

Escola Politécnica da USP

Projeto de Formatura

Software

Cálculo de Perda de Carga e Carga Térmica em Tubulações de Camisa Dupla

Autor: Patrick Henry de Macedo
Prof. Orientador: J.R. Simões

Índice

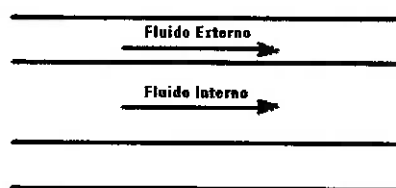
I. Introdução	1
II. Fundamentos Teóricos	2
II.a. Perda de Carga	2
II.b. Malha de Tubos	3
II.c. Método de efetividade	5
III. Programa	8
IV. Bibliografia	19
V. Apêndice - Módulo de Cálculo do programa	20

I. Introdução

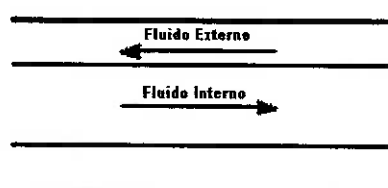
Tubulações de camisa dupla são trocadores de calor bastante simples, que visam manter a temperatura de um fluido interno entre certos limites, determinados pelo fluido que escoa pela tubulação externa, com a finalidade de, por exemplo, manter o fluido interno aquecido para que se mantenha fundido.

A motivação deste trabalho, na verdade, foi a tubulação de camisa dupla de uma fábrica de chocolates, em que o chocolate escoa na tubo interno e é aquecido por água escoando no tubo externo. Isso faz com que o chocolate, que na temperatura ambiente permanece no estado sólido, possa ser transportado por tubulações no estado líquido.

Os fluidos podem escoar no mesmo sentido ou em sentidos opostos, e isso afeta a característica do trocador, como será visto nos capítulos posteriores. A figura abaixo ilustra essa situação.



(a) Trocador de Corrente Paralela



(b) Trocador de Correntes Cruzadas

Além da determinação das temperaturas de saída, também é importante nesses tipos de trocador se determinar a perda de carga nas tubulações. Resolvendo esses problemas, dimensionamos as bombas e os aquecedores ou resfriadores do fluido que escoa no tubo externo, ou seja, resolvemos o problema de perda de carga e de carga térmica.

II. Fundamentos Teóricos

II.a. Perda de Carga

Num escoamento incompressível, a perda de carga é representada pela soma da perda de carga distribuída e da perda de carga localizada. A perda de carga distribuída de um fluido que escoar a uma velocidade constante v é dada pela célebre equação de Darcy-Weisbach, e a perda de carga localizada devido a uma singularidade é característica do elemento:

$$h_d = f_{d-w} \cdot L/D \cdot v^2 / (2g_c) \text{ - Equação de Darcy-Weisbach}$$

onde: f_{d-w} - fator de atrito

L - comprimento da tubulação

D - diâmetro da tubulação

v - velocidade de escoamento do fluido

g_c - aceleração da gravidade local

$$h_L = C \cdot v^2 / (2g_c)$$

onde: C - coeficiente de perda de carga localizada

v - velocidade de escoamento do fluido

g_c - aceleração da gravidade local

O fator de atrito é função do número de Reynolds e da rugosidade específica da tubulação:

$$Re_D = v \cdot D / \nu$$

onde: Re_D - número de Reynolds

D - diâmetro hidráulico

$D = 4 \cdot (\text{área transversal} / \text{perímetro molhado})$

ν - viscosidade cinemática

$$\epsilon = \epsilon / D$$

onde: ϵ - rugosidade relativa

ϵ - rugosidade específica da tubulação

Para o escoamento laminar ($Re_D < 2400$), o fator de atrito é dado pela equação de Darcy-Weisbach para fator de atrito:

$$f = 64 / Re_D$$

Para o escoamento turbulento completamente desenvolvido, o fator de atrito é dado pela equação de Colebrook:

$$1/(f)^{1/2} = \log(\epsilon/3,7D + 2,51/(Re_D \cdot (f)^{1/2}))^{-2}$$

A solução desta equação é obtida gráfica ou iterativamente. No nosso caso, utilizamos o método de Newton-Raphson para chegar na sua solução. O método consiste em, através da derivada da equação, chegar rapidamente à solução da seguinte forma: dá-se um palpite inicial para f e se substitui o mesmo em y e em y' . Calcula-se então um $\Delta = y/y'$. Então calculamos $f - \Delta$ para se chegar ao novo f , até que tenhamos uma precisão determinada entre f e o f calculado anteriormente.

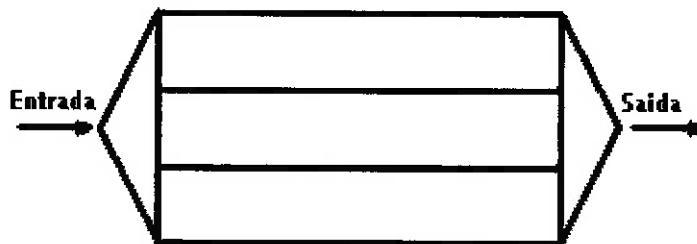
$$y = 1/(f)^{1/2} + 2 \cdot \log((0,27 \cdot \epsilon/D \cdot Re_D \cdot (f)^{1/2} + 2,51 \cdot D)/(D \cdot Re_D \cdot (f)^{1/2}))$$

$$y' = -0,5 \cdot (f)^{-1,5} + (2,51 \cdot D)/(0,27 \cdot \epsilon/D \cdot Re_D \cdot (f)^{1/2} + 2,51 \cdot D) \cdot \ln(10)$$

O coeficiente de perda de carga localizada C é obtido experimentalmente, e pode ser obtidos em diversos livros para vários tipos de singularidades. Os coeficientes utilizados no programa foram obtidos no livro "*Analysis and Design of Energy Systems*" do autor B. K. Hodge.

II.b. Malha de Tubos

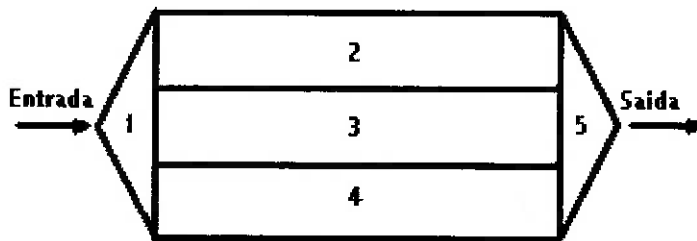
Quando temos uma malha de tubos, não temos como determinar por processos convencionais a vazão que vai para cada tubo em função da vazão de entrada e da de saída:



Para obtermos esta solução utilizamos o método de Hardy-Cross. Este é um método iterativo, baseado na conservação da massa em cada nó (encontro das tubulações) e na igualdade das pressões nos mesmos. Fazendo uma analogia com o método de Newton-Raphson, o método de Hardy-Cross também se utiliza da derivada da equação de perda de carga nos tubos para convergir mais rapidamente:

$$\Delta Q_i = - \frac{\sum_{j=1}^J \left\{ f_{ij} \frac{L_{ij}}{D_{ij}} \frac{V_{ij}|V_{ij}|}{2g_i} + \text{SGN}(Q_{ij})A_{ij} + \sum_{m=1}^M B_{ijm} Q_{ij}^m |Q_{ij}|^{m-1} \right\}}{\sum_{j=1}^J \left\{ f_{ij} \frac{L_{ij}}{D_{ij}} \frac{|V_{ij}|}{g_i A_{ij}} + \sum_{m=1}^M B_{ijm} m |Q_{ij}|^{m-1} \right\}}$$

Os coeficientes A_{ij} e B_{ijm} correspondem a singularidades que levam a uma perda de carga proporcional à vazão nos tubos. Os índices i e j correspondem a tubo j do loop i .



