

**Centro Universitário de Brasília – UniCEUB**  
**Faculdade de Ciências Exatas e Tecnologia – FAET**  
**Engenharia de Computação**

**SIMULADOR COMPUTACIONAL**  
**PARA FLUIDOS**  
**INCOMPRESSÍVEIS**

*Brasília – DF, julho de 2006*

John Franklin L Madeira

**SIMULADOR COMPUTACIONAL  
PARA FLUIDOS  
INCOMPRESSÍVEIS**

Monografia apresentada à  
Faculdade de Ciências Exatas e  
Tecnologia do Centro Universitário  
de Brasília (UniCEUB) como  
requisito parcial à obtenção do grau  
de Engenheiro da Computação.

Professor Orientador: Carmo Gonçalves

*Brasília – DF, julho de 2006*

À minha esposa Elizabeth e minha filha Júlia.

## **AGRADECIMENTOS**

Agradeço ao Professor Carmo Gonçalves pelas orientações, pelas dicas sempre pertinentes e por sua paciência.

Ao meu amigo e colega Amando dos Santos Neto e sua esposa Mônica pela revisão desta monografia.

E todos que direta ou indiretamente contribuíram para a realização deste trabalho.

“A imaginação é mais importante que o conhecimento”

Albert Einstein

## **RESUMO**

Nosso interesse principal na apresentação desta monografia é mostrar os aspectos práticos que envolvem a análise do escoamento de fluidos incompressíveis em condutos forçados, uniformes e de seção circular, em regime permanente apresentado em um sistema computacional que simulará cálculos das variações de diâmetro, vazão e perda de carga.

Essa reunião de condições representa a maioria das situações com as quais uma grande parte dos projetistas de hidráulica se defronta no seu dia-a-dia. Por essa razão é que esse tema está aqui apresentado.

Contudo, não tem o presente trabalho a intenção de esgotar o assunto, e nem de apresentar, questionar ou demonstrar teorias hidráulicas, pois isto já é amplamente tratado na literatura corrente sobre o assunto, mas sim mostrar tópicos práticos relevantes sobre o dimensionamento hidráulico de condutos forçados e das perdas de carga contínua.

## **ABSTRACT**

Our main interest in the presentation of this monograph is to show the practical aspects that they involve the analysis of the non compressible fluid draining in forced conduits uniforms and of circle section, in presented permanent regimen in a computational system that will simulate calculations of variations of diameter, outflow and loss of load.

This meeting of conditions represents the majority of the situations with which a great part of the hydraulical designers if confrots day-by-day in its. For this reason it is that this subject here is presented.

However, it does not have the present work the intention to deplete the subject, and nor to present, to question or to demonstrate hydraulical theories, therefore this already widely is treated in current literature on the subject, but yes to show excellent practical topics on the hydraulical sizing of forced conduits and head loss.

## LISTA DE FIGURAS

Fig. 1 – Exemplo de conduto forçado utilizado numa hidroelétrica .....	4
Fig. 2 – Tipos de escoamento.....	6
Fig. 3 – Viscosidade .....	8
Fig. 4 – Fluxograma para cálculos de equações implícitas usando o método iterativo.....	15
Fig. 5 – Diagrama de caso de uso.....	36
Fig. 6 – Diagrama de Atividades.....	37
Fig. 7 – Tela de entrada usando o browser Microsoft Internet Explorer 6.0 .....	62
Fig. 8 – Módulo de cálculo do diâmetro .....	63
Fig. 9 – Simulação do cálculo do diâmetro - usando concreto rugoso.....	64
Fig. 10 – Simulação do cálculo do diâmetro - usando glicerina .....	65
Fig. 11 – Tela do simulador para o cálculo da vazão. ....	66
Fig. 12 – Tela do simulador para o cálculo da perda de carga unitária .....	66
Fig. 13 – Exemplo de cálculo da vazão para diferentes simulações de condutos forçados .....	67
Fig. 14 – Simulação de cálculo de perda de carga unitária para diferentes valores de viscosidade do fluido. ....	68
Fig. 15 – Tela do sistema para cálculo da perda de carga unitária.....	72
Fig. 16 – Tela do sistema para cálculo do diâmetro. ....	75
Fig. 17 – Tela do sistema para cálculo da vazão .....	77

## SUMÁRIO

INTRODUÇÃO.....	3
1. CONCEITOS INICIAIS .....	4
1.1 – Classificação do Escoamento .....	5
1.2 – Viscosidade.....	7
1.3 – Rugosidade .....	10
2. EQUAÇÕES FUNDAMENTAIS .....	13
2.1 – Darcy-Weisbach.....	13
2.2 – Colebrook-White.....	13
3. A HISTÓRIA DA EQUAÇÃO DE DARCY-WEISBACH .....	16
4. EQUAÇÕES EXPLÍCITAS PARA O FATOR DE ATRITO DE DARCY- WEISBACH .....	19
5. FÓRMULA DE COLEBROOK: ANTIGA, MAS ATUAL. SOLUÇÕES EXPLÍCITAS .....	22
6. O MÉTODO ITERATIVO.....	32
7. CONSTRUÇÃO DO SIMULADOR.....	35
7.1 – Unified Modeling Language (UML).....	35
7.2 – Programando com Java .....	38
7.2.1 – Arquivos de Entrada .....	41
7.2.2 – Código de Cálculo.....	46
7.2.3 – Códigos Recicláveis .....	49
8. RESULTADOS OBTIDOS.....	62
9. CONFIRMAÇÃO DOS RESULTADOS ESPERADOS .....	69
9.1 – Perda de Carga Desconhecida.....	69
9.2 – Diâmetro Desconhecido .....	73
9.3 – Vazão Desconhecida .....	76
CONCLUSÃO.....	78
BIBLIOGRAFIA .....	80
APÊNDICE A – DIAGRAMA DE MOODY .....	83
APÊNDICE B – DEMONSTRAÇÃO DAS PRINCIPAIS EQUAÇÕES UTILIZADAS NOS DIMENSIONAMENTOS E SIMULAÇÕES NUMÉRICAS.....	84

# INTRODUÇÃO

O objetivo deste trabalho é o desenvolvimento de um simulador computacional que adote os conceitos de mecânica dos fluidos para escoamento em regime turbulento interno de fluidos não-compressíveis efetuando cálculos de diâmetro, vazão e perda de carga unitária fazendo uso dos métodos iterativos de cálculo numérico.

Os métodos numéricos, embora aproximativos, são capazes de resolver equações implícitas com a precisão que se desejar. São métodos basicamente computacionais, pois incorrem em operações matemáticas repetidas. Encontram, contudo, muita utilidade em hidráulica.

O uso da linguagem de programação Java, implementa uma série de componentes chamados APIs que nos auxiliam nos cálculo desses métodos. É o caso dos métodos iterativos nos quais se ordena adequadamente a equação, e arbitra-se um valor inicial qualquer para a variável procurada que está no seu segundo membro. Com o valor inicial já arbitrado, calcula-se um novo valor para esta mesma variável procurada, mas para a que está no primeiro membro. Se a diferença entre o valor inicial e o novo valor calculado estiver fora da precisão desejada, repete-se esta operação, porém colocando como valor inicial o novo valor calculado. Se a diferença aumentar diz-se que os valores estão divergindo, e se diminuir diz-se que os valores estão convergindo para a solução.

O número de repetições, isto é, o número de iterações poderá ser pequeno ou não, dependendo do método a ser utilizado, e se sucederá até que a diferença seja suficientemente pequena ou compatível com a precisão desejada. É uma técnica manual, cuja solução se torna trabalhosa e enfadonha. Com o uso de um ambiente computacional baseado numa arquitetura web, tal como o que será apresentado, a resolução torna-se simples, automática e rápida.

# 1. CONCEITOS INICIAIS

Entende-se por conduto forçado aquele no qual o fluido escoar à plena seção e sob pressão. Os condutos de seção circular são chamados de tubos ou condutos (fig. 1). Um conduto é dito uniforme quando a sua seção transversal não varia com o seu comprimento. Se a velocidade do fluido em qualquer seção do conduto não variar com o tempo, o regime de escoamento é dito permanente. Devido à reduzidíssima variação da densidade, nos escoamentos de líquidos em regime permanente considera-se que os mesmos se comportam como incompressíveis. Neste contexto inclui-se: querosene, gasolina, álcool, óleo diesel, água, vinho, vinhoto, leite, etc [6].

Para uso nas simulações, serão utilizados os dados relacionados nas tabelas 1 e 2 que mais adiante relacionam diferentes líquidos (e suas respectivas temperaturas) com os principais tipos de materiais utilizados na fabricação de condutos forçados.



Fig. 1 – Exemplo de conduto forçado utilizado numa hidroelétrica

Como este trabalho tratará de um tema da engenharia relacionado especificamente à hidráulica de tubulações, e alguns de seus aspectos dimensionais, pode-se iniciá-lo apresentando a equação da continuidade em sua forma mais comum:

$$Q = AV \quad (1)$$

Onde:

Q = vazão no tubo ( $\text{m}^3/\text{s}$ )

$A = \pi.D^2/4$  = área da seção transversal do tubo ( $\text{m}^2$ )

D = diâmetro interno do tubo (m)

V = velocidade do líquido no interior do tubo (m/s)

## 1.1 – Classificação do Escoamento

É conveniente ressaltar que um escoamento se classifica também como turbulento ou laminar. No escoamento laminar há um caminhamento disciplinado das partículas fluidas, seguindo trajetórias regulares, sendo que as trajetórias de duas partículas vizinhas não se cruzam. Já no escoamento turbulento a velocidade num dado ponto varia constantemente em grandeza e direção, com trajetórias irregulares, e podendo uma mesma partícula ora localizar-se próxima ao eixo do tubo, ora próxima à parede do tubo.

A figura 2 abaixo demonstra os dois tipos de escoamento:

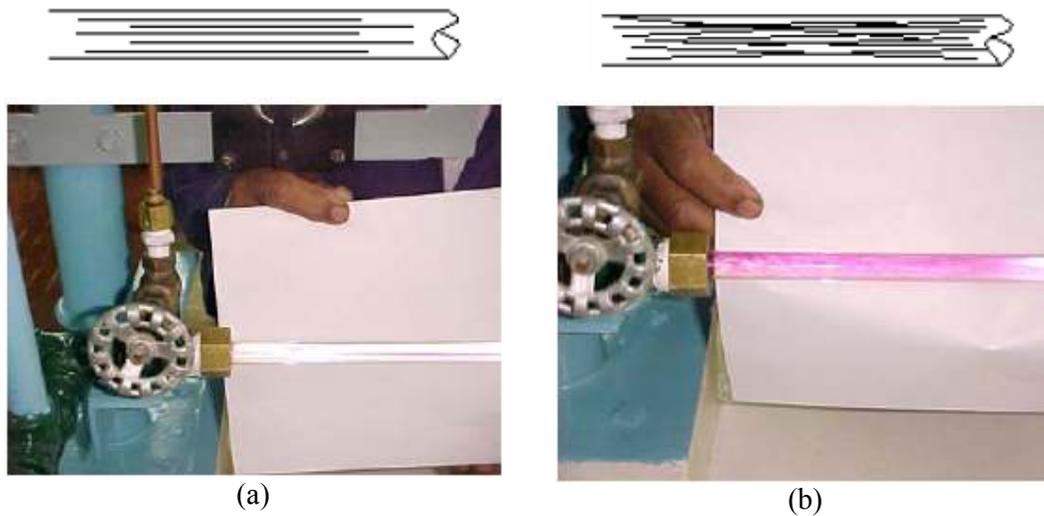


Fig. 2 – Tipos de escoamento: laminar (a) e turbulento (b). As fotos mostram a visualização do escoamento laminar (filete contínuo de permanganato de potássio - filete em tom de rosa) e depois em escoamento turbulento em laboratório de mecânica dos fluidos

O critério para determinar se o escoamento é turbulento ou laminar é especificado pelo número adimensional de Reynolds, onde o regime é considerado escoamento laminar para Reynolds menor que 2000 ( $R_e < 2000$ ) e turbulento para Reynolds maior que 4000 ( $R_e > 4000$ ). Para Reynolds entre 2000 e 4000 ( $2000 < R_e < 4000$ ) o regime é considerado transitório. Este conceito também é discutido no Apêndice A com maiores detalhes. A equação de Reynolds é mostrada abaixo:

$$R_e = \frac{VD}{\nu} \quad (2)$$

Substituindo (1) em (2), a equação de Reynolds assume a conveniente forma:

$$R_e = \frac{4Q}{\pi D \nu} \quad (3)$$

Onde:

$R_e$  = número de Reynolds (adimensional)

$V$  = velocidade do líquido no interior do tubo (m/s)

$D$  = diâmetro interno do tubo (m)

$Q$  = vazão no tubo (m<sup>3</sup>/s)

$\nu$  = viscosidade cinemática do líquido (m<sup>2</sup>/s)

## 1.2 – Viscosidade

A viscosidade é uma força volumétrica de atrito interno que aparece no deslizamento de camadas fluindo umas sobre as outras, dando origem a tensões tangenciais de cisalhamento. Com isso, atrito interno impede ou oferece uma resistência ao escorregamento das partículas, umas sobre as outras. Todos nós temos uma idéia intuitiva de viscosidade, sabendo reconhecer um líquido mais ou menos viscoso; esta noção intuitiva é dada, por exemplo, pela maior ou menor facilidade com que um líquido escorre sobre uma superfície inclinada.

A viscosidade dos líquidos, como mencionado antes, vem do atrito interno, isto é, das forças de coesão entre moléculas relativamente juntas nos condutos forçados. Desta maneira, enquanto que a viscosidade dos gases cresce com o aumento da temperatura, nos líquidos ocorre o oposto. Fox [14] menciona que com o aumento da temperatura, aumenta a energia cinética média das moléculas, diminuindo por sua vez o intervalo de tempo que as moléculas passam umas junto das outras, menos efetivas se tornam às forças intermoleculares diminuindo a viscosidade e, conseqüentemente, o atrito do fluido com as paredes do conduto. Assim, a viscosidade de um líquido é inversamente proporcional à sua temperatura: quanto maior a temperatura do fluido, em estado líquido, menor será sua viscosidade.

A Lei de Newton da viscosidade relaciona a tensão tangencial  $F/A$  (Força por unidade de área) à taxa de variação espacial de velocidade  $dv/dx$ :  $F/A = \nu (dv/dx)$

onde  $\nu$  é o coeficiente de viscosidade do fluido, o qual depende de sua natureza. O atrito é tanto maior quanto maior a viscosidade do fluido. A unidade de  $\nu$  no sistema MKS é o Pa.s (Pascal  $\times$  segundo), sendo que a unidade mais empregada na prática é o centipoise (cp), dado por:  $1 \text{ cp} = 10^{-2} \text{ poise (CGS)} = \text{mPa.s} (10^{-3} \text{ Pa.s}) = 10^{-3} \text{ m}^2/\text{s} \text{ (MKS)}$

Apesar de tratarmos aqui sobre viscosidade apenas em dutos forçados, podemos entender melhor a natureza da viscosidade nos líquidos dando um exemplo em placas paralelas planas. Malafaya [20]: suponhamos duas placas sólidas planas, uma sobre a outra, com um fluido contínuo entre elas. Aplicando uma força constante a uma das placas, a experiência mostra que ela é acelerada até atingir uma velocidade constante (chamada velocidade terminal). Se a intensidade da força aplicada for duplicada, por exemplo, a velocidade terminal também duplica. A velocidade terminal, desta forma, é proporcional à força aplicada. Pensando que o líquido entre as placas se separa em lâminas paralelas, o efeito da força aplicada é o de produzir diferenças de velocidade entre lâminas adjacentes. A lâmina adjacente à placa móvel se move junto com ela e a lâmina adjacente à placa imóvel permanece também imóvel (fig. 3). O atrito entre lâminas adjacentes causa dissipação de energia mecânica e dá origem ao deslizamento entre as camadas do fluido que dá origem às tensões de cisalhamento.

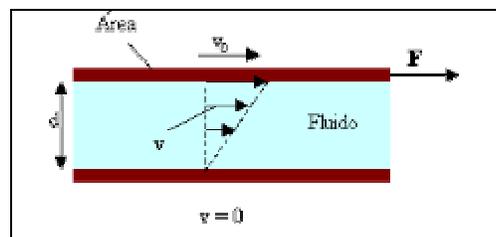


Fig. 3 – Viscosidade

A tabela abaixo, traz a lista da viscosidade dos principais líquidos usados nos cálculos deste trabalho [6]:

Viscosidade cinemática de alguns líquidos		
LÍQUIDO	TEMP. (°C)	VISC. CINEMÁTICA (x10 <sup>-6</sup> m <sup>2</sup> /s)
Água	10	1,31
Água	20	1,00
Água	40	0,66
Água do mar	5	1,61
Água do mar	15	1,22
Água do mar	25	0,97
Álcool Metílico	20	0,727
Asfalto	120	1600
Azeite	38	43
Benzol	20	0,744
Gasolina	20	0,6
Glicerina	20	1180
Glicerina	40	223
Leite	20	1,13
Óleo de algodão	38	38
Óleo de baleia	38	38
Óleo de linhaça	38	30
Óleo de soja	38	35
Óleo SAE-30	30	130
Óleo SAE-30	40	80
Óleo SAE-90	40	250

Tabela 1 – Viscosidade ( $\nu$ ) de alguns líquidos

Como discutido anteriormente, o regime de escoamento num tubo (laminar ou turbulento) é determinado pelo número de Reynolds (equação 3). Pode-se demonstrar, pela clássica experiência de Reynolds<sup>1</sup>, a diferença qualitativa entre escoamento laminar e turbulento. Nessa experiência, a água escoava de um grande reservatório através de um tubo transparente. Um fino filamento de corante injetado na entrada do tubo permite a observação visual do escoamento. As baixas vazões (baixos números de Reynolds), o corante permanece num único filamento; há pouca dispersão de corante porque o escoamento é laminar [14].

À medida que a vazão através do tubo é aumentada, o filamento de corante torna-se instável e parte-se num movimento aleatório; a linha de corante é esticada e torcida numa infinidade de fios enovelados, e rapidamente se dispersa por todo o campo

<sup>1</sup> Essa experiência é demonstrada no filme da NCFMF, *Turbulence*.

de escoamento. Esse comportamento de escoamento turbulento é devido a pequenas flutuações de velocidade, de alta frequência, superpostas ao movimento principal, como previamente ilustrado na figura 2; a mistura de partículas de camadas adjacentes de fluido resulta na rápida dispersão do corante.

Com o cuidado necessário para manter o escoamento isento de perturbações e com superfícies lisas, experiências realizadas até hoje têm sido capazes de manter o escoamento laminar dentro de um tubo até números de Reynolds de cerca de 100.000! Contudo, a maioria das situações de escoamento na engenharia não é controlada com tanto cuidado. Sob condições normais, a transição ocorre em  $Re \approx 2300$  para escoamentos em tubos. A relação entre o número de Reynolds, a rugosidade relativa e o fator de atrito para dutos circulares, é fornecida pelo Diagrama de Moody [21] no Apêndice A.

As equações que aqui são utilizadas se aplicam ao chamado escoamento turbulento. Em geral, o regime de escoamento na condução de líquidos no interior de tubulações é turbulento, exceto em situações especiais, tais como escoamento a baixíssimas vazões, como ocorre em gotejadores de irrigação, onde o escoamento é laminar.

### 1.3 – Rugosidade

Sempre que um líquido escoar no interior de um tubo de um ponto para outro, haverá perda de energia, denominada perda de pressão ou perda de carga. Essa perda de energia é devido ao atrito com as paredes do tubo e devido à viscosidade do líquido em escoamento. Quanto maior for a rugosidade da parede da tubulação, isto é, a altura das asperezas, maior será a turbulência do escoamento e, logo, maior será a perda de carga. Fatores como o tipo de material utilizado no transporte do fluido e sua respectiva rugosidade da parede, a velocidade do escoamento do fluido, o processo de

fabricação do duto, a temperatura e a viscosidade do fluido, influenciam bastante a perda de carga.

