

**Modelagem matemática e simulação computacional
para análise de dispersão de poluentes em um trecho
do Rio Paraíba do Sul**

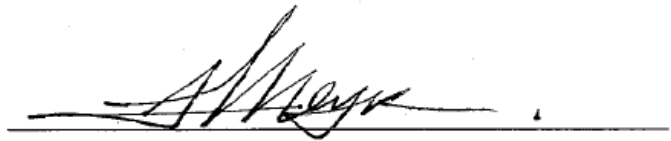
José Ricardo Ferreira de Almeida
joserickardo.almeida14@gmail.com

Campinas-SP, 28 de junho de 2010

Modelagem matemática e simulação computacional para análise de dispersão de poluentes em um trecho do Rio Paraíba do Sul

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por **José Ricardo Ferreira de Almeida** e aprovada pela comissão julgadora.

Campinas, 08 de junho de 2010.



Prof. Dr. João Frederico da Costa Azevedo Meyer
Orientador

Banca Examinadora:

- 1 - Dr. João Frederico da Costa Azevedo Meyer
- 2 - Dr. Vera Lúcia da Rocha Lopes
- 3 - Dr. Rosane Ferreira de Oliveira

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do Título de Mestre em Matemática.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Maria Fabiana Bezerra Müller – CRB8 / 6162

Almeida, José Ricardo Ferreira de

AL64m Modelagem matemática e simulação computacional para análise de dispersão de poluentes em um trecho do Rio Paraíba do Sul/José Ricardo Ferreira de Almeida-- Campinas, [S.P. : s.n.], 2010.

Orientador : João Frederico da Costa Azevedo Meyer

Dissertação (mestrado profissional) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1.Biomatemática. 2.Ecologia - Matemática. 3.Impacto ambiental. 4.Diferenças finitas. 5.Algoritmos de computador. I. Meyer, João Frederico da Costa Azevedo. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Título em inglês: Mathematical models and computer simulation for analysis of pollutant dispersion in a stretch of the Paraíba do Sul River

Palavras-chave em inglês (Keywords): 1. Biomathematics. 2. Ecology – Mathematics. 3.Environmental impact statements. 4. Finite differences. 5. Computer algorithms.

Área de concentração: Matemática Aplicada

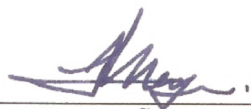
Titulação: Mestre em Matemática

Banca examinadora: Prof. Dr. João Frederico da Costa Azevedo Meyer (IMECC-UNICAMP)
Profª. Drª. Vera Lúcia da Rocha Lopes (IMECC-UNICAMP)
Profª. Drª. Rosane Ferreira de Oliveira (UFRJ)

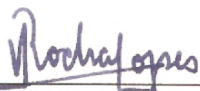
Data da defesa: 08/06/2010

Programa de Pós-Graduação: Mestrado Profissional em Matemática

**Dissertação de Mestrado Profissional defendida em 08 de junho de 2010 e
aprovada pela Banca Examinadora composta pelos Profs. Drs.**



Prof. (a). Dr (a). JOÃO FREDERICO DA COSTA AZEVEDO MEYER



Prof. (a). Dr (a). VERA LÚCIA DA ROCHA LOPES



Prof. (a). Dr (a). ROSANE FERREIRA DE OLIVEIRA

Dedicatória

Dedico esse trabalho à minha esposa Claudiane por toda paciência e tolerância que teve comigo durante meus dias de estudos dedicados à realização de um dos meus sonhos.

Aos meus pais pelo incentivo e apoio constante em toda minha trajetória.

Agradecimentos

Agradeço aos professores, orientadores, João Frederico (Joni) e Rosane por terem, pacientemente, orientado a elaboração deste trabalho.

À AMAN - Academia Militar das Agulhas Negras e à UNICAMP, em especial professoras Sueli Irene (UniCamp), Vera Lopes (UniCamp) e Cel. Tércio (AMAN) por terem implantado em 2008, em Resende-RJ, o curso de Mestrado em Matemática.

À FOA/UniFOA - Centro Universitário de Volta Redonda que autorizou minha ausência das atividades a fim de participar do curso de Mestrado.

Ao prof. e amigo Marcus Víncius Faria de Araújo, pelo apoio e incentivo. Sem contar com as conversas onde eu sempre obtinha informações valiosíssimas para elaboração deste trabalho.

Aos amigos de turma pelo incentivo e união estabelecida durante o curso.

Aos professores do programa de mestrado que nos forneceu belíssimas ferramentas matemáticas.

Aos professores assistentes, Agnaldo, Cristiano, Robson, Simão que estavam sempre de prontidão, aptos a nos ajudar, minimizando nossas dificuldades.

Resumo

ALMEIDA, José Ricardo Ferreira de. **Modelagem matemática e simulação computacional para análise de dispersão de poluentes em um trecho do Rio Paraíba do Sul**. Campinas - SP: Universidade Estadual de Campinas, 2009. Dissertação apresentada como requisito parcial para obtenção do Título de Mestre em Matemática.

Este trabalho usa o modelo clássico de Difusão-Advecção-Reação para simular o comportamento evolutivo bidimensional de manchas de poluentes em um domínio aquático. Em função da ausência de possibilidade de solução analítica (primordialmente pelo tipo de domínio em que se estuda o referido problema), recorre-se a um método de aproximação baseado em diferenças finitas centradas tanto para as variáveis espaciais quanto para a variável temporal (esta, via uso adequadamente explicitado do método de Crank-Nicolson). As condições de contorno são definidas com foco na situação real, considerando a absorção de poluentes nas margens em trechos determinados em função de seu comportamento.

O trecho de rio considerado é o do rio Paraíba do Sul na região de Volta Redonda, RJ, que se caracteriza por ser um trecho reto do referido corpo aquático. Adota-se, como perfil de velocidades, a parábola devida a Poiseuille [11], o que afeta a montagem do sistema de diferenças finitas. Especial atenção é dedicada à precisão numérica resultante da estratégia de aproximação, bem como à possibilidade de visualizar qualitativamente os resultados, criando, assim, um recurso de software que pode ser usado em muitas outras situações minimamente semelhantes de contaminação hídrica sistemática ou emergencial.

Palavras-chave: Biomatemática, Ecologia Matemática, Impacto Ambiental, Método de Diferenças Finitas, Método de Crank-Nicolson, Algoritmos Computacionais.

Abstract

ALMEIDA, José Ricardo Ferreira de. **Mathematical models and computer simulation for analysis of pollutant dispersion in a stretch of the Paraíba do Sul river**. Campinas - SP: Universidade Estadual de Campinas, 2009. Dissertação apresentada como requisito parcial para obtenção do Título de Mestre em Matemática.

This work use the classical model of diffusion-advection-reaction to simulate the evolutionary two-dimensional behavior of patches of pollutants in an aquatic domain. Because of the lack of possibility of analytical solution (primarily by the type of domain where you study the problem), it resorts to an approximation method based on finite differences centered for both spatial variables and for the time variable (this, via the use adequately explained of the Crank-Nicolson method). The boundary conditions are set to focus on the real situation, considering the absorption of pollutants into the bank in determined sections depending on their behavior.

The considered stretch of river is this Paraíba do Sul river in region of Volta Redonda, RJ, which is characterized by a straight stretch of that body of water. It is adopted, such a velocity profile, the parabola due to Poiseuille, which affects the mounting of the system of finite differences. Special attention is devoted to numerical accuracy resulting from the approximation strategy, and the ability to qualitatively visualize the results, thus creating a software feature that can be used in many other similar minimally situations of systematic water contamination or emergency.

Keywords: Biomathematics, Mathematical Ecology, Environmental Impact, Finite difference method, Crank-Nicolson Method, Computacional Algorithms.

Sumário

1	Introdução	1
1.1	Bacia Hidrográfica do Rio Paraíba do Sul - Histórico da região	1
1.2	A importância do Rio Paraíba do Sul para a região sudeste do Brasil	3
1.2.1	Acidentes ambientais no Rio Paraíba do Sul na região Sul Fluminense do Estado do Rio de Janeiro	4
1.3	O trabalho	4
1.4	O modelo, domínio e condições de fronteira	5
1.4.1	Ajustando o modelo	8
2	Discretização do Modelo por Diferenças Finitas	10
2.1	Polinômio de Taylor	10
2.2	Fórmulas de Diferenças Finitas para Aproximações de Derivadas	11
2.3	O Método de Crank-Nicolson	14
2.4	Discretizando o modelo utilizando o método de Crank-Nicolson	16
2.4.1	Aplicando ao modelo as condições de contorno	17
2.4.2	Tratamento das condições de contorno	17
3	A implementação, os parâmetros e as simulações	22
3.1	A Implementação	22
3.2	Os parâmetros	26
3.3	Dificuldades	28
3.4	Cenários e simulações	29
3.4.1	Cenário 1: Emissão constante de óleos nos Córregos dos Coqueiros e dos Carvalhos	30

3.4.2	Cenário 2: Somente um vazamento acidental de óleo oriundo de equipamentos da CSN	30
3.4.3	Cenário 3: Vazamento constante de óleo nos Córregos dos Coqueiros e dos Carvalhos e um vazamento acidental de óleo, oriundo de equipamentos da CSN	33
3.5	Análise dos Resultados	34
4	Conclusão	36
	Apêndice	40

Lista de Figuras

1.1	Bacia Hidrográfica do Rio Paraíba do Sul, Fonte: Agência Nacional de Águas [2], 2009	3
1.2	Peixes mortos por evenenamento no Rio Paraíba do Sul, Fonte: Jornal O Globo [18], 2009	5
1.3	Trecho do Rio Paraíba do Sul adjacente a CSN, Fonte: Google Earth, [12], 2009	6
1.4	Aproximação inicial do domínio para o modelo bi-dimensional	7
1.5	Perfil de velocidades	9
2.1	Malha de <i>nós</i>	12
2.2	Método - Crank-Nicolson	15
2.3	Malha de nós do modelo	16
2.4	<i>Nós</i> contidos na fronteira	18
3.1	Exemplos de <i>Nós</i> utilizados nas simulações	23
3.2	GUI - Graphical User Interface	27
3.3	Trecho do Rio Paraíba do Sul destacando, possíveis, fontes poluidoras, Fonte: Google Earth, [12], 2009	29
3.4	Estado com $t = 0$ hora	31
3.5	Estado com $t = 0.5$ hora	31
3.6	Estado com $t = 1$ hora	31
3.7	Estado com $t = 2$ horas	31
3.8	Estado com $t = 4$ horas	31
3.9	Estado com $t = 8$ horas	31
3.10	Estado com $t = 16$ horas	32
3.11	Estado com $t = 24$ horas	32

