
```

```

/***** adams.c *****/

```

```

#include "externs.h"

```

```

extern FILE *ap;

```

```

void predict()

```

```

{
    vf[4]=vf[3]+(h/24.)*(55*funct1(3)-59*funct1(2)+37*funct1(1)-9*funct1(0));
    vr[4]=vr[3]+(h/24.)*(55*funct2(3)-59*funct2(2)+37*funct2(1)-9*funct2(0));
    efd[4]=efd[3]+(h/24.)*(55*funct3(3)-59*funct3(2)+37*funct3(1)-9*funct3(0));
    vsl[4]=vsl[3]+(h/24.)*(55*funct4(3)-59*funct4(2)+37*funct4(1)-9*funct4(0));
    eqlinha[4]=eqlinha[3]+(h/24.)*(55*funct5(3)-59*funct5(2)+37*funct5(1)-9*funct5(0));
    edlinha[4]=edlinha[3]+(h/24.)*(55*funct6(3)-59*funct6(2)+37*funct6(1)-9*funct6(0));
    w[4]=w[3]+(h/24.)*(55*funct7(3)-59*funct7(2)+37*funct7(1)-9*funct7(0));
    delta[4]=delta[3]+(h/24.)*(55*funct8(3)-59*funct8(2)+37*funct8(1)-9*funct8(0));
    pc[4]=pc[3]+(h/24.)*(55*funct9(3)-59*funct9(2)+37*funct9(1)-9*funct9(0));
    ph[4]=ph[3]+(h/24.)*(55*funct10(3)-59*funct10(2)+37*funct10(1)-9*funct10(0));
    p[4]=p[3]+(h/24.)*(55*funct11(3)-59*funct11(2)+37*funct11(1)-9*funct11(0));
}

```

```

void correct()

```

```

{
    vf[5]=vf[3]+(h/24.)*(9*funct1(4)+19*funct1(3)-5*funct1(2)+funct1(1));
    vr[5]=vr[3]+(h/24.)*(9*funct2(4)+19*funct2(3)-5*funct2(2)+funct2(1));
    efd[5]=efd[3]+(h/24.)*(9*funct3(4)+19*funct3(3)-5*funct3(2)+funct3(1));
    vsl[5]=vsl[3]+(h/24.)*(9*funct4(4)+19*funct4(3)-5*funct4(2)+funct4(1));
    eqlinha[5]=eqlinha[3]+(h/24.)*(9*funct5(4)+19*funct5(3)-5*funct5(2)+funct5(1));
    edlinha[5]=edlinha[3]+(h/24.)*(9*funct6(4)+19*funct6(3)-5*funct6(2)+funct6(1));
    w[5]=w[3]+(h/24.)*(9*funct7(4)+19*funct7(3)-5*funct7(2)+funct7(1));
    delta[5]=delta[3]+(h/24.)*(9*funct8(4)+19*funct8(3)-5*funct8(2)+funct8(1));
    pc[5]=pc[3]+(h/24.)*(9*funct9(4)+19*funct9(3)-5*funct9(2)+funct9(1));
    ph[5]=ph[3]+(h/24.)*(9*funct10(4)+19*funct10(3)-5*funct10(2)+funct10(1));
    p[5]=p[3]+(h/24.)*(9*funct11(4)+19*funct11(3)-5*funct11(2)+funct11(1));
}

```

```

void actual_var_adams()

```

```

{
    int j,k;

```

```

for (j=0;j<4;j++){
    k=j+1;
    t[j]=t[k];
    if (j==3)
        k=5;
    vf[j]=vf[k];
    vr[j]=vr[k];
    efd[j]=efd[k];
    vsl[j]=vsl[k];
    eqlinha[j]=eqlinha[k];
    edlinha[j]=edlinha[k];
    w[j]=w[k];
    delta[j]=delta[k];
    pc[j]=pc[k];
    ph[j]=ph[k];
    p[j]=p[k];
    v[j]=v[k];
    ve[j]=ve[k];
    vrl[j]=vrl[k];
    id[j]=id[k];
    iq[j]=iq[k];
    pe[j]=pe[k];
    pmec[j]=pmec[k];
    pco[j]=pco[k];
}

```

```

void muda_indice0(ind)

```

```

int ind;
{
    vf[0]=vf[ind];
    vr[0]=vr[ind];
    efd[0]=efd[ind];
    vsl[0]=vsl[ind];
    eqlinha[0]=eqlinha[ind];
    edlinha[0]=edlinha[ind];
    w[0]=w[ind];
    delta[0]=delta[ind];
    pc[0]=pc[ind];
    ph[0]=ph[ind];
    p[0]=p[ind];
    v[0]=v[ind];
    ve[0]=ve[ind];
    vrl[0]=vrl[ind];
    id[0]=id[ind];
    iq[0]=iq[ind];
    pe[0]=pe[ind];
    pmec[0]=pmec[ind];
    pco[0]=pco[ind];
}

```

```

int teste_erro(vfr,vrr,efdr,vslr,eqlr,edlr,wr,delt,pcr,phr,pr)

```

```

double vfr,vrr,efdr,vslr,eqlr,edlr,wr,delt,pcr,phr,pr;

```

```

{
    double e1,e2;

```

```

e2=2.*pow(10.,-3.);
e1=e2/32.;
if ( vfr>e2 || vrr>e2 || efdr>e2 || vslr>e2 || eqlr>e2 || wr>e2 || deltr>e2 ||
pcr>e2 || phr>e2 || pr>e2 ){
    h=h/2.;
    return(SIM);
}
if ( vfr<e1 || vrr<e1 || efdr<e1 || vslr<e1 || eqlr<e1 || wr<e1 || deltr<e1 ||
pcr<e1 || phr<e1 || pr<e1 ){
    h=2.*h;
    return(SIM);
}
return(NAO);
}

```

```

int estim_erro()
{
    int r;
    double aux;
    double vfer,vrer,efder,vsler,eqler,edler,wer,delter,pcer,pher,per;
    aux=19./270.;
    vfer=aux*fabs(vf[5]-vf[4]);
    vrer=aux*fabs(vr[5]-vr[4]);
    efd=aux*fabs(efd[5]-efd[4]);
    vsler=aux*fabs(vsl[5]-vsl[4]);
    eqler=aux*fabs(eqlinha[5]-eqlinha[4]);
    edler=aux*fabs(edlinha[5]-edlinha[4]);
    wer=aux*fabs(w[5]-w[4]);
    delter=aux*fabs(delta[5]-delta[4]);
    pcer=aux*fabs(pc[5]-pc[4]);
    pher=aux*fabs(ph[5]-ph[4]);
    per=aux*fabs(p[5]-p[4]);
    r=teste_erro(vfer,vrer,efder,vsler,eqler,edler,wer,delter,pcer,pher,per);
    return(r);
}

```

```

void adams()
{
    int j,i;
    double tnaux;

    tnaux=tn;
    tn=t0+3*h;
    adam=1;
    pontos_adams=0; /* número de itera es */
    for (i=0;i<NUM_MAQUINAS;i++){
        mt=mt+dad_tuF[i].m;
    }
    cond_inic();
    if ((ap=fopen("output","w+"))==(FILE *)NULL){
        Limpa(INSCREEN,CYAN);
        Janela(80,140,469,240,MAGENTA+XOR);
    }
}

```

```

mensagem(70,140,389,100,"N o    possivel abrir o ficheiro output.");
GetKey();
ExitAtega();
exit(1);
}
do{
    cont=0;
    runge_kutta2();
    muda_indice0(5);
    tn=tnaux;
    do{
        predict();
        t[4]=t[3]+h;
        t0=t[4];
        novas_var_aux(4);
        correCt();
        if ( estim_erro()==SIM ){
            t0=t[3];
            tn=t0+3*h;
            if (tn>tnaux)
                tn=tnaux;
            muda_indice0(3);
            alterah=1;
            break;
        }
        novas_var_aux(5);
        envia_result(5);
        actual_var_adams();
        pontos_adams++;
    }while(t[4]<tn);
    }while(t[4]<tn);
    fclose(ap);
}

```

```

/***** menus.c *****/

#include "externs.h"

/* int ncaract; */

int bloco[NUM_MAQUINAS][NUM_BOTOES];

draw () /* Desenha o ecran inicial */
{
    register int f;
    static char *t[10]={"Inserir Dados","Ler Fich. Dados","Alterar Dados","Estudo"
    ,""},
    ,"","", "", "Terminar"};

    for(f=0;f<10;f++)
    {
        botao(550,f*35,90,35,t[f]);
    }
    win(0,0,549,349,CYAN);
}

botao(x,y,larg,altura,texto) /* Desenha um botao em relevo */
{
    int x,y,larg,altura;
    char *texto;

    int f,tx,ty;
    larg--,altura--;
    for(f=0;f<3;f++)
    {
        Line(x+f,y+f,x+larg-f,y+f,15);
        Line(x+f,y+f,x+f,y+altura-f,15);
        Line(x+f,y+altura-f,x+larg-f,y+altura-f,8);
        Line(x+larg-f,y+f,x+larg-f,y+altura-f,8);
    }
    PaintBox(x+3,y+3,x+larg-3,y+altura-3,CYAN);
    GetStringSize(texto,&tx,&ty,&f);
    tx=x+(larg-tx)/2;
    ty=y+(altura-ty)/2+f;
    DrawStringAt(tx-1,ty,texto,15);
    DrawStringAt(tx+1,ty,texto,8);
    DrawStringAt(tx,ty-1,texto,15);
    DrawStringAt(tx,ty+1,texto,8);

    DrawStringAt(tx,ty,texto,INTS+CYAN);
}

Limpa(a,b,c,d,cor) /* Limpa o ecran */
{
    int a,b,c,d;
    int cor;
    {
        PaintBox(a,b,c,d,cor);
    }
}

int window(x,y,x2,y2,cor) /* Desenha uma janela */
{
    int x,y,x2,y2,cor;
    {
        int bloco;
        bloco=SaveBlock(x,y,x2+3,y2+3);
        PaintBox(x+3,y+3,x2+3,y2+3,BLACK);
        win(x,y,x2,y2,cor);
        return(bloco);
    }
}

godown(x,y,x2,y2) /* Baixa o botao */
{
    int x,y,x2,y2;
    {
        int f,t;
        int bloc,bloc2;
        bloc=SaveBlock(x+3,y+3,x2-3,y2-3);
        bloc2=SaveBlock(x,y,x2,y2);
        for(f=0;f<3;f++)
        {
            MoveBlock(x+3+f,y+3+f,bloc);
            GetBlock(bloc);
            Line(x+f,y+f,x2,y+f,BLACK+INTS);
            Line(x+f,y+f,x+f,y2,BLACK+INTS);
            for(t=0;t<5000;t++);
        }
        WaitMouseUp(ALL);
        GetBlock(bloc2);
    }
}

```

```

DelBlock(bloc);
DelBlock(bloc2);
}

Janela(a,b,c,d,cor)
int a,b,c,d,cor;
{
    PaintBox(a+5,b+4,c+5,d+4,BLACK);
    PaintBox(a,b,c,d,INTS+WHITE);
    Box(a+1,b+1,c-1,d-1,cor);
    Box(a+2,b+2,c-2,d-2,cor);
    Box(a+4,b+4,c-4,d-4,cor);
}

int LerInt(x,y,cor,maxdig) /*Le um inteiro */
int x,y,cor,maxdig;
{
    char numero[5];
    int num;

    SetTextControlFlag(NUM);
    ReadPromptStringAt(x,y,"> ",numero,cor,cor,maxdig);
    num=atoi(numero);
    SetTextControlFlag(ALLCHAR);
    return(num);
}

float LerFloat(x,y,corp,cors,maxdig) /*Le um float*/
int x,y,corp,cors,maxdig;
{
    char numero[10];
    float num;

    SetTextControlFlag(DECIMAL+SIGNAL);
    ReadPromptStringAt(x,y,"> ",numero,corp,cors,maxdig);
    /* ncaract=strlen(numero); */
    num=atof(numero);
    SetTextControlFlag(ALLCHAR);
    return(num);
}

DrawFloatAt(x,y,num,cor,numdec)
int x,y,numdec;
double num;
int cor;

{
    char *buffer,temp[20];
    int i,j,decptr,signptr;
    buffer=fcvt(num,numdec,&decptr,&signptr);
    if(!signptr) temp[0]=' ';
    else temp[0]='-';
    if (decptr<=0)
    {
        temp[1]='0';
        temp[2]='.';
        j=3;
        while(decptr++<0) {temp[j]='0';j++;}
        for (i=0;i<=strlen(buffer);i++)
            { temp[j] = buffer[i];j++;}
    }
    else
        for (i=0;i<=strlen(buffer)+1;i++)
        {
            if (i<decptr) temp[i+1] = buffer[i];
            else if (i==decptr) temp[i+1]='.';
            else temp[i+1] = buffer[i-1];
        }
    DrawStringAt(x,y,temp,cor);
}