Já há cerca de dois séculos estudos e pesquisas vêm sendo realizados, procurando estabelecer leis que possam reger as perdas de carga em condutos. Várias fórmulas empíricas foram estabelecidas no passado e algumas empregadas até com alguma confiança em diversas aplicações de engenharia, como as fórmulas de Hazen-Williams, de Manning e de Flamant. Mas, trabalhos de diversos investigadores têm mostrado que, em sua totalidade, são mais ou menos incorretas devido à diferença na precisão dos resultados. A incorreção dessas fórmulas é tanto maior quanto mais amplo é o domínio de aplicação pretendido por seus autores. Podemos acompanhar a evolução das correções e precisões das fórmulas aplicadas em hidráulica de condutos forçados com mais detalhes adiante na seção 5.

A tabela a seguir, lista a rugosidade dos principais materiais que iremos utilizar nos cálculos [6]:

<b>Rugosidade equivalente de alguns materiais de tubos</b>	
<b>MATERIAL</b>	<b>RUG. EQUIVALENTE (m)</b>
Aço comercial	0,00006
Aço galvanizado	0,00016
Aço com ferrugem leve	0,00025
Aço revestido com asfalto	0,00060
Aço revestido com esmalte, vinil, epóxi	0,00006
Alumínio	0,000004
Concreto muito rugoso	0,00200
Concreto rugoso	0,00050
Concreto liso	0,00010
Concreto muito liso	0,00006
Concreto alisado, centrifugado	0,00030
Ferro fundido asfaltado	0,000122
Ferro fundido não revestido novo	0,00050
Ferro fundido com ferrugem leve	0,00150
Ferro fundido com cimento centrifugado	0,00010
Fibrocimento	0,00010
Manilha cerâmica	0,00030
Latão, cobre	0,000007
Plásticos	0,00006

Tabela 2 – Rugosidade ( $k$ ) de alguns tipos de tubos

## 2. EQUAÇÕES FUNDAMENTAIS

### 2.1 – Darcy-Weisbach

Atualmente a expressão mais precisa é usada universalmente para análise de escoamento em tubos, que foi proposta em 1845, é a conhecida equação de Darcy-Weisbach [8][20]:

$$h_l = f \frac{LV^2}{2Dg} \quad (4)$$

que, após substituir valores e realizar operações, também assume uma conveniente forma:

$$\frac{1}{\sqrt{f}} = \sqrt{\frac{8Q^2}{J\pi^2 gD^5}} \quad (5)$$

Onde:

$h_f$  = perda de carga contínua ao longo do comprimento do tubo (mca)

$f$  = fator de atrito (adimensional)

$L$  = comprimento reto do tubo (m)

$V$  = velocidade do líquido no interior do tubo (m/s)

$D$  = diâmetro interno do tubo (m)

$g$  = aceleração da gravidade local (m/s<sup>2</sup>)

$J = h_f/L$  = perda de carga unitária (mca/m)

No Apêndice B, é demonstrado como a fórmula da Darcy-Weisback foi adaptada para uso nos cálculos neste trabalho.

### 2.2 – Colebrook-White

Porém somente em 1939, quase 100 anos depois, é que se estabeleceu definitivamente o fator de atrito  $f$ , através da equação de Colebrook-White [12][27]:

$$\frac{1}{\sqrt{f}} = -2 \log_{10} \left( 0,27 \frac{k}{D} + \frac{2,51}{R_e \sqrt{f}} \right) \quad (6)$$

Onde:

$f$  = fator de atrito (adimensional)

$k$  = rugosidade equivalente da parede do tubo (m)

$D$  = diâmetro interno do tubo (m)

$R_e$  = número de Reynolds (adimensional)

Obviamente, trata-se de uma equação implícita, isto é, a variável  $f$  aparece nos dois membros da equação, de forma não ser possível explicitá-la. Mas isto não sugere que seja impossível resolver equações implícitas. Num esquema básico de cálculo, passo-a-passo, seria algo do tipo:

- 1°. Arbitra-se um valor inicial qualquer para a variável do segundo membro;
- 2°. Calcula-se o novo valor para a mesma variável que está no primeiro membro;
- 3°. Compara-se a diferença entre o valor calculado e o valor inicial com a tolerância estabelecida;
- 4°. Se maior, o novo valor passa a ser o valor inicial, e volta-se para o passo 2. Se menor passa-se para o passo 5;
- 5°. O corrente valor da variável é o valor procurado.

Abaixo está descrito o fluxograma representativo para resolução de equações implícitas:

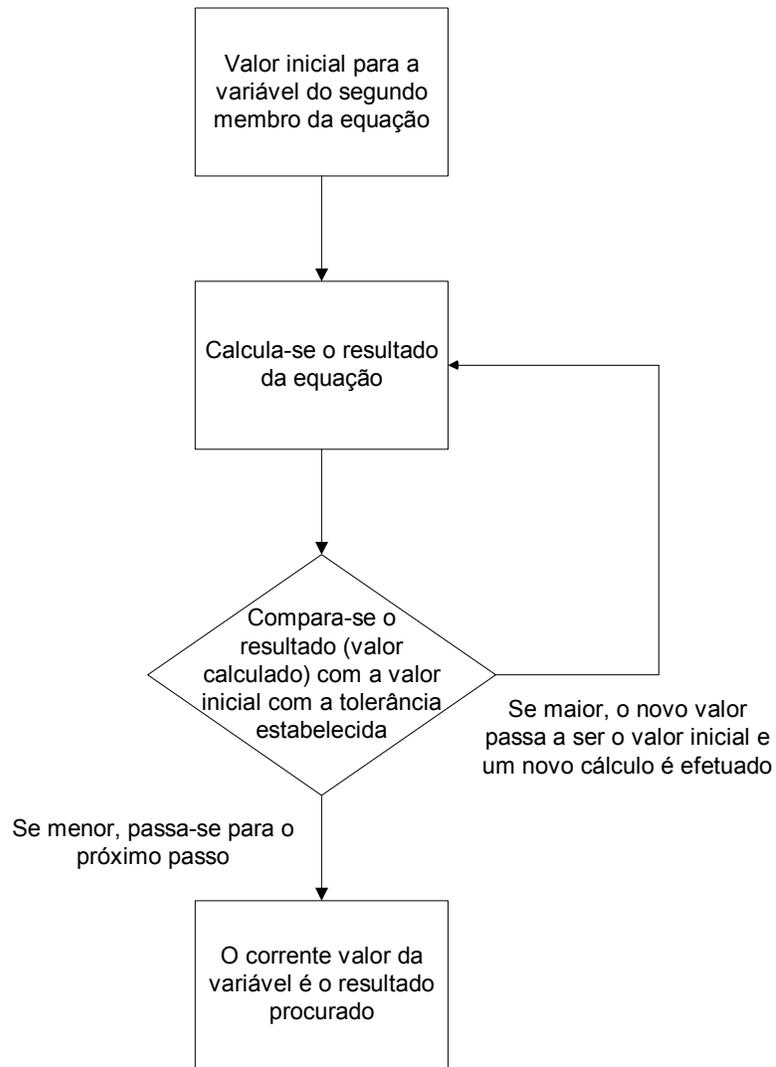


Fig. 4 – Fluxograma para cálculos de equações implícitas usando o método iterativo

### 3. A HISTÓRIA DA EQUAÇÃO DE DARCY-WEISBACH<sup>2</sup>

A equação de Darcy-Weisbach tem uma longa história de desenvolvimento. Ela é conhecida pelo nome de dois grandes engenheiros hidráulicos de meados do século XIX, embora outros nomes tenham dado também importantes contribuições. Julies Weisbach (1806-1871), natural da Saxônia, propôs em 1845 a equação 4 que é utilizada atualmente. Entretanto, Weisbach não proveu dados adequados para a variação de  $f$  com a rugosidade relativa e com a velocidade. Por isso, sua equação teve fraco desempenho comparado com a equação empírica de Prony (Gaspard Clair Francois Marie Riche de Prony, 1755-1839) amplamente usada na ocasião;

$$h_l = \frac{L}{D}(aV + bV^2) \quad (7)$$

Onde  $a$  e  $b$  são fatores de atrito empíricos para a velocidade e para o quadrado da velocidade.

Apesar de Weisbach ter estado à frente da maioria dos outros engenheiros, seu trabalho não foi o primeiro nesta área. Aproximadamente em 1770, Antoine Chézy (1718-1798), um diplomado precoce da l'Ecole des Ponts et Chaussées, publicou uma equação para escoamento em canais abertos que podia ser reduzida à mesma fórmula. Infelizmente, o trabalho de Chézy ficou perdido até 1800 quando seu antigo aluno, Prony, publicou um relato descrevendo-o. Surpreendentemente Prony desenvolveu sua própria equação, mas é sabido que Weisbach estava ciente dos trabalhos de Chézy na publicação de Prony.

---

<sup>2</sup> Glenn Brown, Oklahoma State University. Abril 2001.

Darcy, (aluno de Prony) em 1857 publicou novas relações para os coeficientes de Prony baseado em um grande número de experimentos. Sua nova equação foi:

$$h_l = \frac{L}{D} \left[ (c + d/D^2)V + (d + e/D)V^2 \right] \quad (8)$$

Onde  $c$ ,  $d$  e  $e$  são coeficientes empíricos para um dado tipo de tubo. Darcy desta forma introduziu o conceito de coeficiente de atrito escalonado por diâmetro; o que é conhecido como rugosidade relativa, quando aplicando o Diagrama do Moody (Apêndice A). Portanto, é tradicional chamar  $f$  de "fator  $f$  de Darcy", ainda que Darcy nunca tenha proposto isto naquela fórmula.

Os dois conceitos foram juntados por Fanning por volta de 1880. Ele publicou uma compilação dos valores de  $f$  como uma função do tipo de tubo e da velocidade. Contudo, seria notado que Fanning utilizou o raio hidráulico, ao invés de  $D$  na equação do atrito, e assim os valores do "f de Fanning" são apenas 1/4 dos valores do "f de Darcy". A equação de Darcy-Weisbach não foi universalmente proveitosa até o desenvolvimento do diagrama de Moody [20] que o construiu com base nos trabalhos de Poiseuille, Reynolds, Blasius, Káarmaán, Prandtl, Colebrook, White, Rouse e Nikuradse. Rouse (1946) dá um bom sentido ao desenvolvimento do fator  $f$ , mas ele não faz referência a Moody.

O nome da equação através do tempo é também curioso e pode ser localizado em livros-textos de hidráulica e mecânica dos fluidos. Textos mais antigos geralmente não davam nome à equação. Começando em meados do século 20 alguns autores, incluindo pelo menos um alemão, chamaram-na de "Equação de Darcy", um óbvio ponto de confusão com a "Lei de Darcy". Rouse, em 1946, parece ser o primeiro a chamá-la de "Darcy-Weisbach", porém este nome não se torna universal até perto de

1980. O nome é suficientemente bom, mas como mostrado anteriormente, ele deixa de lado importantes contribuições. Então se realmente quiséssemos dar crédito completo, deveríamos dar o nome confuso de "equação de Chézy - Weisbach - Darcy - Poiseuille - Reynolds - Fanning - Blasius - Kármaán - Prandtl - Colebrook - White - Rouse - Nikuradse - Moody".

De um ponto de vista prático, a equação de Darcy-Weisbach somente tornou-se popular a partir do advento das calculadoras eletrônicas. Ela requer uma grande quantidade de operações quando comparada a relações empíricas, tal como a equação de Hazen-Williams, que são válidas para estreitas faixas de aplicação.

Portanto, por causa de sua precisão geral e ampla faixa de aplicação, a equação de Darcy-Weisbach deveria ser considerada padrão e as outras deveriam ser deixadas para os historiadores.

## 4. EQUAÇÕES EXPLÍCITAS PARA O FATOR DE ATRITO DE DARCY-WEISBACH<sup>3</sup>

Como mencionado anteriormente, com o intuito de estabelecer leis que possam reger as perdas de carga em condutos, já há cerca de dois séculos estudos e pesquisas sendo realizadas. Atualmente a equação 4, é a expressão mais precisa e utilizada universalmente para análise de escoamento em tubos, e que foi proposta em 1845, é a conhecida equação de Darcy-Weisbach.

Mas não se encontrou logo uma maneira segura para determinação do fator de atrito. Somente em 1939, quase 100 anos depois, é que se estabeleceu definitivamente uma lei para fator de atrito  $f$ , através da equação 6 de Colebrook-White [28]:

A equação de Colebrook-White tem sido considerada como a mais precisa lei de resistência ao escoamento e vem sendo utilizada como padrão referencial. Mas, apesar disto, e de todo o fundamentalismo e embasamento teórico agregado à mesma, tem uma particularidade há alguns poucos convenientes: é implícita em relação ao fator de atrito, ou seja, a grandeza  $f$  está presente nos dois membros da equação, sem possibilidade de ser explicitada em relação às demais grandezas. Sua resolução requer um processo iterativo.

Isso resultou em motivos para que muitos pesquisadores, de quase toda parte do mundo, se empenhassem em encontrar equações explícitas, que pudessem ser utilizadas como alternativas à equação de Colebrook-White. Algumas mais compactas e simples, mais fáceis de serem memorizadas, contudo com grandes desvios; outras, menos compactas e complexas, mais difíceis de serem memorizadas, porém com desvios

---

<sup>3</sup> Eng. Prof. Luiz A. Camargo, Junho de 2004.

menores; outras tantas combinando simplicidade e precisão, com erros até bem reduzidos, em relação ao fator de atrito calculado com a equação de Colebrook-White.

Apresenta-se a seguir um pequeno conjunto destas equações explícitas, considerando apenas aquelas que pesquisadores, conforme bibliografia consultada, avaliaram e concluíram terem os menores erros em relação à equação de Colebrook-White:

Sousa-Cunha-Marques (1999):

(Erro: 0,123%)

$$\frac{1}{\sqrt{f}} = -2 \log_{10} \left[ \frac{k}{3.7D} - \frac{5.16}{R_e} \log_{10} \left( \frac{k}{3.7D} + \frac{5.09}{R_e^{0.887}} \right) \right] \quad (9)$$

Haaland (1983):

(Erro: 0,220%)

$$\frac{1}{\sqrt{f}} = -1.8 \log_{10} \left[ \left( \frac{k}{3.7D} \right)^{1.11} + \frac{6.9}{R_e} \right] \quad (10)$$

Barr (1972):

(Erro: 0,375%)

$$\frac{1}{\sqrt{f}} = -2 \log_{10} \left( \frac{k}{3.7D} + \frac{5.15}{R_e^{0.892}} \right) \quad (11)$$

Swamee-Jain (1976):

(Erro: 0,386%)

$$\frac{1}{\sqrt{f}} = -2 \log_{10} \left( \frac{k}{3.7D} + \frac{5.74}{R_e^{0.89}} \right) \quad (12)$$

Churchill (1973):

(Erro: 0,393%)

$$\frac{1}{\sqrt{f}} = -2 \log_{10} \left[ \frac{k}{3.7D} + \left( \frac{7}{R_e} \right)^{0.9} \right] \quad (13)$$

Não se pode dizer que seja uma banalização, mas usando o mesmo critério de alguns autores, daria aqui até para propor, sem maiores dificuldades, outras equações explícitas:

Camargo-Barr-Colebrook-White (2001):

(substituição da eq. de Barr na eq. de Colebrook)

$$\frac{1}{\sqrt{f}} = -2 \log_{10} \left[ \frac{k}{3.7D} - \frac{5.02}{R_e} \log_{10} \left( \frac{k}{3.7D} + \frac{5.15}{R_e^{0.892}} \right) \right] \quad (14)$$

Camargo-Swamee-Jain-Colebrook-White (2001):

(substituição da eq. de Swamee-Jain na eq. de Colebrook)

$$\frac{1}{\sqrt{f}} = -2 \log_{10} \left[ \frac{k}{3.7D} - \frac{5.02}{R_e} \log_{10} \left( \frac{k}{3.7D} + \frac{5.74}{R_e^{0.9}} \right) \right] \quad (15)$$

A conclusão é que, por mais simples ou compactas que possam ser estas equações explícitas, as mesmas requerem algum esforço computacional com operações matemáticas de potenciação, radiciação, logarítmicas, etc.

Contudo, tendo em vista as elevadas velocidades dos processadores dos computadores atuais, praticamente será imperceptível a diferença no esforço computacional do cálculo feito com uma equação implícita ou mesmo com uma equação explícita.

## 5. FÓRMULA DE COLEBROOK: ANTIGA, MAS ATUAL. SOLUÇÕES EXPLÍCITAS<sup>4</sup>

Quando do processo de desenvolvimento de ferramentas da área de tecnologia da informação (TI) para obtenção do equilíbrio hidráulico em sistemas de distribuição de água, os autores deste trabalho sempre se debateram com uma questão de base: "Qual a lei de resistência ao escoamento a incorporar nos modelos matemáticos a serem desenvolvidos?".

No entanto, essa equação apresenta uma característica que a torna pouco atrativa. Devido à sua forma, não é possível expressar explicitamente o fator de resistência em função das restantes grandezas envolvidas no cálculo do mesmo.

Para ultrapassar esse inconveniente, diversos investigadores desenvolveram esforços no sentido de obterem expressões que, sendo explícitas, reproduzissem com algum rigor os valores do fator de resistência estimados pela fórmula de Colebrook-White.

Em 1932 e 1933, Nikuradse desenvolveu várias experiências com o objetivo de estudar os regimes de escoamento em condutos. Estas experiências foram orientadas por Prandtl e von Karman e consistiram na determinação de perdas de carga contínuas provocadas por escoamentos no interior de condutos circulares com rugosidade artificial. Nas experiências realizadas os condutos eram materializados por tubos de vidro e a rugosidade artificial era conseguida colando-se grãos de areia nas paredes desses tubos. Por sua vez, os grãos de areia eram selecionados de forma a obter rugosidades uniformes.

---

<sup>4</sup> José Alfeu A. de Sá Marques - Faculdade de Ciências e Tecnologia da Universidade de Coimbra; Joaquim J. O. Sousa - Instituto Superior de Engenharia do Instituto Politécnico de Coimbra

Ao dispor os resultados obtidos nas suas experiências num diagrama em que figurava o logaritmo do fator de resistência em função do logaritmo do número de Reynolds, Nikuradse observou que existiam três regimes bem distintos:

- O regime laminar, em que o fator de resistência ( $\lambda$ ) depende apenas do número de Reynolds ( $Re$ ), existindo uma relação linear entre ambas as grandezas expressa pela fórmula de Hagen-Poiseuille (1856):

$$\log \lambda = -\log R_e + \log 64 \quad \text{ou} \quad \lambda = \frac{64}{R_e} \quad (16)$$

- Regime turbulento liso, no qual o fator de resistência continua a depender unicamente do número de Reynolds. Nikuradse (1932) confirmou que a relação entre ambos era corretamente definida pela fórmula de Prandtl-von Karman

$$\frac{1}{\sqrt{\lambda}} = 2 \log(R_e \sqrt{\lambda}) - 0,8 \quad \text{ou} \quad \frac{1}{\sqrt{\lambda}} = -2 \log\left(\frac{2,51}{R_e \sqrt{\lambda}}\right) \quad (17)$$

- Regime turbulento rugoso, caracterizado pelo fato de o fator de resistência ser independente do número de Reynolds e depender apenas da rugosidade relativa (quociente entre a rugosidade absoluta e o diâmetro do conduto -  $\varepsilon/D$ ) sendo a relação de dependência expressa pela fórmula de von Karman (1930):

$$\frac{1}{\sqrt{\lambda}} = 2 \log\left(\frac{D}{\varepsilon}\right) + 1,14 \quad \text{ou} \quad \frac{1}{\sqrt{\lambda}} = -2 \log\left(\frac{\varepsilon}{3,7D}\right) \quad (18)$$

Sendo estes regimes separados por regimes de transição:

- Regime de transição laminar-turbulento, em que o fator de resistência tem um comportamento instável;
- Regime de transição turbulento liso-turbulento rugoso, em que o fator de resistência depende simultaneamente do número de Reynolds e da rugosidade relativa;

Não existindo na altura expressões matemáticas que definissem o comportamento destas transições.