3.12 Estado com $t = 0$ hora	32
3.13 Estado com $t = 0,5$ hora	32
3.14 Estado com $t = 1$ hora	33
3.15 Estado com $t = 2$ horas	33
3.16 Estado com $t = 0$ hora	34
3.17 Estado com $t = 0,5$ hora	34
3.18 Estado com $t = 1$ hora	34
3.19 Estado com $t = 2$ horas	34
3.20 Estado com $t = 4$ horas	34
3.21 Estado com $t = 4,5$ horas	34
3.22 Estado com $t = 5$ horas	35
3.23 Estado com $t = 6$ horas	35
3.24 Estado com $t = 12$ horas	35
3.25 Estado com $t = 24$ horas	35

Lista de Tabelas

1.1	Significados dos termos utilizados no modelo	7
3.1	Índices na malha de nós	23
3.2	Termos utilizados na implementação do modelo discretizado	24
3.3	Dimensão da matriz de nós para cálculo de poluição	27
3.4	Parâmetros utilizados nas simulações	28
3.5	Dados cenário 1	30
3.6	Dados cenário 2	32
3.7	Dados cenário 3	33

Capítulo 1

Introdução

Neste capítulo apresentamos, segundo o CEIVAP - Comitê para Integração do Rio Paraíba do Sul[9] (2009), um breve histórico da Bacia Hidrográfica do Rio Paraíba do Sul destacando a importância dessa bacia hidrográfica para o desenvolvimento da Região Sudeste do Brasil.

A fim de ilustrar a real necessidade de um estudo sobre emissões de poluentes no Rio Paraíba do Sul apresentamos duas notícias recentes sobre acidentes ambientais envolvendo grandes empresas dessa região.

Definido então o problema, propomos um modelo com equação diferencial parcial, um domínio de estudo e condições de fronteira para análise de emissão de poluentes em um trecho do Rio Paraíba do Sul.

1.1 Bacia Hidrográfica do Rio Paraíba do Sul - Histórico da região

Apesar de se estender por uma área tão vasta, os diferentes *trechos estaduais que compõem o vale do rio Paraíba do Sul*, que recorta parte do território de São Paulo, Rio de Janeiro e Minas Gerais, no sudeste brasileiro, partilham uma história comum fundamental para o desenvolvimento da região.

trechos estaduais
que compõem o
vale do rio Paraíba
do Sul

Estudos arqueológicos mostram que a região, no período pré-colonial, era ocupada por índios, na sua maioria das tribos Tupi e Guarani. Os vestígios encontrados refletem uma história de mais de mil anos, onde o impacto da população indígena na natureza não parecia ser significativo.

Com a chegada dos colonizadores, e o início do ciclo do ouro em Minas Gerais (1600), o vale adquire uma importância estratégica como corredor comercial, aproximando o interior de Minas à

costa paulista. O vale assistiu assim à construção das primeiras estradas e à formação de pequenos povoados que serviram de suporte aos comerciantes.

Esta dinâmica comercial, nos finais do século XVIII, é substituída pelas culturas do café e da cana-de-açúcar, que se expandem por todo o vale. O cultivo do café deu início ao processo de desmatamento e à ocupação extensiva da bacia, determinando um processo de alteração drástica da paisagem regional. Rapidamente, a bacia do Paraíba tornou-se responsável pela quase totalidade da produção cafeeira do país. Em meados do século XIX, o solo começa a apresentar visíveis sinais de cansaço. Estes, aliados ao fim da escravatura e à crescente dificuldade de obter terras férteis, resultam no declínio da cafeicultura. Com esta mudança assistiu-se, por um lado, à expansão da criação de gado leiteiro, e por outro, a uma migração da população rural para áreas urbanas. A agricultura, praticada geralmente sem respeito pela capacidade de uso das terras, é pouco expressiva e representa uma das mais importantes fontes de poluição dos solos e das águas pelo uso descontrolado de fertilizantes e agrotóxicos. A cana-de-açúcar mantém-se a principal cultura na bacia, embora a sua produção comece também a entrar em declínio.

A estagnação econômica e social resultante da crise do café e da cana-de-açúcar foi gradualmente superada através de um lento processo de industrialização baseado na boa infra-estrutura de transportes herdada da época áurea comercial. Com o início do século XX, a atividade industrial tornou-se o eixo de desenvolvimento da bacia. O processo de industrialização de São Paulo e a implantação, em 1946, da Companhia Siderúrgica Nacional (CSN¹) na cidade de Volta Redonda/RJ permitiram a integração econômica dos estados do Rio de Janeiro e São Paulo, transformando a bacia num dos eixos de comunicação e desenvolvimento cruciais para a região e para o país, graças às condições excepcionais que oferecia suprimento de água, energia suficiente, mercado consumidor e fácil escoamento da produção. A expansão e intensificação do desenvolvimento industrial exigiram a construção de novas rodovias, acelerada pela implantação da indústria automobilística, complementando assim o sistema viário já existente. (Comitê para Integração da Bacia Hidrográfica do Rio Paraíba do Sul - CEIVAP)

¹ CSN - Companhia Siderúrgica Nacional

1.2 A importância do Rio Paraíba do Sul para a região sudeste do Brasil

A bacia do rio Paraíba do Sul (figura 1.1) situa-se na região sudeste do Brasil. De acordo com CEIVAP[10], essa bacia ocupa área de aproximadamente 55.500 km², estendendo-se pelos estados de São Paulo (13.900 km²), Rio de Janeiro (20.900 km²) e Minas Gerais (20.700 km²), abrangendo 180 municípios - 88 em Minas Gerais, 53 no Estado do Rio de Janeiro e 39 no estado de São Paulo. A área da bacia corresponde a cerca de 0,7% da área do país e, aproximadamente, a 6% da região sudeste do Brasil. No Rio de Janeiro, a bacia abrange 63% da área total do estado; em São Paulo, 5% e em Minas Gerais , apenas 4%. O ponto culminante é o Pico das Agulhas Negras (2.787 metros).

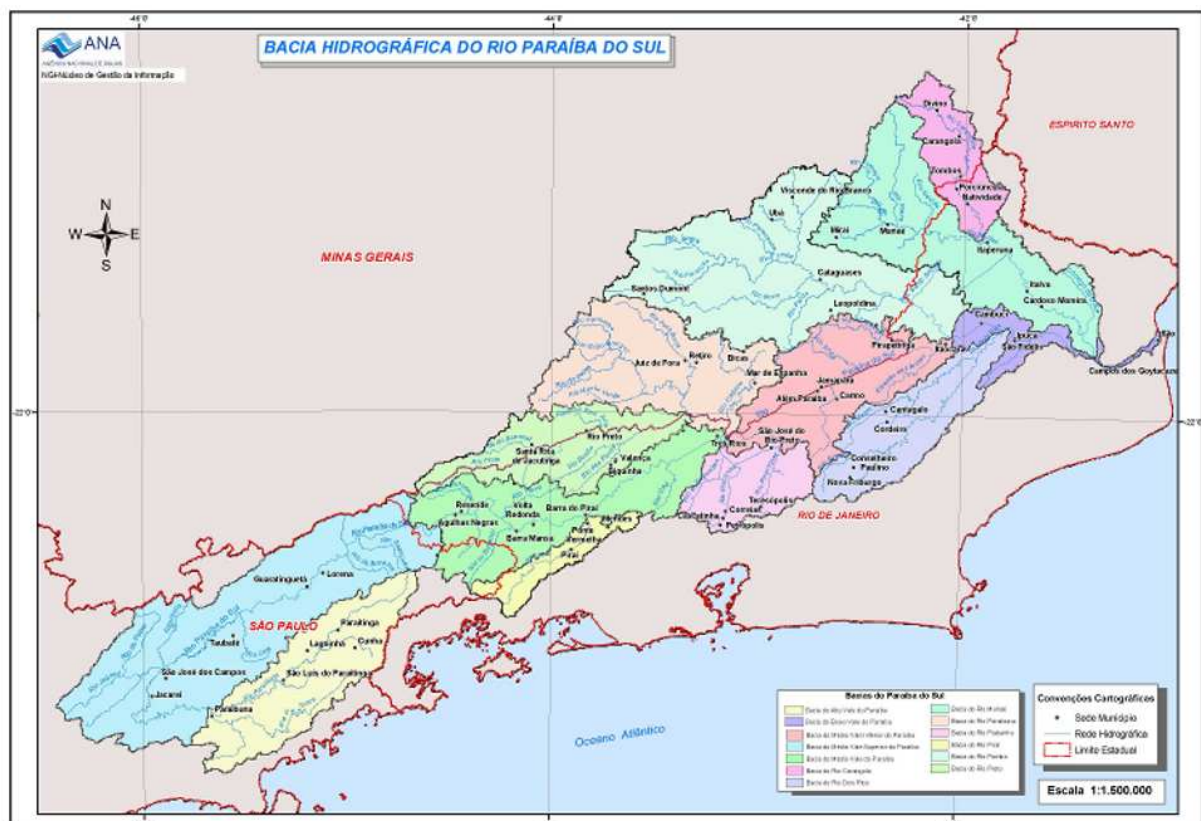


Figura 1.1: Bacia Hidrográfica do Rio Paraíba do Sul, Fonte: Agência Nacional de Águas [2], 2009

Segundo o CEIVAP[10], com relação ao saneamento básico, a situação de degradação é crítica: 1 bilhão de litros de esgotos domésticos, praticamente sem tratamento, são despejados diariamente nos rios da bacia do Paraíba. Aos efluentes domésticos somam-se 150 toneladas de DBO (Demanda Bio-Química de Oxigênio) por dia, correspondente à carga poluidora derivada dos efluentes industriais orgânicos (sem contar os agentes tóxicos, principalmente metais pesados). A carga poluidora total da bacia do Paraíba, de origem orgânica, corresponde a cerca de 300 toneladas de DBO por dia, dos quais cerca de 86% derivam de efluentes domésticos, e 14% industriais.