EraseScreen()
{
    PaintBlock(ALLSCREEN,BLACK);
}

botao_menu(x,y,larg,altura,texto1,texto2) /* Desenha um botao do menu em rele
vo */
int x,y,larg,altura;
char *texto1,*texto2;
{
    int f,tx1,ty1,tx2,ty2;
    larg--,altura--;
    for(f=0;f<3;f++)
    {
        Line(x+f,y+f,x+larg-f,y+f,15);
        Line(x+f,y+f,x+f,y+altura-f,15);
        Line(x+f,y+altura-f,x+larg-f,y+altura-f,8);
        Line(x+larg-f,y+f,x+larg-f,y+altura-f,8);
    }
    PaintBox(x+3,y+3,x+larg-3,y+altura-3,CYAN);
    GetStringSize(texto1,&tx1,&ty1,&f);
    GetStringSize(texto2,&tx2,&ty2,&f);
    tx1=x+(larg-tx1)/2.;
    tx2=x+(larg-tx2)/2.;
    ty1=y+altura/3.-ty1/2.+f;
    ty2=y+2.*altura/3.-ty2/2.+f;
    DrawStringAt(tx1-1,ty1,texto1,15);
    DrawStringAt(tx1+1,ty1,texto1,8);
    DrawStringAt(tx1,ty1-1,texto1,15);
    DrawStringAt(tx1,ty1+1,texto1,8);
}

```

```

DrawStringAt (tx1,ty1,texto1,INTS+CYAN);
DrawStringAt (tx2-1,ty2,texto2,15);
DrawStringAt (tx2+1,ty2,texto2,8);
DrawStringAt (tx2,ty2-1,texto2,15);
DrawStringAt (tx2,ty2+1,texto2,8);
DrawStringAt (tx2,ty2,texto2,INTS+CYAN);
}

draw_menu()          /* Desenha o ecran para inserir dados */
{
    register int f;
    static char *t1[7]={"Menu Inicial","Menu Condi es","Menu Caract.","Menu Carac
t.", "Menu Caract.", "Menu Caract.",
                        "Guardar"};
    static char *t2[7]={"de Dados","de Defeito","Reg. Tens o ","Reg. Veloc.", "Turb
ina","Gerador",
                        "Dados"};
    for (f=0;f<7;f++)
    {
        botao_menu(550,f*50,90,50,t1[f],t2[f]);
    }
    win(0,0,549,349,CYAN);
}

titulo(texto,x1,y1,x2,y2)
int x1,y1,x2,y2;
char *texto;
{
    int tx,ty,b;
    x2--;
    y2--;
    SetPenSize(2,1);
    GetStringSize(texto,&tx,&ty,&b);
    tx=x1+(x2-tx)/2;
    ty=y1+(y2-ty)/2+b;
    DrawStringAt (tx-1,ty,texto,15);
    DrawStringAt (tx+1,ty,texto,8);
    DrawStringAt (tx,ty-1,texto,15);
    DrawStringAt (tx,ty+1,texto,8);
    DrawStringAt (tx,ty,texto,INTS+WHITE);
}

int prossegue_newton()
{
    int dx,dy,b;
    int ghar;
    double qe;

    Limpa (INSCREEN,CYAN);

    qe=newton();

    Janela(80,140,469,240,MAGENTA+XOR);
    DrawStringAt (160,180,"qe=",INTS+BLACK);
    GetStringSize("qe=",&dx,&dy,&b);
    DrawFloatAt (160+dx,180,qe,INTS+BLACK,6);
    DrawString(" (p.u.)");
    DrawStringAt (120,200,"Aceita o valor da Pot ncia Reactiva produzida (S/N)?",BL
ACK);
    do{
        ghar=GetKey();
        if (ghar=='N'||ghar=='n'){
            Limpa (INSCREEN,CYAN);
            return (NAO);
        }
        else{
            if (ghar=='S'||ghar=='s'){
                return (SIM);
            }
        }
    }while (ghar!=ESC);
    return (ESC);
}

int menu_inicial()
{
    int dx,dy,b;
    int x00,y00,b00;
    int ghar;
    int r;

    Limpa (INSCREEN,CYAN);
    SetBackColor (INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo("Menu Inicial de Dados",50,60,449,75);
    SetPenSize(1,1);
    DrawStringAt (80,125,"Pot ncia el ctrica inicial (p.u.):",INTS+WHITE);
    DrawStringAt (80,150,"Tens o no barramento de referencia (p.u.):",INTS+WHITE);
    DrawStringAt (80,175,"Tens o no barramento do gerador (p.u.):",INTS+WHITE);
    DrawStringAt (80,200,"Resist ncia equivalente (p.u.):",INTS+WHITE);
    DrawStringAt (80,225,"React ncia equivalente (p.u.):",INTS+WHITE);

    GetStringSize("Potencia electrica inicial (p.u.):",&dx,&dy,&b);
    dad_ini[i].pe0=LerFloat (80+dx,125,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("Tensao no barramento de referencia (p.u.):",&dx,&dy,&b);
    dad_ini[i].vref=LerFloat (80+dx,150,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("Tensao no barramento do gerador (p.u.):",&dx,&dy,&b);
    dad_ini[i].v0=LerFloat (80+dx,175,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("Resistencia equivalente (p.u.):",&dx,&dy,&b);
    dad_ini[i].ro=LerFloat (80+dx,200,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("Reactancia equivalente (p.u.):",&dx,&dy,&b);
    dad_ini[i].x=LerFloat (80+dx,225,INTS+BLUE,INTS+WHITE,6);

    DrawStringAt (110,260,"Quer alterar algum dado (S/N)?",INTS+WHITE);

```

```

do{
    ghar=GetKey();
    }while(ghar!=ESC && ghar!='S' && ghar!='s' && ghar!='N' && ghar!='n');

do{
    if(ghar=='S' || ghar=='s'){
        r=altera_menu_inicial();
        if (r==ESC)
            return(r);
        else
            ghar='N';
    }
    else{
        if(ghar=='N' || ghar=='n'){
            r=prosseque_newton();
            if (r==ESC)
                return(r);
            else{
                if (r==SIM)
                    return(SIM);
                else
                    ghar='S';
            }
        }
        else
            r=ghar;
    }
    }while(r==NAO);
    return(r);
}

menu_defeito()
{
    int dx,dy,b;
    int x00,y00,b00;
    char ghar;

    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo("Menu de Condi es de Defeito",50,60,449,75);
    SetPenSize(1,1);
    DrawStringAt(80,125,"Durante o defeito:",INTS+WHITE);
    DrawStringAt(120,145,"Resist ncia equivalente (p.u.):",INTS+WHITE);
    DrawStringAt(120,165,"React ncia equivalente (p.u.):",INTS+WHITE);
    DrawStringAt(80,195,"Apes o defeito:",INTS+WHITE);
    DrawStringAt(120,215,"Resist ncia equivalente (p.u.):",INTS+WHITE);
    DrawStringAt(120,235,"React ncia equivalente (p.u.):",INTS+WHITE);

    dad_def[1].rpd=LerFloat(120+dx,215,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("Resist ncia equivalente (p.u.):",&dx,&dy,&b);
    dad_def[1].rp=LerFloat(120+dx,145,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("React ncia equivalente (p.u.):",&dx,&dy,&b);
    dad_def[1].xp=LerFloat(120+dx,165,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("Resist ncia equivalente (p.u.):",&dx,&dy,&b);

    dad_def[1].rpd=LerFloat(120+dx,215,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("React ncia equivalente (p.u.):",&dx,&dy,&b);
    dad_def[1].xpd=LerFloat(120+dx,235,INTS+BLUE,INTS+WHITE,6);

}

titulo2(texto1,texto2,x1,y1,x2,y2)
int x1,y1,x2,y2;
char *texto1,*texto2;
{
    int tx1,ty1,tx2,ty2,b;
    x2--;
    y2--;
    SetPenSize(2,1);
    GetStringSize(texto1,&tx1,&ty1,&b);
    GetStringSize(texto2,&tx2,&ty2,&b);
    tx1=x1+(x2-tx1)/2.;
    tx2=x1+(x2-tx2)/2.;
    ty1=y1+y2/3.-ty1/2.+b;
    ty2=y1+2.*y2/3.-ty2/2.+b;

    DrawStringAt(tx1-1,ty1,texto1,15);
    DrawStringAt(tx1+1,ty1,texto1,8);
    DrawStringAt(tx1,ty1-1,texto1,15);
    DrawStringAt(tx1,ty1+1,texto1,8);
    DrawStringAt(tx1,ty1,texto1,INTS+WHITE);
    DrawStringAt(tx2-1,ty2,texto2,15);
    DrawStringAt(tx2+1,ty2,texto2,8);
    DrawStringAt(tx2,ty2-1,texto2,15);
    DrawStringAt(tx2,ty2+1,texto2,8);
    DrawStringAt(tx2,ty2,texto2,INTS+WHITE);

}

menu_reg_tensao()
{
    int dx,dy,b;
    int x00,y00,b00;
    char ghar;

    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo2("Menu de Caractersticas","do Regulador de Tens o",50,55,449,70);
    SetPenSize(1,1);
    DrawStringAt(80,125,"tr (p.u.):",INTS+WHITE);
    DrawStringAt(80,140,"ka (p.u.):",INTS+WHITE);
    DrawStringAt(80,155,"ta (p.u.):",INTS+WHITE);
    DrawStringAt(80,170,"vrmax (p.u.):",INTS+WHITE);
    DrawStringAt(80,185,"vrmin (p.u.):",INTS+WHITE);
    DrawStringAt(80,200,"aex (p.u.):",INTS+WHITE);
    DrawStringAt(80,215,"bex (p.u.):",INTS+WHITE);

```

```

DrawStringAt(80,230,"ke (p.u.):",INTS+WHITE);
DrawStringAt(80,245,"te (p.u.):",INTS+WHITE);
DrawStringAt(80,260,"ks (p.u.):",INTS+WHITE);
DrawStringAt(80,275,"ts (p.u.):",INTS+WHITE);

}

GetStringSize("tr (p.u.):",&dx,&dy,&b);
dad_ten[1].tr=LerFloat(80+dx,125,INTS+BLUE,INTS+WHITE,6);
GetStringSize("ka (p.u.):",&dx,&dy,&b);
dad_ten[1].ka=LerFloat(80+dx,140,INTS+BLUE,INTS+WHITE,6);
GetStringSize("ta (p.u.):",&dx,&dy,&b);
dad_ten[1].ta=LerFloat(80+dx,155,INTS+BLUE,INTS+WHITE,6);
GetStringSize("vrmax (p.u.):",&dx,&dy,&b);
dad_ten[1].vrmax=LerFloat(80+dx,170,INTS+BLUE,INTS+WHITE,6);
GetStringSize("vrmin (p.u.):",&dx,&dy,&b);
dad_ten[1].vrmin=LerFloat(80+dx,185,INTS+BLUE,INTS+WHITE,6);
GetStringSize("aex (p.u.):",&dx,&dy,&b);
dad_ten[1].aex=LerFloat(80+dx,200,INTS+BLUE,INTS+WHITE,6);
GetStringSize("bex (p.u.):",&dx,&dy,&b);
dad_ten[1].bex=LerFloat(80+dx,215,INTS+BLUE,INTS+WHITE,6);
GetStringSize("ke (p.u.):",&dx,&dy,&b);
dad_ten[1].ke=LerFloat(80+dx,230,INTS+BLUE,INTS+WHITE,6);
GetStringSize("te (p.u.):",&dx,&dy,&b);
dad_ten[1].te=LerFloat(80+dx,245,INTS+BLUE,INTS+WHITE,6);
GetStringSize("ks (p.u.):",&dx,&dy,&b);
dad_ten[1].ks=LerFloat(80+dx,260,INTS+BLUE,INTS+WHITE,6);
GetStringSize("ts (p.u.):",&dx,&dy,&b);
dad_ten[1].ts=LerFloat(80+dx,275,INTS+BLUE,INTS+WHITE,6);

}

menu_reg_velocidade()
{
    int dx,dy,b;
    int x00,y00,b00;
    char ghar;

    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo2("Menu de Caracteristicas","do Regulador de Velocidade",50,55,449,70);
    SetPenSize(1,1);
    DrawStringAt(80,140,"r (p.u.):",INTS+WHITE);
    DrawStringAt(80,160,"ki (p.u.):",INTS+WHITE);
    DrawStringAt(80,180,"tg (p.u.):",INTS+WHITE);

    GetStringSize("r (p.u.):",&dx,&dy,&b);
    dad_vel[1].r=LerFloat(80+dx,140,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("ki (p.u.):",&dx,&dy,&b);
    dad_vel[1].ki=LerFloat(80+dx,160,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("tg (p.u.):",&dx,&dy,&b);
    dad_vel[1].tg=LerFloat(80+dx,180,INTS+BLUE,INTS+WHITE,6);

}

menu_turbina()
{
    int dx,dy,b;
    int x00,y00,b00;
    char ghar;

    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo2("Menu de Caracteristicas","da Turbina",50,55,449,70);
    SetPenSize(1,1);
    DrawStringAt(80,140,"tt (p.u.):",INTS+WHITE);
    DrawStringAt(80,160,"Pot ncia mec nica mxima (p.u.):",INTS+WHITE);
    DrawStringAt(80,180,"m (p.u.):",INTS+WHITE);

    GetStringSize("tt (p.u.):",&dx,&dy,&b);
    dad_tur[1].tt=LerFloat(80+dx,140,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("Pot ncia mec nica mxima (p.u.):",&dx,&dy,&b);
    dad_tur[1].pmecmax=LerFloat(80+dx,160,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("m (p.u.):",&dx,&dy,&b);
    dad_tur[1].m=LerFloat(80+dx,180,INTS+BLUE,INTS+WHITE,6);

}

menu_gerador()
{
    int dx,dy,b;
    int x00,y00,b00;
    char ghar;