Isso porque o método de Hardy-Cross exige que se dê um número para cada malha de tubos fechadas, e um número para cada tubo de cada malha. Assume-se um sentido de escoamento para o fluido no loop (sempre tomando cuidado para que as vazões em cada nó se conservam) e dá-se os palpites iniciais de vazão. A solução é obtida quando ΔQ_i em todos os tubos for menor que uma precisão pré-determinada.

II.c. Método da Efetividade

O método da efetividade ou método de NUT (número de unidades de transferência) é um método de se calcular a temperatura de saída de dois fluidos num trocador de calor tendo dados o coeficiente global de troca de calor entre eles, a vazão e calor específico dos fluidos e suas temperaturas de entrada. O método é baseado no conceito de efetividade, que vem da 2ª Lei da Termodinâmica, que determina a máxima quantidade de calor que pode ser trocada entre dois fluidos, baseando-se no conceito de capacidade térmica:

$$C_{h,c} = m'_{h,c} \cdot c_{p,h,c} \quad \text{onde } m' \text{ é a vazão mássica}$$

c_p é o calor específico

o índice h representa o fluido quente

o índice c representa o fluido frio

O máximo calor que pode ser trocado é determinado pela menor capacidade térmica dos fluidos. Chamando-se $C_{\max}/C_{\min}$ de C, efetividade é determinada pela fórmula:

$$\xi = C \cdot (T_{c,h \text{ saída}} - T_{c,h \text{ entrada}}) / (T_{h,c \text{ entrada}} - T_{c,h \text{ entrada}})$$

Note que assim conseguimos determinar a temperatura de saída do fluido quente e do fluido frio. ξ é determinado pelas fórmulas:

$$\xi = \{ (1 - \exp[-NUT(1 + C)]) / (1 + C) \} - \text{para trocadores de corrente paralela}$$

$$\xi = \{ (1 - \exp[-NUT(1 + C)]) / \{ 1 - C \cdot \exp[-NUT(1 + C)] \} \} - \text{para trocadores de corrente cruzada}$$

O número de unidades de transferência é determinado pela fórmula:

$$NUT = U \cdot A / C_{\min} \quad \text{onde } U - \text{coeficiente global de troca de calor}$$

A - área de troca de calor

A única questão agora é determinar o coeficiente global de troca de calor U. Para o nosso problema desconsideramos as resistividades das tubulações e supomos como importantes na troca de calor apenas a convecção dos fluidos e do ambiente.

Para a troca de calor com um tubo no ambiente, o coeficiente de convecção do ar depende da difusividade, da viscosidade, da condutividade, da densidade, do calor específico e da temperatura do ar, além do diâmetro e da temperatura da superfície do tubo. É obtido pela fórmula:

$$h_{ar} = k_{ar}/D_{tubo} \cdot Nu$$

onde - k_{ar} é a condutividade térmica do ar

Nu é o número de Nusselt

Para um cilindro horizontal comprido o número de Nusselt é obtido por:

$$Nu = \{0,60 + [0,387 \cdot Ra^{1/6} / \{1 + (0,559/Pr)^{9/16}\}^{8/17}]\}^2$$

$$\text{para } 10^{-5} < Ra < 10^{12}$$

onde - Ra é o número de Rayleigh

Pr é o número de Prandtl

O número de Rayleigh e o número de Prandtl para este caso pode ser obtido pelas fórmulas:

$$Ra = g \cdot \beta \cdot (T_s - T_\infty) \cdot D^3 / \nu \cdot \alpha$$

onde - g é a aceleração da gravidade

$$Pr = \nu \cdot \rho \cdot c_p / k$$

β é $(T_s + T_\infty)/2$

T_s é a temperatura superficial do tubo

T_∞ é a temperatura ambiente

ν é a viscosidade do ar

α é a difusividade do ar

ρ é a densidade do ar

Para obtermos o coeficiente de convecção nos tubos basta determinarmos o número de Nusselt. Para o escoamento turbulento utilizamos a equação de Dittus-Boelter:

$$Nu = 0,023 \cdot Re_D^{4/5} \cdot Pr^{1/3}$$

O expoente $1/3$ é na verdade uma aproximação, pois a equação de Dittus-Boelter na verdade exige que ele seja 0,3 no resfriamento e 0,4 no aquecimento. Essa equação é válida para $0,7 \leq Pr \leq 160$, $Re \geq 10000$ e $L/D \geq 10$. Para o escoamento laminar ($Re < 2.400$) o número de

Nusselt é 4,36. No programa assumi a formulação de Dittus-Boelter para $Re \geq 4000$ e $Nu = 4,36$ para $Nu < 4000$, mas isso é apenas uma aproximação.

O coeficiente de convecção é dado por:

$$h = Nu \cdot k / D_H$$

Com os coeficientes de convecção obtidos, fica fácil calcular o coeficiente global de troca de calor. Para isso utilizamos o método das resistências:

$$U = 1 / (\Sigma res)$$

As resistências res são o inverso das condutividades, e são obtidos da seguinte forma:

$$res_{int} = r_o / (r_i \cdot h_i) \quad \text{onde } r_o \text{ é o raio do tubo externo}$$

$$res_{ext} = r_o / (h_e \cdot (r_i + r_e)) \quad r_i \text{ é o raio do tubo interno}$$

$$res_{amb} = 1 / h_{ar} \quad h_i \text{ é o coeficiente de convecção do fluido interno}$$

$$h_e \text{ é o coeficiente de convecção do fluido externo}$$

O coeficiente de troca global é:

$$U = 1 / (res_{int} + res_{ext} + res_{amb})$$

Mais uma vez insisto que a condutividade térmica nas tubulações foi ignorada para simplificar o problema, pois levá-la em conta não alteraria muito o resultado.

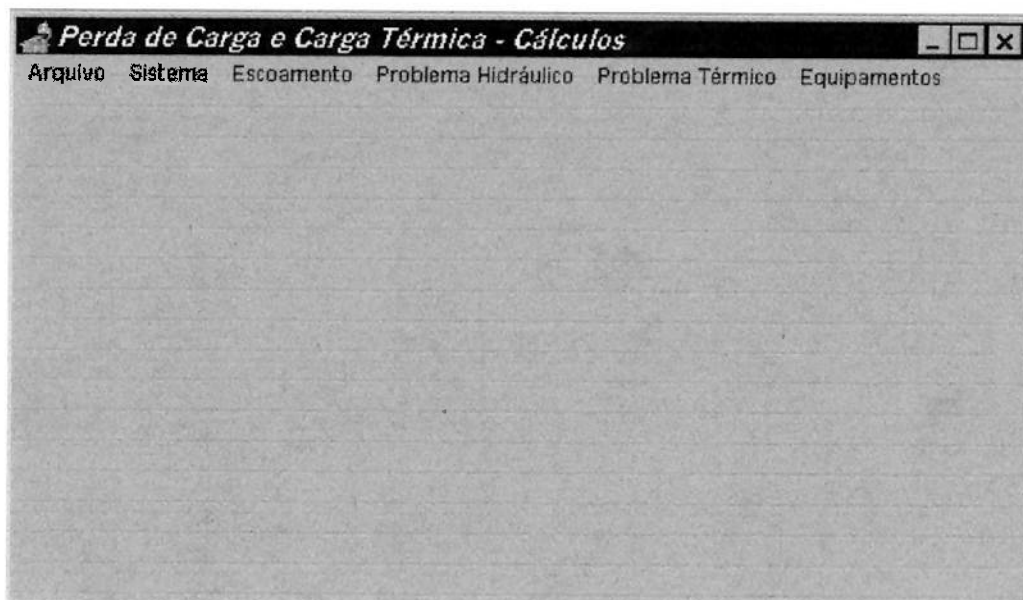
III. O Programa

Agora que já descrevemos os fundamentos que embasaram o programa, vamos ao seu funcionamento.

Primeiramente vale lembrar que os cálculos só funcionam com unidades coerentes. Portanto é fundamental prestar atenção às unidades pedidas nas telas.