1.2.1 Acidentes ambientais no Rio Paraíba do Sul na região Sul Fluminense do Estado do Rio de Janeiro

Portal G1[13], em 2009 noticiou: “Após registrar um novo vazamento de óleo no Rio Paraíba do Sul, em Volta Redonda, na Região Sul Fluminense, a secretária estadual do Ambiente, Marilene Ramos, interditou, na noite desta quinta-feira (06/08/2009), a Unidade Caborquímica da Companhia Siderúrgica Nacional (CSN). O primeiro acidente aconteceu no domingo (03/08/2009).”

O Globo[18], em 2008, noticiou: “A empresa Servatis, responsável pelo vazamento de **1.500 litros de endosulfan** - um inseticida altamente tóxico - nas águas do Rio Pirapitinga, afluente do Paraíba do Sul, na madrugada de terça-feira (18/11/2008), poderá ser multada em mais de R\$ 10 milhões, segundo a secretária estadual do Ambiente, Marilene Ramos. Ela afirmou nesta quinta-feira que a empresa teria agido de forma irresponsável e com má-fé ao manter abertos diques de contenção de vazamentos e não avisar imediatamente à Feema ² sobre o acidente, figura (1.2)”

1.3 O trabalho

É notável que a população da Região do Vale do Paraíba bem como as empresas instaladas nessa região precisam, imediatamente, se preocupar com a emissão de poluentes no rio Paraíba do Sul e seus afluentes e tomar medidas de precaução a fim de minimizar os danos ambientais causados por atividades domésticas e industriais.



Figura 1.2: Peixes mortos por evenenamento no Rio Paraíba do Sul, Fonte: Jornal O Globo [18], 2009

O objetivo deste trabalho é o de criar um instrumento auxiliar relevante para a avaliação de acidentes, de ações de prevenção e de estratégias de ação ou de políticas públicas de controle de impacto ambiental.

Assim, como resultado, este trabalho visa disponibilizar um software em ambiente MATLAB³ que seja viável para criar cenários possíveis de impacto em corpos aquáticos. Nos casos apresentados, o foco é um trecho do Rio Paraíba do Sul.

As simulações serão concentradas no trecho do Rio Paraíba adjacente a Usina Presidente Vargas em Volta Redonda, pertencente ao grupo CSN, pois a CSN é uma das maiores empresas da região do Sul Fluminense do Estado do Rio de Janeiro, e, recentemente, esteve envolvida num acidente ambiental no Rio Paraíba do Sul em Volta Redonda - RJ. Na imagem (figura 1.3) o *trecho em destaque será o utilizado para as simulações*.

trecho em destaque
será o utilizado
para as simulações

1.4 O modelo, domínio e condições de fronteira

De acordo com OLIVEIRA[17] (2003), VÁSQUEZ[19] (2005) e ALEGRIA[1] (2006) uma significativa variedade de fenômenos relativos a problemas ecológicos e ambientais, podem ser modelados

³ MatLab R2008b - The Language of Thecnical Computing, Copyright 1994-2008, The MathWorks inc



Figura 1.3: Trecho do Rio Paraíba do Sul adjacente a CSN, Fonte: Google Earth, [12], 2009

através de uma EDP - *Equação Diferencial Parcial de Difusão-Advecção-Reação*. Nesse caso, para estudar a dispersão de poluentes em um trecho do Rio Paraíba do Sul em Volta Redonda-RJ, deveremos também considerar os fenômenos de decaimento global e as fontes poluidoras.

Equação Diferencial Parcial de Difusão-Advecção-Reação

Nesse trecho do Rio Paraíba do Sul a profundidade é de 4 m a 5 m, a largura de 80 m a 120 m e o comprimento, aproximadamente, 3500 m. Por causa das dimensões: largura (L) e comprimento (C) serem muito maiores do que a profundidade no local estudo, o modelo será descrito de forma bidimensional, ou seja, deconsideraremos a profundidade.

Chamando de $P(x, y, t)$ a concentração de poluentes num ponto $(x, y) \in [0, C] \times \left[-\frac{L}{2}, \frac{L}{2}\right] = \Omega \subset \mathbb{R}^2$ num dado instante $t \in (0, T] = J \subset \mathbb{R}$ com $(x, y, t) \in \Omega \times (0, T]$ temos o seguinte modelo baseado no balanço de poluente

$$\frac{\partial P}{\partial t} = \underbrace{-\text{div}[-\alpha \nabla P]}_{\text{Difusão}} - \underbrace{\text{div}[\vec{v} P]}_{\text{Transporte}} - \underbrace{\delta P}_{\text{Decaimento}} + \underbrace{f(x, y, t)}_{\text{Fonte poluidora}} \quad (1.1)$$

onde,

α	é a constante de difusibilidade efetiva no meio aquático;
$\vec{v}$	é o campo de velocidades no meio aquático Ω (supondo que, em Ω , $div(\vec{v}) = 0$;
δ	é a taxa de decaimento global no meio aquático;
$f(x, y, t)$	é o meio pelo qual a substância é introduzida no meio aquático.

Tabela 1.1: Significados dos termos utilizados no modelo

Iremos, inicialmente, aproximar o domínio destacado da figura 1.3 por um retângulo conforme figura 1.4. Desta forma teremos $\Omega = \left\{ (x, y) | 0 \leq x \leq C \wedge -\frac{L}{2} \leq y \leq \frac{L}{2} \right\}$.

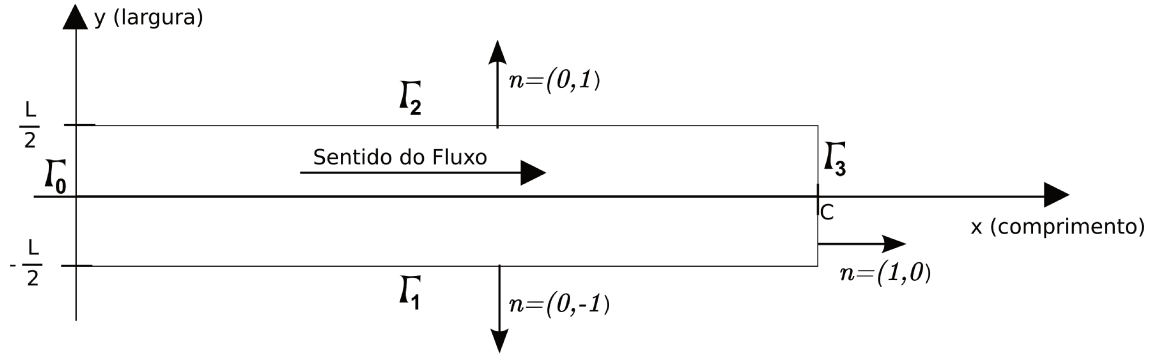


Figura 1.4: Aproximação inicial do domínio para o modelo bi-dimensional

Adotaremos, segundo INFORZATO [14], as *condições de contorno homogênea de Dirichlet (1.2) e de Robin (1.3)*

condições de contorno homogênea de Dirichlet (1.2) e de Robin (1.3)

$$P|_{\Gamma_0} = 0, \quad (1.2)$$

$$aP + b\frac{\partial P}{\partial \eta} = g. \quad (1.3)$$

- Para a **fronteira a montante do rio**, não antigida pela poluição e representada por Γ_0 , temos fronteira a montante do rio

$$P(0, y, t)|_{\Gamma_0} = 0. \quad (1.4)$$

- Para a parte da **fronteira referente às margens direita e esquerda** (representada, respectivamente, por Γ_1 e Γ_2) onde podem ocorrer perda de poluição, temos fronteira referente às margens direita e esquerda

$$\frac{\partial P}{\partial \eta} \Big|_{\Gamma_1} = -\frac{k_1}{\alpha} P \quad (1.5)$$

e

$$\left. \frac{\partial P}{\partial \eta} \right|_{\Gamma_2} = -\frac{k_1}{\alpha} P. \quad (1.6)$$

- Para a **fronteira a jusante do rio** representada por Γ_3 , temos

fronteira a jusante
do rio

$$\left. \frac{\partial P}{\partial \eta} \right|_{\Gamma_3} = -\frac{k_2}{\alpha} P \quad (1.7)$$

Onde $\frac{k_i}{\alpha}$ representa aquela fração de poluente que determina a velocidade de saída do domínio e η é o vetor normal à parte externa da fronteira.

1.4.1 Ajustando o modelo

De acordo com FORTUNA[11] (2000, p. 373) como sabemos a vazão na fronteira de nosso domínio, mas não a exata distribuição espacial das velocidades ao longo dela, a solução é prescrever um **perfil de velocidades** arbitrário, mas compatível com o problema.

perfil de veloci-
dades

Seja v a velocidade normal à fronteira de entrada de fluido e v_0 uma velocidade de referência. O **perfil parabólico**, segundo FORTUNA[11] (2000, p. 373) é o perfil encontrado em escoamentos laminares, estacionários e totalmente desenvolvidos entre duas fronteiras paralelas não escorregadias.

perfil parabólico

Supondo que as fronteiras, margens do rio, estejam separadas por uma distância $L = 2h$, descritas por $[0, C] \times \left\{-\frac{L}{2}\right\}$ e $[0, C] \times \left\{\frac{L}{2}\right\}$, o perfil parabólico, conforme a figura 1.5, segue uma relação do tipo

$$v(y) = v_0 \left(1 - \frac{y^2}{h^2}\right). \quad (1.8)$$

Portanto, segue que o campo $\vec{v}$ é dado por

$$\vec{v}(x, y) = \left(v_0 \left(1 - \frac{y^2}{h^2}\right), 0\right). \quad (1.9)$$