Mais tarde, em 1937, Colebrook e White desenvolveram experiências em condutos comerciais, com o objetivo de estudar a transição entre o regime turbulento liso e o regime turbulento rugoso. Como resultado do estudo empreendido, chegaram à conclusão de que adicionando o argumento da função logaritmo da fórmula de Prandtl-von Karman (regime turbulento liso) ao argumento da função logaritmo da fórmula de von Karman (regime turbulento rugoso) obtinha-se uma expressão que, em todo o domínio dos escoamentos turbulentos, traduzia com bastante rigor (erros de 3 a 5%) os valores obtidos nas experiências realizadas, surgindo assim aquela que é hoje vulgarmente conhecida por fórmula de Colebrook-White (1939)

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left( \frac{k}{3,7D} + \frac{2,51}{R_e \sqrt{\lambda}} \right) \quad (19)$$

Nesta expressão, devido ao fato de os materiais que constituem as paredes dos condutos comerciais não terem rugosidade uniforme, o parâmetro caracterizador da rugosidade de Nikuradse ( $\epsilon$ ) foi substituído por uma rugosidade equivalente ( $k$ ), a qual, para cada tipo de material utilizado, deverá ser determinada experimentalmente a partir de medições de perda de carga.

Da simples observação da fórmula de Colebrook-White, facilmente se depreende que esta não permite explicitar o fator de resistência e função das restantes

grandezas envolvidas no seu cálculo. Como consequência desse fato, tornava-se imperativo o recurso a processos numéricos de cada vez que se desejasse calcular  $\lambda$ .

Para evitar esse inconveniente, com base nas fórmulas de Hagen-Poiseille e de Colebrook-White, Moody (1944) desenvolveu um ábaco que permitia facilmente determinar  $\lambda$  em função do número de Reynolds e da rugosidade relativa (o célebre Ábaco de Moody). Apesar deste avanço, Moody achava que o ideal seria obter uma expressão que permitisse explicitar  $\lambda$ . Determinado a conseguir tal proeza, desenvolveu trabalhos nesse sentido, até que, em 1947, surgiu a fórmula de Moody

$$\lambda = 0,0055 \left[ 1 + \left( 20000 \frac{k}{D} + \frac{10^6}{R_e} \right)^{\frac{1}{3}} \right] \quad (20)$$

Esta fórmula era válida nos limites  $4 \times 10^3 \leq Re \leq 10^8$  e  $0 < k/D < 5 \times 10^{-2}$  e apresentava desvios, relativamente à fórmula de Colebrook e White, da ordem dos  $\pm 15\%$ .

Em 1966, foi a vez de Wood apresentar uma outra forma de explicitar  $\lambda$

$\lambda = a + bR_e^{-c}$  na qual:

$$a = 0,094 \left( \frac{k}{D} \right)^{0,225} + 0,53 \left( \frac{k}{D} \right); b = 88 \left( \frac{k}{d} \right)^{0,44}; c = 1,62 \left( \frac{k}{D} \right)^{0,134} \quad (21)$$

Desta feita, uma expressão bastante complexa, a qual, no interior do domínio definido por  $Re > 10000$  e  $10^{-5} < \varepsilon/D < 0,04$ , apresentava desvios da ordem de  $\pm 5\%$ .

Dois anos mais tarde, em 1968, foi publicado um livro da autoria de Nékrasov em que é apresentada uma expressão, com desvios de ordem de grandeza equivalente aos da fórmula de Moody, sendo a sua autoria atribuída a Altshul:

$$\frac{1}{\sqrt{\lambda}} = -1,8 \log \left( \frac{k}{10D} + \frac{7}{R_e} \right) \quad (22)$$

Foi sugerido mais tarde (1975) que se substituíssem os valores 5,15 e 0,892 por 5,1286 e 0,89, respectivamente, conseguindo uma expressão com desvios entre -0,8 e 3%.

Churchill (1973), seguindo as pisadas de Barr, fez uso da expressão desenvolvida por Nikuradse (1932), já anteriormente utilizada por Altshul que a atribuiu a Konakov

$$\frac{1}{\sqrt{\lambda}} = -1,8 \log \left( \frac{7}{R_e} \right) \quad (23)$$

Obtendo outra expressão simples e bastante precisa (desvios entre -0,6 e 3,4%):

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left[ \frac{k}{3,7D} + \left( \frac{7}{R_e} \right)^{0,9} \right] \quad (24)$$

Três anos mais tarde, Swamee e Jain (1976) publicam um trabalho em que propõem a seguinte expressão (desvios entre -0,7 e 3,4%):

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left( \frac{k}{3,7D} + \frac{5,74}{R_e^{0,9}} \right) \quad (25)$$

Não sendo mais do que a fórmula de Churchill em que  $7^{0,9} = 5,76$  é substituído por 5,74.

Em 1979, N.H. Chen apresentou a sua versão de explicitação da fórmula de Colebrook-White. A expressão apresentada reproduzia fielmente os valores do fator de resistência em todo o domínio dos escoamentos turbulentos (desvios relativos inferiores a  $\pm 0,3\%$ ). No entanto, o seu aspecto não era muito atraente:

$$\frac{1}{\sqrt{\lambda}} = -\log \left[ \frac{k}{3,7065} - \frac{5,0452}{R_e} \log \left( \frac{1}{2,8257} \left( \frac{k}{D} \right)^{1,1098} + \frac{5,8506}{R_e^{0,8981}} \right) \right] \quad (26)$$

Um ano depois, ao analisar a fórmula de Altshul, Round (1980) verificou que esta poderia ser melhorada, especialmente para valores elevados do parâmetro (k/D).

Para tal, introduziu-lhe ligeiras modificações e obteve uma expressão que produzia aproximações ligeiramente melhores para o fator de resistência

$$\frac{1}{\sqrt{\lambda}} = -1,8 \log \left( 0,135 \frac{k}{D} + \frac{6,5}{R_e} \right) \quad (27)$$

Em 1980 Barr reaparece propondo uma expressão algo complexa:

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left[ \frac{k}{3,7D} + \frac{5,02 \log(R_e / 4,518 \log(R_e / 7))}{R_e (1 + R_e^{0,32} (kD)^{0,7} / 29)} \right] \quad (28)$$

Sugerindo mais tarde (1981) que esta fosse substituída por uma outra mais simples

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left( \frac{k}{3,7D} + \frac{4,518 \log(R_e / 7)}{R_e (1 + R_e^{0,32} (kD)^{0,7} / 29)} \right) \quad (29)$$

Com desvios relativos inferiores a  $\pm 0,5\%$

Ainda em 1980, Malafaya-Baptista melhorou a expressão de Barr (1975) e utilizou-a na fórmula de Colebrook-White como aproximação inicial, obtendo uma expressão com erros relativos inferiores a  $\pm 0,15\%$ :

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left[ \frac{k}{3,7D} - \frac{5,02}{R_e} \log \left( \frac{k}{3,7D} + \frac{2,51}{R_e} \frac{1}{0,49 R_e^{-0,11} + 0,18 R_e^{0,1} (kD)^{0,6}} \right) \right] \quad (30)$$

Zigrang e Sylvester (1982), fazendo uso da técnica das aproximações sucessivas, apresentam uma expressão com desvios relativos inferiores a  $0,12\%$ :

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left[ \frac{k}{3,7D} - \frac{5,02}{R_e} \log \left( \frac{k}{3,7D} - \frac{5,02}{R_e} \log \left( \frac{k}{3,7D} + \frac{13}{R_e} \right) \right) \right] \quad (31)$$

Em 1983, Haaland propôs uma expressão que, dentre as mais simples, é a mais rigorosa (desvios relativos inferiores a  $\pm 1,5\%$ ):

$$\frac{1}{\sqrt{\lambda}} = -1,8 \log \left[ \left( \frac{k}{3,7D} \right)^{1,11} + \frac{6,9}{R_e} \right] \quad (32)$$

J.J.J. Chen apercebeu-se de que  $\lambda$  varia aproximadamente com  $(k/D)^{0,3}$ .

Associando esta conclusão à equação de Blasius para o regime turbulento liso:

$$\lambda = 0,3164 \left( 0,11 \frac{k}{D} + \frac{1}{R_e^{0,83}} \right)^{0,3} \quad (33)$$

Em 1984, propôs uma expressão relativamente simples:

$$\lambda = 0,3164 R_e^{-0,25} \quad (34)$$

Que apresenta desvios da ordem de  $\pm 8\%$ . Um ano depois (1985), o mesmo autor sugeriu uma expressão mais precisa (desvios entre  $-0,3$  e  $2,6\%$ ):

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left[ \frac{k}{3,7D} + \frac{4,52}{R_e} \log \left( \frac{R_e}{7} \right) \right] \quad (35)$$

Resultado para se considerar, na fórmula de Colebrook-White, como primeira aproximação para  $\lambda$  o valor dado por uma outra forma da equação de Blasius

$$\lambda = 0,184 R_e^{-0,2} \quad (36)$$

Por fim, em 1988, Nackab desenvolveu uma expressão explícita que se aproximava bastante da fórmula de Prandtl-von Karman para o regime turbulento liso. Seguindo os passos de J.J. Chen, introduziu a sua própria aproximação na fórmula de Colebrook-White e obteve a seguinte expressão (desvios entre  $-1$  e  $3\%$ ):

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left[ \frac{k}{3,7D} + \frac{2,51}{R_e \sqrt{0,4R_e^{-0,3} + 0,0053}} \right] \quad (37)$$

Em 1996, Sousa e Sá Marques efetuaram uma pesquisa na esperança de encontrar na bibliografia da especialidade uma expressão que reproduzisse fielmente os resultados da fórmula de Colebrook-White, fosse simples e ao mesmo tempo explícita. Após analisarem as expressões encontradas ao longo da pesquisa, observaram que existiam dois tipos: expressões simples que produzem desvios consideráveis e expressões bastante rigorosas, mas relativamente complexas. Foi então que decidiram empreender esforços no sentido de chegarem a uma expressão com as características desejadas. O primeiro passo realizado foi estudar a forma como os diversos autores construíram as suas expressões e quantificar os desvios produzidos por cada uma delas. Quanto à forma como os autores construíram as suas expressões, dentre as mais rigorosas, concluiu-se que existem duas distintas:

- Reconstrução da fórmula de Colebrook-White substituindo a fórmula de Prandtl-von Karman, para o regime turbulento liso, por uma aproximação explícita (fórmula de Nikuradse, fórmula de Blasius, ou outra) e posterior calibração da expressão obtida;
- Resolução numérica da fórmula de Colebrook-White, partindo de uma expressão que proporciona uma estimativa inicial e efectuando uma ou mais iterações.

As expressões obtidas pela primeira forma, regra geral, são simples e apresentam desvios relativos máximos da ordem dos 3%. Por sua vez, as expressões obtidas por resolução numérica são mais complexas, mas os desvios relativos máximos produzidos são inferiores a 0,5%.

Os primeiros trabalhos realizados foram no sentido de obter uma expressão pela primeira forma descrita. No entanto, foram infrutíferos já que não se conseguiu obter uma expressão que, com a mesma simplicidade, fosse mais rigorosa que a fórmula de Haaland. Por fim optou-se pela forma iterativa, porém com o objetivo de obter uma expressão que, face às já existentes, fosse mais simples. Ao fim de inúmeras tentativas, chegou-se à conclusão de que a expressão mais rigorosa era a que se obtinha considerando a aproximação inicial

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left( \frac{k}{3,7D} + \frac{5}{R_e^{0,89}} \right) \quad (38)$$

Por sinal, bastante semelhante à fórmula de Barr (1975), resultando a seguinte expressão final:

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left[ \frac{k}{3,7D} - \frac{5,02}{R_e} \log \left( \frac{k}{3,7D} + \frac{5}{R_e^{0,89}} \right) \right] \quad (39)$$

Correspondendo-lhe um desvio relativo máximo inferior a 0,2%, podendo-se visualizar, no diagrama de Moody no Apêndice A, a perfeita concordância entre a expressão proposta e a fórmula de Colebrook-White.

Após a realização desse trabalho os autores concluem que hoje em dia existem várias expressões que, sendo explícitas, reproduzem com bastante rigor os valores do fator de resistência estimados pela fórmula de Colebrook-White, pelo que, a utilização desta é perfeitamente desnecessária.

Dentre as diversas expressões que se encontram neste trabalho sugere-se o seguinte:

- Quando se exigir uma expressão que reproduza fielmente os valores da fórmula de Colebrook-White, aconselha-se o uso da expressão proposta neste trabalho, Eq. 39 (desvios relativos inferiores a  $\pm 0,2\%$ );
- Se o objetivo é uma expressão simples e não é exigido tanto rigor, aconselha-se o uso da fórmula de Haaland, Eq. 32 (desvios relativos inferiores a  $\pm 1,5\%$ ).

## 6. O MÉTODO ITERATIVO

Métodos iterativos como de Newton são muito potentes e convergem muito rapidamente, podendo alcançar resultados altamente precisos com duas ou três iterações. Mas não serão aqui mostrados, pois requerem outros conhecimentos que nos tirariam do rumo aqui almejado. Neste texto se pretende mostrar em linguagem simples, como resolver a equação implícita de Colebrook em simultaneidade com a equação de Darcy, a equação da continuidade e o número de Reynolds, empregando um método iterativo simples, fácil e capaz de convergir com razoável rapidez quando aplicado a essas equações.

Na prática, em termos específicos, sempre se depara com três variáveis relevantes nos dimensionamentos hidráulicos, que são listadas a seguir:

1. Diâmetro da tubulação;
2. A vazão (ou velocidade);
3. A perda de carga;

Essas são em síntese, as três variáveis principais envolvidas no cálculo hidráulico, pois as demais (material do tubo, tipo de líquido, temperatura, etc), são básicas. Por qualquer método que viermos a empregar, para se determinar qualquer uma dessas três variáveis, as duas demais deverão ser conhecidas.

Então, substituindo-se as equações 3 e 5 na equação 6 e realizando operações, tem-se:

$$D_{n+1} = \frac{0,7267Q^{0,4}}{(Jg)^{0,2}} \left[ -\log_{10} \left( 0,27 \frac{k}{D_n} + \frac{1,7748\nu}{\sqrt{JgD_n^3}} \right) \right]^{-0,4} \quad (40)$$

$$Q = -2,2214\sqrt{JgD^5} \log_{10} \left( 0,27 \frac{k}{D} + \frac{1,7748\nu}{\sqrt{JgD^3}} \right) \quad (41)$$

$$J_{n+1} = \frac{0,2026Q^2}{D^5 g} \left[ -\log_{10} \left( 0,27 \frac{k}{D} + \frac{1,7748\nu}{\sqrt{gD^3 J_n}} \right) \right]^{-2} \quad (42)$$

Como já dito anteriormente:

$D$  = diâmetro interno do tubo (m)

$Q$  = vazão no tubo (m<sup>3</sup>/s)

$J = h_f/L$  = perda de carga unitária (mca/m)

$k$  = rugosidade equivalente da parede do tubo (m)

$\nu$  = viscosidade cinemática do líquido (m<sup>2</sup>/s)

$g$  = aceleração da gravidade local (m/s<sup>2</sup>)

Com isso passa-se a dispor do conjunto de equações que melhor se aplica a análise, ao cálculo e ao dimensionamento hidráulico de condutos forçados, uniformes, de seção circular, com qualquer diâmetro e comprimento, em regime permanente, para fluidos incompressíveis.

Ao invés de se calcular primeiramente o valor de  $f$ , estas equações possibilitam calcular logo diretamente os valores do diâmetro  $D$ , da vazão  $Q$  e da perda de carga unitária  $J$ , respectivamente. Note-se que a equação 44 é explícita. Já as equações 43 e 45 são implícitas e por isto, para facilitar a resolução, as variáveis recebem o índice " $n$ ", que indica um valor qualquer, ou inicial, e o índice " $n+1$ ", que indica o valor imediatamente subsequente. Em suma: atribui-se valor para a variável com índice " $n$ " e calcula-se o valor da variável com índice " $n+1$ ", repetindo-se este procedimento até alcançar a precisão desejada.

O cálculo da espessura do conduto não é relevante para o estudo aqui apresentado. Porém, o estudo em condutos que necessitem do cálculo da espessura poderá ser objeto de estudos futuros além desta monografia.

Dessa forma, finalizamos a teoria referente à hidráulica aplicada a fluidos incompressíveis em dutos forçados necessária para a construção do simulador computacional. A seguir, trataremos dos conceitos teóricos e práticos que envolvem a programação e construção do código fonte aplicando-se os conceitos de mecânica dos fluidos.

## 7. CONSTRUÇÃO DO SIMULADOR

Para definição do escopo do projeto, indicaremos os diagramas de caso de uso e o diagrama da classe. Essa definição é necessária para o entendimento inicial do “o que” o programa faz, não “como” ele faz.

### 7.1 – Unified Modeling Language (UML)

Segundo Ivar Jacobson [17], podemos dizer que um caso de uso é um "documento narrativo que descreve a seqüência de eventos de um ator que usa um sistema para completar um processo". Um caso de uso é uma técnica de modelagem usada para descrever o que um novo sistema deve fazer. Ele é construído através de um processo interativo no qual as discussões entre o cliente e os desenvolvedores conduzem a uma especificação do sistema da qual todos estão de acordo.

O caso de uso tem por objetivo:

- Decidir e descrever os requisitos funcionais do sistema.
- Fornecer uma descrição clara e consistente do que o sistema deve fazer.
- Permitir descobrir os requisitos funcionais das classes e operações do sistema.

*(Casos de uso não são requisitos)*

A seguir, mostraremos o diagrama de caso de uso e o diagrama de atividades que representam o sistema através do conceito UML. O simulador (programa) constará então de três processos no qual o usuário poderá interagir: cálculo do diâmetro, da vazão e da perda de carga.

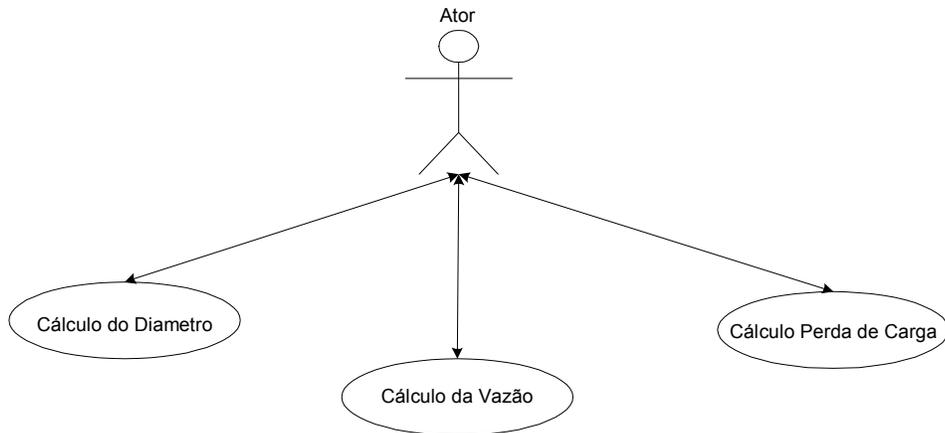


Fig. 5 – Diagrama de caso de uso

Outro diagrama necessário para o projeto do simulador, é o diagrama da atividade. Um diagrama de atividade é essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Na maior parte, isso envolve a modelagem das etapas seqüências em um processo computacional.

Os diagramas de atividade não são importantes somente para a modelagem de aspectos dinâmicos de um sistema ou um fluxograma, mas também para a construção de sistemas executáveis por meio de engenharia de produção reversa.



Fig. 6 – Diagrama de Atividades

Vale ressaltar que UML não é um método de desenvolvimento, o que significa que ela não diz o que fazer primeiro e em seguida ou como desenhar seu sistema, mas ela lhe auxilia a visualizar seu desenho e a comunicação entre objetos. UML é uma linguagem de modelagem não proprietária de terceira geração.

Assim, finalizamos a modelação do sistema e podemos partir para a programação do código fonte.

## 7.2 – Programando com Java

Para a programação, utilizaremos a linguagem JAVA. Java é uma linguagem computacional completa, adequada para o desenvolvimento de aplicações baseadas em tecnologia Web, redes fechadas ou ainda programas *stand-alone*.

Foi desenvolvida na primeira metade da década de 90 nos laboratórios da Sun Microsystems com o objetivo de ser mais simples e eficiente do que suas predecessoras. O alvo inicial era a produção de software para produtos eletrônicos de consumo (fornos de microondas, agendas eletrônicas, etc.). Um dos requisitos para esse tipo de software é ter código compacto e de arquitetura neutra.