1. A Tela Inicial

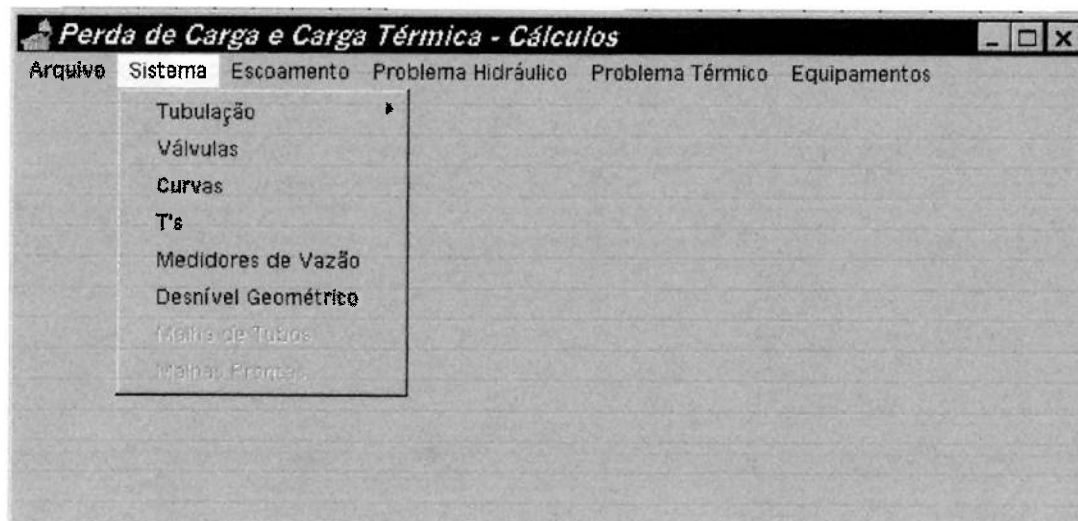
A tela inicial contém os menus onde se deve fazer a entrada de dados. Pode-se tanto abrir um arquivo previamente gravado no programa, ou iniciar uma nova entrada de dados.



A tela inicial

Para começar, você deve entrar com os dados da tubulação. Esta pode ser tanto de camisa dupla como simples. Em seguida você pode também entrar com as singularidades ou desnível geométrico.

Note que os outros menus estão desabilitados. Você pode também entrar com as singularidades. Se qualquer dos dados for zero, o programa não continuará, ou seja, você deve entrar com todos os dados.



O menu *Malha de Tubos* e *Malhas Prontas* fica desabilitado até que se entre com os dados da tubulação simples. Ele utiliza esses dados para obter a rugosidade da tubulação.

A screenshot of a software window titled "Tubulação". The window contains several input fields and two buttons. The input fields are labeled as follows: "Comprimento da Tubulação [m]", "Diâmetro da Tubulação Externa [m]", "Espessura da Tubulação Externa [m]", "Rugosidade da Tubulação Externa [m]", "Diâmetro da Tubulação Interna [m]", "Espessura da Tubulação Interna [m]", and "Rugosidade da Tubulação Interna [m]". The "Rugosidade da Tubulação Externa [m]" field is currently selected. To the right of the input fields are two buttons: "Gravar" and "Limpar".

Em seguida você deve entrar com os dados dos fluidos, clicando no menu *Fluidos* que está no menu *Escoamento*.

The screenshot shows a software window titled "Fluidos". It is divided into two main sections: "Fluido Interno" (Internal Fluid) and "Fluido Externo" (External Fluid). Each section has an "Escoamento" (Flow) subsection and a "Propriedades do Fluido" (Fluid Properties) subsection. The "Escoamento" subsection contains input fields for "Vazão [m³/s]" (Flow rate) and "Temperatura de Entrada [°C]" (Inlet temperature). The "Propriedades do Fluido" subsection contains input fields for "Viscosidade Cinemática [m²/s]" (Kinematic viscosity), "Densidade [kg/m³]" (Density), "Condutividade Térmica [kW/m.K]" (Thermal conductivity), and "Calor Específico [kJ/kg.K]" (Specific heat). At the bottom of the "Fluido Externo" section, there is a "Sentido dos Escoamentos" (Flow direction) subsection with two radio buttons: "Trocador de Correntes Paralelas" (Parallel flow exchanger) and "Trocador de Correntes Cruzadas" (Crossed flow exchanger). The "Gravar" (Save) and "Limpar" (Clear) buttons are located at the bottom left of the window.

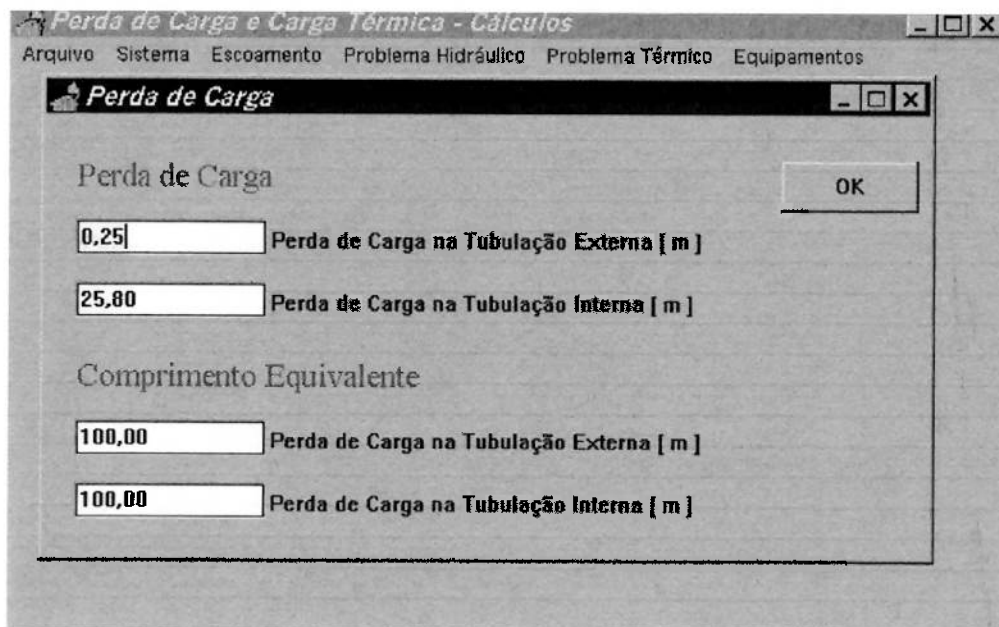
Fluido Interno	Fluido Externo
Escoamento	Escoamento
Vazão [m³/s]: 0,00110	Vazão [m³/s]: 0,00140
Temperatura de Entrada [°C]: 40,0	Temperatura de Entrada [°C]: 65,0
Propriedades do Fluido	Propriedades do Fluido
Viscosidade Cinemática [m²/s]: 0,0000050	Viscosidade Cinemática [m²/s]: 0,0000007
Densidade [kg/m³]: 1,3000	Densidade [kg/m³]: 1,0000
Condutividade Térmica [kW/m.K]: 0,1440000	Condutividade Térmica [kW/m.K]: 0,6200000
Calor Específico [kJ/kg.K]: 2,240	Calor Específico [kJ/kg.K]: 4,180
Sentido dos Escoamentos	
☒ Trocador de Correntes Paralelas	
☐ Trocador de Correntes Cruzadas	

No exemplo acima, escolhemos uma tubulação de camisa dupla, portanto é necessário se preencher os dados dos dois fluidos. Antes de partir para o cálculo, devemos informar a temperatura ambiente. Isso é feito no menu *Escoamento*, item *Temperatura Ambiente*.

The screenshot shows a software window titled "Temperatura Ambiente". It contains a single input field for "Temperatura Ambiente [°C]" with the value "20,00". To the right of the input field are two buttons: "Grava" (Save) and "Limpa" (Clear).