Com essas suposições a equação 1.1 pode ser reescrita como

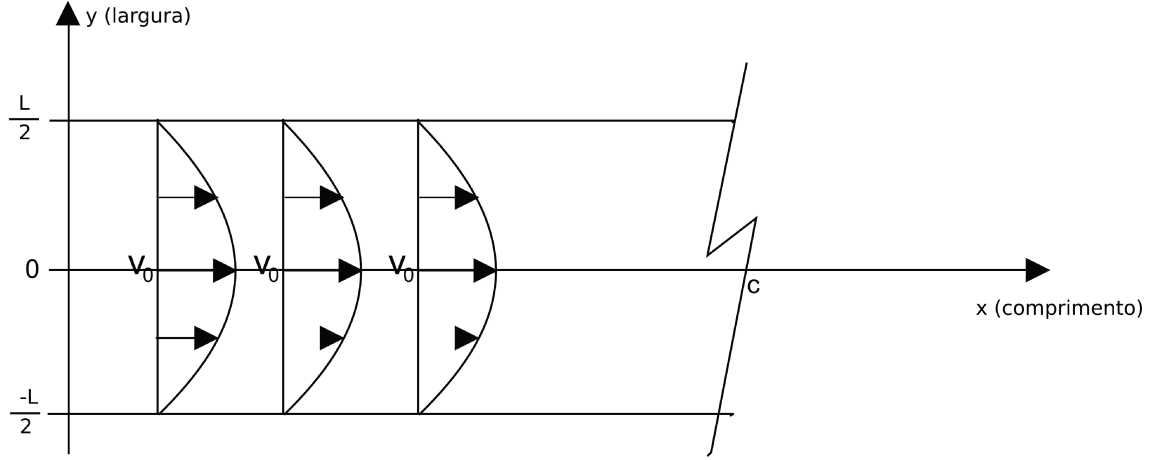


Figura 1.5: Perfil de velocidades

$$\begin{aligned}
 \frac{\partial P}{\partial t} &= \alpha \operatorname{div}_{xy} [\nabla_{xy} P] - \operatorname{div}_{xy} [\vec{v} P] - \delta P + f, \\
 \frac{\partial P}{\partial t} &= \alpha \left(\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \right) - \vec{v} \nabla_{xy} P - \delta P + f, \\
 \frac{\partial P}{\partial t} &= \alpha \left(\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \right) - v_0 \left(1 - \frac{y^2}{h^2} \right) \frac{\partial P}{\partial x} - \delta P + f.
 \end{aligned} \tag{1.10}$$

Portanto se tomarmos a equação 1.10, o domínio Ω e as condições representadas pelas equações 1.4, 1.5, 1.6 e 1.7 temos a formulação do problema.

Uma caracterização de fluxo de fluidos incompressíveis é que $\operatorname{div}(\vec{v}(x, y)) = 0$. Uma condição necessária para os procedimentos numéricos a serem usados nas aproximações da desejada ou, neste caso, da necessária solução.

Passaremos, a seguir, para a aproximação do problema por Diferenças Finitas.

Capítulo 2

Discretização do Modelo por Diferenças Finitas

Nesse capítulo, inicialmente, julgamos necessário apresentar, de acordo com LIMA[15] (2008, p.151-159) e BORTOLOSSI[5] (2002, p. 378-382), o Polinômio de Taylor para funções de várias variáveis. Isto porque de acordo com CARNAHAN, LUTHER e WILKES[8] (1969), NOVAIS e CUNHA[16] (2003) e BURDEN e FAIRES[7] (2008), o método de aproximação de solução de equações diferenciais parciais por diferenças finita utiliza aproximações de funções por polinômios de Taylor.

Descrevemos, segundo CARNAHAN, LUTHER e WILKES[8] (1969) o método de diferenças finitas para aproximação de solução de equações diferenciais parciais e utilizamos esse método para discretizar nosso modelo.

2.1 Polinômio de Taylor

Teorema 2.1.1 *Considere uma função $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ de classe C^k e $a = (a_1, a_2, \dots, a_n)$ um ponto do interior de D . Então existe um único polinômio p_k de grau k que satisfaz as condições:*

1. *Os valores das funções p_k e f coincidem em a , ou seja,*
$$p_k(a_1, a_2, \dots, a_n) = f(a_1, a_2, \dots, a_n)$$
2. *Sendo $c \in \mathbb{N}$ e $1 \leq c \leq k$ então os valores das derivadas de ordem c de p_k e f coincidem em a .*

*Esse polinômio recebe o nome de **Polinômio de Taylor**. Sabendo que $x \in D \subset \mathbb{R}^n$, ou*

Polinômio de Taylor

seja, $x = (x_1, x_2, \dots, x_n)$ e fazendo $\vec{\Delta x} = x - a = (x_1 - a_1, x_2 - a_2, \dots, x_n - a_n) \Rightarrow \vec{\Delta x} = (\Delta x_1, \Delta x_2, \dots, \Delta x_n)$ podemos representar o Polinômio de Taylor da seguinte forma:

$$f(x) = \underbrace{f(a) + \partial f(a) \left(\vec{\Delta x} \right) + \frac{\partial^2 f(a) \cdot \left(\vec{\Delta x} \right)^2}{2!} + \dots + \frac{\partial^k f(a) \cdot \left(\vec{\Delta x} \right)^k}{k!}}_{p_k} + r_k \left(\vec{\Delta x} \right) \quad (2.1)$$

Onde:

$$\frac{\partial^k f(a) \cdot \left(\vec{\Delta x} \right)^k}{k!} = \frac{1}{k!} \sum_{i_1=1, i_2=1, \dots, i_k=1}^n \frac{\partial^k f(a)}{\partial x_{i_1} \partial x_{i_2} \dots \partial x_{i_k}} \Delta x_{i_1} \Delta x_{i_2} \dots \Delta x_{i_k}$$

e o resto de Lagrange é dado por:

$$r_k \left(\vec{\Delta x} \right) = \frac{1}{(k+1)!} \partial^{k+1} f(a + \xi \Delta x) (\Delta x)^{k+1} \text{ com } 0 < \xi < 1$$

2.2 Fórmulas de Diferenças Finitas para Aproximações de Derivadas

Para simplificar assumiremos $u = u(x, y)$, ou seja, u é uma função de duas variáveis reais. Supondo que u tenha um número suficiente de derivadas parciais e que existam valores de u nos dois pontos (x, y) e $(x + h, y + k)$ poderemos representar u , utilizando notação de CARNAHAN, LUTHER e WILKES [[8]], como um Polinômio de Taylor

$$u(x + h, y + k) = u(x, y) + \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right) u(x, y) + \frac{1}{2!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^2 u(x, y) + \dots + \quad (2.2)$$

$$+ \frac{1}{(n-1)!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^{n-1} u(x, y) + R_n,$$

onde o termo R_k é dado por

$$R_n = \frac{1}{n!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^n u(x + \xi h, y + \xi k), \quad 0 < \xi < 1. \quad (2.3)$$

Portanto, segue que

$$R_n = O \left[(|h| + |k|)^n \right]. \quad (2.4)$$

Da equação (2.4) nós podemos dizer que existe uma constante positiva M tal que $|R_n| \leq M(|h| + |k|)^n$ pois ambas h e k tendem para zero.

O ponto $(i\Delta x, j\Delta y)$, também chamado de *nó* (i, j) conforme mostrado na malha representada na figura 2.1, será considerado nas expansões polinomiais.

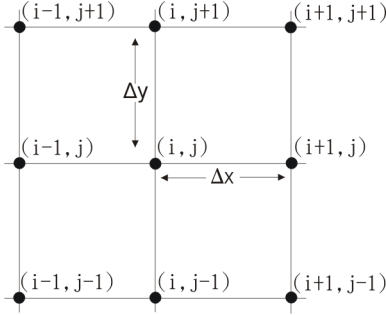


Figura 2.1: Malha de nós

Expandindo as séries de Taylor para $u_{i-1,j}$, $u_{i+1,j}$ sobre o valor central $u_{i,j}$, nós obtemos

$$u_{i-1,j} = u_{i,j} - \Delta x u_x + \frac{(\Delta x)^2}{2!} u_{xx} - \frac{(\Delta x)^3}{3!} u_{xxx} + \frac{(\Delta x)^4}{4!} u_{xxxx}, \quad (2.5)$$

$$u_{i+1,j} = u_{i,j} + \Delta x u_x + \frac{(\Delta x)^2}{2!} u_{xx} + \frac{(\Delta x)^3}{3!} u_{xxx} + \frac{(\Delta x)^4}{4!} u_{xxxx}, \quad (2.6)$$

sendo $u_x \equiv \frac{\partial u}{\partial x}$, $u_{xx} \equiv \frac{\partial^2 u}{\partial x^2}$, etc., e todas as derivadas são calculadas no nó (i, j) . Operando adequadamente com as equações (2.5) e (2.6), obteremos as seguintes fórmulas de **diferenças finitas** de primeira e segunda ordens em (i, j) :

diferenças finitas

$$\text{de (2.6): } \left. \frac{\partial u}{\partial x} \right|_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + O(\Delta x), \quad (2.7)$$

$$\text{de (2.5): } \left. \frac{\partial u}{\partial x} \right|_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + O(\Delta x), \quad (2.8)$$

$$\text{de (2.6) - (2.5): } \left. \frac{\partial u}{\partial x} \right|_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + O[(\Delta x)^2], \quad (2.9)$$

$$\text{de (2.6) + (2.5): } \left. \frac{\partial^2 u}{\partial x^2} \right|_{i,j} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2} + O[(\Delta x)^2]. \quad (2.10)$$

As fórmulas (2.8), (2.9) e (2.7) são conhecidas, respectivamente, por forma de diferença **atrasada**, **central** e **avançada**, respectivamente. Existem formas análogas para $\frac{\partial u}{\partial y}$ e $\frac{\partial^2 u}{\partial y^2}$. Por exemplo, ainda para a malha de nós da figura 2.1, podemos obter a forma para $\frac{\partial^2 u}{\partial x \partial y}$ da seguinte maneira.