    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo2("Menu de Caracteristicas","do Gerador",50,55,449,70);
    SetPenSize(1,1);
    DrawStringAt(80,135,"xd (p.u.):",INTS+WHITE);
    DrawStringAt(80,155,"xdlinha (p.u.):",INTS+WHITE);
    DrawStringAt(80,175,"xq (p.u.):",INTS+WHITE);
    DrawStringAt(80,195,"xqlinha (p.u.):",INTS+WHITE);
    DrawStringAt(80,215,"tdolinha (p.u.):",INTS+WHITE);

    GetStringSize("xd (p.u.):",&dx,&dy,&b);
    dad_ger[1].xd=LerFloat(80+dx,135,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("xdlinha (p.u.):",&dx,&dy,&b);
    dad_ger[1].xdlinha=LerFloat(80+dx,155,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("xq (p.u.):",&dx,&dy,&b);
    dad_ger[1].xq=LerFloat(80+dx,175,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("xqlinha (p.u.):",&dx,&dy,&b);

```

```

dad_ger[i].xqlinha=LerFloat(80+dx,195,INTS+BLUE,INTS+WHITE,6);
GetStringSize("tdolinha (p.u.):",&dx,&dy,&b);
dad_ger[i].tdolinha=LerFloat(80+dx,215,INTS+BLUE,INTS+WHITE,6);

}

mensagem2(texto1,texto2,x1,y1,x2,y2)
int x1,y1,x2,y2;
char *texto1,*texto2;
{
    int tx1,ty1,tx2,ty2,b;
    x2--;
    y2--;
    SetPenSize(1,1);
    GetStringSize(texto1,&tx1,&ty1,&b);
    GetStringSize(texto2,&tx2,&ty2,&b);
    tx1=x1+(x2-tx1)/2.;
    tx2=x1+(x2-tx2)/2.;
    ty1=y1+y2/3.-ty1/2.+b;
    ty2=y1+2.*y2/3.-ty2/2.+b;

    DrawStringAt(tx1,ty1,texto1,INTS+BLACK);
    DrawStringAt(tx2,ty2,texto2,INTS+BLACK);

}

int bothao[NUM_BOTOES];

esquecimento()
{
    int k,i;
    for (i=0;i<NUM_BOTOES;i++)
        if (bothao[i]==0){
            Janela(80,140,469,240,MAGENTA+XOR);
            switch(i){
                case 0: mensagem2("Falta introduzir os dados do","Menu Inicial de Dados"
,80,140,389,100);
                    GetKey();
                    Limpa(INSCREEN,CYAN);
                    k=menu_inicial();
                    if (k==ESC)
                        return(k);
                    break;
                case 1: mensagem2("Falta introduzir os dados do","Menu de Condi es de D
efeito",80,140,389,100);
                    GetKey();
                    Limpa(INSCREEN,CYAN);
                    menu defeito();
                    break;
                case 2: mensagem2("Falta introduzir os dados do","Menu de Caracteristica
s do Regulador de Tens o",80,140,389,100);
                    GetKey();
                    Limpa(INSCREEN,CYAN);
                    menu_reg_tensao();
                    break;
                case 3: mensagem2("Falta introduzir os dados do","Menu de Caracteristica
s do Regulador de Velocidade",80,140,389,100);
                    GetKey();
                    Limpa(INSCREEN,CYAN);
                    menu_reg_velocidade();
                    break;
                case 4: mensagem2("Falta introduzir os dados do","Menu de Caracteristica
s da Turbina",80,140,389,100);
                    GetKey();
                    Limpa(INSCREEN,CYAN);
                    menu_turbina();
                    break;
                case 5: mensagem2("Falta introduzir os dados do","Menu de Caracteristica
s do Gerador",80,140,389,100);
                    GetKey();
                    Limpa(INSCREEN,CYAN);
                    menu_gerador();
                    break;
            }
            Limpa(INSCREEN,CYAN);
        }

}

void mensagem(x,y,larg,altura,texto)
int x,y,larg,altura;
char *texto;
{
    int f,tx,ty;
    larg--,altura--;
    GetStringSize(texto,&tx,&ty,&f);
    tx=x+(larg-tx)/2;
    ty=y+(altura-ty)/2+f;
    DrawStringAt(tx,ty,texto,INTS+BLACK);
}

void guarda_dados()
{
    FILE *fp;
    int i;
    char *fname;

    SetBackColor(INTS+WHITE);
    Janela(80,140,469,240,MAGENTA+XOR);
    DrawStringAt(120,190,"Nome do ficheiro:",INTS+BLACK);
    SetTextControlFlag(FLUSH+ALFANUM+TOUPPER);
    ReadString(filename,NUM_MAX_CHARACT);
    strcpy(fname,filename);
    strcat(filename,".DAD");
}

```

```

if ((fp=fopen(filename,"wb"))==NULL) {
    Limpa(INSCREEN,CYAN);
    Janela(80,140,469,240,MAGENTA^XOR);
    mensagem(70,140,389,100,"N~o    possivel abrir o ficheiro ");
    DrawString(fname);
    GetKey();
    ExitAtega();
    exit(1);
}
for (i=0;i<NUM_MAQUINAS;i++){
    fwrite(&dad_ini[i],sizeof(m_ini),1,fp);
    fwrite(&dad_def[i],sizeof(m_def),1,fp);
    fwrite(&dad_ten[i],sizeof(m_ten),1,fp);
    fwrite(&dad_vel[i],sizeof(m_vel),1,fp);
    fwrite(&dad_tur[i],sizeof(m_tur),1,fp);
    fwrite(&dad_ger[i],sizeof(m_ger),1,fp);
}
fclose(fp);
dados=1;
posicao++;
}

```

```

int insere()

```

```

{
    int opcao2=0;
    int x,y,x00,y00,b,b00;
    int ghar;
    int i,k;
    int dx,dy;

```

```

    for (i=0;i<NUM_BOTOES;i++){
        bothao[i]=0;
    }

```

```

    Limpa(ALLSCREEN,CYAN);
    draw_menu();

```

```

    Getmouse(&x00,&y00,&b00);

```

```

    while(opcao2!=7 && (ghar=InKey()) !=ESC )
    {

```

```

        Getmouse(&x,&y,&b);
        if (x!=x00 || y!=y00)
        { DrawCursor(x,y);
          x00=x;y00=y;b00=b;
          DrawCursor(x,y);
        }
    }

```

```

    if (b==LEFT && x>=550)
    { opcao2=y/50+1;
      UndrawCursor();
      godown(550,opcao2*50-50,639,opcao2*50-1);
    }

```

```

switch (opcao2)

```

```

{
    case 1:WaitMouseUp(LEFT);
        if (bothao[0]==1){
            k=altera_menu_inicial();
            if (k==NAO)
                break;
            else{
                return(k);
            }
        }
        bothao[0]=1;
        k=menu_inicial();
        if (k==ESC){
            Limpa(ALLSCREEN,CYAN);
            return(k);
        }
        Limpa(INSCREEN,CYAN);
        Setmouse(560,75);
        break;
    case 2:WaitMouseUp(LEFT);
        if (bothao[1]==1){
            k=altera_menu_defeito();
            if (k==NAO)
                break;
            else
                return(k);
        }
        bothao[1]=1;
        menu_defeito();
        Limpa(INSCREEN,CYAN);
        Setmouse(560,125);
        break;
    case 3:WaitMouseUp(LEFT);
        if (bothao[2]==1){
            k=altera_menu_reg_tensao();
            if (k==NAO)
                break;
            else
                return(k);
        }
        bothao[2]=1;
        menu_reg_tensao();
        Limpa(INSCREEN,CYAN);
        Setmouse(560,175);
        break;
    case 4:WaitMouseUp(LEFT);
        if (bothao[3]==1){
            k=altera_menu_reg_velocidade();
            if (k==NAO)
                break;
            else
                return(k);
        }
        bothao[3]=1;
        menu_reg_velocidade();
        Limpa(INSCREEN,CYAN);
        Setmouse(560,225);
        break;
    case 5:WaitMouseUp(LEFT);
        if (bothao[4]==1){

```

```

        k=altera_menu_turbina();
        if (k==NÃO)
            break;
        else
            return(k);
    }
    bothao[4]=1;
    menu_turbina();
    Limpa(INSCREEN,CYAN);
    Setmouse(560,275);
    break;
case 6:WaitMouseUp(LEFT);
    if (bothao[5]==1){
        k=altera_menu_gerador();
        if (k==NÃO)
            break;
        else
            return(k);
    }
    bothao[5]=1;
    menu_gerador();
    Limpa(INSCREEN,CYAN);
    Setmouse(560,325);
    break;
}
WaitMouseUp(LEFT);
DrawCursor(x,y);
}
}
if (ghar==ESC)
    return(ESC);
else {
    UnDrawCursor();
    Limpa(INSCREEN,CYAN);
    esquecimento();
    /* GetKey(); */
    guarda_dados();
    Limpa(ALLSCREEN,CYAN);
    draw();
    return(SIM);
    GetKey();
}
}

escolhe_metodo()
{
    int xa,ya,xbeg,ybeg,b1;
    int ch;
    while ( (ch=InKey()) !=ESC ){
        GetMouse(&xa,&ya,&b1);
        if ( (b1==LEFT) && (xa>319) && (xa<331) && (ya>192) && (ya<204) ){
            UnDrawCursor();
            PaintBox(320,193,330,203,INTS+WHITE);
            WaitMouseUp(LEFT);
            runge_kutta();
            return(SIM);
        }
        if ( (b1==LEFT) && (xa>319) && (xa<331) && (ya>212) && (ya<224) ){
            UnDrawCursor();
            PaintBox(320,213,330,223,INTS+WHITE);
            WaitMouseUp(LEFT);
            adams();
            return(SIM);
        }
    }

    if ( ( xa != xbeg ) || ( ya != ybeg ) )
        DrawCursor(xbeg=xa,ybeg=ya);
}

WaitMouseUp(LEFT);
UnDrawCursor();
SetMouse(100,100);
Limpa(INSCREEN,CYAN);
return(ESC);
}

estuda()
{
    int dx,dy,b;
    int k;

    Limpa(ALLSCREEN,CYAN);

    if (dados==0){
        Janela(80,140,469,240,MAGENTA+XOR);
        mensagem(70,140,389,100,"Falta introduzir os dados.");
        GetKey();
        Limpa(ALLSCREEN,CYAN);
        return(NAO);
    }

    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    DrawStringAt(80,100,"Tempo de dura ˆo do defeito:",INTS+WHITE);
    DrawStringAt(80,120,"Tempo de estudo:",INTS+WHITE);
    DrawStringAt(80,140,"Passo da integra ˆo:",INTS+WHITE);
    DrawStringAt(80,180,"M todo de clculo:",INTS+WHITE);
    DrawStringAt(120,200,"M todo de runge kutta",INTS+WHITE);
    DrawStringAt(120,220,"M todo de adams",INTS+WHITE);

    GetStringSize("Tempo de dura ˆo do defeito:",&dx,&dy,&b);
    tp=LerFloat(80+dx,100,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("Tempo de estudo:",&dx,&dy,&b);
    tn=LerFloat(80+dx,120,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("Passo da integra ˆo:",&dx,&dy,&b);
    h=LerFloat(80+dx,140,INTS+BLUE,INTS+WHITE,6);
    GetStringSize("M todo de clculo:",&dx,&dy,&b);
    Box(320,193,330,203,INTS+WHITE);
    Box(320,213,330,223,INTS+WHITE);

```

```

k=escolhe_metodo();
return(k);
}

void le()
{
    FILE *fp;
    int i;
    int dx,dy,b;
    double qe;
    char fname[9];

    lista_fich();
    if (strcmp(filename,"Cancelar")==0){
        Limpa(INSCREEN,CYAN);
        return;
    }
    else{
        strcpy(fname,filename);
        strcat(fname,".DAD");
    }
    if ((fp=fopen(filename,"rb"))==NULL){
        Limpa(INSCREEN,CYAN);
        Janela(80,140,469,240,MAGENTA+XOR);
        mensagem(70,140,389,100,"No possivel abrir o ficheiro ");
        DrawString(fname);
        GetKey();
        ExitAtega();
        exit(1);
    }
    for (i=0;i<NUM_MAQUINAS;i++){
        fread(&dad_ini[i],sizeof(m_ini),1,fp);
        fread(&dad_def[i],sizeof(m_def),1,fp);
        fread(&dad_ten[i],sizeof(m_ten),1,fp);
        fread(&dad_vel[i],sizeof(m_vel),1,fp);
        fread(&dad_tur[i],sizeof(m_tur),1,fp);
        fread(&dad_ger[i],sizeof(m_ger),1,fp);
    }
    fclose(fp);
    dados=1;
    mostra_menu_inicial();
    qe=newton();
    DrawStringAt(80,250,"Pot ncia reactiva (p.u.):",INTS+WHITE);
    GetStringSize("Pot ncia reactiva (p.u.):",&dx,&dy,&b);
    DrawFloatAt(80+dx,250,qe,INTS+WHITE,6);
    mensagem(50,250,449,50,"Prima qualquer tecla para continuar");
    GetKey();
    mostra_menu defeito();
    mensagem(50,235,449,65,"Prima qualquer tecla para continuar");
    GetKey();
    mostra_menu_reg_tensao();
    GetKey();
    mostra_menu_reg_velocidade();
    mensagem(50,180,449,120,"Prima qualquer tecla para continuar");

```