Temperatura Ambiente
Temperatura Ambiente [°C]: 20,00
Grava
Limpa

Com isso podemos partir para a próxima etapa, que é obter a perda de carga na tubulação. Para isso, devemos selecionar o menu *Problema Hidráulico*, no item *Cálculos*. Prontamente obteremos a perda de carga nas tubulações e o comprimento equivalente.



Perda de Carga e Carga Térmica - Cálculos

Arquivo Sistema Escoamento Problema Hidráulico Problema Térmico Equipamentos

Perda de Carga OK

Perda de Carga

0,25 Perda de Carga na Tubulação Externa [m]

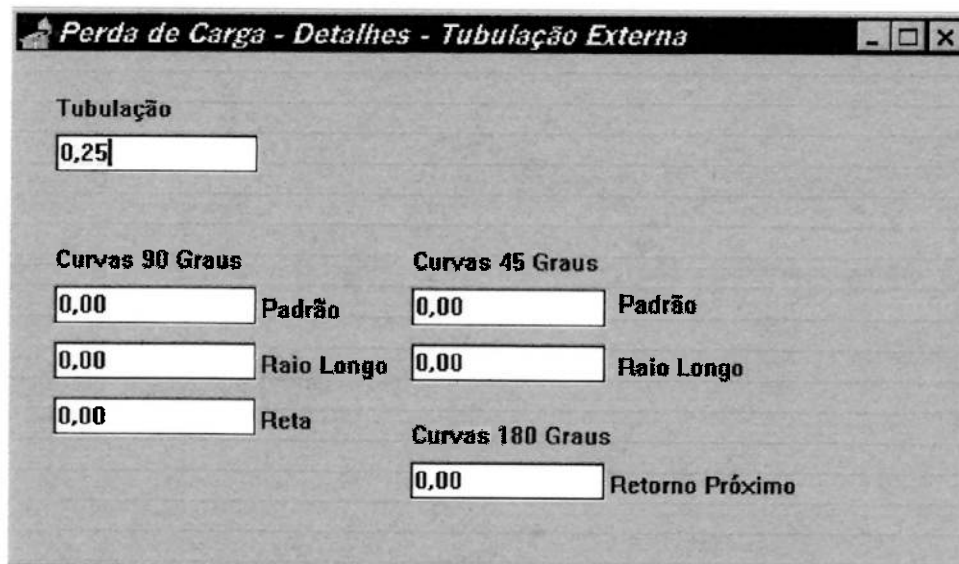
25,80 Perda de Carga na Tubulação Interna [m]

Comprimento Equivalente

100,00 Perda de Carga na Tubulação Externa [m]

100,00 Perda de Carga na Tubulação Interna [m]

Caso queiramos, podemos visualizar a perda de carga em cada componente da tubulação. Basta no menu *Problema Hidráulico* selecionar *Visualizar Perdas*.



Perda de Carga - Detalhes - Tubulação Externa

Tubulação

0,25

Curvas 90 Graus

0,00 Padrão

0,00 Raio Longo

0,00 Reta

Curvas 45 Graus

0,00 Padrão

0,00 Raio Longo

Curvas 180 Graus

0,00 Retorno Próximo

Perda de Carga - Detalhes - Tubulação Interna

Tubulação		Válvulas de Retenção	
25,80		0,00 Disco	0,00 Esfera
Curvas 90 Graus	Medidores de Vazão	Válvulas de Diafragma	
0,00 Padrão	0,00 Disco	0,00 25% Aberta	
0,00 Raio Longo	0,00 Pistão	0,00 50% Aberta	
0,00 Reta	0,00 Rotativo	0,00 75% Aberta	
Curvas 45 Graus	0,00 Turbina	0,00 100% Aberta	
0,00 Padrão		Válvulas Borboleta	
0,00 Raio Longo		0,00 5o Aberta	
Curvas 180 Graus	Válvula de Pé	0,00 10o Aberta	
0,00 Retorno Próximo	0,00	0,00 20o Aberta	
T's		0,00 40o Aberta	

O próximo passo é resolver o problema térmico. Para isso selecionamos o menu *Problema Térmico*, no item calcular. Imediatamente o programa nos fornecerá a resposta sobre a carga térmica que ocorre no sistema.

Perda de Carga e Carga Térmica - Cálculos

Arquivo Sistema Escoamento Problema Hidráulico **Problema Térmico** Equipamentos

Carga Térmica

6,355	Coefficiente de Convecção Ambiente [kW / m² . K]
2,468	Coefficiente Global de Troca de Calor [kW / m² . K]
0,0792	Efetividade
40,00	Temperatura de Entrada do Fluido Frio [o C]
65,00	Temperatura de Entrada do Fluido Quente [o C]
41,98	Temperatura de Saída do Fluido Frio [o C]
64,17	Temperatura de Saída do Fluido Quente [o C]
0,005	Carga Térmica [kW]

Com isso já podemos resolver problemas de perda de carga e carga térmica em tubulações de camisa dupla. Agora precisamos aprender a resolver problemas de redes de tubos, possível no nosso programa.

Primeiramente devemos selecionar o menu *Sistema*, e entrar com os dados em *Tubulação Simples*.

Dados da Tubulação

10 Comprimento da Tubulação [m]

0,05 Diâmetro da Tubulação [m]

0,003 Espessura da Tubulação [m]

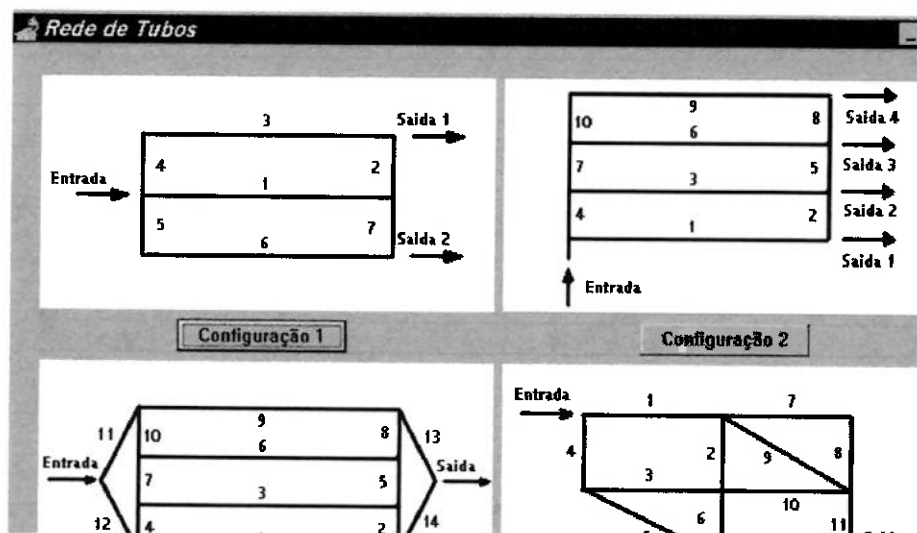
0,0001 Rugosidade da Tubulação [m]

Gravar

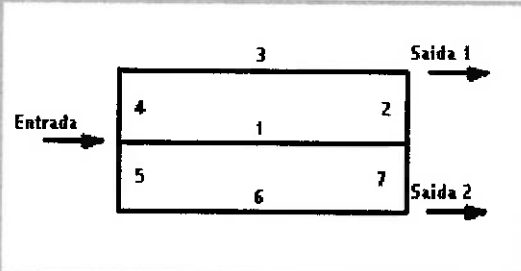
Limpar

Isso porque é desse menu que o programa obtém a rugosidade da tubulação da rede.

Em seguida, com os menus *Malhas Prontas* e *Malhas de Tubos* habilitados, selecionamos um para efetuar os cálculos. No nosso caso vou selecionar uma malha pronta.



Escolherei a configuração 1, que é a mais simples, mas é bem elucidativa.