A linguagem obteve sucesso em cumprir os requisitos de sua especificação, mas apesar de sua eficiência não conseguiu sucesso comercial. Com a popularização da rede Internet, os pesquisadores da Sun Microsystems perceberam que aquele seria um nicho ideal para aplicar a recém criada linguagem de programação. A partir disso, adaptaram o código Java para que pudesse ser utilizado em microcomputadores conectados a rede Internet, mais especificamente no ambiente da World Wide Web. Java permitiu a criação de programas batizados applets, que trafegam e trocam dados através da Internet e se utiliza a interface gráfica de um web browser. Implementaram também o primeiro browser compatível com a linguagem, o HotJava, que fazia a interface entre as aplicações Java e o sistema operacional dos computadores.

Com isso, a linguagem conseguiu uma popularização fora de série, passando a ser usada amplamente na construção de documentos web que permitam maior interatividade. Os principais web browsers disponíveis comercialmente passaram a dar suporte aos programas Java, e outras tecnologias em áreas como computação gráfica e

banco de dados também buscaram integrar-se com o novo paradigma proposto pela linguagem: aplicações voltadas para o uso de redes de computadores.

Java é uma linguagem poderosa em ambientes distribuídos complexos como a rede Internet. Mas sua versatilidade permite ao programador ir além, oferecendo uma poderosa linguagem de programação de uso geral, com recursos suficientes para a construção de uma variedade de aplicativos que podem ou não, conforme a necessidade do projeto, depender do uso de recursos de conectividade.

As principais vantagens da adoção do Java são:

- Simplicidade e eficiência de código orientado a objetos: Java é uma linguagem simples, de fácil aprendizado ou migração, pois possui um reduzido número de construções. A diminuição das construções mais suscetíveis a erros de programação, tais como ponteiros e gerenciamento de memória via código de programação também faz com que a programação em Java seja mais eficiente. Contém um conjunto de bibliotecas que fornecem grande parte da funcionalidade básica da linguagem, incluindo rotinas de acesso à rede e criação de interface gráfica.
- Código Interpretado e Portável: A linguagem Java é tanto compilada como interpretada. Após escrever um programa em Java, utilizando um editor de textos qualquer, se salva o programa como código fonte. A seguir, pode-se compilar esse código fonte a fim de produzir um tipo de arquivo binário chamado de arquivo de classe. Esses arquivos não são executados diretamente, pois eles não contêm instruções que são entendidas diretamente pelos processadores atualmente disponíveis no mercado. Os programas Java são compilados em um formato intermediário chamado bytecodes. Assim, esses programas podem ser executados

em qualquer sistema através de um interpretador Java (runtime environment). Com isso, o código precisa ser escrito e compilado apenas uma vez, pois os bytecodes gerados serão executados da mesma forma em qualquer plataforma de hardware e software.

- Segurança: Por ter seu projeto voltado para a simplicidade de código, as possibilidades de erro de programação em Java são reduzidas. Apesar disso, a linguagem traz outros recursos para tornar seu código ainda mais eficiente. O processo de compilação - geração de bytecodes - é projetado para a detecção prévia dos possíveis erros, evitando que os erros se manifestem em tempo de execução. O uso de código para tratamento de exceções - *exception handling* - permite manter a consistência da aplicação no caso de erros.
- Aplicações distribuídas e processamento paralelo: Com a larga utilização dos recursos da rede Internet, a linguagem teve seu projeto voltado para as aplicações em rede. Assim, a linguagem traz classes para o suporte a vários níveis de conectividade: acesso a URLs (padrão Internet), uso de conexões em sockets, criação de protocolos, criação de clientes e servidores. A introdução desses conceitos se deu de forma simples para o programador, pois permite a ele o acesso às informações da rede com a mesma facilidade do acesso aos arquivos locais.

Para a construção do simulador, estamos utilizando três páginas .jsp iniciais (conforme descrevemos no diagrama de caso de uso) responsáveis pela apresentação inicial das telas dos módulos de cálculos do diâmetro, vazão e perda de carga. Assim, temos uma página (arquivos .jsp) responsável para atender cada parte do cálculo. A seguir, apresenta-se o código fonte em separado. Todo o código fonte contém os

comentários sobre sua descrição de atividades ou de suas funções em relação ao sistema global.

## 7.2.1 – Arquivos de Entrada

O arquivo index.jsp é responsável para apresentação de tela inicial do sistema:

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>
<%@ taglib uri="http://struts.apache.org/tags-logic" prefix="logic" %>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean" %>
<%@ taglib uri="http://displaytag.sf.net" prefix="display"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://struts.apache.org/tags-nested" prefix="n"%>
<!--
    Página inicial que contém as três tabs para realização dos calculos.
    Está página também apresenta os dados e resultados em forma de tabela.
-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<html:base/>
<head>
    <script src="/js/behaviour.js" language="javascript"></script>
    <script src="/js/jspcontrols.js" language="javascript"></script>
    <LINK rel="stylesheet" type="text/css" HREF="<html:rewrite page='/css/estilo.css'/>">
    <LINK rel="stylesheet" type="text/css" HREF="<html:rewrite page='/css/tigristabs.css'/>">
    <LINK rel="stylesheet" type="text/css" HREF="<html:rewrite page='/css/displaytag.css'/>">

    <script>
        function submitForm( pValor ){
            document.forms[0].metodoExecutar.value = pValor;
            document.forms[0].submit();
        }

        /* function iniciar() {
            document.all("descricao").selectedIndex = 0;
            document.all("vicosidadeCinematica").style.display = "none";
            document.all("parametroBean.rugosidade").selectedIndex = 0;
            for(i = 0; i < document.forms[0].length; i++){
                if ( document.forms[0].elements[i].type == "text" ) {
                    document.forms[0].elements[i].value = "0.0";
                }
            }
        }
    */

        function validar(){
            if ( document.all("metodoExecutar").value != "consultarTemperatura" ){
                for(i = 0; i < document.forms[0].length; i++){
                    if ( document.forms[0].elements[i].type == "text" &&
document.forms[0].elements[i].value == 0.0 ) {
```

```

        document.forms[0].elements[i].select();
        alert(document.forms[0].elements[i].title);
        return false;
    }
    else if (document.forms[0].elements[i].type == "select-one"){
        if ( document.forms[0].elements[i].value == -1 ){
            document.forms[0].elements[i].focus();
            alert(document.forms[0].elements[i].title);
            return false;
        }
    }
}
}
return true;
}

function abrirNovaTela() {
    window.open("", "novaTela",
"width=450,height=400,left=0,top=100,screenX=0,screenY=100");
}
</script>
</head>

<body>
<html:form action="/simularCalculo" method="post" onsubmit="return validar();"
    <P align="center" class="titulo">
        <bean:message key="nome.aplicacao"/>
    </P>

    <div align="center" style="border: 1px solid navy; height: 500px;background: white; ">
        <table>
            <tr>
                <td class="label"><bean:message key="viscosidade.label"/></td>
                <td>
                    <html:select property="descricao"
onchange="submitForm('consultarTemperatura');" styleClass="campo" title="Selecione a Viscosidade
Cinemática.">
                        <html:optionsCollection property="viscosidadeList" value="value"
label="label"/>
                    </html:select>
                    <logic:notEmpty property="viscosidadeTempList" name="simuldorForm">
                        <div id="vicosidadeCinematica">
                            <html:select
property="parametroBean.vicosidadeCinematica">
                                <html:optionsCollection
property="viscosidadeTempList" value="value" label="label"/>
                            </html:select>
                        </div>
                    </logic:notEmpty>
                </td>
            </tr>
            <tr>
                <td class="label"><bean:message key="rugosidade.label"/></td>
                <td>
                    <html:select property="rugosidadeDescricao" title="Selecione a Rugosidade.">
                        <html:optionsCollection property="rugosidadeList" value="value"
label="label"/>
                    </html:select>
                </td>
            </tr>
        </table>
    </div>

```



O arquivo diâmetro.jsp é responsável pela composição dos campos para entrada dos valores referente ao cálculo do diâmetro.

```

<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>
<%@ taglib uri="http://struts.apache.org/tags-logic" prefix="logic" %>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean" %>

<!-- Página que contém os campos usados no cálculo do diâmetro -->
<table align="center" style="border: 1pt solid black;">
  <tr>
    <td class="label"><bean:message key="vazao.label"/></td>
    <td><html:text title="Digite a Vazão no Tubo." property="parametroBean.vazaoQ"
maxlength="10" size="10" styleClass="campo"/></td>
  </tr>
  <tr>
    <td class="label"><bean:message key="perda.label"/></td>
    <td><html:text title="Digite a Perda de Carga Unitária."
property="parametroBean.perdaCargaJ" maxlength="10" size="10" styleClass="campo"/></td>
  </tr>
  <logic:equal value="true" property="apresentaDiamentroResultado" name="simuldorForm">
    <tr>
      <td colspan="2" align="center" class="resposta">
        <bean:message key="resultado.label"/><bean:write
format="<%= (String)request.getAttribute("formato2") %>" property="parametroBean.resultado"
name="simuldorForm"/>&nbsp;m.
      </td>
    </tr>
  </logic:equal>
</table>
<html:hidden property="classeID" value="1"/>
<html:hidden property="metodoExecutar" value="calcular"/>

```

O arquivo perda\_carga.jsp é responsável pela composição dos campos para a entrada dos valores referentes ao cálculo da perda de carga unitária.

```

<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>
<%@ taglib uri="http://struts.apache.org/tags-logic" prefix="logic" %>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean" %>

<!-- Página que contém os campos usados no cálculo da perda de carga -->
<table align="center" style="border: 1pt solid black;">
  <tr>
    <td class="label"><bean:message key="diametro.label"/></td>
    <td><html:text title="Digite o Diâmetro Interno do Tubo."
property="parametroBean.diametroD" maxLength="10" size="10" styleClass="campo"/></td>
  </tr>
  <tr>
    <td class="label"><bean:message key="vazao.label"/></td>
    <td><html:text title="Digite a Vazão no Tubo." property="parametroBean.vazaoQ"
maxLength="10" size="10" styleClass="campo"/></td>
  </tr>

  <logic:equal value="true" property="apresentaDiamentroResultado" name="simuldorForm">
    <tr>
      <td colspan="2" align="center" class="resposta">
        <bean:message key="resultado.label"/><bean:write
format="<%= (String)request.getAttribute("formato2") %>" property="parametroBean.resultado"
name="simuldorForm"/>&nbsp;   mca/m.
      </td>
    </tr>
  </logic:equal>
</table>
<html:hidden property="classeID" value="3"/>
<html:hidden property="metodoExecutar" value="calcular"/>

```

O arquivo vazao.jsp é responsável pela composição dos campos para a entrada dos valores referentes ao cálculo da vazão.

```

<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>
<%@ taglib uri="http://struts.apache.org/tags-logic" prefix="logic" %>
<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean" %>

<!-- Página que contém os campos usados no cálculo da vazão -->
<table align="center" style="border: 1pt solid black;">
  <tr>
    <td class="label"><bean:message key="diametro.label"/></td>
    <td><html:text title="Digite o Diâmetro Interno do Tubo."
property="parametroBean.diametroD" maxLength="10" size="10" styleClass="campo"/></td>
  </tr>
  <tr>
    <td class="label"><bean:message key="perda.label"/></td>
    <td><html:text title="Digite a Perda de Carga Unitária."
property="parametroBean.perdaCargaJ" maxLength="10" size="10" styleClass="campo"/></td>
  </tr>
  <tr>
    <td colspan="2" align="center" class="resposta">
      <bean:message key="resultado.label"/><bean:write
format="<%= (String)request.getAttribute("formato2") %>" property="parametroBean.resultado"
name="simuldorForm"/>&nbsp;m&sup3;/s.
    </td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <logic:equal value="true" property="apresentaDiamentroResultado" name="simuldorForm">
        <td colspan="2" align="center" class="resposta">
          <bean:message key="resultado.label"/><bean:write
format="<%= (String)request.getAttribute("formato2") %>" property="parametroBean.resultado"
name="simuldorForm"/>&nbsp;m&sup3;/s.
        </td>
      </td>
    </tr>
  </tr>
</table>
<html:hidden property="classeID" value="2"/>
<html:hidden property="metodoExecutar" value="calcular"/>

```

Em seguida, foram criados três java classes para efetuar o cálculo (usando a teoria e as equações 40, 41 e 42 mencionadas anteriormente) para cada equação:

### 7.2.2 – Código de Cálculo

Arquivo: CalcularDiametro.java

```

package servico.impl;
import bean.ParametroBean;
import bean.Util;
import servico.interfaces.Calcular;
/**
 *
 * Classe usa para realizar o cálculo do Diâmetro

```

```

*
*/
public class CalcularDiametro implements Calcular {

    public double executar(ParametroBean bean) {
        double J          = bean.getPerdaCargaJ(); // bean.getComprimentoL();
        double viscosidade = bean.getVicosidadeCinematica() * Math.pow(10, -6);
        double Dn = 1;
        String numFormatado = Util.formatar( Dn );

        int contador = 1;
        while(true){

            double c1      = Util.NUM_CONST1 * Math.pow( bean.getVazaoQ(), 0.4);
//0.7267 * D ^ 0,4
            double c2      = Math.pow(( J * Util.GRAVIDADE),0.2); // (J * g) ^ 0.2
            double c3      = c1 / c2;

            double c4      = Util.NUM_CONST3 * (bean.getRugosidade() / Dn); // 0.27
* (K / D)
            double c5      = Util.NUM_CONST2 * viscosidade; //1.7748 * ni
            double c6      = J * Util.GRAVIDADE * (Math.pow(Dn,3)); // J * g * (D ^3)
            double c7      = Math.sqrt( c6 ); //raiz quadrada( c6 )
            double c8      = c5 / c7;
            double c9      = -Math.log10( c4 + c8 ); // Log10( c4 + c7 )
            double c10     = Math.pow( c9, -0.4); //c8 ^ -0.4
            Dn             = c3 * c10;
            System.out.println("Dn[" + contador + "]: " + Util.formatar(Dn));
            if ( numFormatado.equals(Util.formatar( Dn )) ){
                //encerra o loop quando encontrar um padrão
                break;
            }
            contador++;
            numFormatado = Util.formatar( Dn );
        }
        return Dn;
    }
}

```

Arquivo: CalcularPerdaCarga.java

```

package servico.impl;

import bean.ParametroBean;
import bean.Util;
import servico.interfaces.Calcular;

/**
 * Classe usa para realizar o cálculo da Perda de Carga
 */
public class CalcularPerdaCarga implements Calcular {

    public double executar(ParametroBean bean) {
        double viscosidade = bean.getVicosidadeCinematica() * Math.pow(10, -6); //ni
        double Jn          = 1.0; //?
        int contador      = 1;
        String numFormatado = Util.formatar( Jn );
    }
}

```

```

        while(true){
            double c1 = 0.2026 * ( Math.pow(bean.getVazaoQ(),2) ); // 0.2026 * ( Q ^ 2 )
            double c2 = (Math.pow(bean.getDiametroD(),5)) * Util.GRAVIDADE; // ( D ^
5) * g

            double c3 = c1 / c2;
            double c4 = 0.27 * ( bean.getRugosidade() / bean.getDiametroD() ); //0.27 * ( K
/ D)

            double c5 = 1.7748 * viscosidade; //1.7748 * ni
            double c6 = Math.sqrt( Util.GRAVIDADE *
(Math.pow(bean.getDiametroD(),3)) * Jn); //Raiz Quadrada( g * (D^3) * Jn)
            double c7 = c5 / c6;
            double c8 = c4 + c7;
            double c9 = -Math.log10( c8 );
            Jn
                = c3 * ( Math.pow(c9,-2)); // c3 * ( Math.sqrt( c9 ));

            System.out.println("Jn[" + contador + "]: " + Util.formatar(Jn));
            if ( numFormatado.equals(Util.formatar( Jn )) ){
                //encerra o loop quando encontrar um padrão
                break;
            }
            contador++;
            numFormatado = Util.formatar( Jn );
        }
        return Jn;
    }
}

```

Arquivo: CalcularVazao.java

```

package servico.impl;

import bean.ParametroBean;
import bean.Util;
import servico.interfaces.Calcular;

/**
 *
 * Classe usada para realizar o cálculo da Vazão
 *
 */
public class CalcularVazao implements Calcular {

    public double executar(ParametroBean bean) {
        double viscosidade = bean.getVicosidadeCinematica() * Math.pow(10, -6); // NI letra
grega

        double J = bean.getPerdaCargaJ(); /// bean.getComprimentoL();

        double c1 = J * Util.GRAVIDADE * Math.pow(bean.getDiametroD(),5); // J * g * D ^ 5
        double c2 = -2.2214 * Math.sqrt( c1 ); // -2.2214 * raiz quadrada(c1)

        double c3 = Math.sqrt( J * Util.GRAVIDADE * Math.pow(bean.getDiametroD(),3) );// J
* g * D ^ 3

        double c4 = (Util.NUM_CONST2 * viscosidade) / ( c3 ); // (1.7748 * ni) / C3
        double c5 = Math.log10( 0.27 * bean.getRugosidade()/bean.getDiametroD() + c4);

        double resultado = c2 * c5;
    }
}

```

```

        return resultado;
    }
}

```

A saída emprega o conceito de interface, que nada mais é do que um conjunto de declarações de métodos (nome, tipo de retorno, tipos dos argumentos) desprovidos de implementação. Cabe ao programador que deseja implementar a interface em questão providenciar uma implementação destes métodos na classe que ele está desenvolvendo. Todas as classes mencionadas anteriormente, implementam uma classe de interface como mostrado a seguir:

Arquivo: Calcular.java

```

package servico.interfaces;
import bean.ParametroBean;

/**
 *
 * Interface que define um contrato a ser implementado pela classe de negócio.
 * Essa interface utiliza o design pattern Command que permite encapsular uma
 * requisição como um objeto.
 * Diferentes tipos de requisições podem ser executados utilizando-se uma mesma
 * interface criando um alto nível de abstração.
 *
 */

public interface Calcular {
    /**
     * Realiza o calculo
     * @param bean  Dados digitados pelo usuario
     * @return      resultado do cálculo
     */
    public double executar( ParametroBean bean );
}

```

### 7.2.3 – Códigos Recicláveis

Durante a programação do simulador, foram construídos arquivos auxiliares para a organização e melhoria da produtividade de codificação. Estes arquivos são responsáveis, por exemplo, pelo armazenamento de *labels* que são centralizados e depois

utilizados como “variáveis globais” do sistema, assim como o aproveitamento da “reusabilidade” de código-fonte para pontos específicos como, por exemplo, a criação de combo-box e campos de dados.

A seguir, estão listados alguns exemplos de código-fonte reutilizáveis que foram implementados neste sistema:

Arquivo: CalcularAction.java

```
package simulador.action;

import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import bean.ParametroBean;
import bean.Util;

import servico.interfaces.Calcular;
import simulador.form.SimuladorForm;
import xml.parse.ParseXML;

public class CalcularAction extends Action {
    private ParseXML xml = null;
    private final static String PATH_CLASS = "servico.impl.";
    public CalcularAction(){
        super();
        xml = ParseXML.getInstance();
    }

    /**
     * Cria uma ComboBox com os dados da temperatura ao selecionar a Viscosidade Cinemática
     * @param simuladorForm
     * @throws Exception
     */
    public void consultarTemperatura(SimuladorForm simuladorForm) throws Exception {

        List temperaturas = xml.parse( simuladorForm.getDescricao()
);
        simuladorForm.setViscosidadeTempList( temperaturas );
    }

    /**
     * Trata a requisição do cliente ao clicar no botão Calcular
     */
}
```

```

public ActionForward execute(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response) throws Exception {

    SimuladorForm simuladorForm = (SimuladorForm)form;
    String metodoExecutar = simuladorForm.getMetodoExecutar();

    //Endereço mapeado no arquivo struts-config.xml
    String telaDestino = "sucesso";

    if ( "calcular".equalsIgnoreCase(metodoExecutar) ){
        calcular( simuladorForm );
    }
    else if ( "consultarTemperatura".equalsIgnoreCase(metodoExecutar) ){
        consultarTemperatura(simuladorForm);
    }
    else if ( "limpar".equalsIgnoreCase(metodoExecutar)){
        limpar( simuladorForm );
    }
    else if ( "apresentarResumo".equalsIgnoreCase(metodoExecutar)){
        telaDestino = "resumo_tela";
        ParametroBean parametroBean = (ParametroBean)
simuladorForm.getResultados().get( simuladorForm.getIndice() );
        simuladorForm.setBeanResumo( parametroBean );
        simuladorForm.setMetodoExecutar(null);
    }
    else {
        simuladorForm.setApresentaDiamentroResultado( false );
        simuladorForm.getParametroBean().setResultado(0);
    }
    request.setAttribute("formato1", Util.FORMATO_TABELA);
    request.setAttribute("formato2", Util.FORMATO);
    return mapping.findForward(telaDestino);
}