Expandindo as séries de Taylor para $u_{i+1,j+1}$, $u_{i+1,j-1}$, $u_{i-1,j+1}$ e $u_{i-1,j-1}$ sobre o valor central $u_{i,j}$, nós obtemos

$$u_{i+1,j+1} = u_{i,j} + \Delta x u_x + \Delta y u_y + \frac{(\Delta x)^2}{2!} u_{xx} + 2 \frac{\Delta x \Delta y}{2!} u_{xy} + \frac{(\Delta y)^2}{2!} u_{yy}, \quad (2.11)$$

$$u_{i+1,j-1} = u_{i,j} + \Delta x u_x - \Delta y u_y + \frac{(\Delta x)^2}{2!} u_{xx} - 2 \frac{\Delta x \Delta y}{2!} u_{xy} + \frac{(\Delta y)^2}{2!} u_{yy}, \quad (2.12)$$

$$u_{i-1,j+1} = u_{i,j} - \Delta x u_x + \Delta y u_y + \frac{(\Delta x)^2}{2!} u_{xx} - 2 \frac{\Delta x \Delta y}{2!} u_{xy} + \frac{(\Delta y)^2}{2!} u_{yy}, \quad (2.13)$$

$$u_{i-1,j-1} = u_{i,j} - \Delta x u_x - \Delta y u_y + \frac{(\Delta x)^2}{2!} u_{xx} + 2 \frac{\Delta x \Delta y}{2!} u_{xy} + \frac{(\Delta y)^2}{2!} u_{yy}, \quad (2.14)$$

sendo $u_x \equiv \frac{\partial u}{\partial x}$, $u_{xx} \equiv \frac{\partial^2 u}{\partial x^2}$, $u_y \equiv \frac{\partial u}{\partial y}$, $u_{yy} \equiv \frac{\partial^2 u}{\partial y^2}$ e $u_{xy} \equiv \frac{\partial^2 u}{\partial x \partial y}$ e todas as derivadas são calculadas no nó (i, j) . Operando adequadamente com as equações (2.11), (2.12), (2.13) e (2.14) obteremos a seguinte fórmula de **diferenças finitas** para $\frac{\partial^2 u}{\partial x \partial y}$ em (i, j)

diferenças finitas

$$\text{de } ((2.11) - (2.13)) + ((2.14) - (2.12)): \frac{\partial^2 u}{\partial x \partial y} = \frac{u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1} + u_{i-1,j-1}}{4\Delta x \Delta y} + O\left[(\Delta x + \Delta y)^2\right]. \quad (2.15)$$

Para o Laplaciano em duas dimensões ($\nabla \cdot \nabla = \nabla^2 \equiv \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$) e uma malha quadrada (isto é, na qual $\Delta x = \Delta y$) com nove nós de aproximação, utilizaremos esses nove nós, sabendo que $u_{xx} + u_{yy} = 0$ é resolvido por

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\begin{bmatrix} u_{i-1,j+1} + 4u_{i,j+1} + u_{i+1,j+1} \\ + 4u_{i-1,j} - 20u_{i,j} + 4u_{i+1,j} \\ + u_{i-1,j-1} + 4u_{i,j-1} + u_{i+1,j-1} \end{bmatrix}}{6(\Delta x)^2} + O\left[16(\Delta x)^4\right]. \quad (2.16)$$

Numa malha poderemos obter mais e mais *nós* e portanto é enorme a gama de possíveis escolhas para aproximações deste tipo; muitas têm, porém, interesse apenas teórico, não se prestando a práticas computacionais viáveis. As formas abaixo são as mais compactas e por conveniência, o **operador de diferença central** δ_x pode ser utilizado. Essas formas são definidas por

operador de diferença central

$$\delta_x u_{i,j} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} \quad (2.17)$$

e

$$\delta_x^2 u_{i,j} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2}, \quad (2.18)$$

garantindo aproximações locais de 2^a ordem.

2.3 O Método de Crank-Nicolson

Nas aproximações temporais de Equações ou Sistemas de Equações de Derivadas Parciais, a escolha recaí com certa frequência sobre o Método de Crank-Nicolson. E com bons motivos: a ordem da aproximação, o custo computacional, a relativa simplicidade algorítmica e, por último, mas não menos, o bom comportamento numérico.

O Método de Crank-Nicolson apóia-se, por um lado em uma aproximação de $\frac{\partial u}{\partial t} \left(x_i, y_j, t_{n+\frac{1}{2}} \right)$ portanto uma fórmula de diferenças centradas, além de ter uma estimativa de idêntica ordem de aproximação para $u \left(x_i, y_j, t_{n+\frac{1}{2}} \right)$. Estas são dadas por, respectivamente,

$$\frac{\partial u}{\partial t} \left(x_i, y_j, t_{n+\frac{1}{2}} \right) = \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} \quad (2.19)$$

$$u \left(x_i, y_j, t_{n+\frac{1}{2}} \right) = \frac{u_{i,j}^{n+1} + u_{i,j}^n}{2} \quad (2.20)$$

Se $\Omega \in \mathbb{R}^1$ e $u = u(x, t)$, tomando a equação (2.9), obtemos uma aproximação para $\frac{\partial u}{\partial t}$ de ordem $O \left[(\Delta t)^2 \right]$. Podemos, conseqüentemente, escrever a seguinte equação

$$\frac{\partial u}{\partial t} = \frac{u_{i,n+1} - u_{i,n}}{\Delta t} + O \left[(\Delta t)^2 \right] \quad (2.21)$$

para a derivada $\frac{\partial u}{\partial t}$ no ponto médio $\left(i, n + \frac{1}{2}\right)$. Teríamos, esquematicamente (figura 2.2), para x_{i-1} , x_i e x_{i+1} .

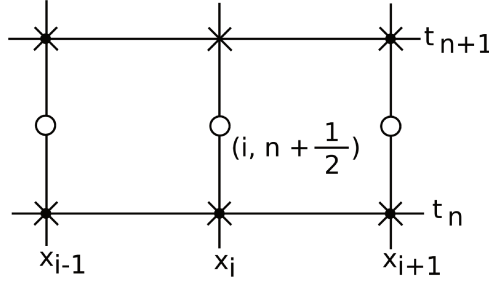


Figura 2.2: Método - Crank-Nicolson

Ainda considerando $\Omega \in \mathbb{R}^1$, temos, para a derivada de 2ª ordem na variável espacial x na notação de CARNAHAN e outros [8]:

$$\frac{\partial^2 u}{\partial x^2} \cong \theta \delta_x^2 u_i^{n+1} + (1 - \theta) \delta_x^2 u_i^n, \quad (2.22)$$

onde o valor de $\theta \in [0, 1]$, ou seja, $0 \leq \theta \leq 1$. O operador de diferença-central δ_x é utilizado por conveniência. Este operador está definido em (2.17).

No **método de Crank-Nicolson**, o θ escolhido é $\frac{1}{2}$ e a equação de diferença-finita, correspondente a (2.21) para EDP (Equação Diferencial Parcial) $u_t = u_{xx}$ é

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{2} \delta_x^2 u_i^{n+1} + \frac{1}{2} \delta_x^2 u_i^n. \quad (2.23)$$

Segundo CARNAHAN, LUTHER e WILKES[8] (1969) o método de Crank-Nicolson é estável para valores de $\lambda < \frac{1}{2}$ tais que $\lambda = \frac{\Delta t}{(\Delta x)^2}$, e é convergente com erro de discretização de ordem $O[(\Delta t)^2 + (\Delta x)^2]$.

Expandindo a equação (2.23), obteremos

$$-\lambda u_{i-1}^{n+1} + 2(1 + \lambda) u_i^{n+1} - \lambda u_{i+1}^{n+1} = \lambda u_{i-1}^n + 2(1 - \lambda) u_i^n + \lambda u_{i+1}^n. \quad (2.24)$$

2.4 Discretizando o modelo utilizando o método de Crank-Nicolson

Para aproximar a solução da equação 1.10 com as condições 1.4, 1.5, 1.6 e 1.7, uma malha com *nós* é, primeiramente, estabelecida em toda a região $\Omega \times (0, T]$ conforme ilustrado na figura 2.3, com espaços entre os nós $\Delta x = \frac{C}{M}$, $\Delta y = \frac{L}{N}$ e $\Delta t = \frac{T}{Q}$, onde M , N e Q são inteiros arbitrários.

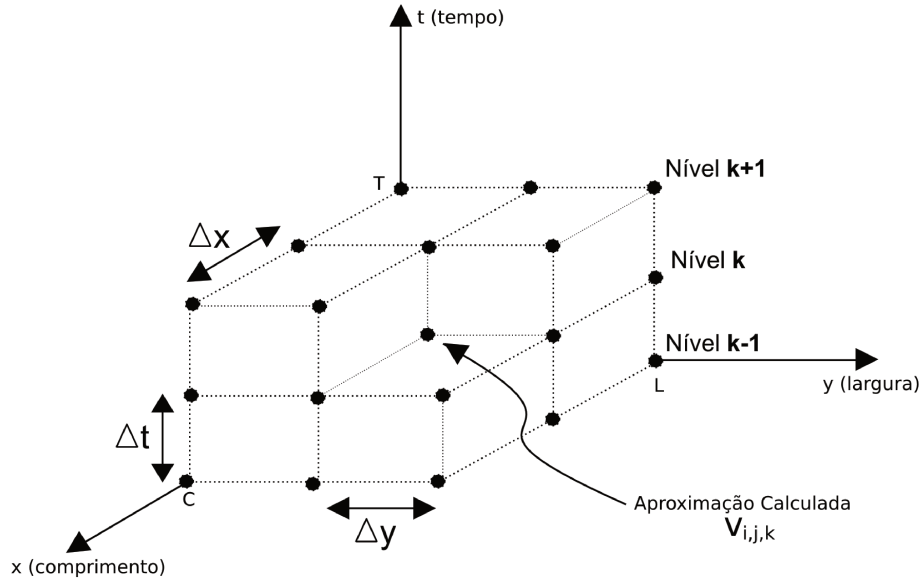


Figura 2.3: Malha de nós do modelo

Para qualquer nó (i, j, k) , interior as derivadas da equação (1.10) são agora rescritas na forma de diferença finita e nas aproximações temporais utilizando o **método de Crank-Nicolson**.