Entrada →

Saída 1 →

Saída 2 →

OK

Limpa

0,3 Vazão de Entrada

0,5 Diâmetro do Tubo 1

50 Comprimento do Tubo 1

0,3 Diâmetro do Tubo 2

30 Comprimento do Tubo 2

0,4 Diâmetro do Tubo 3

50 Comprimento do Tubo 3

0,5 Diâmetro do Tubo 4

30 Comprimento do Tubo 4

0,6 Diâmetro do Tubo 5

60 Comprimento do Tubo 5

0,5 Diâmetro do Tubo 6

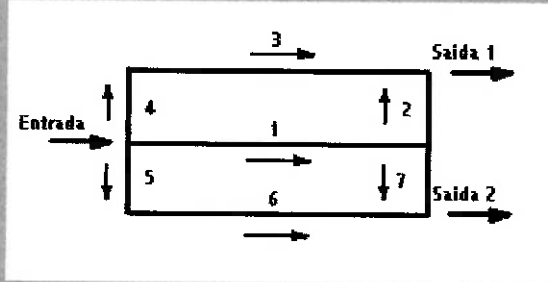
50 Comprimento do Tubo 6

0,5 Diâmetro do Tubo 7

60 Comprimento do Tubo 7

Inserir Singularidade

Com os dados inseridos, clicamos em OK e a resposta será dada.



Entrada →

Saída 1 →

Saída 2 →

OK

0,110 Vazão do Tubo 1

0,029 Vazão do Tubo 2

0,071 Vazão do Tubo 3

0,071 Vazão do Tubo 4

0,118 Vazão do Tubo 5

0,118 Vazão do Tubo 6

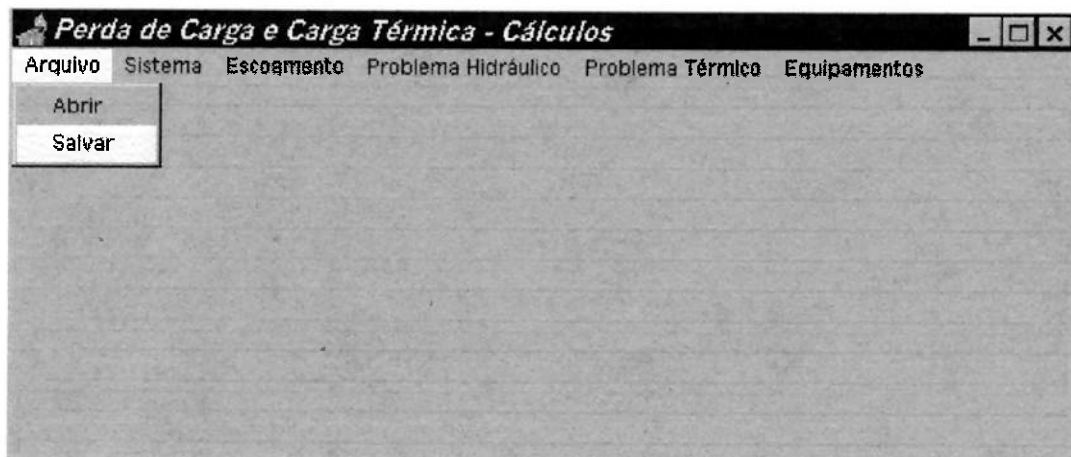
0,082 Vazão do Tubo 7

0,100 Vazão na Saída 1

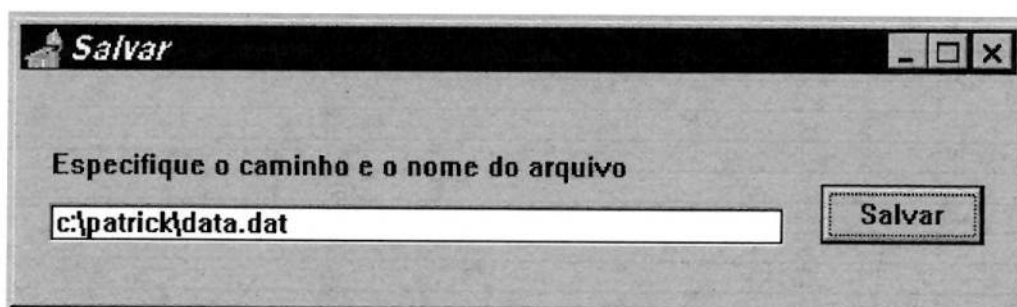
0,200 Vazão na Saída 2

O grande problema é que o programa assume as vazões de saída das tubulações. Mais a frente iremos abordar a solução deste problema.

Agora podemos gravar os dados em arquivo. Isso é feito no menu *Arquivo*, no item *Salvar*.



Uma janela pedindo o caminho para o arquivo aparecerá. Ali você deve especificar o disco, o diretório e o nome do arquivo, com extensão inclusive.

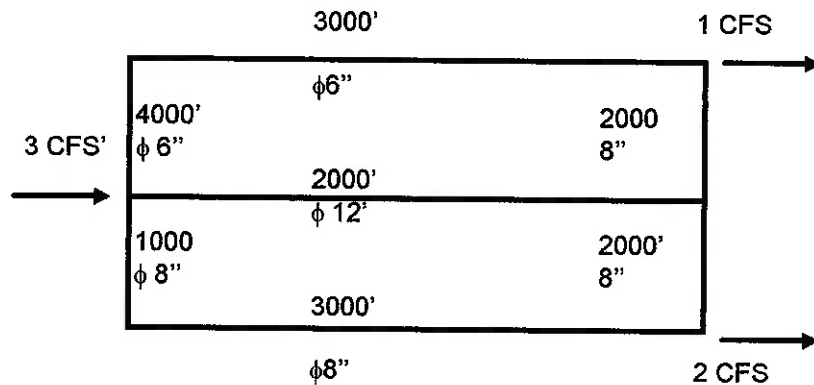


Para abrir o arquivo, basta especificar o mesmo caminho, e os dados armazenados serão novamente lidos.

Quando a malha de tubos não está pré-definida, devemos no menu *Sistema* selecionar o item *Malha de Tubos*. Uma janela pedindo o número de loops aparecerá.

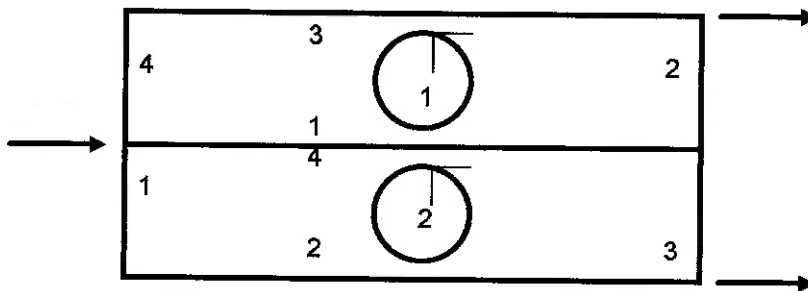
Loop é definido como a malha fechada de um sistema de tubulações. No exemplo da página 5 vemos que a tubulação tem 5 loops. O próximo passo é numerar os tubos de cada malha e definir um sentido de escoamento. As vazões são definidas pela continuidade da vazão nos nós. Deve-se entrar com os dados dos tubos e com as vazões. Veja o exemplo a seguir.

Exemplo: Calcule a vazão nos tubos na malha abaixo:



O fluido passando nos tubos é a água e a rugosidade dos tubos é de 0,0001ft.

Solução: Primeiramente deve-se numerar as malhas e os tubos, dando-se um sentido de escoamento para cada malha e uma vazão, verificando a continuidade nos nós.