```

GetKey();
mostra_menu turbina();
mensagem(50,180,449,120,"Prima qualquer tecla para continuar");
GetKey();
mostra_menu gerador();
mensagem(50,215,449,85,"Prima qualquer tecla para continuar");
GetKey();
Limpa(INSCREEN,CYAN);
}

```

```

void val_ini()
{
    t0=0.0;
    i=0;
    mt=0.0;
    adam=0;
    cont=0;
    alterah=0;
}

```

```

arranca()
{
    int opcao=0;
    int x,y,x00,y00,b,b00;
    char ghar;
    int l;

```

```

draw();
Limpa(INSCREEN,CYAN);

```

```

Setmouse(320,175);
Getmouse(&x00,&y00,&b00);

```

```

while(opcao!=10 && (ghar=InKey()) !=ESC )
{
    Getmouse(&x,&y,&b);
    if (x!=x00 || y!=y00)
    { DrawCursor(x,y);
      x00=x;y00=y;b00=b;
      DrawCursor(x,y);
    }

```

```

    if(b==LEFT && x>=550)
    { opcao=y/35+1;
      UndrawCursor();
      godown(550,opcao*35-35,639,opcao*35-1);
      switch(opcao)
      {
          case 1:WaitMouseUp(LEFT);
            l=insere();
            if (l==ESC){

```

```
        UnDrawCursor();
        Limpa(ALLSCREEN, CYAN);
        MouseSpan(1, 1, 639, 349);
        draw();
        val_ini();
    }
    Getmouse(&x00, &y00, &b00);
    break;
case 2:
    WaitMouseUp(LEFT);
    le();
    Getmouse(&x00, &y00, &b00);
    break;
case 3: WaitMouseUp(LEFT);
    l=altera();
    if (l==ESC){
        UnDrawCursor();
        Limpa(ALLSCREEN, CYAN);
        draw();
        MouseSpan(1, 1, 639, 349);
        val_ini();
    }
    Getmouse(&x00, &y00, &b00);
    break;
case 4:
    WaitMouseUp(LEFT);
    l=estuda();
    if (l==SIM){
        UnDrawCursor();
        Limpa(ALLSCREEN, CYAN);
        if (adam==1)
            Arran_Graf_Adams();
        else
            Arran_Graf();
        UnDrawCursor();
    }
    draw();
    val_ini();
    MouseSpan(1, 1, 639, 349);
    Getmouse(&x00, &y00, &b00);
    break;
case 5:
    break;
case 6:
    break;
case 7:
    break;
case 8:
    break;
case 9:
    break;
}
WaitMouseUp(LEFT);
DrawCursor(x, y);
}
}
```

```

/***** altera.c *****/

#include "externs.h"

#define COMP_MAX_MENU_INI 5
#define COMP_MAX_MENU_DEF 4
#define COMP_MAX_MENU_TEN 11
#define COMP_MAX_MENU_VEL 3

void mostra_menu_inicial()
{
    int xa,ya,xbeg,ybeg,bl;
    int dx,dy,b;
    int x00,y00,b00;
    int ghar;

    Limpa(INSCREEN,CYAN);
    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo("Menu Inicial de Dados",50,60,449,75);
    SetPenSize(1,1);
    DrawStringAt(80,125,"Pot ncia el ctrica inicial (p.u.):",INTS+WHITE);
    DrawStringAt(80,150,"Tens o no barramento de referencia (p.u.):",INTS+WHITE);
    DrawStringAt(80,175,"Tens o no barramento do gerador (p.u.):",INTS+WHITE);
    DrawStringAt(80,200,"Resist ncia equivalente (p.u.):",INTS+WHITE);
    DrawStringAt(80,225,"React ncia equivalente (p.u.):",INTS+WHITE);

    GetStringSize("Potencia electrica inicial (p.u.):",&dx,&dy,&b);
    DrawFloatAt(80+dx,125,dad_ini[i].pe0,INTS+WHITE,6);
    GetStringSize("Tensao no Barramento de referencia (p.u.):",&dx,&dy,&b);
    DrawFloatAt(80+dx,150,dad_ini[i].vref,INTS+WHITE,6);
    GetStringSize("Tensao no Barramento do gerador (p.u.):",&dx,&dy,&b);
    DrawFloatAt(80+dx,175,dad_ini[i].v0,INTS+WHITE,6);
    GetStringSize("Resistencia equivalente (p.u.):",&dx,&dy,&b);
    DrawFloatAt(80+dx,200,dad_ini[i].ro,INTS+WHITE,6);
    GetStringSize("Reactancia equivalente (p.u.):",&dx,&dy,&b);
    DrawFloatAt(80+dx,225,dad_ini[i].x,INTS+WHITE,6);
}

void mostra_menu_defeito()
{
    int xa,ya,xbeg,ybeg,bl;
    int dx,dy,b;

    Limpa(INSCREEN,CYAN);
    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo("Menu de Condi es de Defeito",50,60,449,75);
    SetPenSize(1,1);
    DrawStringAt(80,125,"Durante o defeito:",INTS+WHITE);
    DrawStringAt(120,145,"Resist ncia equivalente (p.u.):",INTS+WHITE);
    DrawStringAt(120,165,"React ncia equivalente (p.u.):",INTS+WHITE);
    DrawStringAt(80,195,"Ap s o defeito:",INTS+WHITE);
    DrawStringAt(120,215,"Resist ncia equivalente (p.u.):",INTS+WHITE);
    DrawStringAt(120,235,"React ncia equivalente (p.u.):",INTS+WHITE);

    GetStringSize("Resist ncia equivalente (p.u.):",&dx,&dy,&b);
    DrawFloatAt(120+dx,145,dad_def[i].rp,INTS+WHITE,6);
    GetStringSize("React ncia equivalente (p.u.):",&dx,&dy,&b);
    DrawFloatAt(120+dx,165,dad_def[i].xp,INTS+WHITE,6);
    GetStringSize("Resist ncia equivalente (p.u.):",&dx,&dy,&b);
    DrawFloatAt(120+dx,215,dad_def[i].rpd,INTS+WHITE,6);
    GetStringSize("React ncia equivalente (p.u.):",&dx,&dy,&b);
    DrawFloatAt(120+dx,235,dad_def[i].xpd,INTS+WHITE,6);
}

void mostra_menu_reg_tensao()
{
    int xa,ya,xbeg,ybeg,bl;
    int dx,dy,b;
    int x00,y00,b00;
    int ghar;

    Limpa(INSCREEN,CYAN);
    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo2("Menu de Caracteristicas","do Regulador de Tens o",50,55,449,70);
    SetPenSize(1,1);
    DrawStringAt(80,125,"tr (p.u.):",INTS+WHITE);
    DrawStringAt(80,140,"ka (p.u.):",INTS+WHITE);
    DrawStringAt(80,155,"ta (p.u.):",INTS+WHITE);
    DrawStringAt(80,170,"vrmax (p.u.):",INTS+WHITE);
    DrawStringAt(80,185,"vrmin (p.u.):",INTS+WHITE);
    DrawStringAt(80,200,"aex (p.u.):",INTS+WHITE);
    DrawStringAt(80,215,"bex (p.u.):",INTS+WHITE);
    DrawStringAt(80,230,"ke (p.u.):",INTS+WHITE);
    DrawStringAt(80,245,"te (p.u.):",INTS+WHITE);
    DrawStringAt(80,260,"ks (p.u.):",INTS+WHITE);
    DrawStringAt(80,275,"ts (p.u.):",INTS+WHITE);

    GetStringSize("tr (p.u.):",&dx,&dy,&b);
    DrawFloatAt(80+dx,125,dad_ten[i].tr,INTS+WHITE,6);
    GetStringSize("ka (p.u.):",&dx,&dy,&b);

```

```

DrawFloatAt (80+dx,140,dad_ten[i].ka,INTS+WHITE,6);
GetStringSize("ta (p.u.):",&dx,&dy,&b);
DrawFloatAt (80+dx,155,dad_ten[i].ta,INTS+WHITE,6);
GetStringSize("vrmax (p.u.):",&dx,&dy,&b);
DrawFloatAt (80+dx,170,dad_ten[i].vrmax,INTS+WHITE,6);
GetStringSize("vrmin (p.u.):",&dx,&dy,&b);
DrawFloatAt (80+dx,185,dad_ten[i].vrmin,INTS+WHITE,6);
GetStringSize("aex (p.u.):",&dx,&dy,&b);
DrawFloatAt (80+dx,200,dad_ten[i].aex,INTS+WHITE,6);
GetStringSize("bex (p.u.):",&dx,&dy,&b);
DrawFloatAt (80+dx,215,dad_ten[i].bex,INTS+WHITE,6);
GetStringSize("ke (p.u.):",&dx,&dy,&b);
DrawFloatAt (80+dx,230,dad_ten[i].ke,INTS+WHITE,6);
GetStringSize("te (p.u.):",&dx,&dy,&b);
DrawFloatAt (80+dx,245,dad_ten[i].te,INTS+WHITE,6);
GetStringSize("ks (p.u.):",&dx,&dy,&b);
DrawFloatAt (80+dx,260,dad_ten[i].ks,INTS+WHITE,6);
GetStringSize("ts (p.u.):",&dx,&dy,&b);
DrawFloatAt (80+dx,275,dad_ten[i].ts,INTS+WHITE,6);
}

void mostra_menu_reg_velocidade()
{
    int xa,ya,xbeg,ybeg,bl;
    int dx,dy,b;
    int x00,y00,b00;
    int ghar;

    Limpa(INSCREEN,CYAN);
    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo2("Menu de Caracteristicas","do Regulador de Velocidade",50,55,449,70);
    SetPenSize(1,1);
    DrawStringAt(80,140,"r (p.u.):",INTS+WHITE);
    DrawStringAt(80,160,"ki (p.u.):",INTS+WHITE);
    DrawStringAt(80,180,"tg (p.u.):",INTS+WHITE);

    GetStringSize("r (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,140,dad_vel[i].r,INTS+WHITE,6);
    GetStringSize("ki (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,160,dad_vel[i].ki,INTS+WHITE,6);
    GetStringSize("tg (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,180,dad_vel[i].tg,INTS+WHITE,6);
}

void mostra_menu_turbina()
{
    int xa,ya,xbeg,ybeg,bl;
    int dx,dy,b;

    int x00,y00,b00;
    int ghar;

    Limpa(INSCREEN,CYAN);
    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo2("Menu de Caracteristicas","da Turbina",50,55,449,70);
    SetPenSize(1,1);
    DrawStringAt(80,140,"tt (p.u.):",INTS+WHITE);
    DrawStringAt(80,160,"Pot ncia mec nica mxima (p.u.):",INTS+WHITE);
    DrawStringAt(80,180,"m (p.u.):",INTS+WHITE);

    GetStringSize("tt (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,140,dad_tur[i].tt,INTS+WHITE,6);
    GetStringSize("Pot ncia mec nica mxima (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,160,dad_tur[i].pmecmax,INTS+WHITE,6);
    GetStringSize("m (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,180,dad_tur[i].m,INTS+WHITE,6);
}

void mostra_menu_gerador()
{
    int xa,ya,xbeg,ybeg,bl;
    int dx,dy,b;
    int x00,y00,b00;
    int ghar;

    Limpa(INSCREEN,CYAN);
    SetBackColor(INTS+BLUE);
    win(50,50,499,299,INTS+BLUE);
    titulo2("Menu de Caracteristicas","do Gerador",50,55,449,70);
    SetPenSize(1,1);
    DrawStringAt(80,135,"xd (p.u.):",INTS+WHITE);
    DrawStringAt(80,155,"xdlinha (p.u.):",INTS+WHITE);
    DrawStringAt(80,175,"xq (p.u.):",INTS+WHITE);
    DrawStringAt(80,195,"xqlinha (p.u.):",INTS+WHITE);
    DrawStringAt(80,215,"tdolinha (p.u.):",INTS+WHITE);

    GetStringSize("xd (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,135,dad_ger[i].xd,INTS+WHITE,6);
    GetStringSize("xdlinha (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,155,dad_ger[i].xdlinha,INTS+WHITE,6);
    GetStringSize("xq (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,175,dad_ger[i].xq,INTS+WHITE,6);
    GetStringSize("xqlinha (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,195,dad_ger[i].xqlinha,INTS+WHITE,6);
    GetStringSize("tdolinha (p.u.):",&dx,&dy,&b);
    DrawFloatAt (80+dx,215,dad_ger[i].tdolinha,INTS+WHITE,6);
}

```