método de Crank-Nicolson

Sendo $v(y) = v_0 \left(1 - \frac{y^2}{h^2}\right)$ e fazendo $P(x, y, t) \approx u(x_i, y_j, t_k) \equiv u_{i,j}^k$ temos

$$\frac{\partial P}{\partial t} = \frac{u_{i,j}^{k+1} - u_{i,j}^k}{\Delta t} + O[\Delta t], \quad (2.25)$$

$$\frac{\partial P}{\partial x} = \frac{u_{i+1,j}^{k+\frac{1}{2}} - u_{i-1,j}^{k+\frac{1}{2}}}{2\Delta x} = \frac{u_{i+1,j}^{k+1} - u_{i-1,j}^{k+1} + u_{i+1,j}^k - u_{i-1,j}^k}{4\Delta x} + O[(\Delta x)^2], \quad (2.26)$$

$$\frac{\partial^2 P}{\partial x^2} = \frac{u_{i-1,j}^{k+1} - 2u_{i,j}^{k+1} + u_{i+1,j}^{k+1} + u_{i-1,j}^k - 2u_{i,j}^k + u_{i+1,j}^k}{2(\Delta x)^2} + O[(\Delta x)^2] \text{ e} \quad (2.27)$$

$$\frac{\partial^2 P}{\partial y^2} = \frac{u_{i,j-1}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j+1}^{k+1} + u_{i,j-1}^k - 2u_{i,j}^k + u_{i,j+1}^k}{2(\Delta y)^2} + O[(\Delta y)^2]. \quad (2.28)$$

$$(2.29)$$

Substituindo na equação 1.10 obtemos

$$\begin{aligned}
\frac{u_{i,j}^{k+1} - u_{i,j}^k}{\Delta t} = & \alpha \left(\frac{u_{i-1,j}^{k+1} - 2u_{i,j}^{k+1} + u_{i+1,j}^{k+1} + u_{i-1,j}^k - 2u_{i,j}^k + u_{i+1,j}^k}{2\Delta x^2} + \right. \\
& \left. + \frac{u_{i,j-1}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j+1}^{k+1} + u_{i,j-1}^k - 2u_{i,j}^k + u_{i,j+1}^k}{2\Delta y^2} \right) \\
& - v(y_j) \left(\frac{u_{i+1,j}^{k+1} - u_{i-1,j}^{k+1} + u_{i+1,j}^k - u_{i-1,j}^k}{4\Delta x} \right) + \\
& - \delta \left(\frac{u_{i,j}^{k+1} + u_{i,j}^k}{2} \right) + f(k + \frac{1}{2}).
\end{aligned} \tag{2.30}$$

Reorganizando a equação 2.30 com a finalidade de escrever o nível $k + 1$ do tempo em função do nível k obteremos a **equação do modelo discretizada** conforme equação (2.31)

equação do modelo
discretizada

$$\begin{aligned}
u_{i-1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + u_{i,j-1}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i,j}^{k+1} \left(1 + \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) + \delta \frac{\Delta t}{2} \right) + \\
+ u_{i,j+1}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i+1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) = \\
u_{i-1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) + u_{i,j-1}^k \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i,j}^k \left(1 - \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) - \delta \frac{\Delta t}{2} \right) + \\
+ u_{i,j+1}^k \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i+1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + f(k + \frac{1}{2})\Delta t.
\end{aligned} \tag{2.31}$$

2.4.1 Aplicando ao modelo as condições de contorno

Nosso próximo passo é a utilização das condições de contorno (1.4), (1.5), (1.6) e (1.7), para ajustarmos a discretização. A figura 2.4 ilustra os “problemas” ao calcularmos a quantidade de poluição nos nós contidos nas fronteiras.

2.4.2 Tratamento das condições de contorno

Para os contornos Γ_1 , Γ_2 e Γ_3 , ver figura (1.4), que formam, juntos, a fronteira $\partial\Omega$, temos

$$\left. \frac{\partial P}{\partial \eta} \right|_{\Gamma_l} = -\frac{k_l}{\alpha} P, \quad l=1, 2, 3 \tag{2.32}$$

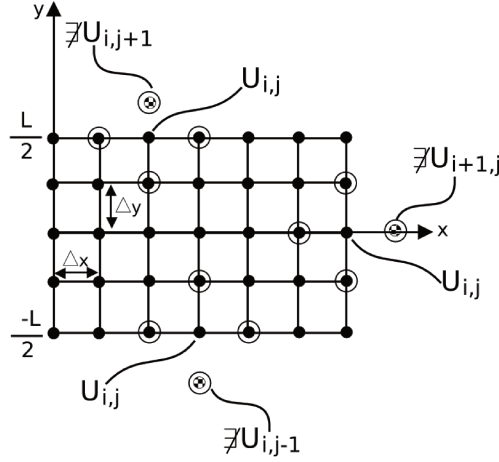


Figura 2.4: Nós contidos na fronteira

O desafio é tratar estas equações ainda como aproximação de 2ª ordem, incorporando-as, onde convier, ao sistema indicado por (2.31).

Em 1º lugar, temos, de acordo com as equações (1.5) a (1.6) e figura (1.4):

$$\left. \frac{\partial P}{\partial \eta} \right|_{\Gamma_1} = \left(- \frac{\partial P}{\partial y} \right) \Big|_{\Gamma_1}, \quad (2.33)$$

$$\left. \frac{\partial P}{\partial \eta} \right|_{\Gamma_2} = \left. \frac{\partial P}{\partial y} \right|_{\Gamma_2}, \quad (2.34)$$

$$\left. \frac{\partial P}{\partial \eta} \right|_{\Gamma_3} = \left. \frac{\partial P}{\partial x} \right|_{\Gamma_3}. \quad (2.35)$$

A aproximação indicada em (2.26) apresenta a dificuldade de exigir mais um valor de P **fora** do domínio para manter um erro de 2ª ordem (ver figura 2.4).

Nesse caso, o artifício numérico é de, a partir da relação dada, genericamente por:

$$\frac{\partial P}{\partial z} = \frac{P_{\text{fora do domínio}} - P_{\text{interno}}}{2\Delta z} = k_l P_{\text{fronteira}}, \quad (2.36)$$

obtém-se

$$P_{\text{fora do domínio}} = 2\Delta z k_l P_{\text{fronteira}} + P_{\text{interno}}. \quad (2.37)$$

Nessas condições teríamos, usando a igualdade (2.37) nas aproximações de $\frac{\partial P}{\partial x}$, $\frac{\partial^2 P}{\partial x^2}$ e $\frac{\partial^2 P}{\partial y^2}$ o seguinte:

Para Γ_1

$$\begin{aligned}\frac{\#u_{i,j+1} - u_{i,j-1}}{2\Delta y} &= -\frac{k_1}{\alpha}u_{i,j} \\ \#u_{i,j+1} &= u_{i,j-1} - \frac{2k_1\Delta y}{\alpha}u_{i,j}\end{aligned}\quad (2.38)$$

$$\begin{aligned}u_{i-1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + u_{i,j-1}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i,j}^{k+1} \left(1 + \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) + \delta \frac{\Delta t}{2} \right) + \\ + \left(u_{i,j-1}^{k+1} - \frac{2k_1\Delta y}{\alpha}u_{i,j}^{k+1} \right) \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i+1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) = \\ u_{i-1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) + u_{i,j-1}^k \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i,j}^k \left(1 - \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) - \delta \frac{\Delta t}{2} \right) + \\ + \left(u_{i,j-1}^k - \frac{2k_1\Delta y}{\alpha}u_{i,j}^k \right) \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i+1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + f(k + \frac{1}{2})\Delta t\end{aligned}\quad (2.39)$$

$$\begin{aligned}u_{i-1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + u_{i,j-1}^{k+1} \left(-\alpha \frac{\Delta t}{\Delta y^2} \right) + \\ + u_{i,j}^{k+1} \left(1 + \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) + \delta \frac{\Delta t}{2} + \frac{k_1\Delta t}{\Delta y} \right) + 0 + u_{i+1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) = \\ u_{i-1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) + u_{i,j-1}^k \left(\alpha \frac{\Delta t}{\Delta y^2} \right) + \\ + u_{i,j}^k \left(1 - \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) - \delta \frac{\Delta t}{2} - \frac{k_1\Delta t}{\Delta y} \right) + 0 + u_{i+1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + \\ + f(k + \frac{1}{2})\Delta t\end{aligned}\quad (2.40)$$

Para Γ_2

$$\begin{aligned}\frac{u_{i,j+1} - \#u_{i,j-1}}{2\Delta y} &= \frac{k_1}{\alpha}u_{i,j} \\ \#u_{i,j-1} &= -\frac{2k_1\Delta y}{\alpha}u_{i,j} + u_{i,j+1}\end{aligned}\quad (2.41)$$

$$\begin{aligned}
& u_{i-1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + \left(-\frac{2k_1\Delta y}{\alpha} u_{i,j}^{k+1} + u_{i,j+1}^{k+1} \right) \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + \\
& \quad + u_{i,j}^{k+1} \left(1 + \alpha\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) + \delta \frac{\Delta t}{2} \right) + \\
& \quad + u_{i,j+1}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i+1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) = \\
& u_{i-1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) + \left(-\frac{2k_1\Delta y}{\alpha} u_{i,j}^k + u_{i,j+1}^k \right) \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + \\
& \quad + u_{i,j}^k \left(1 - \alpha\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) - \delta \frac{\Delta t}{2} \right) + \\
& \quad + u_{i,j+1}^k \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i+1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + f(k + \frac{1}{2})\Delta t
\end{aligned} \tag{2.42}$$

$$\begin{aligned}
& u_{i-1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + 0 \\
& + u_{i,j}^{k+1} \left(1 + \alpha\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) + \delta \frac{\Delta t}{2} + \frac{k_1\Delta t}{\Delta y} \right) + \\
& + u_{i,j+1}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i+1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) = \\
& u_{i-1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) + 0 \\
& + u_{i,j}^k \left(1 - \alpha\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) - \delta \frac{\Delta t}{2} - \frac{k_1\Delta t}{\Delta y} \right) + \\
& + u_{i,j+1}^k \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i+1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + f(k + \frac{1}{2})\Delta t
\end{aligned} \tag{2.43}$$