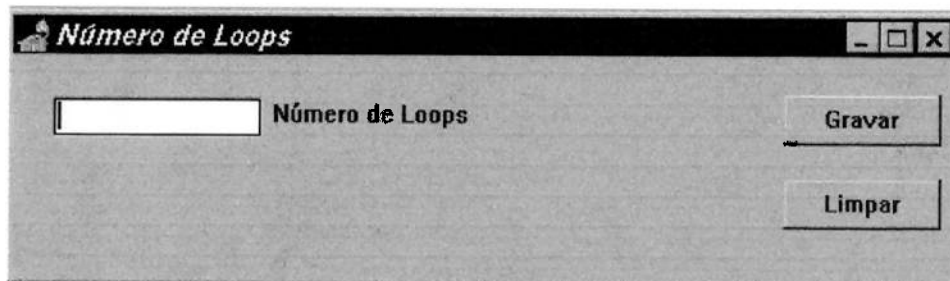


Vamos agora supor vazões em cada tubo. Por exemplo, $Q_{1,2}$ é a vazão do tubo 2 no loop 1. Deve se observar o sentido do loop e a vazão chutada. Se tiverem sentidos opostos, Q deve ter sinal negativo.

Vazão de entrada: $3 \text{ m}^3/\text{s}$

$Q_{1,1} = -Q_{2,4} = 0,8 \text{ m}^3/\text{s}$	$Q_{1,4} = -1,2 \text{ m}^3/\text{s}$	$Q_{2,3} = -1,0 \text{ m}^3/\text{s}$
$Q_{1,2} = -0,2 \text{ m}^3/\text{s}$	$Q_{2,1} = 1,0 \text{ m}^3/\text{s}$	
$Q_{1,3} = -1,2 \text{ m}^3/\text{s}$	$Q_{2,2} = 1,0 \text{ m}^3/\text{s}$	

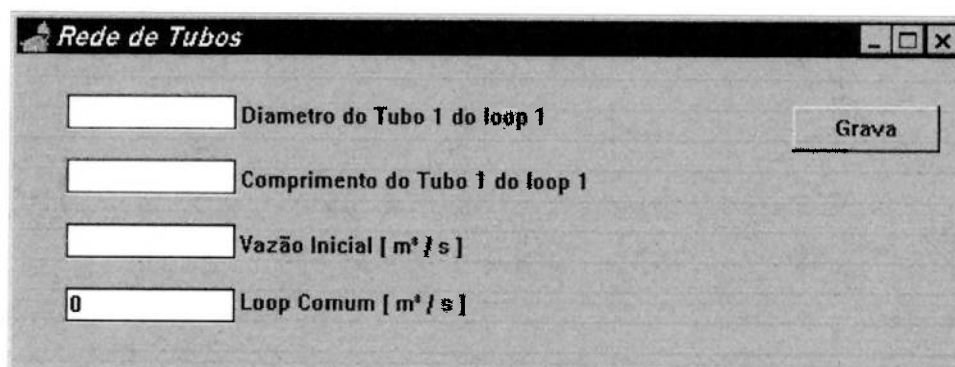
Vamos resolver o problema com o programa. Entre com a rugosidade no menu *Sistema* item *Tubulação Simples*, como explicado anteriormente. Agora o menu *Malha de Tubos* está habilitado. Clique nele.



A screenshot of a software window titled "Número de Loops". It features a text input field on the left, currently empty. To its right is the label "Número de Loops". On the right side of the window, there are two buttons: "Gravar" (Save) and "Limpar" (Clear).

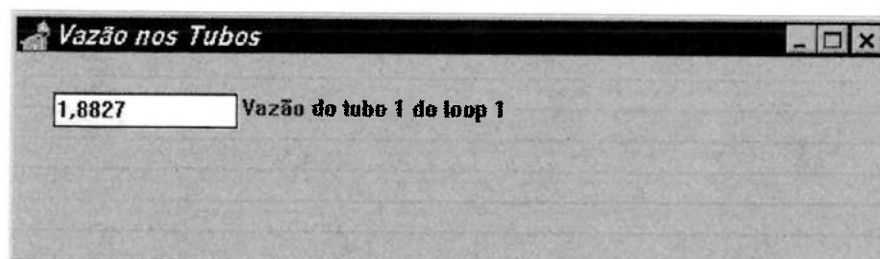
O número de loops é 2. Na sequência, o número de tubos no loop 1 é 4 e no loop 2 é 4 também.

Agora o programa solicita os dados dos tubos. Basta preenchê-los.



A screenshot of a software window titled "Rede de Tubos". It contains four text input fields stacked vertically. The first is labeled "Diâmetro do Tubo 1 do loop 1", the second "Comprimento do Tubo 1 do loop 1", the third "Vazão Inicial [m³ / s]", and the fourth "Loop Comum [m³ / s]" with the value "0" entered. A "Grava" (Save) button is located on the right side.

Após todos os dados serem digitados, as vazões nos tubos serão fornecidas uma a uma, na seguinte tela:



A screenshot of a software window titled "Vazão nos Tubos". It shows a single text input field containing the value "1,8827". To the right of the field is the label "Vazão do tubo 1 do loop 1".

Só para ilustrar as respostas são:

$$\begin{array}{lll} Q_{1,1} = -Q_{2,4} = 1.8827 \text{ m}^3/\text{s} & Q_{1,4} = -0,2308 \text{ m}^3/\text{s} & Q_{2,3} = -1,1135 \text{ m}^3/\text{s} \\ Q_{1,2} = 0,7692 \text{ m}^3/\text{s} & Q_{2,1} = 0,8865 \text{ m}^3/\text{s} & \\ Q_{1,3} = -0,2308 \text{ m}^3/\text{s} & Q_{2,2} = 0,8865 \text{ m}^3/\text{s} & \end{array}$$

Para sistemas mais complicados o sistema é o mesmo, bastando apenas tomar cuidado em alguns casos onde o método de Hardy-Cross pode não convergir.

IV. Conclusão

O programa facilita o projeto de tubulações, dimensionando bombas e estimando a temperatura de saída de fluidos em tubulação de camisa dupla. Pode ser aplicado facilmente em qualquer instalação hidráulica, e tem fácil aprendizado.

Caso haja interesse, nos disquetes estão todos os códigos de programação. Para aperfeiçoá-lo basta possuir um compilador Delphi.

Bibliografia:

Hodge, B.K. - Analysis and Design of Energy Systems - Segunda Edição - Prentice Hall

Frank, P. Incropera e David P. de Witt - Fundamentos de Transferência de Calor e de Massa - Terceira Edição - Guanabara Koogan

Mesquita, Telmo João Martins Mesquita - Turbo Pascal - Primeira Edição - Editora Érica

Cornell, Gary e Strain, Troy - Delphi Segredos e Soluções - Primeira Edição - Makron

Apostila de PMC-513 - Modelagem e Simulação de Sistemas Termofluidos

Apostila de PMC-331 - Mecânica dos Fluidos

Apêndice - Módulo de Cálculo do Programa

Pascal

```
uses
```

```
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  
Forms, Dialogs, StdCtrls, Menus, Tubosimp, Numtub, Numloop, Camisadu,  
Resphard, Tempamb1, Fluidos, Tubosim, Fluid;
```

```
var numtubos: array[1..10] of integer;  
    diam, comp, perdar, derperdar, q, a, b, c: array[1..10,1..10] of real;  
    loopcomm: array[1..10,1..10] of integer;  
    vazao: real;
```

```
    numloops, cont, conf: integer;
```

```
function min(a,b:real):real;  
function friction(vazao,diam,rugos,visc:real):real;  
function carga(vazao,diam,comp,rugosidade,viscosidade:real):real;  
function dercarga(vazao,diam,comp,rugosidade,viscosidade:real):real;  
function area(diam:real):real;  
function sgn(num:real):real;  
function reynolds(vazao,diam,visc:real):real;  
function veloc(vazao,diam:real):real;  
function cargasing(vazao,k,diamext,diamint:real):real;  
function diamhid(diamext,diamint:real):real;  
function compequival(perda,vazao,diam,rugos,visc:real):real;  
function elev(num,expon:real):real;  
function potencia(densidade,vazao,carga:real):real;  
procedure leitura;  
procedure hardycalc;  
procedure hardyresult;  
procedure inicializa1;  
procedure inicializa2;  
procedure inicializa3;  
procedure inicializa4;  
function alfa(temperatura:real):real;  
function viscosidadear(temperatura:real):real;  
function conductividadear(temperatura:real):real;  
function PrAr(temperatura:real):real;  
function hamb(tempamb,temp sup,diam:real):real;  
function Pr(visc,cond,dens,caloresp:real):real;  
function conductividadeagua(temperatura:real):real;  
function Nusselt(vazao,diam,visc,temperatura,dens,caloresp,cond:real):real;  
function h(vazao,diam,visc,temperatura,cond,dens,caloresp:real):real;  
function U(hint,hext,rint,rent,espirit,espect,hamb:real):real;  
function efetiv(vazoint,vazaoext,cpint,cpext,U,A:real):real;  
function cargaterm(efetiv,vazoint,vazaoext,cpint,cpext,tentint,tentext:real):real;  
function tempsaidaqueante(temp,cargaterm,vazao,cp:real):real;  
function tempsaidafrio(temp,cargaterm,vazao,cp:real):real;
```