/**
 * Delega o calculo para a classe especifica
 * @param mapping
 * @param form
 * @param request
 * @param response
 * @return
 * @throws Exception
 */
public void calcular(SimuladorForm simuladorForm) throws Exception {

    Calculo calcular = null;
    String nomeClasse = getNomeClasse( simuladorForm );
    double resultado = -1.0;
    double valorRugosidade = xml.getValorDescricaoRugosidade(
simuladorForm.getRugosidadeDescricao());
    String temperatura = xml.getTemperatura(
simuladorForm.getDescricao(), simuladorForm.getParametroBean().getVicosidadeCinematica());
    ParametroBean bean = null;
    String labelRugosidade = xml.getComboDescricao("rugosidadeList",
simuladorForm.getRugosidadeDescricao());
    String labelViscosidade = xml.getComboDescricao("viscosidadeList",
simuladorForm.getDescricao());
}

```

```

simuladorForm.getParametroBean().setRugosidade( valorRugosidade );
simuladorForm.getParametroBean().setTemperatura( temperatura );

calcular = (Calcular) Class.forName( PATH_CLASS + nomeClasse).newInstance();
resultado = calcular.executar( simuladorForm.getParametroBean() );
simuladorForm.getParametroBean().setResultado( resultado );
simuladorForm.getParametroBean().setLabelRugosidade( labelRugosidade );
simuladorForm.getParametroBean().setLabelViscosidade( labelViscosidade );

simuladorForm.setApresentaDiamentroResultado( true );
bean = (ParametroBean) simuladorForm.getParametroBean().clone();
simuladorForm.getResultados().add( bean );

simuladorForm.setMetodoExecutar(null);
consultarTemperatura( simuladorForm );
}

private void limpar(SimuladorForm simulador ){
    simulador.setDescricao( null );
    simulador.setApresentaDiamentroResultado(false);
    simulador.setParametroBean( new ParametroBean() );
    simulador.setRugosidadeDescricao( null );
    simulador.getViscosidadeTempList().clear();
    simulador.getResultados().clear();
}

/**
 * Identifica qual classe que será instanciada em tempo de runtime
 * @param simulador    Dados do parametros
 * @return             Nome da classe
 */
private String getNomeClasse( SimuladorForm simulador ){
    String nomeClasse = null;
    String nomeCalculo = null;

    int id = simulador.getClasseID();

    if ( id == 1 ){
        nomeClasse = "CalcularDiametro";
        nomeCalculo = "Cálculo do Diâmetro";
        simulador.getParametroBean().setDiametroD(0);
    }
    else if ( id == 2 ){
        nomeClasse = "CalcularVazao";
        nomeCalculo = "Cálculo da Vazão";
        simulador.getParametroBean().setVazaoQ(0);
    }
    else if ( id == 3 ){
        nomeClasse = "CalcularPerdaCarga";
        nomeCalculo = "Cálculo da Perda de Carga";
        simulador.getParametroBean().setPerdaCargaJ(0);
    }
    //System.out.println("NOME DA CLASSE: " + nomeClasse);
    simulador.getParametroBean().setNomeCalculo( nomeCalculo );
    return nomeClasse;
}
}
}

```

Arquivo: SimuladorForm.java

```
package simulador.form;

import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;

import xml.parse.ParseXML;

import bean.ParametroBean;
/**
 * Classe que armazena os dados digitado pelo cliente por meio da interface gráfica.
 * @version 1.0
 * Data: 07/05/2006
 */
public class SimuladorForm extends ActionForm {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    private ParametroBean parametroBean = null;
    private ParametroBean beanResumo = null;
    private int classeID = 0;
    private List rugosidadeList = null;
    private List viscosidadeList = null;
    private List viscosidadeTempList = null;
    private String descricao = "";
    private String metodoExecutar = null;
    private boolean apresentaDiametroResultado = false;
    private ParseXML xml = null;
    private String rugosidadeDescricao = null;
    private List resultados = null;
    private int indice = 0;

    /**
     * Retorna o índice da resultado do cálculo na coleção
     * @return índice da resultado do cálculo
     */
    public int getIndice() {
        return indice;
    }

    /**
     * Informa o índice da resultado do cálculo na coleção
     * @param indice
     */
    public void setIndice(int indice) {
        this.indice = indice;
    }
}
```

```

/**
 * Construtor. Onde se iniciam as variaveis.
 *
 */
public SimuladorForm() {
    super();
    xml = ParseXML.getInstance();
    loadViscosidadeLista();
    loadRugosidadeLista();
    parametroBean = new ParametroBean();
    resultados = new ArrayList(3);
}

/**
 * Retorna uma lista de resultados dos calculos
 * @return lista de resultados dos calculos
 */
public List getResultados() {
    return resultados;
}

/**
 * Informa lista de resultados dos calculos
 * @param resultados
 */
public void setResultados(List resultados) {
    this.resultados = resultados;
}

/**
 * Retorna a descrição da rugosidade
 * @return descrição da rugosidade
 */
public String getRugosidadeDescricao() {
    return rugosidadeDescricao;
}

/**
 * Informa a descrição da rugosidade
 * @param rugosidadeDescricao
 */
public void setRugosidadeDescricao(String rugosidadeDescricao) {
    this.rugosidadeDescricao = rugosidadeDescricao;
}

public boolean isApresentaDiamentroResultado() {
    return apresentaDiamentroResultado;
}

/**
 * Informa se pode apresentar o resultado do calculo na tela
 * @param apresentaDiamentroResultado
 */
public void setApresentaDiamentroResultado(boolean apresentaDiamentroResultado) {
    this.apresentaDiamentroResultado = apresentaDiamentroResultado;
}

/**

```

```

* Retorna o valor selecionado da combo box Viscosidade Cinemática
* @return      valor selecionado da combo box Viscosidade Cinemática
*/
public String getDescricao() {
    return descricao;
}

/**
* Informa o valor a combo box Viscosidade Cinemática
* @param descricao
*/
public void setDescricao(String descricao) {
    this.descricao = descricao;
}

/**
* Retorna uma lista de temperaturas ao selecionar a Viscosidade Cinemática
* @return      lista de temperaturas
*/
public List getViscosidadeTempList() {
    return viscosidadeTempList;
}

/**
* Informa uma lista de temperaturas ao selecionar a Viscosidade Cinemática
* @param viscosidadeTempList
*/
public void setViscosidadeTempList(List viscosidadeTempList) {
    this.viscosidadeTempList = viscosidadeTempList;
}

/**
* Retorna uma lista de Rugosidade
* @return      lista de Rugosidade
*/
public List getRugosidadeList() {
    return rugosidadeList;
}

/**
* Informa lista de Rugosidade
* @param rugosidadeList
*/
public void setRugosidadeList(List rugosidadeList) {
    this.rugosidadeList = rugosidadeList;
}

/**
* Retorna Identificador da classe que realiza o calculo
* @return Identificador da classe que realiza o calculo
*/
public int getClasseID() {
    return classeID;
}

/**
* Informa Identificador da classe que realiza o calculo
* @param classeID
*/

```

```

public void setClasseID(int classeID) {
    this.classeID = classeID;
}

/**
 * Armazena todos os dados, informados pelo cliente, usado na calculo
 * @return      dados para calculo
 */
public ParametroBean getParametroBean() {
    return parametroBean;
}

/**
 * Armazena todos os dados, informados pelo cliente, usado na calculo
 * @param paramBean
 */
public void setParametroBean(ParametroBean paramBean) {
    this.parametroBean = paramBean;
}

/**
 * Carrega a lista de viscosidade usando um arquivo XML
 *
 */
public void loadViscosidadeLista(){
    List lista = xml.parse( "viscosidadeList" );
    setViscosidadeList( lista );
}

/**
 * Carrega a lista de Rugosidade usando um arquivo XML
 *
 */
public void loadRugosidadeLista(){
    List lista = xml.parse( "rugosidadeList" );
    setRugosidadeList( lista );
}

/**
 * Retorna uma lista com dados da Viscosidade
 * @return      Retorna uma lista de Viscosidade
 */
public List getViscosidadeList() {
    return viscosidadeList;
}

/**
 * Informa uma lista com dados da Viscosidade
 * @param viscosidadeList
 */
public void setViscosidadeList(List viscosidadeList) {
    this.viscosidadeList = viscosidadeList;
}

/**
 * Identifica qual a ação a ser tomada
 * @return      ação a ser tomada
 */
public String getMetodoExecutar() {

```

```

        return metodoExecutar;
    }

    /**
     * Identifica qual a ação a ser tomada
     * @param metodoExecutar
     */
    public void setMetodoExecutar(String metodoExecutar) {
        this.metodoExecutar = metodoExecutar;
    }

    /**
     * Retorna todos os dados para apresentar na tela de resumo
     * @return      dados para apresentar na tela de resumo
     */
    public ParametroBean getBeanResumo() {
        return beanResumo;
    }

    /**
     * Informa todos os dados para apresentar na tela de resumo
     * @param beanResumo
     */
    public void setBeanResumo(ParametroBean beanResumo) {
        this.beanResumo = beanResumo;
    }
}

```

#### Arquivo: Viscosidade.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Esse XML simula um banco de dados que armazena dados da viscosidade e rugosidade -->
<tabela-valores>
    <!-- Lista a temperatura -->
    <agua>
        <option label="10º" value="1.31"/>
        <option label="20º" value="1.00"/>
        <option label="40º" value="0.66"/>
        <option label="80º" value="0.37"/>
    </agua>
    <aguadomar>
        <option label="5º" value="1.61"/>
        <option label="15º" value="1.22"/>
        <option label="25º" value="0.97"/>
    </aguadomar>
    <alcoholmetilico>
        <option label="20º" value="0.727"/>
    </alcoholmetilico>
    <asfalto>
        <option label="120º" value="1600"/>
    </asfalto>
    <azeite>
        <option label="38º" value="43"/>
    </azeite>
    <benzol>
        <option label="20º" value="0.744"/>
    </benzol>
    <gasolina>

```

```

        <option label="20" value="0.6"/>
</gasolina>
<glicerina>
        <option label="0" value="8310"/>
        <option label="20" value="1180"/>
        <option label="40" value="223"/>
</glicerina>
<leite>
        <option label="20" value="1.13"/>
</leite>
<oleobrutodens0855>
        <option label="30" value="5.5"/>
        <option label="40" value="4.5"/>
        <option label="60" value="3.5"/>
        <option label="80" value="2.7"/>
        <option label="100" value="2.1"/>
        <option label="120" value="1.7"/>
        <option label="150" value="1.5"/>
</oleobrutodens0855>
<oleocombdens0940>
        <option label="30" value="400"/>
        <option label="40" value="180"/>
        <option label="60" value="60"/>
        <option label="80" value="25"/>
        <option label="100" value="13"/>
        <option label="120" value="8"/>
</oleocombdens0940>
<oleocombdens0968>
        <option label="40" value="1200"/>
        <option label="60" value="300"/>
        <option label="80" value="80"/>
        <option label="100" value="35"/>
        <option label="120" value="18.5"/>
        <option label="150" value="10"/>
</oleocombdens0968>
<oleodealgodao>
        <option label="38" value="38"/>
</oleodealgodao>
<oleodebaleia>
        <option label="38" value="38"/>
</oleodebaleia>
<oleodesoja>
        <option label="38" value="35"/>
</oleodesoja>
<oleodelinhaca>
        <option label="38" value="30"/>
</oleodelinhaca>
<oleoSAE-10>
        <option label="20" value="80"/>
        <option label="30" value="45"/>
        <option label="40" value="30"/>
        <option label="60" value="15"/>
        <option label="80" value="10"/>
        <option label="100" value="5"/>
        <option label="120" value="3"/>
</oleoSAE-10>
<oleoSAE-30>
        <option label="20" value="250"/>
        <option label="30" value="130"/>
        <option label="40" value="80"/>

```

```

    <option label="60" value="35"/>
    <option label="80" value="19"/>
    <option label="100" value="10"/>
    <option label="120" value="6.5"/>
</oleoSAE-30>
<oleoSAE-90>
    <option label="40" value="250"/>
</oleoSAE-90>
<tetracloretocarbono>
    <option label="20" value="0.612"/>
</tetracloretocarbono>

```

<!-- Tabela de Viscosidade -->

```

<viscosidadeList>
    <option value="-1" label="[Selecione]"/>
    <option value="agua" label="Água"/>
    <option value="aguadomar" label="Água do mar"/>
    <option value="alcoolmetilico" label="Álcool metílico"/>
    <option value="asfalto" label="Asfalto"/>
    <option value="azeite" label="Azeite"/>
    <option value="benzol" label="Benzol"/>
    <option value="gasolina" label="Gasolina"/>
    <option value="glicerina" label="Glicerina"/>
    <option value="leite" label="Leite"/>
    <option value="oleobrutodens0855" label="Óleo bruto dens. 0855"/>
    <option value="oleocombdens0940" label="Óleo comb. dens. 0940"/>
    <option value="oleodealgodao" label="Óleo de algodão"/>
    <option value="oleodebaleia" label="Óleo de baleia"/>
    <option value="oleodelinhaca" label="Óleo de linhaça"/>
    <option value="oleodesoja" label="Óleo de soja"/>
    <option value="oleoSAE-10" label="Óleo SAE-10"/>
    <option value="oleoSAE-30" label="Óleo SAE-30"/>
    <option value="oleoSAE-90" label="Óleo SAE-90"/>
    <option value="tetracloretocarbono" label="Tetracloro carbono"/>
</viscosidadeList>

```

<!-- Tabela de Rugosidade -->

```

<rugosidadeList>
    <option label="[Selecione]" value="-1"/>
    <option label="Aço comercial" value="acocomercial"/>
    <option label="Aço galvanizado" value="acogalvanizado"/>
    <option label="Aço com ferrugem leve" value="acocomferrugemleve"/>
    <option label="Aço com grandes incrustações" value="acocomgrandesincrustacoes"/>
    <option label="Aço com cimento centrifugado" value="acocomcimentocentrifugado"/>
    <option label="Aço revestido com asfalto" value="acorevestidocomasfalto"/>
    <option label="Aço rev. c/esmalte, vinil, epoxi" value="acorevesmaltevinilepoxi"/>
    <option label="Alumínio" value="aluminio"/>
    <option label="Concreto muito rugoso" value="concretomuitorugoso"/>
    <option label="Concreto rugoso" value="concretorugoso"/>
    <option label="Concreto liso" value="concretoliso"/>
    <option label="Concreto muito liso" value="concretomuitoliso"/>
    <option label="Concreto alisado centrifugado" value="concretoalisadocentrifugado"/>
    <option label="Concreto liso formas metálicas" value="concretolisoforasmetalicas"/>
    <option label="Ferro fundido asfaltado" value="ferrofundidoasfaltado"/>
    <option label="Ferro galvanizado" value="ferrogalvanizado"/>
    <option label="Ferro fund. não revestido novo" value="ferrofundnaorevestidonovo"/>
    <option label="Ferro fund. com ferrugem leve" value="ferrofundcomferrugemleve"/>
    <option label="Ferro fund. c/cim. centrifugado" value="ferrofundcimcentrifugado"/>
    <option label="Fibrocimento" value="fibrocimento"/>
    <option label="Manilha cerâmica" value="manilhaceramica"/>

```

```

        <option label="Latão cobre" value="lataocobre"/>
        <option label="Plásticos" value="plasticos"/>
        <option label="Rocha (galeria) não revestida" value="rocha(galeria)naorevestida"/>
    </rugosidadeList>

    <!-- Tabela que traduz a descrição da rugosidade para o valor correspondente -->
    <rugosidadeValor>
        <option label="[Selecione]" value="-1"/>
        <option label="acocomercial" value="0.00006"/>
        <option label="acogalvanizado" value="0.00016"/>
        <option label="acocomferrugemleve" value="0.00025"/>
        <option label="acocomgrandesincrustacoes" value="0.007"/>
        <option label="acocomcimentocentrifugado" value="0.0001"/>
        <option label="acorevestidocomasfalto" value="0.0006"/>
        <option label="acorevesmaltevinilepoxi" value="0.00006"/>
        <option label="aluminio" value="0.000004"/>
        <option label="concretomuitorugoso" value="0.002"/>
        <option label="concretorugoso" value="0.0005"/>
        <option label="concretoliso" value="0.0001"/>
        <option label="concretomuitoliso" value="0.00006"/>
        <option label="concretoalisadocentrifugado" value="0.0003"/>
        <option label="concretolisoformasmetalicas" value="0.00012"/>
        <option label="ferrofundidoasfaltado" value="0.000122"/>
        <option label="ferrogalvanizado" value="0.00015"/>
        <option label="ferrofundnaorevestidonovo" value="0.0005"/>
        <option label="ferrofundcomferrugemleve" value="0.0015"/>
        <option label="ferrofundcimcentrifugado" value="0.0001"/>
        <option label="fibrocimento" value="0.0001"/>
        <option label="manilhaceramica" value="0.0003"/>
        <option label="lataocobre" value="0.000007"/>
        <option label="plasticos" value="0.00006"/>
        <option label="rocha(galeria)naorevestida" value="0.35"/>
    </rugosidadeValor>
</tabela-valores>

```

## Arquivo: ApplicationResources.properties

# Define de forma centralizada os labels requisitados

nome.aplicacao=SIMULADOR PARA DETERMINAR VALORES DE DIÂMETRO, VAZÃO E PARA FLUÍDOS INCOMPRESSÍVEIS

viscosidade.label=Viscosidade Cinemática de alguns líquidos (¶):

rugosidade.label=Rugosidade Equivalente de alguns materias de tubos (k):

perda.label=Perda de Carga Unitária (mca/m) :

resultado.label=Resultado: &nbsp;

vazao.label=Vazão no Tubo (m<sup>3</sup>/s):

diametro.label=Diâmetro Interno do Tubo (m):

valor.mascara=0.00000

## Arquivo: util.java

```
package bean;
```

```
import java.text.DecimalFormat;
```

```
/**
 *
```

```

* Classe utilitaria usa para definição de valores e formatação de valores
*
*/
public class Util {

    public final static String FORMATO = "0.00000000";
    public final static String FORMATO_TABELA = "{0,number,0.00000000}";
    public final static double NUM_CONST1 = 0.7267;
    public final static double NUM_CONST2 = 1.7748;
    public final static double NUM_CONST3 = 0.27;
    public final static double DN = 1.0;
    public final static double GRAVIDADE = 9.81;

    /**
    * Formata o número
    * @param num Número a ser formatado
    * @return Número formatado
    */
    public static String formatar( double num ){
        DecimalFormat decFormat = new DecimalFormat(FORMATO);
        return decFormat.format( num );
    }
}

```

## 8. RESULTADOS OBTIDOS

A seguir, apresentam-se as principais telas com os resultados de algumas simulações desenvolvidas. Abaixo, está a tela de entrada do sistema. Nela, o usuário pode escolher um dos três módulos disponíveis para início dos cálculos/simulações: diâmetro, vazão ou perda de carga unitária.

Também podemos perceber que os valores da viscosidade dos fluidos (mais a sua temperatura) e a rugosidade do conduto aparecem em caixa tipo “combo box” que foram apresentadas nas tabelas 1 e 2 anteriormente.

Para efetivação do cálculo, é necessária a entrada dos valores da vazão e da perda de carga unitária iniciais nos campos específicos conforme mostrado na fig. 7. Após estes procedimentos o cálculo poderá ser executado clicando-se em “calcular”.