```

int altera_menu_inicial()
{
    int xa,ya,xbeg,ybeg,b1;
    int b;
    int ch;
    int r;
    int dx[COMP_MAX_MENU_INI],dy[COMP_MAX_MENU_INI];
    mostra_menu_inicial();
    mensagem2("Prima o bot o direito do rato para continuar","Prima Esc para c
    ar",50,245,449,50);
    GetStringSize("Potencia electrica inicial (p.u.):",&dx[0],&dy[0],&b);
    GetStringSize("Tensao no barramento de referencia (p.u.):",&dx[1],&dy[1],&
    b);
    GetStringSize("Tensao no barramento do gerador (p.u.):",&dx[2],&dy[2],&b);
    GetStringSize("Resistencia equivalente (p.u.):",&dx[3],&dy[3],&b);
    GetStringSize("Reactancia equivalente (p.u.):",&dx[4],&dy[4],&b);
    MouseSpan(51,51,499,299);
    SetMouse(240,220);
    GetMouse(&xa,&ya,&b1);
    DrawCursor(xbeg=xa,ybeg=ya);
    while ( (ch=InKey()) !=ESC ){
        GetMouse(&xa,&ya,&b1);
        if (MouseDown(RIGHT)){
            UndrawCursor();
            Limpa(INSCREEN,CYAN);
            MouseSpan(1,1,639,349);
            return(NAO);
        }
        if ((b1==LEFT) && (xa>80+dx[0]-1) && (xa<176+dx[0]+1) && (ya>128-dy[0]-1) && (ya<133
    )){
            UndrawCursor();
            PaintBox(80+dx[0],128-dy[0],176+dx[0],132,INTS+BLUE);
            WaitMouseUp(LEFT);
            dad_ini[i].pe0=LerFloat(80+dx[0],125,INTS+BLUE,INTS+WHITE,6);
            mostra_menu_inicial();
            mensagem2("Prima o bot o direito do rato para continuar","Prima Esc para c
            ancelar",50,245,449,50);
        }
        if ((b1==LEFT) && (xa>80+dx[1]-1) && (xa<176+dx[1]+1) && (ya>153-dy[1]-1) && (ya<158
    )){
            UndrawCursor();
            PaintBox(80+dx[1],153-dy[1],176+dx[1],157,INTS+BLUE);
            WaitMouseUp(LEFT);
            dad_ini[i].vref=LerFloat(80+dx[1],150,INTS+BLUE,INTS+WHITE,6);
            mostra_menu_inicial();
            mensagem2("Prima o bot o direito do rato para continuar","Prima Esc para c
            ancelar",50,245,449,50);
        }
        if ((b1==LEFT) && (xa>80+dx[2]-1) && (xa<176+dx[2]+1) && (ya>178-dy[2]-1) && (ya<183
    )){
            UndrawCursor();
            PaintBox(80+dx[2],178-dy[2],176+dx[2],182,INTS+BLUE);
            WaitMouseUp(LEFT);
            dad_ini[i].v0=LerFloat(80+dx[2],175,INTS+BLUE,INTS+WHITE,6);
            mostra_menu_inicial();
            mensagem2("Prima o bot o direito do rato para continuar","Prima Esc para c
            ancelar",50,245,449,50);
        }
    }
}

```

```

    if ((b1==LEFT) && (xa>80+dx[3]-1) && (xa<176+dx[3]+1) && (ya>203-dy[3]-1) && (ya<208
    )){
        UndrawCursor();
        PaintBox(80+dx[3],203-dy[3],176+dx[3],207,INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ini[i].ro=LerFloat(80+dx[3],200,INTS+BLUE,INTS+WHITE,6);
        mostra_menu_inicial();
        mensagem2("Prima o bot o direito do rato para continuar","Prima Esc para c
        ancelar",50,245,449,50);
    }
    if ((b1==LEFT) && (xa>80+dx[4]-1) && (xa<176+dx[4]+1) && (ya>228-dy[4]-1) && (ya<233
    )){
        UndrawCursor();
        PaintBox(80+dx[4],228-dy[4],176+dx[4],232,INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ini[i].x=LerFloat(80+dx[4],225,INTS+BLUE,INTS+WHITE,6);
        mostra_menu_inicial();
        mensagem2("Prima o bot o direito do rato para continuar","Prima Esc para c
        ancelar",50,245,449,50);
    }
    if ( ( xa != xbeg ) || ( ya != ybeg ) )
        DrawCursor(xbeg=xa,ybeg=ya);
    WaitMouseUp(LEFT);
    UndrawCursor();
    SetMouse(100,100);
    Limpa(INSCREEN,CYAN);
    return(ESC);
}

int altera_menu_defeito()
{
    int xa,ya,xbeg,ybeg,b1;
    int b;
    int ch;
    int r;
    int dx[COMP_MAX_MENU_DEF],dy[COMP_MAX_MENU_DEF];
    mostra_menu_defeito();
    mensagem2("Prima o bot o direito do rato para continuar","Prima ESC para c
    ar",50,245,449,50);
    GetStringSize("Resist ncia equivalente (p.u.):",&dx[0],&dy[0],&b);
    GetStringSize("React ncia equivalente (p.u.):",&dx[1],&dy[1],&b);
    GetStringSize("Resist ncia equivalente (p.u.):",&dx[2],&dy[2],&b);
    GetStringSize("React ncia equivalente (p.u.):",&dx[3],&dy[3],&b);
    MouseSpan(51,51,499,299);
    SetMouse(240,220);
    GetMouse(&xa,&ya,&b1);
    DrawCursor(xbeg=xa,ybeg=ya);
    while ( (ch=InKey()) !=ESC ){
        GetMouse(&xa,&ya,&b1);
        if (MouseDown(RIGHT)){
            UndrawCursor();
            Limpa(INSCREEN,CYAN);
            MouseSpan(1,1,639,349);
            return(NAO);
        }
        if ((b1==LEFT) && (xa>120+dx[0]-1) && (xa<216+dx[0]+1) && (ya>148-dy[0]-1) && (ya<15
    )){

```

```

UnDrawCursor();
PaintBox(120+dx[0],148-dy[0],216+dx[0],152,INTS+BLUE);
WaitMouseUp(LEFT);
dad_def[i].rp=LerFloat(120+dx[0],145,INTS+BLUE,INTS+WHITE,6);
mostra_menu_defeito();
mensagem2("Prima o bot o direito do rato para continuar","Prima ESC para c
ancelar",50,245,449,50);
}
if ((b1==LEFT)&&(xa>120+dx[1]-1)&&(xa<216+dx[1]+1)&&(ya>168-dy[1]-1)&&(ya<17
3)) {
    UnDrawCursor();
    PaintBox(120+dx[1],168-dy[1],216+dx[1],172,INTS+BLUE);
    WaitMouseUp(LEFT);
    dad_def[i].rp=LerFloat(120+dx[1],165,INTS+BLUE,INTS+WHITE,6);
    mostra_menu_defeito();
    mensagem2("Prima o bot o direito do rato para continuar","Prima ESC para c
ancelar",50,245,449,50);
}
if ((b1==LEFT)&&(xa>120+dx[2]-1)&&(xa<216+dx[2]+1)&&(ya>218-dy[2]-1)&&(ya<22
3)) {
    UnDrawCursor();
    PaintBox(120+dx[2],218-dy[2],216+dx[2],222,INTS+BLUE);
    WaitMouseUp(LEFT);
    dad_def[i].rp=LerFloat(120+dx[2],215,INTS+BLUE,INTS+WHITE,6);
    mostra_menu_defeito();
    mensagem2("Prima o bot o direito do rato para continuar","Prima ESC para c
ancelar",50,245,449,50);
}
if ((b1==LEFT)&&(xa>120+dx[3]-1)&&(xa<216+dx[3]+1)&&(ya>238-dy[3]-1)&&(ya<24
3)) {
    UnDrawCursor();
    PaintBox(120+dx[3],238-dy[3],216+dx[3],242,INTS+BLUE);
    WaitMouseUp(LEFT);
    dad_def[i].xpd=LerFloat(120+dx[3],235,INTS+BLUE,INTS+WHITE,6);
    mostra_menu_defeito();
    mensagem2("Prima o bot o direito do rato para continuar","Prima ESC para c
ancelar",50,245,449,50);
}
if ( ( xa != xbeg ) || ( ya != ybeg ) )
    DrawCursor(xbeg=xa,ybeg=ya);
}
WaitMouseUp(LEFT);
UnDrawCursor();
SetMouse(100,100);
Limpa(INSCREEN,CYAN);
return(ESC);
}

int altera_menu_reg_tensao()
{
    int xa,ya,xbeg,ybeg,b1;
    int b;
    int ch;
    int r;
    int dx[COMP_MAX_MENU_TEN],dy[COMP_MAX_MENU_TEN];
    mostra_menu_reg_tensao();
    GetStringSize("tr (p.u.):",&dx[0],&dy[0],&b);
    GetStringSize("ka (p.u.):",&dx[1],&dy[1],&b);

```

```

GetStringSize("ta (p.u.):",&dx[2],&dy[2],&b);
GetStringSize("vrmax (p.u.):",&dx[3],&dy[3],&b);
GetStringSize("vrmin (p.u.):",&dx[4],&dy[4],&b);
GetStringSize("aex (p.u.):",&dx[5],&dy[5],&b);
GetStringSize("bex (p.u.):",&dx[6],&dy[6],&b);
GetStringSize("ke (p.u.):",&dx[7],&dy[7],&b);
GetStringSize("te (p.u.):",&dx[8],&dy[8],&b);
GetStringSize("ks (p.u.):",&dx[9],&dy[9],&b);
GetStringSize("ts (p.u.):",&dx[10],&dy[10],&b);
MouseSpan(51,51,499,299);
SetMouse(240,220);
GetMouse(&xa,&ya,&b1);
DrawCursor(xbeg=xa,ybeg=ya);
while ( (ch=InKey()) !=ESC ) {
    GetMouse(&xa,&ya,&b1);
    if (MouseDown(RIGHT)) {
        UnDrawCursor();
        Limpa(INSCREEN,CYAN);
        MouseSpan(1,1,639,349);
        return(NAO);
    }
    if ((b1==LEFT)&&(xa>80+dx[0]-1)&&(xa<176+dx[0]+1)&&(ya>128-dy[0]-1)&&(ya<133
)) {
        UnDrawCursor();
        PaintBox(80+dx[0],128-dy[0],176+dx[0],132,INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].tr=LerFloat(80+dx[0],125,INTS+BLUE,INTS+WHITE,6);
        mostra_menu_reg_tensao();
    }
    if ((b1==LEFT)&&(xa>80+dx[1]-1)&&(xa<176+dx[1]+1)&&(ya>143-dy[1]-1)&&(ya<148
)) {
        UnDrawCursor();
        PaintBox(80+dx[1],143-dy[1],176+dx[1],147,INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].ka=LerFloat(80+dx[1],140,INTS+BLUE,INTS+WHITE,6);
        mostra_menu_reg_tensao();
    }
    if ((b1==LEFT)&&(xa>80+dx[2]-1)&&(xa<176+dx[2]+1)&&(ya>158-dy[2]-1)&&(ya<163
)) {
        UnDrawCursor();
        PaintBox(80+dx[2],158-dy[2],176+dx[2],162,INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].ta=LerFloat(80+dx[2],155,INTS+BLUE,INTS+WHITE,6);
        mostra_menu_reg_tensao();
    }
    if ((b1==LEFT)&&(xa>80+dx[3]-1)&&(xa<176+dx[3]+1)&&(ya>173-dy[3]-1)&&(ya<178
)) {
        UnDrawCursor();
        PaintBox(80+dx[3],173-dy[3],176+dx[3],177,INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].vrmax=LerFloat(80+dx[3],170,INTS+BLUE,INTS+WHITE,6);
        mostra_menu_reg_tensao();
    }
    if ((b1==LEFT)&&(xa>80+dx[4]-1)&&(xa<176+dx[4]+1)&&(ya>188-dy[4]-1)&&(ya<193
)) {
        UnDrawCursor();
        PaintBox(80+dx[4],188-dy[4],176+dx[4],192,INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].vrmin=LerFloat(80+dx[4],185,INTS+BLUE,INTS+WHITE,6);
        mostra_menu_reg_tensao();
    }
}

```

```

    if ((b1==LEFT) && (xa>80+dx[5]-1) && (xa<176+dx[5]+1) && (ya>203-dy[5]-1) && (ya<208)) {
        UnDrawCursor();
        PaintBox(80+dx[5], 203-dy[5], 176+dx[5], 207, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].aex=LerFloat(80+dx[5], 200, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_reg_tensao();
    }
    if ((b1==LEFT) && (xa>80+dx[6]-1) && (xa<176+dx[6]+1) && (ya>218-dy[6]-1) && (ya<223)) {
        UnDrawCursor();
        PaintBox(80+dx[6], 218-dy[6], 176+dx[6], 222, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].bex=LerFloat(80+dx[6], 215, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_reg_tensao();
    }
    if ((b1==LEFT) && (xa>80+dx[7]-1) && (xa<176+dx[7]+1) && (ya>233-dy[7]-1) && (ya<238)) {
        UnDrawCursor();
        PaintBox(80+dx[7], 233-dy[7], 176+dx[7], 237, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].kex=LerFloat(80+dx[7], 230, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_reg_tensao();
    }
    if ((b1==LEFT) && (xa>80+dx[8]-1) && (xa<176+dx[8]+1) && (ya>248-dy[8]-1) && (ya<253)) {
        UnDrawCursor();
        PaintBox(80+dx[8], 248-dy[8], 176+dx[8], 252, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].tex=LerFloat(80+dx[8], 245, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_reg_tensao();
    }
    if ((b1==LEFT) && (xa>80+dx[9]-1) && (xa<176+dx[9]+1) && (ya>263-dy[9]-1) && (ya<268)) {
        UnDrawCursor();
        PaintBox(80+dx[9], 263-dy[9], 176+dx[9], 267, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].ks=LerFloat(80+dx[9], 260, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_reg_tensao();
    }
    if ((b1==LEFT) && (xa>80+dx[10]-1) && (xa<176+dx[10]+1) && (ya>278-dy[10]-1) && (ya<283)) {
        UnDrawCursor();
        PaintBox(80+dx[10], 278-dy[10], 176+dx[10], 282, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ten[i].ts=LerFloat(80+dx[10], 275, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_reg_tensao();
    }
    if ( ( xa != xbeg ) || ( ya != ybeg ) )
        DrawCursor(xbeg=xa, ybeg=ya);
    WaitMouseUp(LEFT);
    UnDrawCursor();
    SetMouse(100, 100);
    Limpa(INSCREEN, CYAN);
    return(ESC);
}