Implementation

```
const g=9.81;
```

```

function min(a,b:real):real;
begin
  if a<b then
    min:=a
  else
    min:=b;
  end;
end;

function max(a,b:real):real;
begin
  if a>b then
    max:=a
  else
    max:=b;
  end;
end;

function log(num:real):real;
begin
  log:=ln(num)/ln(10);
end;

function sgn(num:real):real;
begin
  sgn:=abs(num)/num;
end;

function elev(num,expo:real):real;
var aux1,aux2,res:real;
begin
  aux1:=0;
  if (expo>=1) or (expo<=-1) then
    aux1:=expo/int(expo);
    if aux1=1 then
      begin
        aux2:=expo;
        res:=num;
        while abs(aux2)>1 do
          begin
            res:=num*res;
            if aux2<0 then
              aux2:=aux2+1
            else
              aux2:=aux2-1;
            end;
          if (expo<0) and (num<>0) then
            elev:=1/res
          else
            elev:=res;
          end
        end
      end
    end
  end
end;

```



```

begin
  if expoen>0 then
    elev:=exp(ln(num)*expoen);
  if expoen=0 then
    elev:=1;
  if expoen<0 then
    elev:=1/exp(ln(num)*expoen*(-1));
  end;
  if num=0 then
  begin
    if expoen>0 then
      elev:=0;
    if expoen=0 then
      elev:=1;
    if expoen<0 then
      begin
        ShowMessage('Houve erro na Exponenciacao');
        elev:=4;
      end;
    end;
  if num<0 then
  begin
    ShowMessage('Houve erro na Exponenciacao');
    elev:=1;
  end;
  end;

function area(diam:real):real;
begin
  area:=elev(diam,2)*pi/4;
end;

function veloc(vazao,diam:real):real;
begin
  veloc:=vazao/area(diam);
end;

function reynolds(vazao,diam,visc:real):real;
begin
  reynolds:=abs(veloc(vazao,diam)*diam/visc);
end;

function rr(diam,rugos:real):real;
begin
  rr:=rugos/diam;
end;

function funcao(f,rugos,vazao,diam,visc:real):real;
var aux: real;

```

```

end;

function derivada(f,diam,rugos,vazao,visc:real):real;
begin
    derivada:=-0.5*elev(f,-1.5)+(2.51*diam/(0.27*rugos*reynolds(vazao,diam,visc)*sqrt(f)+2.51*diam)/ln(10));
end;

function friction(vazao,diam,rugos,visc:real):real;
var delta,f,fant: real;
begin
    cont:=0;
    f:=0.02;
    delta:=funcao(f,rugos,vazao,diam,visc)/derivada(f,diam,rugos,vazao,visc);
    if reynolds(vazao,diam,visc)<2400 then
        friction:=64/reynolds(vazao,diam,visc)
    else
    begin
        repeat
            begin
                fant:=f;
                delta:=funcao(f,rugos,vazao,diam,visc)/derivada(f,diam,rugos,vazao,visc);
                cont:=cont+1;
            end;
            until (f-fant<prec);
            friction:=f;
        end;
    end;

function carga(vazao,diam,comp,rugosidade,viscosidade:real):real;
var v,re,ff,aux:real;
begin
    carga:=friction(vazao,diam,rugosidade,viscosidade)*comp*veloc(vazao,diam)*abs(veloc(vazao,diam))/(diam*2*g);
end;

function dercarga(vazao,diam,comp,rugosidade,viscosidade:real):real;
var v,re,ff,aux:real;
begin
    dercarga:=friction(vazao,diam,rugosidade,viscosidade)*comp*abs(veloc(vazao,diam))/(diam*area(diam)*g);
end;

function cargasing(vazao,k,diamext,diamint:real);
begin
    cargasing:=k*elev(veloc(vazao,diamhid(diamext,diamint)),2)/(2*g);
end;

function diamhid(diamext,diamint:real);
var areasecao,perimolhado:real;
begin
    areasecao:=area(diamext)-area(diamint);

```

```

function compequival(perda,vazao,diam,rugos,visc: real):real;
begin
    compequival:=perda*diam*2*g/(friction(vazao,diam,rugos,visc)*elev(veloc(vazao,diam),2));
end;

function potencia(densidade,vazao,carga:real):real;
begin
    potencia:=densidade*g*vazao*carga;
end;

procedure leitura;
var i,j,numloops:integer;
begin
    loops.showmodal;
    numloops:=numloopsaux;
    for i:=1 to numloops do
    begin
        numtubo.label1.caption:=('Número de tubos do loop '+inttostr(i));
        numtubo.showmodal;
        numtubos[i]:=numtubosaux;
    end;
    for i:=1 to numloops do
        for j:=1 to numtubos[i] do
            begin
                tubo.label1.caption:=('Diametro do Tubo '+inttostr(j)+' do loop '+inttostr(i));
                tubo.label2.caption:=('Comprimento do Tubo '+inttostr(j)+' do loop '+inttostr(i));
                tubo.showmodal;
                diam[i,j]:=diamaux;
                comp[i,j]:=compaux;
                q[i,j]:=qaux;
                loopcomum[i,j]:=loopcom;
                a[i,j]:=0;
                b[i,j]:=0;
                c[i,j]:=0;
            end;
        end;
    end;

procedure hardycalc;
var st, sb, dq: array[1..10] of real;
    tt,tb:real;
    i, j, ic, it: integer;
begin
    ic:=0;
    it:=0;
    while ic=0 do
    begin
        for i:=1 to numloops do
            begin
                st[i]:=0;

```

```

tt:=0;
tb:=0;
tt:=a[i,j]*sgn(q[i,j])+b[i,j]*veloc(q[i,j],diam[i,j])+c[i,j]*q[i,j]*abs(q[i,j]);
tb:=b[i,j]+2*c[i,j]*abs(q[i,j]);
perdacar[i,j]:=carga(q[i,j],diam[i,j],comp[i,j],rugos,viscosidade);
derperdacar[i,j]:=dercarga(q[i,j],diam[i,j],comp[i,j],rugos,viscosidade);
sb[i]:=sb[i]+derperdacar[i,j]+tb;
st[i]:=st[i]+perdacar[i,j]+tt;
end;
dq[i]:=-st[i]/sb[i];
end;
for i:=1 to numloops do
  for j:=1 to numtubos[i] do
    begin
      d2:=0;
      if loopcomum[i,j]>0 then
        d2:=-dq[loopcomum[i,j]];
        q[i,j]:=q[i,j]+dq[i]+d2
      end;
      ic:=0;
      for i:=1 to numloops do
        if abs(dq[i])<0.0001 then
          ic:=1;
          it:=it+1;
          if it>10000 then
            begin
              Showmessage('Excesso de Iterações - Programa Abortado');
              ic:=1;
            end;
          end;
        end;
      end;
    end;
  end;
end;

procedure hardyresult;
begin
  for i:=1 to numloops do
    for j:=1 to numtubos[i] do
      begin
        resposta.labell.caption:=('Vazão do tubo '+inttostr(j)+' do loop '+inttostr(i));
        resposta.edit1.text:=format('%*.n',[3,4,q[i,j]]);
        resposta.showmodal;
      end;
    end;
  end;
end;

procedure inicializar;
begin
  q[1,1]:= 0.8*vazao/3;
  q[1,2]:= -0.2*vazao/3;
  q[1,3]:= -1.2*vazao/3;
  q[1,4]:= -1.2*vazao/3;
  q[2,1]:= vazao/3;