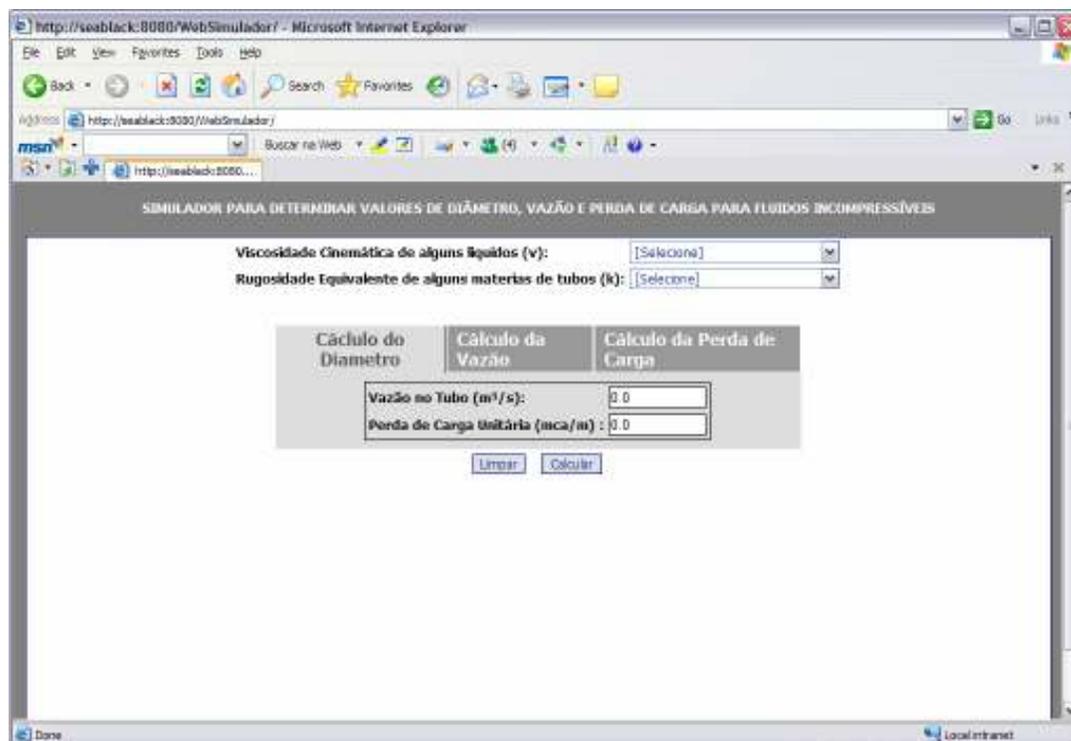


Fig. 7 – Tela de entrada usando o browser Microsoft Internet Explorer 6.0 – Módulo de cálculo do diâmetro

Percebe-se aqui, que é possível a comparação do cálculo do diâmetro com diferentes saídas. Neste exemplo, o primeiro cálculo foi efetuado usando a água como fluido inicial de entrada à temperatura de 20° Celsius, conduto forçado de aço comercial, e os valores de entrada para a vazão de 6,7 m³/s e a perda de carga unitária de 1,27550 m.c.a./m. O resultado obtido na primeira simulação foi  $D = 0,51482202$  m (fig. 8).

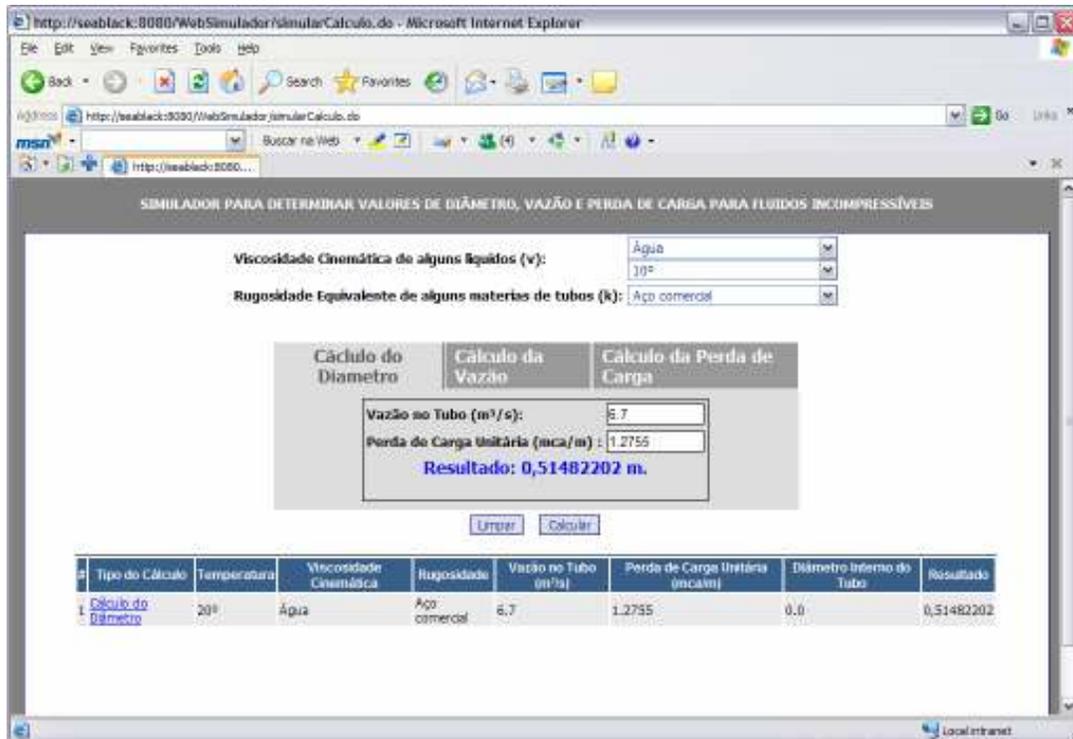


Fig. 8 – Módulo de cálculo do diâmetro

Na simulação seguinte, foi alterado o parâmetro de rugosidade, usando-se concreto rugoso como conduto forçado. Neste caso o resultado foi  $D = 0,56104674$  (fig. 9). Os demais valores não foram alterados.

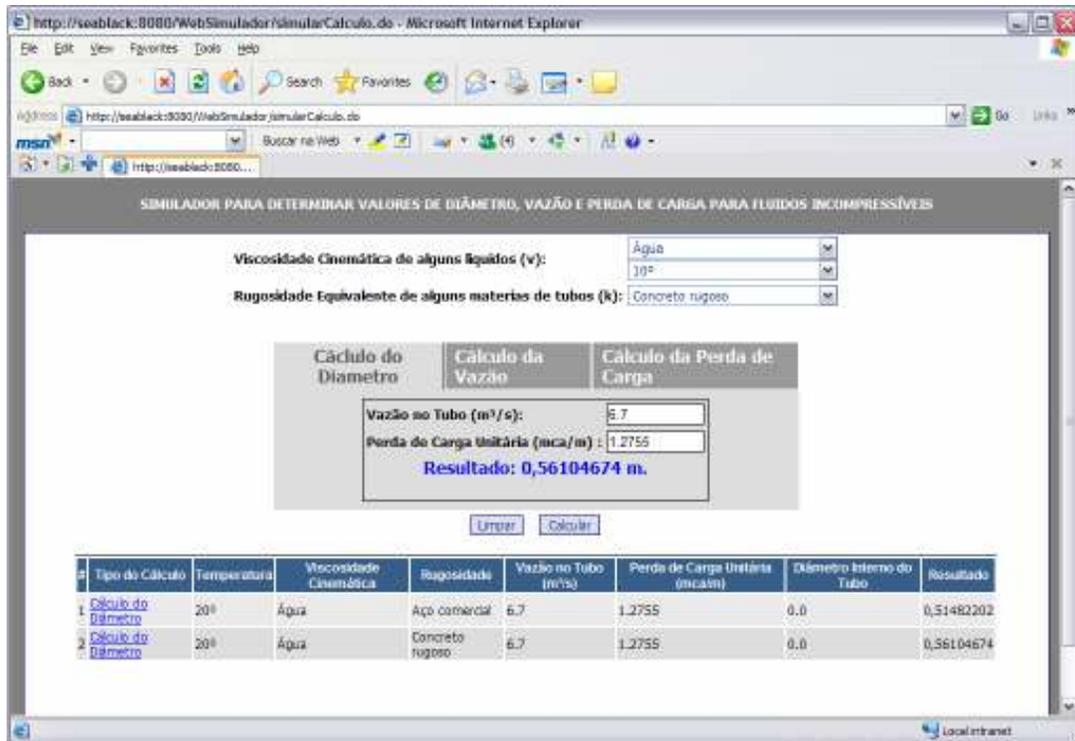


Fig. 9 – Simulação do cálculo do diâmetro - usando concreto rugoso.

Na terceira simulação foi utilizado um outro tipo de fluido, neste caso a glicerina, alterando-se assim o valor da viscosidade e mantendo-se fixos as demais entradas. Para esta simulação, o novo valor de diâmetro foi de 0,61779766 m (fig. 10).

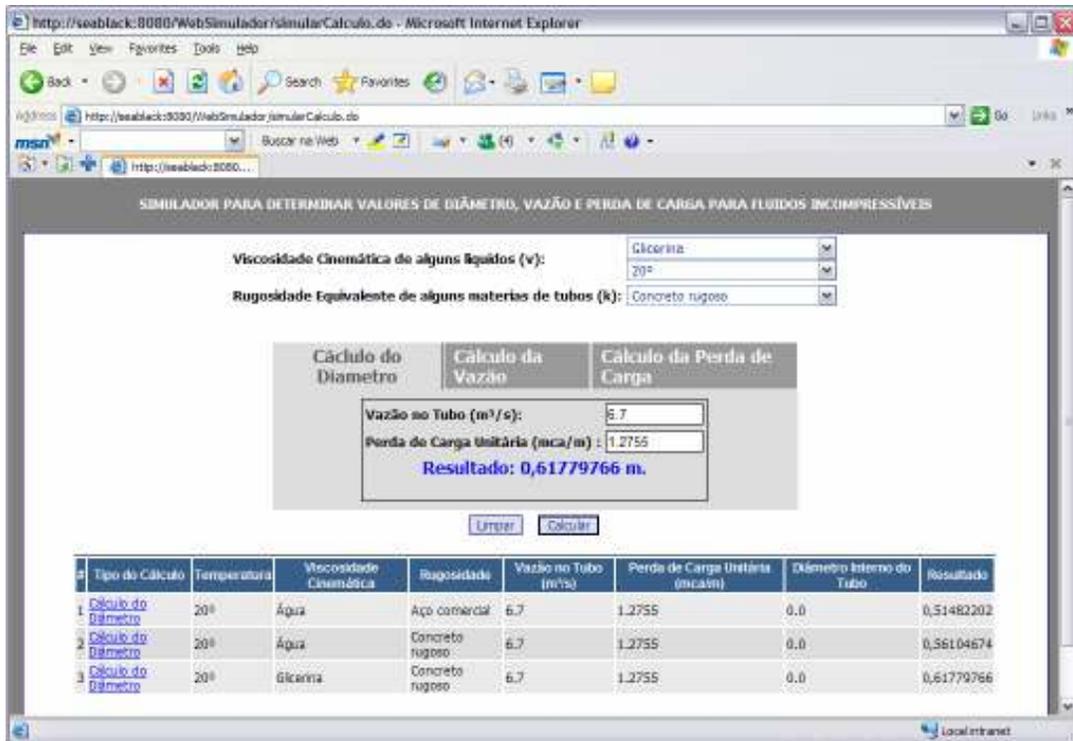


Fig. 10 – Simulação do cálculo do diâmetro - usando glicerina

O simulador monta uma estrutura em tabela para a consolidação dos resultados obtidos com os valores iniciais imputados. Assim, podemos comparar os resultados obtidos:

No.	Tipo do Cálculo	Temperatura	Viscosidade Cinemática	Rugosidade	Vazão no Tubo (m³/s)	Perda de Carga Unitária (mca/m)	Diâmetro Interno do Tubo	Resultado
1	Cálculo do Diâmetro	20°	Água	Aço comercial	6,70000	1,2755		0,51482202
2	Cálculo do Diâmetro	20°	Água	Concreto rugoso	6,70000	1,2755		0,56104674
3	Cálculo do Diâmetro	20°	Glicerina	Concreto rugoso I	6,70000	1,2755		0,61779766

Tabela 3 – Resultado consolidado referente ao cálculo do diâmetro

Este mesmo conceito é aplicado no cálculo da vazão e da perda de carga unitária (fig 11 e 12). Não há restrição da quantidade de cálculos que podem ser feitos pelo sistema. Todos os cálculos serão armazenados na tabela de valores consolidados (tabela 3) e apresentados na tela enquanto o “browser” estiver aberto. Se necessário, os dados podem ser exportados (usando-se o recurso CTRL+C/CTRL+V) para um software de diagramação de gráficos e relatórios gerenciais mais específicos.

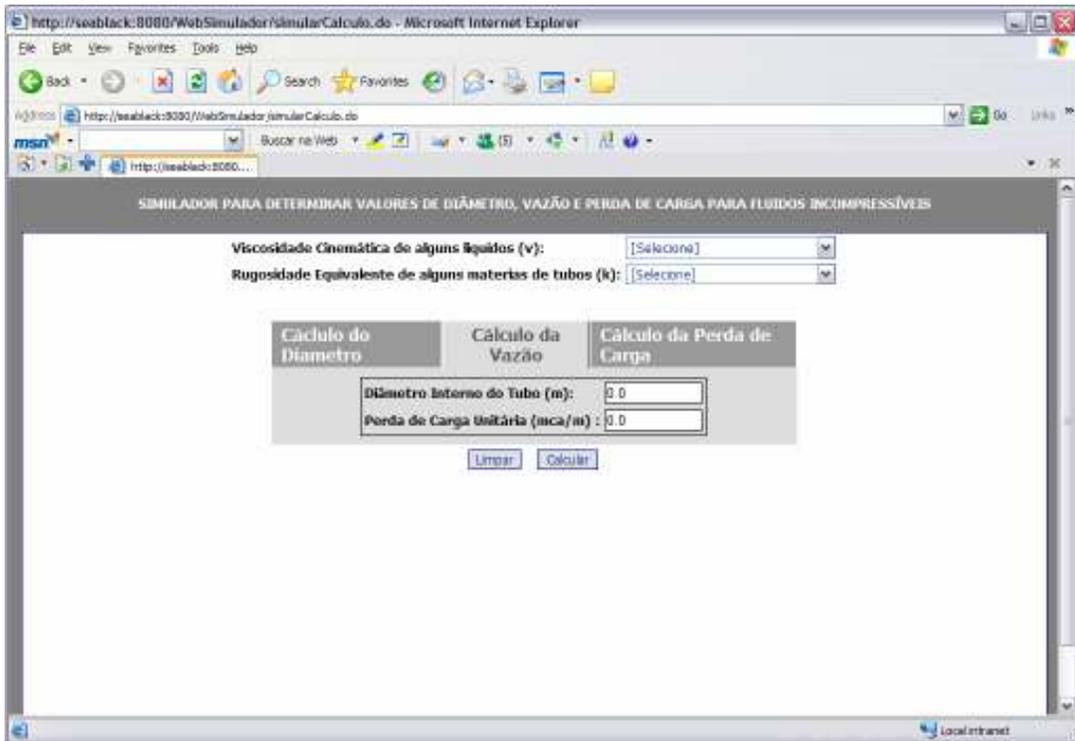


Fig. 11 – Tela do simulador para o cálculo da vazão.

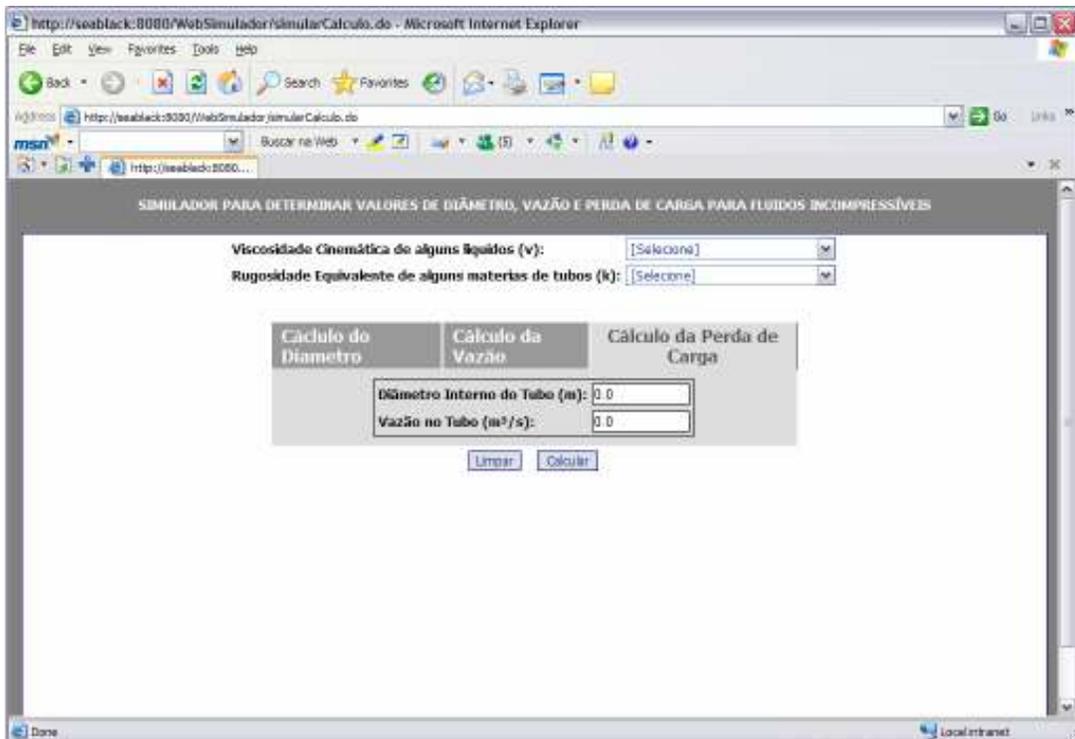


Fig. 12 – Tela do simulador para o cálculo da perda de carga unitária

A seguir, um exemplo para a simulação do cálculo da vazão (fig. 13) com diferentes valores de condutos alterando-se, assim, os valores do parâmetro de rugosidade do conduto:

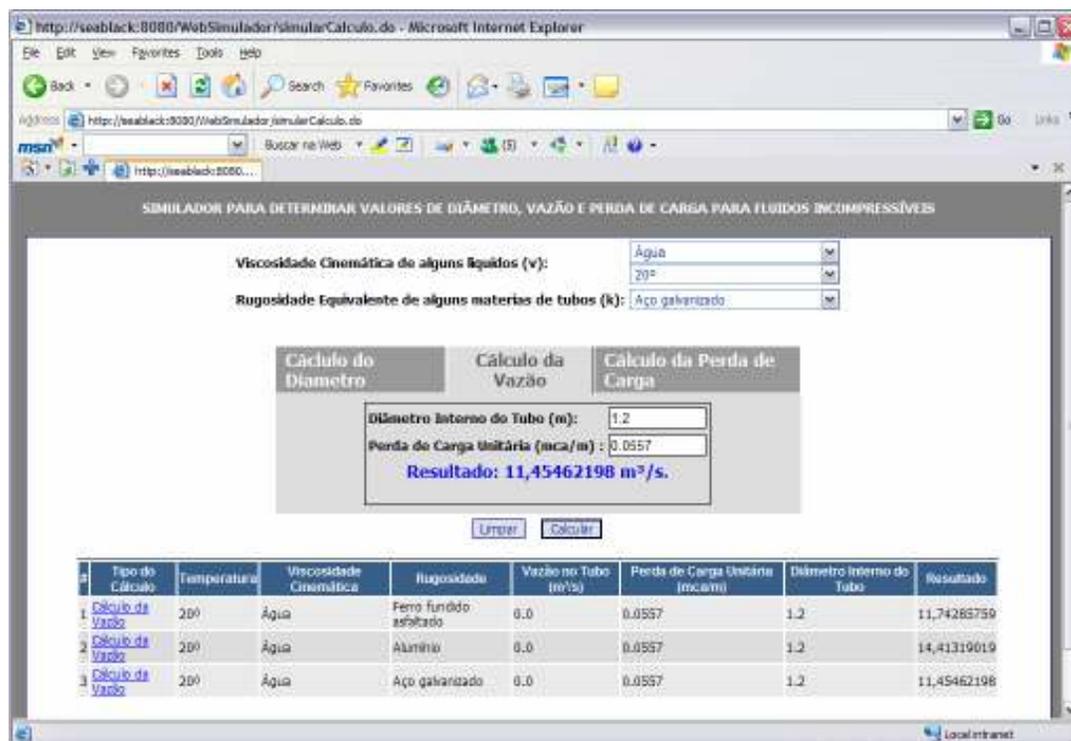


Fig. 13 – Exemplo de cálculo da vazão para diferentes simulações de condutos forçados

Aqui simulações para o cálculo da perda de carga unitária para diferentes valores de viscosidade (fig. 14).