int altera_menu_reg_velocidade()
{
    int xa, ya, xbeg, ybeg, b1;
    int b;
    int ch;
    int r;
    int dx[COMP_MAX_MENU_VEL], dy[COMP_MAX_MENU_VEL];
    mostra_menu_reg_velocidade();
    GetStringSize("F (p.u.):", &dx[0], &dy[0], &b);
    GetStringSize("k1 (p.u.):", &dx[1], &dy[1], &b);
    GetStringSize("tg (p.u.):", &dx[2], &dy[2], &b);
    MouseSpan(51, 51, 499, 299);
    SetMouse(240, 220);
    GetMouse(&xa, &ya, &b1);
    DrawCursor(xbeg=xa, ybeg=ya);
    while ( (ch=InKey()) !=ESC ) {
        GetMouse(&xa, &ya, &b1);
        if (MouseDown(RIGHT)) {
            UnDrawCursor();
            Limpa(INSCREEN, CYAN);
            MouseSpan(1, 1, 639, 349);
            return(NAO);
        }
    }
    if ((b1==LEFT) && (xa>80+dx[0]-1) && (xa<176+dx[0]+1) && (ya>143-dy[0]-1) && (ya<148)) {
        UnDrawCursor();
        PaintBox(80+dx[0], 143-dy[0], 176+dx[0], 147, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_vel[i].r=LerFloat(80+dx[0], 140, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_reg_velocidade();
    }
    if ((b1==LEFT) && (xa>80+dx[1]-1) && (xa<176+dx[1]+1) && (ya>163-dy[1]-1) && (ya<168)) {
        UnDrawCursor();
        PaintBox(80+dx[1], 163-dy[1], 176+dx[1], 167, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_vel[i].ki=LerFloat(80+dx[1], 160, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_reg_velocidade();
    }
    if ((b1==LEFT) && (xa>80+dx[2]-1) && (xa<176+dx[2]+1) && (ya>183-dy[2]-1) && (ya<188)) {
        UnDrawCursor();
        PaintBox(80+dx[2], 183-dy[2], 176+dx[2], 187, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_vel[i].tg=LerFloat(80+dx[2], 180, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_reg_velocidade();
    }
    if ( ( xa != xbeg ) || ( ya != ybeg ) )
        DrawCursor(xbeg=xa, ybeg=ya);
    WaitMouseUp(LEFT);
    UnDrawCursor();
    SetMouse(100, 100);
    Limpa(INSCREEN, CYAN);
    return(ESC);
}

int altera_menu_turbina()

```

```

int xa, ya, xbeg, ybeg, bl;
int b;
int ch;
int r;
int dx[COMP_MAX_MENU_VEL], dy[COMP_MAX_MENU_VEL];
mostra_menu_turbina();
GetStringSize("tt (p.u.):", &dx[0], &dy[0], &b);
GetStringSize("Potencia maxima (p.u.):", &dx[1], &dy[1], &b);
GetStringSize("m (p.u.):", &dx[2], &dy[2], &b);
MouseSpan(51, 51, 499, 299);
SetMouse(240, 220);
GetMouse(&xa, &ya, &bl);
DrawCursor(xbeg=xa, ybeg=ya);
while ( (ch=InKey()) !=ESC ) {
    GetMouse(&xa, &ya, &bl);
    if (MouseDown(RIGHT)) {
        UnDrawCursor();
        Limpa(INSCREEN, CYAN);
        MouseSpan(1, 1, 639, 349);
        return(NAO);
    }
    if ((bl==LEFT) && (xa>80+dx[0]-1) && (xa<176+dx[0]+1) && (ya>143-dy[0]-1) && (ya<148)) {
        UnDrawCursor();
        PaintBox(80+dx[0], 143-dy[0], 176+dx[0], 147, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_tur[i].tt=LerFloat(80+dx[0], 140, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_turbina();
    }
    if ((bl==LEFT) && (xa>80+dx[1]-1) && (xa<176+dx[1]+1) && (ya>163-dy[1]-1) && (ya<168)) {
        UnDrawCursor();
        PaintBox(80+dx[1], 163-dy[1], 176+dx[1], 167, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_tur[i].pmecmax=LerFloat(80+dx[1], 160, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_turbina();
    }
    if ((bl==LEFT) && (xa>80+dx[2]-1) && (xa<176+dx[2]+1) && (ya>183-dy[2]-1) && (ya<188)) {
        UnDrawCursor();
        PaintBox(80+dx[2], 183-dy[2], 176+dx[2], 187, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_tur[i].m=LerFloat(80+dx[2], 180, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_turbina();
    }
    if ( ( xa != xbeg ) || ( ya != ybeg ) )
        DrawCursor(xbeg=xa, ybeg=ya);
    WaitMouseUp(LEFT);
    UnDrawCursor();
    SetMouse(100, 100);
    Limpa(INSCREEN, CYAN);
    return(ESC);
}

int altera_menu_gerador()
{

```

```

int xa, ya, xbeg, ybeg, bl;
int b;
int ch;
int r;
int dx[COMP_MAX_MENU_INI], dy[COMP_MAX_MENU_INI];
mostra_menu_gerador();
GetStringSize("xd (p.u.):", &dx[0], &dy[0], &b);
GetStringSize("xlinha (p.u.):", &dx[1], &dy[1], &b);
GetStringSize("xq (p.u.):", &dx[2], &dy[2], &b);
GetStringSize("xqlinha (p.u.):", &dx[3], &dy[3], &b);
GetStringSize("tdolinha (p.u.):", &dx[4], &dy[4], &b);
MouseSpan(51, 51, 499, 299);
SetMouse(240, 220);
GetMouse(&xa, &ya, &bl);
DrawCursor(xbeg=xa, ybeg=ya);
while ( (ch=InKey()) !=ESC ) {
    GetMouse(&xa, &ya, &bl);
    if (MouseDown(RIGHT)) {
        UnDrawCursor();
        Limpa(INSCREEN, CYAN);
        MouseSpan(1, 1, 639, 349);
        return(NAO);
    }
    if ((bl==LEFT) && (xa>80+dx[0]-1) && (xa<176+dx[0]+1) && (ya>138-dy[0]-1) && (ya<143)) {
        UnDrawCursor();
        PaintBox(80+dx[0], 138-dy[0], 176+dx[0], 142, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ger[i].xd=LerFloat(80+dx[0], 135, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_gerador();
    }
    if ((bl==LEFT) && (xa>80+dx[1]-1) && (xa<176+dx[1]+1) && (ya>158-dy[1]-1) && (ya<163)) {
        UnDrawCursor();
        PaintBox(80+dx[1], 158-dy[1], 176+dx[1], 162, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ger[i].xlinha=LerFloat(80+dx[1], 155, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_gerador();
    }
    if ((bl==LEFT) && (xa>80+dx[2]-1) && (xa<176+dx[2]+1) && (ya>178-dy[2]-1) && (ya<183)) {
        UnDrawCursor();
        PaintBox(80+dx[2], 178-dy[2], 176+dx[2], 182, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ger[i].xq=LerFloat(80+dx[2], 175, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_gerador();
    }
    if ((bl==LEFT) && (xa>80+dx[3]-1) && (xa<176+dx[3]+1) && (ya>198-dy[3]-1) && (ya<203)) {
        UnDrawCursor();
        PaintBox(80+dx[3], 198-dy[3], 176+dx[3], 202, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ger[i].xqlinha=LerFloat(80+dx[3], 195, INTS+BLUE, INTS+WHITE, 6);
        mostra_menu_gerador();
    }
    if ((bl==LEFT) && (xa>80+dx[4]-1) && (xa<176+dx[4]+1) && (ya>218-dy[4]-1) && (ya<223)) {
        UnDrawCursor();
        PaintBox(80+dx[4], 218-dy[4], 176+dx[4], 222, INTS+BLUE);
        WaitMouseUp(LEFT);
        dad_ger[i].tdolinha=LerFloat(80+dx[4], 215, INTS+BLUE, INTS+WHITE, 6);
    }
}

```

```

    mostra_menu_gerador();
}
if ( ( xa != xbeg ) || ( ya != ybeg ) )
    DrawCursor(xbeg=xa,ybeg=ya);
}
WaitMouseUp(LEFT);
UnDrawCursor();
SetMouse(100,100);
Limpa(INSCREEN,CYAN);
return(ESC);
}

int altera()
{
    FILE *fp;
    int i,k;
    int opcao2=0;
    int x,y,x00,y00,b,b00;
    int ghar;
    int dx,dy;
    int ativo=0;
    double qe;
    char fname[9];

    lista_fich();
    if (strcmp(filename,"Cancelar")==0) {
        Limpa(INSCREEN,CYAN);
        return(NAO);
    }
    else {
        strcpy(fname,filename);
        strcat(filename,".DAD");
    }
    if ((fp=fopen(filename,"rb"))==NULL) {
        Limpa(INSCREEN,CYAN);
        Janela(80,140,469,240,MAGENTA+XOR);
        mensagem(70,140,389,100,"N-o possivel abrir o ficheiro ");
        DrawString(fname);
        GetKey();
        ExitAtega();
        exit(1);
    }
    for (i=0;i<NUM_MAQUINAS;i++){
        fread(&dad_ini[i],sizeof(m_ini),1,fp);
        fread(&dad_def[i],sizeof(m_def),1,fp);
        fread(&dad_ten[i],sizeof(m_ten),1,fp);
        fread(&dad_vel[i],sizeof(m_vel),1,fp);
        fread(&dad_tur[i],sizeof(m_tur),1,fp);
        fread(&dad_ger[i],sizeof(m_ger),1,fp);
    }
    fclose(fp);

    Limpa(ALLSCREEN,CYAN);
    draw_menu();
    GetMouse(&x00,&y00,&b00);

    while(opcao2!=7 && (ghar=InKey()) !=ESC)
    {
        Getmouse(&x,&y,&b);
        if (x!=x00 || y!=y00)
        { DrawCursor(x,y);
          x00=x;y00=y;b00=b;
          DrawCursor(x,y);
        }

        if (b==LEFT && x>=550)
        { opcao2=y/50+1;
          UnDrawCursor();
          godown(550,opcao2*50-50,639,opcao2*50-1);
          switch(opcao2)
          {
              case 1:ativo=1;
                    WaitMouseUp(LEFT);
                    do{
                        k=altera_menu_inicial();
                        if (k==ESC)
                            return(k);
                        else{
                            k=prosseque_newton();
                            if (k==ESC)
                                return(k);
                            else
                                if (k==SIM)
                                    break;
                        }
                    }while(k==NAO);
                    Limpa(INSCREEN,CYAN);
                    Setmouse(560,75);
                    break;
              case 2:WaitMouseUp(LEFT);
                    k=altera_menu_defeito();
                    if (k==NAO) {
                        Limpa(INSCREEN,CYAN);
                        Setmouse(560,125);
                        break;
                    }
                    else
                        return(k);
              case 3:WaitMouseUp(LEFT);
                    k=altera_menu_reg_tensao();
                    if (k==NAO) {
                        Limpa(INSCREEN,CYAN);
                        Setmouse(560,175);
                        break;
                    }
                    else
                        return(k);
              case 4:WaitMouseUp(LEFT);
                    k=altera_menu_reg_velocidade();
                    if (k==NAO) {
                        Limpa(INSCREEN,CYAN);
                        Setmouse(560,225);
                        break;
                    }
                    else

```

```
        return(k);
    case 5:WaitMouseUp(LEFT);
        k=altera_menu_turbina();
        if (k==NÃO){
            Limpa(INSCREEN,CYAN);
            Setmouse(560,275);
            break;
        }
        else
            return(k);
    case 6:WaitMouseUp(LEFT);
        k=altera_menu_gerador();
        if (k==NÃO){
            Limpa(INSCREEN,CYAN);
            Setmouse(560,325);
            break;
        }
        else
            return(k);
    }
    WaitMouseUp(LEFT);
    DrawCursor(x,y);
}
if (ghar==ESC)
    return(ESC);
else {
    UnDrawCursor();
    Limpa(INSCREEN,CYAN);
    guarda_dados();
    if (ativo==0)
        qe=newton();
    Limpa(ALLSCREEN,CYAN);
    draw();
    return(SIM);
}
```

```

/***** lista.c *****/

#include "externs.h"

#define NUM_MAX_FICH 10

FILE *ffic;
char fil[NUM_MAX_FICH][9];

void Abrir_Fich(num)
short int num;
{
    int xa, ya, bl, xbeg, ybeg, i, flag=0;

    MouseSpan(180, 63, 360, 113+13*num);
    SetMouse(240, 220);
    GetMouse(&xa, &ya, &bl);
    DrawCursor(xbeg=xa, ybeg=ya);
    while (flag==0)
    {
        GetMouse(&xa, &ya, &bl);
    }

    /* ficheiros */

    for(i=0; i<=num; i++)
    {
        if ((bl==LEFT) && (xa>275) && (xa<300) && (ya>91+13*i) && (ya<101+13*i))
        {
            UnDrawCursor();
            flag=1;
            if (i==num)
            {
                PaintBox(281, 94+13*i, 294, 102+13*i, RED); WaitMouseUp(LEFT);
                PaintBox(281, 94+13*i, 294, 102+13*i, WHITE);
                strcpy(filename, "Cancelar");
                Limpa(INSCREEN, CYAN);
            }
            else
            {
                PaintBox(281, 92+13*i, 294, 100+13*i, BLACK);
                WaitMouseUp(LEFT);
                PaintBox(281, 92+13*i, 294, 100+13*i, WHITE);
                Limpa(INSCREEN, CYAN);
                strcpy(filename, fil[i]);
            }
        }
        if (( xa != xbeg) || ( ya != ybeg ) )
            DrawCursor(xbeg=xa, ybeg=ya);
    }