Para Γ_3

$$\begin{aligned}
\frac{\#u_{i+1,j} - u_{i-1,j}}{2\Delta x} &= -\frac{k_2}{\alpha} u_{i,j} \\
\#u_{i+1,j} &= u_{i-1,j} - \frac{2k_2\Delta x}{\alpha} u_{i,j}
\end{aligned} \tag{2.44}$$

$$\begin{aligned}
& u_{i-1,j}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + u_{i,j-1}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i,j}^{k+1} \left(1 + \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) + \delta \frac{\Delta t}{2} \right) + \\
& \quad + u_{i,j+1}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + \left(u_{i-1,j}^{k+1} - \frac{2k_2\Delta x}{\alpha} u_{i,j}^{k+1} \right) \left(-\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) = \\
& u_{i-1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x} \right) + u_{i,j-1}^k \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + u_{i,j}^k \left(1 - \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) - \delta \frac{\Delta t}{2} \right) + \\
& \quad + u_{i,j+1}^k \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + \left(u_{i-1,j}^k - \frac{2k_2\Delta x}{\alpha} u_{i,j}^k \right) \left(\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x} \right) + f(k + \frac{1}{2})\Delta t
\end{aligned} \tag{2.45}$$

$$\begin{aligned}
& u_{i-1,j}^{k+1} \left(-\alpha \frac{\Delta t}{\Delta x^2} \right) + u_{i,j-1}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + \\
& + u_{i,j}^{k+1} \left(1 + \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) + \delta \frac{\Delta t}{2} + \frac{k_2\Delta t}{\Delta x} - \frac{v(y_j)k_2\Delta t}{2\alpha} \right) + \\
& \quad + u_{i,j+1}^{k+1} \left(-\alpha \frac{\Delta t}{2\Delta y^2} \right) + 0 = \\
& u_{i-1,j}^k \left(\alpha \frac{\Delta t}{2\Delta x^2} \right) + u_{i,j-1}^k \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + \\
& + u_{i,j}^k \left(1 - \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) - \delta \frac{\Delta t}{2} - \frac{k_2\Delta t}{\Delta x} + \frac{v(y_j)k_2\Delta t}{2\alpha} \right) + \\
& \quad + u_{i,j+1}^k \left(\alpha \frac{\Delta t}{2\Delta y^2} \right) + 0 + f(k + \frac{1}{2})\Delta t
\end{aligned} \tag{2.46}$$

Vale observar que, se necessário, podemos fazer $f(k + \frac{1}{2})\Delta t = \Delta t \frac{f(k+1) + f(k)}{2}$.

Para o funcionamento adequado deste método é necessário respeitar as limitações numéricas dadas pelo **Núcleo de Péclet** que segundo FORTUNA [11] (2000, p.166) é dado, no presente caso, pela desigualdade

$$v \frac{\Delta x}{\alpha} < 2 \text{ ou } \Delta x < \frac{2\alpha}{v}. \tag{2.47}$$

Temos então informações suficientes para desenvolver uma implementação computacional e efetuar as simulações.

Capítulo 3

A implementação, os parâmetros e as simulações

Neste capítulo, inicialmente, comentaremos a implementação computacional do algoritmo de aproximação numérica do modelo adotado. Em seguida, proporemos cenários a partir dos quais a implementação do modelo será executada efetuando simulações que serão comentadas.

Ressaltamos que uma das maiores dificuldades na Modelagem Matemática é a obtenção dos parâmetros, pois nem sempre conseguimos na literatura e/ou com os órgãos que possam ter interesse na análise do problema.

Neste capítulo não apresentaremos o código fonte da implementação pois esse código será apresentado, na íntegra, como apêndice.

3.1 A Implementação

A implementação do modelo foi realizada no **MatLab versão 7.7.0.471 (R2008b)** sobre a plataforma operacional **Ubuntu Linux versão 9.10 (Karmic Koala)** num notebook Toshiba com processador Intel Core 2 Duo Processor T5550 de 4 Gigabytes de memória RAM.

Para ilustrar a passagem do modelo matemático para a *implementação algorítmica*, adotaremos um retângulo com nove nós conforme figura (3.1).

implementação al-
gorítmica

Na aproximação, surge de modo natural, a expressão das equações (2.31), (2.43), (2.40) e (2.46) de modo matricial e, assim, vale a pena expressar estas aproximações espaço-temporais como sucessivos valores $u_{i_{\text{novo}}}^{n+1}$, onde $i_{\text{novo}} = 1$ a ni .

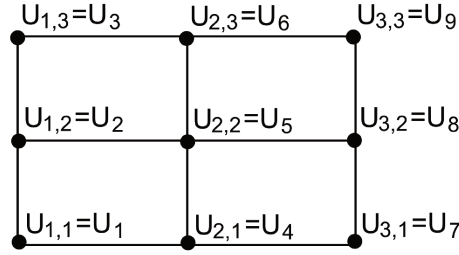


Figura 3.1: Exemplos de Nós utilizados nas simulações

Para implementação, portanto, a primeira mudança que realizamos foi a de índice, partimos do índice bidimensional para o índice unidimensional utilizando:

$$i_{\text{novo}} = (i - 1) * nny + j \quad (3.1)$$

onde:

i	coluna do nó na malha
j	linha do nó na malha
nny	número de nós numa coluna da malha
i_{novo}	novo índice obtido
ni	número total de incógnitas (nós)

Tabela 3.1: Índices na malha de nós

Deste ponto reescrevemos a equação (2.31) utilizando os termos descritos na tabela (3.2) e obtemos a equação (3.2).

$$u_{i-1,j}^{k+1}(dsile) + u_{i,j-1}^{k+1}(dsipe) + u_{i,j}^{k+1}(dpe) + u_{i,j+1}^{k+1}(dsspe) + u_{i+1,j}^{k+1}(dssle) = \\ u_{i-1,j}^k(dsild) + u_{i,j-1}^k(dsipd) + u_{i,j}^k(dpd) + u_{i,j+1}^k(dsspd) + u_{i+1,j}^k(dssld) + f(k + \frac{1}{2})\Delta t \quad (3.2)$$

dsspe	D agonal S ecundária S uperior P erto E squerda	$-\alpha \frac{\Delta t}{2\Delta y^2}$
dsipe	D agonal S ecundária I nferior P erto E squerda	$-\alpha \frac{\Delta t}{2\Delta y^2}$
dssle	D agonal S ecundária S uperior L onge E squerda	$-\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x}$
dsile	D agonal S ecundária I nferior L onge E squerda	$-\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x}$
dpe	D agonal P rincipal E squerda	$1 + \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) + \delta \frac{\Delta t}{2}$
dsspd	D agonal S ecundária S uperior P erto D ireita	$\alpha \frac{\Delta t}{2\Delta y^2}$
dsipd	D agonal S ecundária I nferior P erto D ireita	$\alpha \frac{\Delta t}{2\Delta y^2}$
dssld	D agonal S ecundária S uperior L onge D ireita	$\alpha \frac{\Delta t}{2\Delta x^2} - v(y_j) \frac{\Delta t}{4\Delta x}$
dsild	D agonal S ecundária I nferior L onge D ireita	$\alpha \frac{\Delta t}{2\Delta x^2} + v(y_j) \frac{\Delta t}{4\Delta x}$
dpd	D agonal P rincipal E squerda	$1 - \alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) - \delta \frac{\Delta t}{2}$

Tabela 3.2: Termos utilizados na implementação do modelo discretizado

Utilizando, como exemplo, os nove nós da malha representada na figura (3.1) poderemos escrever a equação (3.2) utilizando matrizes conforme equação (3.3).

$$\begin{aligned}
& \begin{bmatrix} dpe & dsspe & 0 & dssle & 0 & 0 & 0 & 0 & 0 \\ dsipe & dpe & dsspe & 0 & dssle & 0 & 0 & 0 & 0 \\ 0 & dsipe & dpe & dsspe & 0 & dssle & 0 & 0 & 0 \\ dsile & 0 & dsipe & dpe & dsspe & 0 & dssle & 0 & 0 \\ 0 & dsile & 0 & dsipe & dpe & dsspe & 0 & dssle & 0 \\ 0 & 0 & dsile & 0 & dsipe & dpe & dsspe & 0 & dssle \\ 0 & 0 & 0 & dsile & 0 & dsipe & dpe & dsspe & 0 \\ 0 & 0 & 0 & 0 & dsile & 0 & dsipe & dpe & dsspe \\ 0 & 0 & 0 & 0 & 0 & dsile & 0 & dsipe & dpe \end{bmatrix} \begin{bmatrix} U_1^{k+1} \\ U_2^{k+1} \\ U_3^{k+1} \\ U_4^{k+1} \\ U_5^{k+1} \\ U_6^{k+1} \\ U_7^{k+1} \\ U_8^{k+1} \\ U_9^{k+1} \end{bmatrix} = \\
& \begin{bmatrix} dpd & dsspd & 0 & dssld & 0 & 0 & 0 & 0 & 0 \\ dsipd & dpd & dsspd & 0 & dssld & 0 & 0 & 0 & 0 \\ 0 & dsipd & dpd & dsspd & 0 & dssld & 0 & 0 & 0 \\ dsild & 0 & dsipd & dpd & dsspd & 0 & dssld & 0 & 0 \\ 0 & dsild & 0 & dsipd & dpd & dsspd & 0 & dssld & 0 \\ 0 & 0 & dsild & 0 & dsipd & dpd & dsspd & 0 & dssld \\ 0 & 0 & 0 & dsild & 0 & dsipd & dpd & dsspd & 0 \\ 0 & 0 & 0 & 0 & dsild & 0 & dsipd & dpd & dsspd \\ 0 & 0 & 0 & 0 & 0 & dsild & 0 & dsipd & dpd \end{bmatrix} \begin{bmatrix} U_1^k \\ U_2^k \\ U_3^k \\ U_4^k \\ U_5^k \\ U_6^k \\ U_7^k \\ U_8^k \\ U_9^k \end{bmatrix} + \begin{bmatrix} f_1^{k+\frac{1}{2}} \\ f_2^{k+\frac{1}{2}} \\ f_3^{k+\frac{1}{2}} \\ f_4^{k+\frac{1}{2}} \\ f_5^{k+\frac{1}{2}} \\ f_6^{k+\frac{1}{2}} \\ f_7^{k+\frac{1}{2}} \\ f_8^{k+\frac{1}{2}} \\ f_9^{k+\frac{1}{2}} \end{bmatrix} \Delta t
\end{aligned} \tag{3.3}$$

A equação (3.3) pode então ser escrita na forma matricial:

$$M_e U^{n+1} = M_d U^n + F^{n+\frac{1}{2}} \tag{3.4}$$

Para a implementação do programa em MatLab utilizamos então a representação matricial do problema em sua formulação discreta.