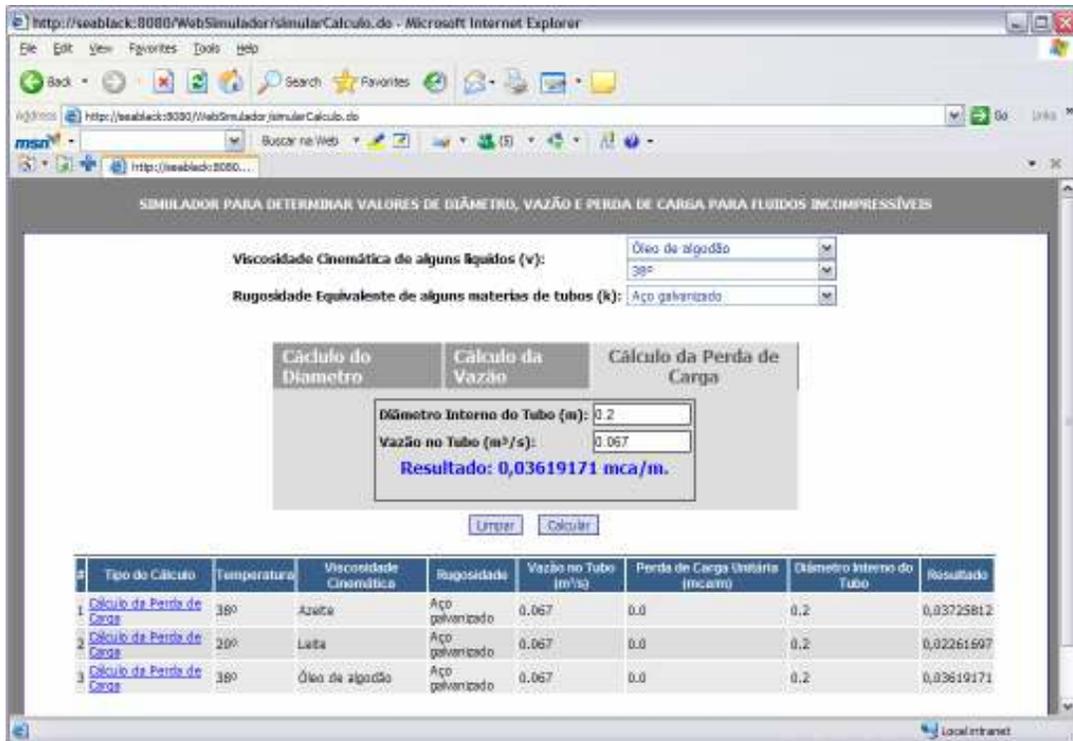


Fig. 14 – Simulação de cálculo de perda de carga unitária para diferentes valores de viscosidade do fluido.

Também, podemos simular cálculos com diferentes valores para vazão, diâmetro interno ou mesmo perda da carga unitária para qualquer um dos três módulos de cálculo, bastando para isso entrar com os valores iniciais nos respectivos campos.

## 9. CONFIRMAÇÃO DOS RESULTADOS ESPERADOS

A seguir serão apresentadas as três formas de cálculos utilizados no sistema (cálculo do diâmetro, cálculo da vazão e o cálculo da perda de carga unitária) empregando-se algumas metodologias de resolução para obtenção dos resultados.

Para validação dos resultados, cada cálculo descrito acima é tipificado em um problema cujo resultado é conhecido. Os problemas propostos foram apresentados por Luiz A Camargo [6] em seu texto “Análise de escoamento em condutos forçados”.

Para cada problema, o cálculo é efetuado usando-se, primeiramente, a forma manual (tradicional) de solução e usando as equações básicas de mecânica dos fluidos. Depois é apresentado o problema resolvido [6], fazendo uso das equações implícitas de Colebrook-White para o cálculo. Em seguida, é apresentado o resultado baseado no sistema computacional desenvolvido, também usando as equações implícitas de Colebrook-White. O resultado final dos três métodos é apresentado ao final de cada problema para comparações dos resultados obtidos.

O objetivo aqui é demonstrar que a precisão dos resultados com a utilização do sistema computacional é a mesma quando utilizado o método manual junto com as equações implícitas de Colebrook-White e também mostrar que, quando os cálculos são feitos com as equações básicas, há uma grande divergência no resultado final

### 9.1 – Perda de Carga Desconhecida

Num conduto cilíndrico de fibrocimento de comprimento igual a 100 m, de diâmetro igual a 0,20 m e de rugosidade equivalente uniforme de  $10^{-4}$  m, está escoando água à temperatura de 20 °C ( $\nu = 10^{-6}$  m<sup>2</sup>/s) com a vazão de 62,8 litros/s. Pede-se a perda de carga.

Dados do problema:

$$\nu = 0,000001 \text{ m}^2/\text{s}$$

$$k = 0,0001 \text{ m}$$

$$D = 0,20 \text{ m}$$

$$Q = 0,0628 \text{ m}^3/\text{s}$$

$$L = 100 \text{ m}$$

$$g = 9,81 \text{ m/s}^2$$

Na solução Tradicional, usamos a equação 4 para o cálculo da perda de carga:

$$h_l = f \frac{LV^2}{2Dg}$$

Conforme demonstrado a seguir:

É necessário descobrir o número de Reynolds para achar o valor do atrito ( $f$ ) usando a equação 3:

$$R_e = \frac{4Q}{\pi D \nu}, \text{ desta forma temos: } R_e = \frac{4 \cdot 0,0628}{3,14 \cdot 0,20 \cdot 0,000001} = \frac{0,2512}{0,000000628} = 400.000$$

Consultando-se o diagrama de Moody (Apêndice A), obtém-se o coeficiente de atrito relacionando-se o resultado obtido do número de Reynolds com a rugosidade dada. Pela tabela:  $f = 0,015$

A velocidade pode ser obtida através da equação 2:  $R_e = \frac{VD}{\nu}$

$$V = \frac{R_e \nu}{D} = \frac{400000 \cdot 0,000001}{0,20} = 2 \text{ m/s}$$

Agora os valores na equação 4 são substituídos:

$$h_l = f \frac{LV^2}{2Dg} = 0,015 \frac{100 \cdot (2)^2}{2 \cdot 0,20 \cdot 9,81} = 1,5290 \text{ m.c.a.}$$

Comparando-se com o cálculo de um problema já conhecido [6] que emprega o método iterativo na equação de Colebrook-White.

Para determinar a perda de carga, utiliza-se a equação 42:

$$J_{n+1} = \frac{0,2026Q^2}{D^3g} \left[ -\log_{10} \left( 0,27 \frac{k}{D} + \frac{1,7748\nu}{\sqrt{gD^3J_n}} \right) \right]^{-2}$$

$$J_{n+1} = \frac{0,2026 \cdot 0,0628^2}{0,20^3 \cdot 9,81} \left[ -\log_{10} \left( 0,27 \frac{0,0001}{0,20} + \frac{1,7748 \cdot 0,000001}{\sqrt{0,20^3 \cdot 9,81 \cdot J_n}} \right) \right]^{-2}$$

$$J_{n+1} = 0,25453 \cdot [-\log_{10}(0,000135 + 0,000006335 \cdot J_n^{-0,5})]^{-2}$$

Neste exemplo será usada a precisão de 4 casas decimais, ou seja, o valor encontrado será considerado como aceito quando o erro for na ordem de  $10^{-4}$ , isto é, 10 milésimos, ou 0,01%. Como valor inicial será adotado  $J = 1$ .

Então:

Como valor inicial tem-se  $J_0 = 1,0000$

Iteração 1: com  $J_0 = 1,0000$  tem-se  $J_1 = 0,0185$

Iteração 2: com  $J_1 = 0,0185$  tem-se  $J_2 = 0,0182$

Iteração 3: com  $J_2 = 0,0182$  tem-se  $J_3 = 0,0182$

Aqui interrompe-se o cálculo, uma vez que  $J_3$  é igual a  $J_2$ , o que significa que para esta tolerância o valor procurado convergiu com duas iterações. Logo, como resposta tem-se:

$J = 0,0182$  m.c.a./m

O cálculo aqui obtido é o da perda unitária. Para ser encontrado a perda total, basta multiplicar a perda unitária pelo comprimento, isto é,  $h_f = J \cdot L = 0,01826 \times 100 = 1,826$  m.c.a.

A seguir, o mesmo cálculo usando-se o sistema computacional proposto:

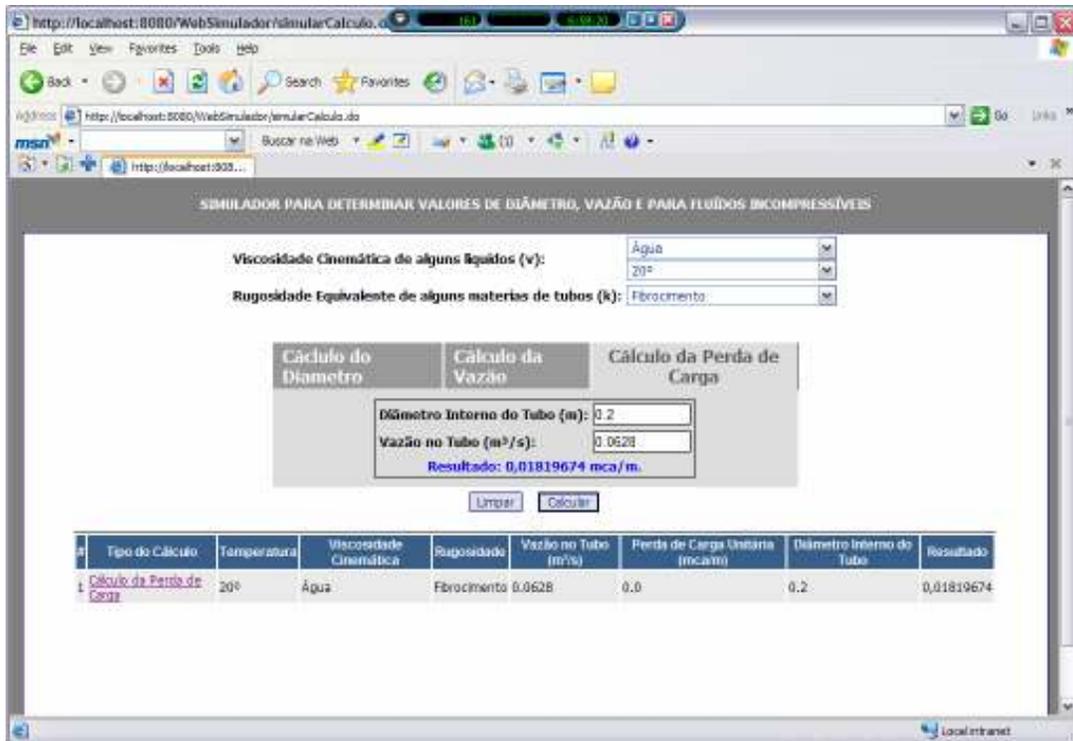


Fig. 15 – Tela do sistema para cálculo da perda de carga unitária

Para o cálculo da perda de carga unitário é obtido o valor de  $J = 0,01819674 \text{ m.c.a./m}$

Logo  $h_f = 0,01819674 \times 100 = 1,819674 \text{ m.c.a.}$

Comparando os três resultados:

Resultado manual:  $h_f = 1,5290 \text{ m.c.a.}$

Resultado manual com método iterativo  $h_f = 1,8260 \text{ m.c.a.}$

Resultado computacional com mét. iterativo  $h_f = 1,8196 \text{ m.c.a.}$

## 9.2 – Diâmetro Desconhecido

Um conduto cilíndrico, de ferro fundido com cimento centrifugado, comprimento 360 m, rugosidade equivalente de  $10^{-4}$  m, conduz água à temperatura de  $20\text{ }^{\circ}\text{C}$  ( $n = 10^{-6}$  m<sup>2</sup>/s), com vazão de 12 m<sup>3</sup>/s e sob a diferença de carga piezométrica nas seções extremas de 3,9 m. Pede-se o diâmetro. Dado:  $Re = 8780000$

Solução:

Dados do problema:

$$\begin{aligned} n &= 0,000001 \text{ m}^2/\text{s} \\ k &= 0,0001 \text{ m} \\ Q &= 12 \text{ m}^3/\text{s} \\ h_f &= 3,9 \text{ m.c.a.} \\ L &= 360 \text{ m} \\ J &= hf/L = 0,0108333 \text{ m.c.a./m} \\ g &= 9,81 \text{ m/s}^2 \end{aligned}$$

Na solução Tradicional, usamos a equação 3, isolando-se o diâmetro:

$$R_e = \frac{4Q}{\pi D v} \Rightarrow D = \frac{4Q}{\pi R_e v} = \frac{4 \cdot 12}{3,14 \cdot 8780000 \cdot 0,000001} = 1,7410 \text{ m}$$

Comparando-se com o cálculo de um problema já conhecido que emprega o método iterativo na equação de Colebrook-White:

Para determinar o diâmetro utiliza-se a eq. 4:

$$\begin{aligned} D_{n+1} &= \frac{0,7267 Q^{0,4}}{(Jg)^{0,2}} \left[ -\log_{10} \left( 0,27 \frac{k}{D_n} + \frac{1,7748 v}{\sqrt{Jg D_n^3}} \right) \right]^{-0,4} \\ D_{n+1} &= \frac{0,7267 \cdot 12^{0,4}}{(0,010833 \cdot 9,81)^{0,2}} \left[ -\log_{10} \left( 0,27 \frac{0,0001}{D_n} + \frac{1,7748 \cdot 0,000001}{\sqrt{0,010833 \cdot 9,81 \cdot D_n^3}} \right) \right]^{-0,4} \\ D_{n+1} &= 3,074284 \cdot [-\log_{10}(0,000027 \cdot D_{n-1} + 0,000005444 \cdot D_n^{-1,5})]^{-0,4} \end{aligned}$$

Será usada neste exemplo a precisão de 4 casas decimais, ou seja, o valor encontrado será dado como aceito quando o erro for na ordem de  $10^{-4}$ , isto é, 10 milésimos, ou 0,01%.

Como valor inicial será adotado  $D = 1$ .

Então:

Como valor inicial tem-se  $D_0 = 1,0000$

Iteração 1: com  $D_0 = 1,0000$  tem-se  $D_1 = 1,6861$  (isto é,  $D_n = 1,0000$  resulta em  $D_{n+1} = 1,6861$ )

Iteração 2: com  $D_1 = 1,6861$  tem-se  $D_2 = 1,6508$

Iteração 3: com  $D_2 = 1,6508$  tem-se  $D_3 = 1,6521$

Iteração 4: com  $D_3 = 1,6521$  tem-se  $D_4 = 1,6521$

Neste ponto o cálculo pode ser interrompido já que  $D_4$  é igual a  $D_3$ . Isto significa que para esta tolerância o valor procurado já convergiu na terceira iteração. Logo, como resposta tem-se:

$D = 1,6521$  m

Usando o sistema para efetuar o cálculo usando-se os valores acima:

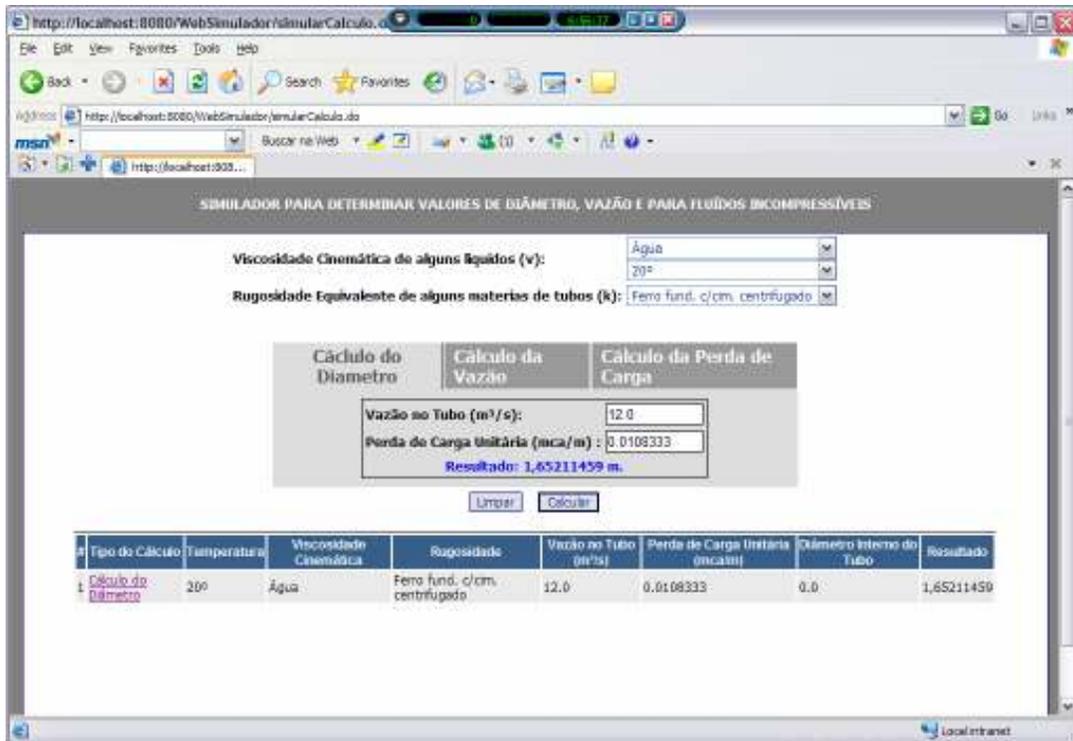


Fig. 16 – Tela do sistema para cálculo do diâmetro.

Comparando os três resultados:

Resultado manual:  $D = 1,7410 \text{ m}$

Resultado manual com método iterativo  $D = 1,6521 \text{ m}$

Resultado computacional com mét. Iterativo  $D = 1,6521 \text{ m}$

### 9.3 – Vazão Desconhecida

Num conduto cilíndrico longo de concreto alisado centrifugado de diâmetro igual a 0,10 m e rugosidade equivalente de  $3,0 \cdot 10^{-4}$  m, está escoando água à temperatura de  $37^\circ\text{C}$  ( $\nu = 7,0 \cdot 10^{-7} \text{ m}^2/\text{s}$ ) com perda de carga unitária de  $J = h_f/L = 0,0115 \text{ m.c.a./m}$ . Pede-se a vazão.