    UnDrawCursor();
    MouseSpan(1, 1, 639, 349);
    SetMouse(100, 100);
    Limpa(INSCREEN, CYAN);
}

void lista_fich()
{
    short int iret, result, t, i;
    int tx, ty, f, j;
    char al[9], a2[5], cadeia[80];

    Limpa(INSCREEN, CYAN);

    if(system("dir *.dad>fich.lis")!=0){
        Win(170, 110, 390, 160, RED);
        DrawStringAt(190, 130, "N o fez o comando 'system' !", INTS+WHITE);
    }

    /* Le o arquivo "dirrede.lis" e armazena os casos */
    /* na estrutura casos. */

    if ( (ffic=fopen("fich.lis", "r")) != NULL ){
        fscanf(ffic, "%s[\n]\n");
        fscanf(ffic, "%s[\n]\n");
        fscanf(ffic, "%s[\n]\n");
        fscanf(ffic, "%s[\n]\n");
        t = 0;
        while ( fgets(cadeia, 80, ffic) != NULL ){
            sscanf(cadeia, "%s %s", al, a2);
            result = strcmpi(a2, "DAD");
            if ( result == 0 ){
                strcpy(fil[t], al);
                t++;
            }
        }
        Janela(180, 63, 360, 113+13*t, MAGENTA+XOR);
        mensagem(180, 63, 180, 40, "Lista de Ficheiros");
        for(i=0; i<t; i++){
            DrawStringAt(200, 100+13*i, fil[i], BLACK);
            Box(280, 91+13*i, 295, 101+13*i, BLACK);
        }
        DrawStringAt(200, 103+13*i, "Cancelar", RED);
        Box(280, 93+13*i, 295, 103+13*i, RED);
        fclose(ffic);
        remove("fich.lis"); /* Eliminar o arquivo "fich.lis" */
    }

    Abrir_Fich(t);
}
else {
    DrawStringAt(230, 145, "Prima LEFT para sair", INTS+WHITE);
    WaitMouseUp(LEFT);
}

```

```
WaitMouseDown(LEFT);  
Limpa(INSCREEN,CYAN);  
return;  
}
```

```

/***** graficos.c *****/

```

```

#include "externs.h"
#include <float.h>
#include <ctype.h>
#include <conio.h>
#include <process.h>
#include <malloc.h>
#include <io.h>

```

```

int noimp,nogers,nogerr,nobar,serv,conta_tempo,zoom;
int pontos,precisao[9],maxrest,curv_activ[9];
double difmaxmin[9],passo[9];
double minim[9],maxim[9];
double limitmax[9];
double limitmin[9];
unsigned char pix[9][510];
double mom_ini,mom_fim,interval;

```

```

int guarda[510];
int maxx;

```

```

FILE *fout;

```

```

DrawFloatAt2(x,y,num,cor,numdec)
int x,y,numdec;
double num;
int cor;

```

```

{
char *buffer,temp[20];
int i,j,decptr,signptr;
buffer=fcvt(num,numdec,&decptr,&signptr);
if(!signptr) temp[0]=' ';
else temp[0]='-';
if (decptr<=0)
{
temp[1]='0';
temp[2]='.';
j=3;
while(decptr++<0) {temp[j]='0';j++;}
for (i=0;i<=strlen(buffer);i++)
{ temp[j] = buffer[i];j++;}
}
else
for (i=0;i<=strlen(buffer)+1;i++)
{

```

```

if (i<decptr) temp[i+1] = buffer[i];
else if (i==decptr) temp[i+1]='.';
else temp[i+1] = buffer[i-1];
}
DrawStringAt(x,y,temp,cor);
}

```

```

DrawFloatAx2(x,y,num,cor,numdec)
int x,y,numdec;
double num;
int cor;

```

```

{
char *buffer,temp[20];
int i,j,decptr,signptr;
buffer=fcvt(num,numdec,&decptr,&signptr);
if(!signptr) temp[0]=' ';
else temp[0]='-';
if (decptr<=0)
{
temp[1]='.';
j=2;
while(decptr++<0) {temp[j]='0';j++;}
for (i=0;i<=strlen(buffer);i++)
{ temp[j] = buffer[i];j++;}
}
else
for (i=0;i<=strlen(buffer)+1;i++)
{
if (i<decptr) temp[i+1] = buffer[i];
else if (i==decptr) temp[i+1]='.';
else temp[i+1] = buffer[i-1];
}
DrawStringAt(x,y,temp,cor);
}

```

```

int cin(x,y,xl,yi,xf,yf) /* Verifica se o cursor esta dentro de um determinado
rectangulo */
int x,y,xl,yi,xf,yf;
{
return((x<=xf && x>=xl) && (y<=yf && y>=yi));
}

```

```

Tipo(texto)
char *texto;
{
int tx,ty,f;

```

```

GetStringSize(texto,&tx,&ty,&f);
Janela(300,15,320+tx,37,YELLOW+XOR);
DrawStringAt(309,28,texto,INTS+BLUE);
}

```

```

/*****
Zoom
*****/

void Ampliar(zoom)
int zoom;
{
    PaintBox(210,11,333,33,WHITE);
    if (zoom==0)
    {
        PaintBox(233,14,333,30,INTS+BLACK);
        PaintBox(230,11,330,27,INTS+WHITE);
        DrawStringAt(255,21,"zoom",BLACK);
    }

    else
    {
        PaintBox(213,14,333,30,INTS+BLACK);
        PaintBox(210,11,330,27,INTS+WHITE);
        DrawStringAt(217,22,"curva completa",BLACK);
    }
    Box(305,24,315,17,BLACK);
}

/*****
Modificar Escala
*****/

void Escala(opc)

int *opc;

{ int xa,ya,xbeg,ybeg,bl;
  char ch;

  PaintBox(160,11,383,28,WHITE);
  PaintBox(183,14,383,28,INTS+BLACK);
  PaintBox(180,11,380,25,INTS+WHITE);
  DrawStringAt(191,20,"O.K.",BLACK);
  DrawStringAt(244,20,"Repetir",BLACK);
  DrawStringAt(310,20,"Cancelar",BLACK);
  Box(218,20,228,13,BLACK);
  Box(291,20,301,13,BLACK);
  Box(360,20,370,13,BLACK);
  SetMouse(300,20);
  GetMouse(&xa,&ya,&bl);

  DrawCursor(xbeg=xa,ybeg=ya);

  while ( (ch=InKey()) != ESC )

```

```

{
    GetMouse(&xa,&ya,&bl);

    if ((bl==LEFT) && (xa>213) && (xa<233) && (ya>9) && (ya<24))
    {
        UnDrawCursor();
        Box(219,19,227,14,INTS+RED);
        *opc=1;
        break;
    }

    if ((bl==LEFT) && (xa>286) && (xa<306) && (ya>9) && (ya<24))
    {
        UnDrawCursor();
        Box(292,19,300,14,INTS+RED);
        *opc=2;
        break;
    }

    if ((bl==LEFT) && (xa>355) && (xa<375) && (ya>9) && (ya<24))
    {
        UnDrawCursor();
        Box(361,21,369,16,INTS+RED);
        *opc=3;
        break;
    }

    if ( ( xa != xbeg ) || ( ya != ybeg ) )
        DrawCursor(xbeg=xa,ybeg=ya);
}

    PaintBox(160,11,383,30,WHITE);
    WaitMouseUp(LEFT);
    UnDrawCursor();
    MouseSpan(ALLSCREEN);

}

```

```

/*****
Proc. do Eixo X
*****/

void Eixos_X(yzero)

int yzero;

{
    int i;
    int posi;
    double coef;

    Line(35,yzero,535,yzero,INTS+WHITE);
    DrawStringAt(516,yzero+20,"seg.",INTS+WHITE);

    coef=(double)500.0/interval;
    i=(int)mom_ini;
    posi=(int)(coef*(i-(int)mom_ini)+35);
}

```

```

while(posi<=535)
{
    Line(posi,yzero,posi,yzero+4,INTS+WHITE);
    if (i==(int)mom_ini) posi=posi+5;
    if (posi>=533) posi=posi-5;
    DrawIntAt(posi-3,yzero+13,i,INTS+WHITE);
    i++;
    posi=(int)(coef*(i-(int)mom_ini)+35);
}
if (interval <= 5)
{
    coef=(double)50.0/interval;
    i=(int)mom_ini;
    posi=(int)(coef*(i-(int)mom_ini)+35);
    while (posi<=535)
    {
        Line(posi,yzero,posi,yzero+3,INTS+WHITE);
        i++;
        posi=(int)(coef*(i-(int)mom_ini)+35);
    }
}
coef = 500.0/4/interval;
if (interval>=25)
{
    posi=35+(int)coef;
    while (posi<=535)
    {
        line(posi,yzero,posi,yzero+2,INTS+WHITE);
        posi += 2*coef;
    }
}

/*
Proc. que calcule a escala
*/

void Limit_Eixos(graf)

int graf;

{
    double difer;

    difer = limitmax[graf] - limitmin[graf];
    if (difer==0.000) difer += 0.00001;
    precisao[graf] = 1;
    while (floor(difer)<1) {difer *=10; precisao[graf] *= 10;}
    maxim[graf] = ceil(precisao[graf]*limitmax[graf])/precisao[graf];
    if (maxim[graf] == floor(precisao[graf]*limitmax[graf])/precisao[graf]) maxim[graf] += (1.0/precisao[graf]);
    minim[graf] = floor(precisao[graf]*limitmin[graf])/precisao[graf];

    difmaxmin[graf] = maxim[graf]-minim[graf];
}

/*
Proc. do Eixo Y
*/

void Eixos_Y(graf, str)
int graf;
char *str;

{
    int i,j,yzero;
    int posy=500;
    int posx=2;

    i=0;
    DrawStringAt(12,18,str,INTS+WHITE);
    Line(35,30,35,280,INTS+WHITE);
    passo[graf]=1/(double)precisao[graf];
    while (posy>30)
    {
        if (precisao[graf]<0.01) posx=-2; else posx=0;
        posy=(int)(280-(passo[graf]*i)*250/difmaxmin[graf]);
        Line(31,posy,34,posy,INTS+WHITE);
        if (fabs(minim[graf]+passo[graf]*i)>0.000001)
            DrawFloatAx2(posx,posy-2,minim[graf]+passo[graf]*i,INTS+WHITE,(int)(log10((double)precisao[graf])));
        else DrawIntAt(20,posy,0,INTS+WHITE);
        i++;
    }
    if (((int)(difmaxmin[graf]/passo[graf])+1)<5)
    {
        posy=(int)(280-(passo[graf]/2)*250/difmaxmin[graf]);
        i=0;
        while (posy>30)
        {
            Line(32,posy,34,posy,INTS+WHITE);
            if (precisao[graf]<0.01) posx=-2; else posx=0;
            if ((difmaxmin[graf]/passo[graf]+1)*2<10)
                DrawFloatAx2(posx,posy,minim[graf]+passo[graf]/2+passo[graf]*i,INTS+WHITE,(int)(log10((double)precisao[graf])+1));
            else
                posy=(int)(posy-(passo[graf]*250/difmaxmin[graf]));
            i++;
        }
    }
    if ((difmaxmin[graf]/passo[graf]+1)*2<10)
    {
        posy=(int)(280-(passo[graf]/4)*250/difmaxmin[graf]);
        while (posy>30)
        {
            Line(32,(int)posy,34,(int)posy,INTS+WHITE);
            posy=(int)(posy-(passo[graf]*125/difmaxmin[graf]));
        }
    }
}

```

```

if (minim[graf]>=0)
    yzero = 280;
else
    yzero = (int) ( 30 + (float)maxim[graf]*250.0/(float)difmaxmin[graf]);
Eixos_X(yzero);

}

/*
*****
Proc. Escolher zona a ampliar
*****
*/

Definir_Zoom(option)
int *option;

{
    int i,b,x,y,xbb,ybb,xx,yy,xx0,yy0,cor,zz1,zz2;
    int aa,cc,dd,opcao,yy1,yy2,pnt1,pnt2,pnum1,pnum2;