```

```
loopcomum[1,1]:= 2;  
loopcomum[2,4]:= 1;  
and;
```

```
procedure inicializa2;  
begin  
  q[1,1]:= 0.3*vazao;  
  q[1,2]:= -0.1*vazao;  
  q[1,3]:= -0.48*vazao;  
  q[1,4]:= -0.7*vazao;  
  q[2,1]:= 0.48*vazao;  
  q[2,2]:= 0.08*vazao;  
  q[2,3]:= -0.1*vazao;  
  q[2,4]:= -0.22*vazao;  
  q[3,1]:= 0.1*vazao;  
  q[3,2]:= -0.02*vazao;  
  q[3,3]:= -0.12*vazao;  
  q[3,4]:= -0.12*vazao;  
  loopcomum[1,3]:=2;  
  loopcomum[2,1]:=1;  
  loopcomum[2,3]:=3;  
  loopcomum[3,1]:=2;  
and;
```

```
procedure inicializa3;  
begin  
  q[1,1]:= 0.3*vazao;  
  q[1,2]:= 0.1*vazao;  
  q[1,3]:= -0.1*vazao;  
  q[1,4]:= -0.3*vazao;  
  q[1,5]:= -0.7*vazao;  
  q[2,1]:= 0.3*vazao;  
  q[2,2]:= 0.2*vazao;  
  q[2,3]:= 0.1*vazao;  
  q[2,4]:= -0.4*vazao;  
  q[3,1]:= 0.1*vazao;  
  q[3,2]:= 0.2*vazao;  
  q[3,3]:= -0.1*vazao;  
  q[3,4]:= -0.2*vazao;  
  q[4,1]:= -0.1*vazao;  
  q[4,2]:= 0.2*vazao;  
  q[4,3]:= -0.2*vazao;  
  q[4,4]:= -0.2*vazao;  
  q[5,1]:= 0.5*vazao;  
  q[5,2]:= -0.5*vazao;  
  q[5,3]:= -0.1*vazao;  
  q[5,4]:= 0.1*vazao;  
  q[5,5]:= 0.2*vazao;  
  loopcomum[1,2]:=4;  
  loopcomum[1,3]:=3;
```

```

loopcomum[2,3]:=5;
loopcomum[3,1]:=1;
loopcomum[3,2]:=4;
loopcomum[3,3]:=5;
loopcomum[3,4]:=2;
loopcomum[4,1]:=1;
loopcomum[4,3]:=5;
loopcomum[4,4]:=3;
loopcomum[5,3]:=2;
loopcomum[5,4]:=3;
loopcomum[5,5]:=4;
and;

```

```

procedure inicializa4;
begin
  q[1,1]:= 0.4*vazao;
  q[1,2]:= 0.1*vazao;
  q[1,3]:= -0.3*vazao;
  q[1,4]:= -0.6*vazao;
  q[2,1]:= 0.3*vazao;
  q[2,2]:= -0.3*vazao;
  q[2,3]:= -0.1*vazao;
  q[3,1]:= 0.3*vazao;
  q[3,2]:= 0.1*vazao;
  q[3,3]:= -0.2*vazao;
  q[4,1]:= 0.2*vazao;
  q[4,2]:= -0.1*vazao;
  q[4,3]:= -0.1*vazao;
  q[5,1]:= 0.3*vazao;
  q[5,2]:= 0.6*vazao;
  q[5,3]:= -0.4*vazao;
  q[5,4]:= -0.1*vazao;
  loopcomum[1,2]:=2;
  loopcomum[1,3]:=3;
  loopcomum[2,2]:=5;
  loopcomum[2,3]:=1;
  loopcomum[3,1]:=1;
  loopcomum[3,2]:=5;
  loopcomum[3,3]:=4;
  loopcomum[4,1]:=3;
  loopcomum[5,1]:=2;
  loopcomum[5,4]:=3;
and;

```

```

function alfa(temperatura:real):real;
begin
  alfa:=(0.000179*elev(temperatura,2)+0.029738*temperatura-2.509143)/1000000;
and;

function viscosidadear(temperatura:real):real;

```

```

function condutividadear(temperatura:real):real;
begin
    condutividadear:=(0.0783*temperatura+2.2338)/1000;
and;

function PrAr(temperatura:real):real;
begin
    PrAr:=0.00000068*elev((temperatura),2)-0.00065431*temperatura+0.84237148;
and;

function hamb(tempamb,tempup,diam:real):real;
var Ra, beta, temperatura, Nu: real;
begin
    temperatura:=(tempamb+tempup)/2;
    beta:=1/temperatura;
    Ra:=(g*beta*(tempup-tempamb)*elev(diam,3))/(viscosidadear(temperatura)*alfa(temperatura));
    Nu:=elev(0.60+(0.387*elev(Ra,1/6))/elev(1+elev(0.559/PrAr(temperatura),9/16),8/27),2);
    hamb:=condutividadear(temperatura)/diam*Nu;
and;

function Pr(visc,cond,dens,caloresp:real):real;
begin
    Pr:=visc*dens*caloresp/cond;
and;

function condutividadeagua(temperatura:real):real;
begin
    condutividadeagua:=(-0.006525*elev(temperatura,2)+5.362318*temperatura-408.101272)/1000;
and;

function Nusselt(vazao,diam,visc,temperatura,dens,caloresp,cond:real):real;
begin
    if Reynolds(vazao,diam,visc)<4000 then
        Nusselt:=4.36
    else
        Nusselt:=0.023*elev(Reynolds(vazao,diam,visc),4/5)*elev(Pr(visc,cond,dens,caloresp),1/3);
and;

function h(vazao,diam,visc,temperatura,cond,dens,caloresp:real):real;
begin
    h:=Nusselt(vazao,diam,visc,temperatura,dens,caloresp,cond)*cond/diam;
and;

function U(hint,hext,rint,rect,espint,espect,hamb:real):real;
var resint, resext, resamb:real;
begin
    resint:=(rect+espect)/(hint*rint);
    resext:=(rect+espect)/(hext*((rint+espint)+rect));

```

```

function efetiv(vazaoint,vazaoext,cpint,cpext,U,A:real):real;
var NUT, cmin, cmax:real;
begin
  cmin:=min(vazaoint*cpint,vazaoext*cpext);
  cmax:=max(vazaoint*cpint,vazaoext*cpext);
  NUT:=U*A*cmin;
  if not cc then
    efetiv:=(1-exp(-NUT*(1+cmin/cmax)))/(1+cmin/cmax)
  else
    efetiv:=(1-exp(-NUT*(1+cmin/cmax)))/(1-cmin/cmax*exp(-NUT*(1-cmin/cmax)));
  and;

function cargaterm(efetiv,vazaoint,vazaoext,cpint,cpext,tentint,tentext:real):real;
begin
  cargaterm:=efetiv*min(vazaoint*cpint,vazaoext*cpext)*abs((tentint-tentext));
  and;

function tempsaidaquente(temp,cargaterm,vazao,cp:real):real;
begin
  tempsaidaquente:=temp - cargaterm / (vazao*cp) - 273.15
  and;

function tempsaidafrio(temp,cargaterm,vazao,cp:real):real;
begin
  tempsaidafrio:=temp + cargaterm / (vazao*cp) - 273.15
  and;
  and.

```