Dados do problema:

$$\nu = 7,0 \cdot 10^{-7} \text{ m}^2/\text{s}$$

$$k = 0,0003 \text{ m}$$

$$D = 0,10 \text{ m}$$

$$J = h_f/L = 0,0115 \text{ m.c.a./m}$$

$$g = 9,81 \text{ m/s}^2$$

$$V = 0,91 \text{ m/s}$$

Solução tradicional

Usando a equação 3 é obtido a vazão:

$$R_e = \frac{4Q}{\pi D \nu} \Rightarrow Q = \frac{R_e \pi D \nu}{4}$$

É necessário descobrir o valor do número de Reynolds para o cálculo da vazão. Podemos obter pela equação 4 isolando-se o fator de atrito e consultado o diagrama de Moody junto com o valor da rugosidade dado:

$$R_e = \frac{VD}{\nu} = \frac{0,91 \cdot 0,10}{7,0 \cdot 10^{-7}} = 130000$$

$$\text{Substituindo na equação 3: } Q = \frac{R_e \pi D \nu}{4} = \frac{130000 \cdot 3,14 \cdot 0,10 \cdot 7,0 \cdot 10^{-7}}{4} = 0,00714 \text{ m}^3/\text{s}$$

Comparando-se com o cálculo de um problema já conhecido que emprega o método iterativo na equação de Colebrook-White:

Para determinar a vazão utiliza-se a equação 41:

$$Q = -2,2214 \sqrt{JgD^5} \log_{10} \left( 0,27 \frac{k}{D} + \frac{1,778\nu}{\sqrt{JgD^3}} \right)$$

$$Q = -2,2214 \sqrt{0,0115 \cdot 9,81 \cdot 0,10^3} \cdot \log_{10} \left( 0,27 \frac{0,0003}{0,10} + \frac{1,7748 \cdot 7 \cdot 0,0000001}{\sqrt{0,0115 \cdot 9,81 \cdot 0,10^3}} \right)$$

Como a equação é explícita, o cálculo aqui é imediato:  $Q = 0,0071560 \text{ m}^3/\text{s}$

Usando o simulador para efetuar o cálculo usando-se os valores acima:

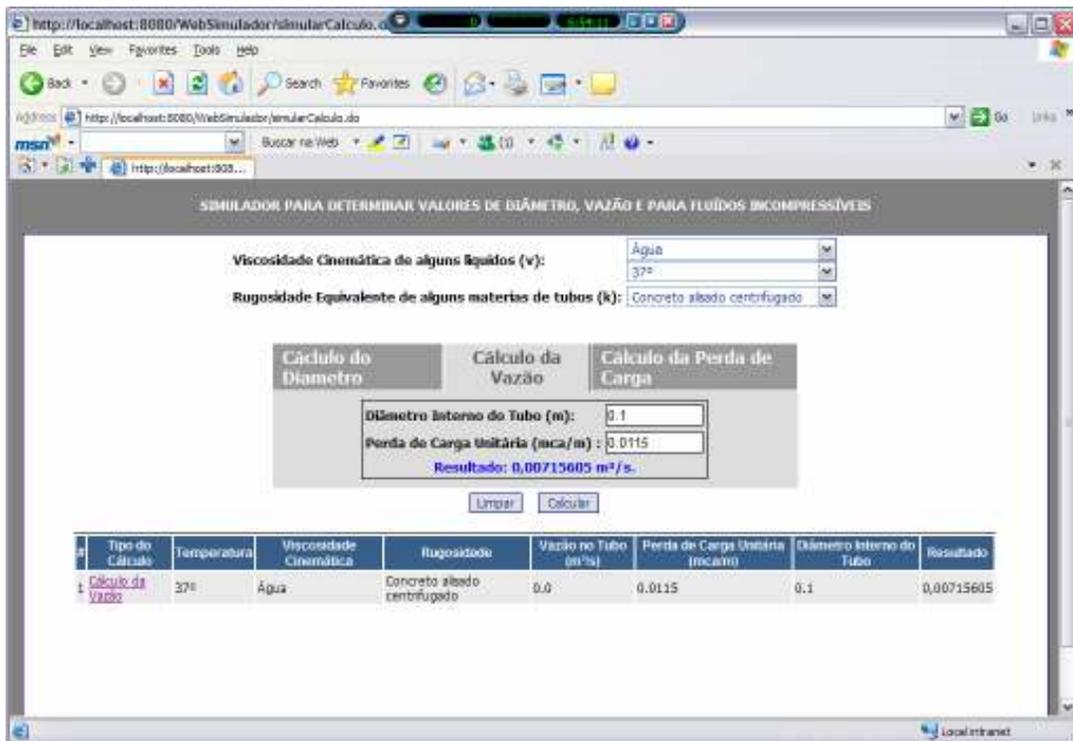


Fig. 17 – Tela do sistema para cálculo da vazão

Comparando os três resultados:

Resultado manual:  $Q = 0,0071435 \text{ m}^3/\text{s}$

Resultado manual com método iterativo  $Q = 0,0071560 \text{ m}^3/\text{s}$

Resultado computacional com mét. iterativo  $Q = 0,0071560 \text{ m}^3/\text{s}$

## CONCLUSÃO

Nesta monografia, procurou-se transmitir de forma direta a maioria dos conceitos utilizados em mecânica dos fluidos aplicada a condutos forçados em regime permanente com a finalidade da construção de um sistema computacional amigável e de utilização flexível.

Toda a teoria mencionada foi, desta forma, traduzida numa linguagem de programação que pudesse ser seguida com estes conceitos para que o usuário do sistema consiga fazer uso do sistema sem necessidade do entendimento de teorias e fórmulas complexas que regem os conceitos de mecânica dos fluidos. Como consequência, obteve-se um ganho direto na aplicação das teorias e dos conceitos matemáticos que foram transformados numa metodologia sistemática e repetitiva, através do método de cálculo iterativo para solução de equações implícitas e das equações de Colebrook-White e Darcy-Weisbach, que é ideal para ser aplicado em sistemas computacionais devido a grande facilidade desses ambientes na execução de tarefas rotineiras.

Uma grande preocupação durante o desenvolvimento deste trabalho, foi o uso do melhor método matemático para realização dos cálculos com o maior grau de precisão possível. Neste caso, observou-se que as equações diferem-se no resultado apenas 0,02% dos casos estudados. Outro fator que contribuiu para que o resultado esperado seja satisfatório, é a possibilidade da comparação de resultados em tempo real fazendo uso do simulador desenvolvido. A possibilidade de se acrescentar novas funcionalidades ao sistema (novos módulos de cálculos, emissão de relatórios e gráficos analíticos, por exemplo) são praticamente infinitas, já que o sistema é facilmente adaptável quando surgirem novas necessidades. Isso se deve muito ao fato de que o

sistema ser planejado, antes mesmo do início da programação do código-fonte, para ser flexível e portátil.

Este é um assunto com grande possibilidade de crescimento. A adoção de equações que contribuam para os cálculos de perdas localizadas ainda não consta neste trabalho e poderão, com certeza, ser exploradas pelos futuros estudantes sobre este tema. Outra possibilidade é a adição de módulos para outros tipos de cálculos, como, por exemplo, para fluidos compressíveis ou mesmo fluidos em condutos abertos.

Assim, por mais simples ou compactas que possam ser estas equações implícitas, as mesmas requerem algum esforço computacional com operações matemáticas de potenciação, radiciação, logarítmicas, etc. Contudo, tendo em vista as elevadas velocidades dos processadores dos computadores atuais, praticamente será imperceptível a diferença no esforço computacional do cálculo feito com uma equação implícita ou mesmo com uma equação explícita. Acredita-se que nosso principal objetivo foi alcançado, pois o usuário de hidráulica aplicada aos fluidos em condutos forçados dispõe de uma ferramenta simples e funcional, que permite ajudá-lo nos cálculos dos diversos parâmetros como vazão, perda de carga ou diâmetro das tubulações, fazendo-se uso de um sistema computacional como ferramenta.

## BIBLIOGRAFIA

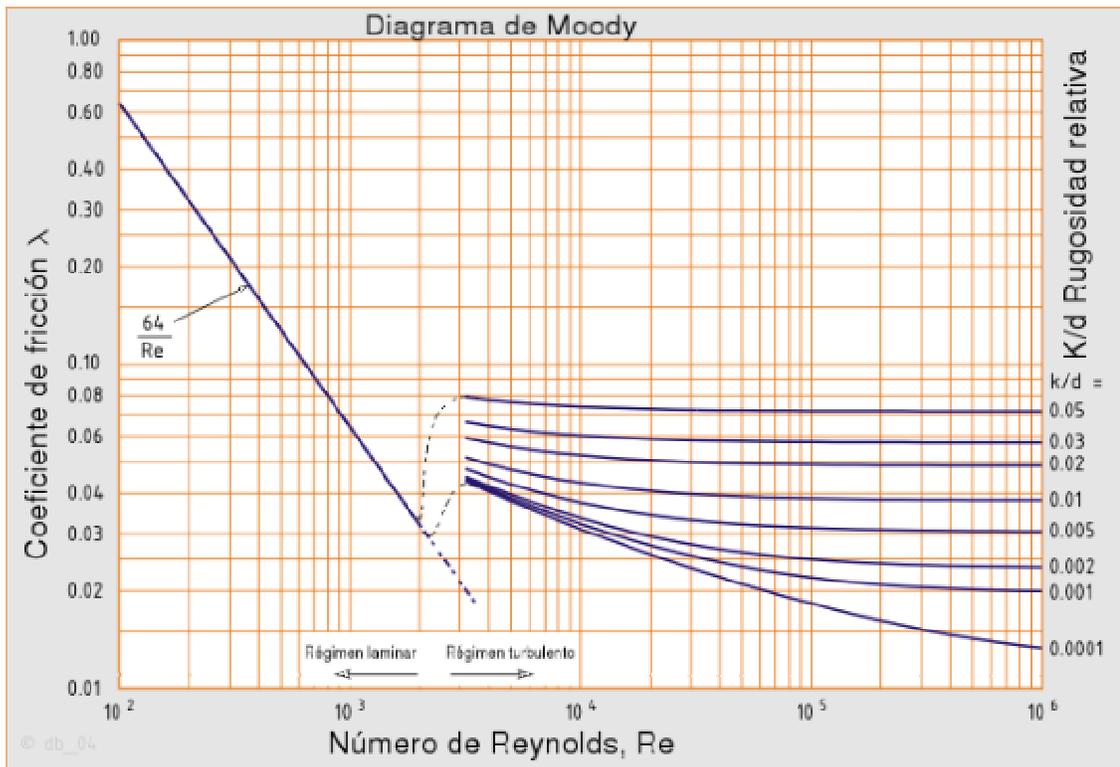
- [1] ADIDA, B. It all starts at the server. **IEEE Internet Computing**, Los Alamitos, Feb. 1997. p. 75.
- [2] BARR, D.I.H. **New forms of equations for the correlation of pipe resistance data**, Proc. Instn. Civ. Engrs., Part 2, 1972. pp. 383-390.
- [3] BARR, D.I.H. **Two additional methods of direct solution of the Colebrook-White function**, Proc. Instn. Civ. Engrs., Part 2, 59, 1975. pp. 827-835.
- [4] BARR, D.I.H. **The transition from laminar to turbulent flow**, Proc. Instn. Civ. Engrs., Part 2, 69, 1980. pp. 555-562.
- [5] BARR, D.I.H. **Solutions of the Colebrook-White function for resistance to uniform turbulent flow**, Proc. Instn. Civ. Engrs., Part 2, 1981. pp. 529-535.
- [6] CAMARGO, Luiz A. **Análise de escoamento em condutos forçados. Uso das equações de Darcy-Weisbach e de Colebrook-White**. 2001. Disponível na internet em <http://paginas.terra.com.br/servicos/hidrotec/condutos.htm>. Acesso em 15 abril, 2006.
- [7] CAMPIONE, M; WALRATH, K. **The Java Tutorial: Object-Oriented Programming for the Internet**. [S.l.]: SunSoft Press, 1996.
- [8] CHEN, J.J.J. **A simple explicit formula for the estimation of pipe friction factor**, Proc. Instn. Civ. Engrs., Part 2, 77, 1984. pp. 49-55.
- [9] CHEN, J.J.J. **Systematic explicit solutions of the Prandtl and Colebrook-White equation for pipe flow**, Proc. Instn. Civ. Engrs., Part 2, 79, 1985. 383-389.

- [10] CHEN, N.H. **An explicit equation for friction factor in pipe**, Ind. Eng. Chem. Fund., A.C.S., Vol. 18, No. 3, 1979. pp. 296-297.
- [11] CHURCHILL, S.W. **Empirical expression for the shear stress in turbulent flow in commercial pipe**, Am. Inst. Ch. Engrs. J., Vol. 19, No. 2, 1973. pp. 375-376.
- [12] COLEBROOK, C.F. **Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws**, J. Instn. Civ. Engrs., 11, 1939. pp. 133-156.
- [13] CHANG, P.I. **Java Web Server Ships**. JavaWorld, IDG Communications, June 1997. Disponível na internet em <http://www.javaworld.com/javaworld/jw-06-1997/jw-06-webserver.html>. Acesso em 15 abril, 2006.
- [14] FOX, R. W., McDonald, A.T., **Introdução a Mecânica dos Fluidos**, 5ª edição, LTC, 1995, pp 217-256.
- [15] GOSLING, J.; JOY, B.; STEELE,G. **The Java Language Specification**. Sun Microsystems, July 1996. Disponível na Internet em [http://java.sun.com/doc/language\\_specification.html](http://java.sun.com/doc/language_specification.html). Acesso em 15 Abril, 2006.
- [16] HAALAND, S.E. (1983) - **Simple and explicit formulas for the friction factor in turbulent pipe flow**, J. Fluids Eng. Trans. ASME, 1983. pp. 89-90.
- [17] JACOBSON, Ivar – UML, **Unified Modeling Language**. Rio de Janeiro, 2000.
- [18] KRAMER, D. **JDK Documentation**. [S.l.]: Sun Microsystems, 1997.
- [19] KRAMER, D. **The Java Platform: A White Paper**. [S.l.]: Sun Microsystems, 1996.
- [20] MALAFAYA-BAPTISTA, M. **Crítérios de explicitação da expressão de Colebrook-White. Novas perspectivas**, Laboratório de Hidráulica, FEUP. 1980

- [21] MOODY, L.F. **An approximate formula for pipe friction factors**, Mech. Engng., New York, 1947. pp. 1005-1006.
- [22] MUNSON, Bruce R.; Young, Donald F; Okiishi, Theodore H.; **Fundamentos da Mecânica dos Fluidos** – Versão SI – Volume I – Edgard Blücher LTDA – 1997.
- [23] NACKAB, J. **Calcul direct, sans itération, de la perte de charge en conduite par la formule de Colebrook**, La Houille Blanche, 1988. pp. 61.
- [24] NEKRASOV, B. **Hydraulics**, Peace Publishers, Moscow, 1968. pp. 95-101.
- [25] ROUND, G.F. **An explicit approximation for the friction factor-Reynolds number relation for rough and smooth pipes**, The Canadian Journal of Chemical Engineering, Vol. 58, 1980. pp. 122-123.
- [26] SWAMEE, P.K, e JAIN, A.K. **Explicit equations for pipe-flow problems**, Journal of the Hydraulics Division, Á.S.C.E., Vol 102, No. HY5, 1976. pp. 657-664.
- [27] WHITE, C.M. e COLEBROOK, C.F. **Fluid friction in roughened pipes**, Proc. Roy. Soc. A., 1937. pp. 367-381.
- [28] WOOD, D.J. **An explicit friction factor relationship**, Civ. Engng, Am. Soc, Cív. Engrs., 36, No. 12, 1966. pp. 60-61.
- [29] WUTKA, M. **Java: Técnicas Profissionais**. [S.l.]: Berkeley, 1997.
- [30] ZIGRANG, D.J. e SYLVESTER, N.D. **Explicit approximations to the solution of Colebrook's friction factor equation**, Am. Inst. Ch. Engrs. J., Vol. 28, No. 3, 1982. pp. 514-515.

## APÊNDICE A – DIAGRAMA DE MOODY<sup>5</sup>

Abaixo está representado o diagrama de Moody que relaciona o coeficiente de atrito ( $f$ ) com a rugosidade relativa ( $k$ ) e o número de Reynolds ( $R_e$ ) para condutos circulares forçados. O diagrama mostra os regimes de escoamento laminar para  $R_e < 2000$  e turbulento para  $R_e > 4000$ . Para  $2000 < R_e < 4000$  o regime é considerado transitório.



The Moody Diagram

<sup>5</sup> Diagrama obtido via Internet do web site: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_Moody](http://es.wikipedia.org/wiki/Diagrama_de_Moody)

## APÊNDICE B – DEMONSTRAÇÃO DAS PRINCÍPIAS EQUAÇÕES UTILIZADAS NOS DIMENSIONAMENTOS E SIMULAÇÕES NUMÉRICAS

### 1. Equação de Darcy-Weisbach

A equação de Darcy-Weisbach é apresentada da seguinte forma:

$$h_f = f \frac{LV^2}{2Dg}$$

Fazendo  $h_f = L.J$  e inserindo a equação da continuidade, temos:

$$\frac{1}{f} = \frac{V^2}{2JDg} \quad \longrightarrow \quad \sqrt{\frac{1}{f}} = \sqrt{\frac{\left(\frac{Q}{A}\right)^2}{2JDg}}$$

$$\frac{1}{\sqrt{f}} = \sqrt{\frac{Q^2}{2JDg\left(\frac{\pi D^2}{4}\right)^2}} \quad \longrightarrow \quad \frac{1}{\sqrt{f}} = \sqrt{\frac{8Q^2}{J\pi^2 gD^5}}$$

c.q.d.

2. Equação de Colebrook-White utilizada para o cálculo da vazão

Para determinar a vazão usando a fórmula de Colebrook-White são necessárias três equações básicas:

A equação de perda de carga de Darcy-Weisbach:

$$h_f = f \frac{LV^2}{2Dg}$$

A equação de Reynolds

$$R_e = \frac{4Q}{\pi D \nu}$$

E a equação de Darcy-Weisbach modificada (demonstrada acima):

$$\frac{1}{\sqrt{f}} = \sqrt{\frac{8Q^2}{J\pi^2 gD^5}} \Rightarrow \sqrt{f} = \frac{\sqrt{J\pi^2 gD^5}}{\sqrt{8Q^2}}$$

Assim, inserindo estas equações na equação de Colebrook-White obtém-se a equação da vazão:

$$\frac{1}{\sqrt{f}} = -2 \log \left( 0,27 \frac{k}{D} + \frac{2,51}{R_e \sqrt{f}} \right) \longrightarrow \frac{\sqrt{8Q^2}}{\sqrt{J\pi^2 gD^5}} = -2 \log \left( 0,27 \frac{k}{D} + \frac{2,51}{\frac{4Q}{\pi D \nu} \cdot \frac{\sqrt{J\pi^2 gD^5}}{\sqrt{8Q^2}}} \right)$$

$$Q = \frac{-2\pi\sqrt{JgD^5}}{2,8284} \log \left( 0,27 \frac{k}{D} + \frac{2,51}{\frac{4Q}{\pi D \nu} \cdot \frac{\sqrt{JgD^3}}{2,8284Q}} \right)$$

$$Q = -2,2213\sqrt{JgD^5} \log \left( 0,27 \frac{k}{D} + \frac{2,51 \cdot 2,8284}{4 \cdot \sqrt{JgD^3}} \right)$$

$$Q = -2,2213\sqrt{JgD^5} \log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)$$

c.q.d.

### 3. Equação de Colebrook-White para o cálculo do diâmetro

A seqüência para o cálculo do diâmetro é a mesma adotada para vazão,

assim:

$$\frac{\sqrt{8Q^2}}{\sqrt{J\pi^2 gD^5}} = -2 \log \left( 0,27 \frac{k}{D} + \frac{2,51}{\frac{4Q}{\pi D \nu} \cdot \frac{\sqrt{J\pi^2 gD^5}}{\sqrt{8Q^2}}} \right)$$

$$\frac{\sqrt{J\pi^2 gD^5}}{\sqrt{8Q^2}} = -0,5 \log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)^{-1}$$

$$\frac{\pi \sqrt{Jg} \cdot \sqrt{D^5}}{2,8284Q} = -0,5 \log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)^{-1}$$

$$D^{\frac{5}{2}} = \frac{-0,5 \cdot 2,8284Q}{\pi \sqrt{Jg}} \cdot \log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)^{-1}$$

$$\left( D^{\frac{5}{2}} \right)^{0,4} = \left( \frac{-0,5 \cdot 2,8284Q}{\pi \sqrt{Jg}} \cdot \log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)^{-1} \right)^{0,4}$$

$$D = \frac{0,7268Q^{0,4}}{(Jg)^{0,2}} \cdot -\log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)^{-0,4}$$

c.q.d.

4. Equação de Colebrook-White utilizada para o cálculo da perda de carga unitária

Seguindo a mesma orientação das equações anteriores, temos:

$$\frac{\sqrt{8Q^2}}{\sqrt{J\pi^2 gD^5}} = -2 \log \left( 0,27 \frac{k}{D} + \frac{2,51}{\frac{4Q}{\pi D \nu} \cdot \frac{\sqrt{J\pi^2 gD^5}}{\sqrt{8Q^2}}} \right)$$

$$\frac{\sqrt{J\pi^2 gD^5}}{\sqrt{8Q^2}} = -0,5 \log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)^{-1}$$

$$\sqrt{J} = \frac{-0,5 \cdot Q \cdot 2,8284 \log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)^{-1}}{\pi \sqrt{gD^5}}$$

$$(\sqrt{J})^2 = \left( \frac{-0,5 \cdot Q \cdot 2,8284 \log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)^{-1}}{\pi \sqrt{gD^5}} \right)^2$$

$$J = \frac{-0,2028 \cdot Q^2 \log \left( 0,27 \frac{k}{D} + \frac{1,7748 \cdot \nu}{\sqrt{JgD^3}} \right)^{-2}}{gD^5}$$

c.q.d.