    MouseSpan(36,31,535,280);
    SetMouse(240,120);
    GetMouse(&xbb,&ybb,&b);
    DrawCursor(xbb,ybb);
    while(MouseUp(LEFT))
    {
        GetMouse(&x,&y,&b);
        if ( ( x != xbb ) || ( y != ybb ) )
            DrawCursor(xbb=x,ybb=y);
    }
    MouseSpan(36,31,535,280);
    SetLineStyle(0x0033);
    GetMouse(&x,&y,&b);
    xx=xx0=x; yy=yy0=y;
    Box(xx,yy,x,y,XOR+INTS+WHITE);
    while ( ( b == LEFT ) )
    { if ( ( x != xx0 ) || ( y != yy0 ) )
        {
            UnDrawCursor();
            Box(xx,31,xx0,279,XOR+INTS+WHITE);
            Box(xx,31,x,279,XOR+INTS+WHITE);
            aa=xx;cc=xx0;dd=x;
            DrawCursor(xx0=x,yy0=y);
        }
        GetMouse(&x,&y,&b);
    }
    SetLineStyle(255);
    aa=xx;cc=xx0;
    MouseSpan(180,11,380,30);
    Escala(&opcao);
    SetLineStyle(0x0033);
    switch(opcao)
    {
        case 1:mom_fim=ceil((float)(cc-35)/500.0*interval+mom_ini);
              mom_ini=floor(mom_ini+(float)(aa-35)/500.0*interval);
              interval=mom_fim-mom_ini;
              SetLineStyle(255);
              break;
        case 2:SetLineStyle(255);
              Box(aa,31,cc,279,WHITE);
              for(i=0;i<=8;i++)
              {
                  if (curv_activ[i]==1) Tracar_Tens(i);
              }
              SetLineStyle(0x0033);
              break;
        case 3:Limpa(36,30,535,279,WHITE);
              SetLineStyle(255);
              Ecran_graf();
              zoom=0;
              break;
    }
    *option=opcao;
}

/*
*****
void Val_Max_Min_Fich()

{
    int i,k,l,j,n,part,intpol;
    double valor_lido,leit_mom;

    fout=fopen("output","r+");
    if (!fout)
    {
        win(140,140,400,230,INTS+BLUE);
        DrawStringAt(188,162,"O Ficheiro n'õ existe.",INTS+WHITE);
        DrawStringAt(210,213,"Prima LEFT,para continuar",INTS+MAGENTA);
        WaitMouseDown(LEFT);
        Limpa(INSCREEN,WHITE);
        ExitAtega();
        Exit(-1);
    }
    for (n=1;n<=8;n++)
    {
        limitmax[n]=-999;
        limitmin[n]=999;
    }
    fscanf(fout,"%lf",&leit_mom);
    while(leit_mom<mom_ini)
    {
        fscanf(fout,"%s[^\n]\n");
        fscanf(fout,"%lf",&leit_mom);
    }
    Janela (190,140,414,224,YELLOW+XOR);
    DrawStringAt(224,186,"0%",GREEN);
    DrawStringAt(370,186,"100%",GREEN);
}

```

```

Box(227,190,377,205,BLACK);

if (adam==1)
    pontos=pontos_adams;
else
    pontos = (int)((interval)/h);    /* num. total de iter.*/

part = pontos/5;
DrawStringAt(217,160,"Os dados est o a ser recolhidos",GREEN);
for (i=0;i<=pontos;i++)
{
    for (j=1;j<=8;j++)
    {
        fscanf(fout," %lf",&valor_lido);
        if(valor_lido>limitmax[j]) limitmax[j]=valor_lido;
        if(valor_lido<limitmin[j]) limitmin[j]=valor_lido;
    }
    if((i%part)==0)
    {
        intpol=(int)((228+150.0*(leit_mom-mom_ini)/interval);
        PaintBox(228,191,intpol,204,GREEN);
    }
    fscanf(fout,"%*[^\\n\\n]");
    fscanf(fout," %lf",&leit_mom);
}
fclose(fout);
if (limitmax[4]>limitmax[7]) limitmax[7]=limitmax[4];
else limitmax[4]=limitmax[7];
if (limitmin[4]<limitmin[7]) limitmin[7]=limitmin[4];
else limitmin[4]=limitmin[7];
}

/****** */

void Leit_Fich()
{
    int i,j,pontnum,k=0,l=0,graf,part,intpol,tx,ty,f;
    double leit_mom,valor_lido;

    Val_Max_Min_Fich();
    fout = fopen("output","r+");
    if (!fout)
    {
        win(140,140,400,230,INTS+BLUE);
        DrawStringAt(188,162,"O Ficheiro n'õ existe.",INTS+WHITE);
        DrawStringAt(210,213,"Prima LEFT,para continuar",INTS+MAGENTA);
        WaitMouseDown(LEFT);
        Limpa(INSCREEN,WHITE);
        ExitAtega();
        Exit(-1);
    }
    maxrest = (int)(pontos/500);

    if (maxrest>1) maxrest--;
    for (j=1;j<=8;j++) Limit_Eixos(j);
    fscanf(fout," %lf",&leit_mom);
    while(leit_mom<mom_ini)
    {
        fscanf(fout,"%*[^\\n\\n]");
        fscanf(fout," %lf",&leit_mom);
    }
    Janela (190,140,414,224,YELLOW+XOR);
    DrawStringAt(224,186,"0%",BLUE);
    DrawStringAt(370,186,"100%",BLUE);
    Box(227,190,377,205,BLACK);
    DrawStringAt(217,160,"Os dados est o a ser analisados",BLUE);
    part = pontos/5;
    for (i=0;i<=pontos;i++)
    {
        pontnum = (int)((float)(i+maxrest)/(float)pontos*500.0);
        /* posic X */
        /* PaintBox(279,175,350,200,INTS+WHITE);
        DrawFloatAt2(280,190,leit_mom,BLUE,3);*/
        if((i%part)==0)
        {
            intpol=(int)((228+150*(leit_mom-mom_ini)/interval);
            PaintBox(228,191,intpol,204,BLUE);
        }
        for (j=1;j<=8;j++)
        {
            fscanf(fout," %lf",&valor_lido);
            pix[j][pontnum] = (unsigned char)((maxim[j]-valor_lido)*250.0/difmaxmin[j]);
        }
        fscanf(fout,"%*[^\\n\\n]");
        fscanf(fout," %lf",&leit_mom);
        if(MouseDown(RIGHT)) exit(-1);
    }
    fclose(fout);
}

/****** */

Ecran_graf()
{
    int i;
    static char *t[10]={" Efd"," Tens"," Corr"," Pe"," ",
                        " Pmec"," ", " Limp"," Sair"};
    Limpa(INSCREEN,WHITE);

    if (adam==0){
        maxrest = (int)(pontos/500);
        if (maxrest>1) maxrest--;
    }

    WaitMouseUp(LEFT);
    Janela(20,300,535,334,YELLOW+XOR);
    for (i=0;i<=9;i++)

```

```

{
    DrawStringAt(46+i*50,314,t[i],INTS+BLUE);
    Box(60+i*50,325,70+i*50,317,BLACK);
    PaintBox(61+i*50,324,69+i*50,318,INTS+WHITE);
} /* endfor */
MouseSpan(INSCREEN);
}

/***** */

void Plot_Graf(graf)
int graf;

{
    int pontnum,antx,anty,k,cora,corb,tempo;

    if ( graf==5 || graf==6 )
        Eixos_Y(graf,"rad.");
    else
        Eixos_Y(graf,"p.u.");

    antx=0;anty=pix[graf][0];
    cora=15-graf;
    corb=cora;
    if(cora==9) {cora=INTS+BLUE;corb=INTS+BLUE;}
    if(cora==8) {cora=GREEN;corb=GREEN;}
    if(cora==7) {cora=INTS+WHITE;corb=WHITE;}
    PaintBox(61+(graf-1)*50,324,69+(graf-1)*50,318,corb);

    if (adam==1){
        for (k=0;k<=maxx;k++){
            tempo = guarda[k];
            Line(35+antx,anty+30,35+tempo,pix[graf][tempo]+30,cora);
            antx = tempo; anty = pix[graf][tempo];
        }
    }
    else{
        for (k=0;k<=pontos;k++){
            pontnum = (int)((float)(k+maxrest)/(float)pontos*500.0);
            Line(35+antx,anty+30,35+pontnum,pix[graf][pontnum]+30,cora);
            antx = pontnum; anty = pix[graf][pontnum];
        }
    }
}

/***** */

void Graficos()
{
    int i,j,k,l,xa,ya,xbeg,ybeg,bl,opcao;
    char ch;

    Ecran_graf();
    DefineCursor(283,BLACK,284,WHITE+INTS);

    /*if ((mom_ini!=0.0)|| (mom_fim!=tn))
        zoom = 1;
    else
        zoom = 0;*/
    SetMouse(40,320);
    GetMouse(&xa,&ya,&bl);
    /*Ampliar(zoom);*/

    DrawCursor(xbeg=xa,ybeg=ya);

    while ( (ch=InKey()) != ESC )
    {
        GetMouse(&xa,&ya,&bl);

        /* zoom */
        /*
        if ((bl==LEFT) && (xa>300) && (xa<320) && (ya>13) && (ya<28))
        {
            UnDrawCursor();
            zoom=1-zoom;
            PaintBox(306,18,314,23,RED);
            WaitMouseUp(LEFT);
            PaintBox(210,11,333,33,WHITE);
            while (zoom==1)
            {
                Definir_Zoom(&opcao);
                if (opcao!=2) break;
            }
            if (zoom==0)
            {
                mom_ini=0.0;
                mom_fim=tn;
                interval=mom_fim-mom_ini;
            }
            MouseSpan(ALLSCREEN);
            if (opcao!=3)
            {
                Leit_resl_tens();
                Ecran_graf();
                for(i=0;i<8;i++) {if (curv_activ[i]==1) Tracar_Tens(i);}
                Ampliar(zoom);
            }
        }
        */
    }
}

```

```

/* Sair */

if ((b1==LEFT) && (xa>510) && (xa<520) && (ya>317) && (ya<325))
{
    PaintBox(511,324,519,318,RED);
    WaitMouseUp(LEFT);
    break;
}

/* Limpar */

if ((b1==LEFT) && (xa>460) && (xa<470) && (ya>317) && (ya<325))
{
    UnDrawCursor();
    PaintBox(461,324,469,318,RED);
    WaitMouseUp(LEFT);
    Limpa(0,0,539,299,WHITE);
    for (i=0;i<=9;i++) PaintBox(61+i*50,324,69+i*50,318,INTS+WHITE);
    Ecran_graf();
}

/* Os Grficos */

for(i=0;i<=6;i++)
    if ((b1==LEFT) && (xa>60+i*50) && (xa<70+i*50) && (ya>317) && (ya<325))
    {
        UnDrawCursor();
        if ((i!=3) && (i!=6)) for(j=0;j<=7;j++) if(j!=i) curv_activ[j]=0;
        curv_activ[i]=1;
        WaitMouseUp(LEFT);
        if ((curv_activ[3]!=1) || (curv_activ[6]!=1))
        {
            for(j=0;j<=7;j++) if(j!=i) curv_activ[j]=0;
            for(j=0;j<=9;j++) if(j!=i) PaintBox(61+j*50,324,69+j*50,318,INTS+WHIT
E);
        }
        PaintBox(151,324,159,318,14); /*
        Limpa(0,0,539,299,WHITE);
        Plot_Graf(i+1);
        Ampliar(zoom); */
        if (( xa != xbeg) || ( ya != ybeg ) )
            DrawCursor(xbeg=xa,ybeg=ya);
        WaitMouseUp(LEFT);
        UnDrawCursor();
        MouseSpan(ALLSCREEN);
        SetMouse(100,100);
        Limpa(INSscreen,WHITE);
    }

/****** */

void Arran_Graf()
{
    Limpa(ALLSCREEN,WHITE);
    mom_ini=0.0;
    mom_fim=tn;
    interval=mom_fim-mom_ini;
    Leit_Fich();
    Graficos();
}

void Leit_Fich_Adams()
{
    int i,j,tempo,k=0,l=0,graf,part,intpol,tx,ty,f;
    double leit_mom,valor_lido;

    Val_Max_Min_Fich();
    fout = fopen("output","r+");
    if (!fout)
    {
        win(140,140,400,230,INTS+BLUE);
        DrawStringAt(188,162,"O Ficheiro n' existe.",INTS+WHITE);
        DrawStringAt(210,213,"Prima LEFT, para continuar",INTS+MAGENTA);
        WaitMouseDown(LEFT);
        Limpa(INSscreen,WHITE);
        ExitAtega();
        Exit(-1);
    }
    for (j=1;j<=8;j++) Limit_Eixos(j);
    fscanf(fout,"%lf",&leit_mom);
    while (leit_mom<mom_ini)
    {
        fscanf(fout,"%s[^\n]\n");
        fscanf(fout,"%lf",&leit_mom);
    }
    Janela (190,140,414,224,YELLOW+XOR);
    DrawStringAt(224,186,"0%",BLUE);
    DrawStringAt(370,186,"100%",BLUE);
    Box(227,190,377,205,BLACK);
    DrawStringAt(217,160,"Os dados est o a ser analisados",BLUE);
    part = pontos/5;

    guarda[0]=0.0;

    for (i=0;i<=pontos;i++)
    {
        tempo = (int)((leit_mom * 500.0)/tn);
        if ((i%part)==0)
        {

```

```
    intpol=(int) (228+150*(leit_mom-mom_ini)/interval);
    PaintBox(228,191,intpol,204,BLUE);
}
for (j=1;j<=8;j++)
{
    fscanf(fout," %lf",&valor_lido);
    plx[j][tempo] =(unsigned char) ((maxim[j]-valor_lido)*250.0/difmaxmin[j]);
}
fscanf(fout,"%*[^\\n]\\n");
fscanf(fout," %lf",&leit_mom);

if (guarda[k]!=tempo){
    k=k+1;
    guarda[k]=tempo;
}

if (MouseDown(RIGHT)) exit (~1);
}

maxx=k;
fclose(fout);
}

void Arran_Graf_Adams()
{
    Limpa(ALLSCREEN,WHITE);
    mom_ini=0.0;
    mom_fim=tn;
    interval=mom_fim-mom_ini;
    Leit_Fich_Adams();
    Graficos();
}
```



FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000101641