Vale destacar que as matrizes da equação (3.3) sofreram alterações para os nós das fronteiras, a fim de ajustá-las às condições de contorno, como indicado anteriormente, nas equações (2.43), (2.40) e (2.46).

Para solucionar o sistema dado pela equação (3.4) utilizamos **Fatoração LU**, que segundo BURDEN e FAIRES [7] (2008, p.371) pode reduzir a quantidade de cálculos em mais de 99%, e a evolução temporal dada através do método de Crank-Nicolson, que permitiram a obtenção de gráficos de contorno e de superfície que utilizamos para efetuar análises das simulações.

A interface gráfica da implementação

Com vistas a tornar o software útil no estudo de problemas e na análise qualitativa de estratégias propusemo-nos a desenvolver uma **GUI - Graphical User Interface** (Interface Gráfica de Usuário). Nessa GUI, figura (3.2), daremos condições para um usuário, independentemente de seu conhecimento em MatLab, de simular emissão de poluentes em meio aquático, levando em consideração duas dimensões (largura e comprimento) mais o tempo, bastando ao usuário informar dados referentes ao rio e ao poluente, além de permitir que a simulação seja feita com até 4 fontes distintas de poluição em pontos à escolha do usuário. Através dessa interface este usuário poderá gravar imagens em formato **PNG**¹ no intervalo de tempo desejado.

*GUI - Graphical
User Interface*

Utilizando o pacote **deploytool**² do MatLab geramos o arquivo executável a fim de permitir ao usuário que se execute a aplicação diretamente no Sistema Operacional³.

deploytool

3.2 Os parâmetros

De acordo com a resolução CONAMA N^o - 357, de 17 de março de 2005 [6] corpos de água superficiais, águas doces, salobras e salinas do Território Nacional, são classificados, segundo a qualidade requerida para os seus usos preponderantes, em treze classes de qualidade, e ainda, as águas doces de classe 2 deverão apresentar o seguinte padrão para óleos e graxas: virtualmente ausentes.

De acordo com a Associação Pró-gestão das Águas da Bacia Hidrográfica do Rio Paraíba do Sul - AGEVAP[4] o trecho do Rio Paraíba do Sul em estudo é classificado em classe 2.

Portanto adotamos com condição inicial:

¹ PNG - Portable Network Graphics: é um formato de dados utilizado para imagens

² Ferramenta do software MatLab utilizada para compilar e empacotar, utilizando a linguagem C. Após o empacotamento é gerado um programa de instalação que possibilita o usuário utilizar a aplicação em um computador que não possua o software MatLab instalado.

³ O arquivo executável gerado foi para plataforma Windows 32 bits.

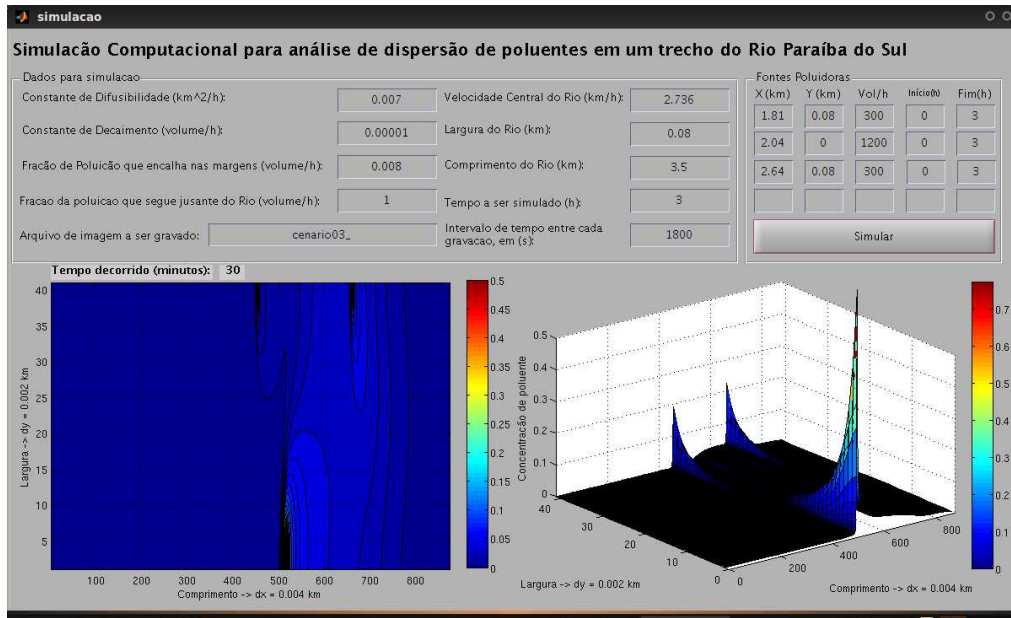


Figura 3.2: GUI - Graphical User Interface

$$U_i = 0 \text{ para } 1 \leq i \leq ni \quad (3.5)$$

onde:

ni	nx * nny
nx	número de nós no eixo x
nny	número de nós no eixo y

Tabela 3.3: Dimensão da matriz de nós para cálculo de poluição

Além da condição inicial devemos ainda definir, visando a execução do programa, alguns dados:

- o movimento da mancha na fronteira jusante, onde vale a condição de contorno, representada pela equação (1.7), iremos utilizar $k_2 = 1 \frac{\text{unidade de volume}}{h}$ para realizarmos os ensaios, pois assumimos passagem integral, no sentido normal, pela fronteira jusante, ou seja, não impedimentos à movimentação da mancha.
- o movimento da mancha na fronteira direita e esquerda, onde valem as condições de contorno, representadas pelas equações (1.5) e (1.6), iremos utilizar, segundo INFORZATO[14] (2008, p.61-63) $k_1 = 0.008 \frac{\text{unidade de volume}}{h}$ para realizarmos os ensaios, admitindo que há perda de poluição da água para as margens do rio.

3. o decaimento do óleo, segundo OLIVEIRA[17] (2003, p.67) ocorre em quantidade maior no início do derrame, estaremos considerando $\delta = 0.00001 \frac{\text{unidade de volume}}{h}$.
4. a difusão, constante, segundo OLIVEIRA[17] (2003, p.67) será $\alpha = 0.007 \text{ km}^2/h$.
5. a fonte de poluição, quando existir, terá vazão, tempo de derrame e local a escolha do usuário.
6. o passo no tempo $\Delta t = \frac{1}{3600}h$, ou seja, Δt equivale a um segundo.
7. $\Delta x = 0.004 \text{ km}$ e $\Delta y = 0.002 \text{ km}$
8. a velocidade central do rio, segundo ARAÚJO[3] (1989, p. J24), nesse trecho, é, em média, $v_0 = 2.736 \text{ km/h}$

Descrição	Constante	Unidade
Perda de poluente pelas margens esquerda e direita	$k_1 = 0.008$	$\frac{\text{unidade de volume}}{h}$
Perda de poluente na fronteira jusante	$k_2 = 1$	$\frac{\text{unidade de volume}}{h}$
Decaimento	$\delta = 0.00001$	$\frac{\text{unidade de volume}}{h}$
Difusividade	$\alpha = 0.007$	km^2/h
Passo no tempo	$\Delta t = \frac{1}{3600}$	h
Passo no eixo x	$\Delta x = 0.004$	km
Passo no eixo y	$\Delta y = 0.002$	km
Velocidade central	$v_0 = 2.736$	km/h

Tabela 3.4: Parâmetros utilizados nas simulações

3.3 Dificuldades

Destacamos a seguir algumas das dificuldades encontradas durante o desenvolvimento deste trabalho.

1. O número de Péclet, $Pe = v \frac{\Delta x}{\alpha}$, neste caso de estudo deve ser menor do que 2.
2. Como o método explícito para Diferenças Finitas, segundo CARNAHAN e outros [8] (1969 p.449), é condicionalmente estável, necessitamos de algo melhor (muito!). Utilizamos o método implícito para Diferenças Finitas que segundo CARNAHAN e outros [8] (1969 p. 450) é incondicionalmente estável.
3. Definição de um domínio que aproxima adequadamente Ω .

- Um dos objetivos deste trabalho é o de visualizar os resultados graficamente daí a necessidade da esparsidade pois trabalhamos com matrizes de ordem de 35875×35875 , ou seja 1287915625 elementos.

3.4 Cenários e simulações

De acordo com ARAÚJO[3] (1989, p. J19) no trecho do Rio Paraíba do Sul em estudo há três possíveis fontes de emissão de poluentes que são: Córrego dos Carvalhos (a 1,81 km - resíduos residenciais e de pequenas empresas), ponto de Emissão Principal da CSN (a 2,04 km - resíduos industriais) e Córrego dos Coqueiros (a 2,64 km - resíduos residenciais e de pequenas empresas). Ilustramos na figura (3.3), obtida pelo software GOOGLE Earth (2009), essas três possíveis fontes de poluição.



Figura 3.3: Trecho do Rio Paraíba do Sul destacando, possíveis, fontes poluidoras, Fonte: Google Earth, [12], 2009

Considerando esses três, possíveis, pontos de emissão de poluição realizaremos simulações a partir dos seguintes cenários:

1. Emissão constante de óleos nos Córregos dos Coqueiros e dos Carvalhos.
2. Somente um vazamento accidental de óleo oriundo de equipamentos da CSN.
3. Vazamento constante de óleo nos Córregos dos Coqueiros e dos Carvalhos e um vazamento accidental de óleo oriundos de equipamentos da CSN.

3.4.1 Cenário 1: Emissão constante de óleos nos Córregos dos Coqueiros e dos Carvalhos

Neste cenário a emissão de óleos (residenciais e industriais) foi considerada constante para os Córregos dos Carvalhos e dos Coqueiro, conforme tabela (3.5). Não havendo emissão de poluentes oriundos da CSN.

Nossa proposta, com este cenário, foi a de possibilitar a análise do estado desse trecho do Rio Paraíba do Sul levando em consideração apenas poluentes que são lançados diariamente, de forma indiscriminada, por representarem pequenas quantidades e não causarem alarmes ambientais. Vale destacar que a cidade de Volta Redonda - RJ, nesse trecho, não possui estação de tratamento de esgoto.

Ponto	Coordenadas (km)		Vazão (vol /h)	Período de Vazamento (h)	
	X	Y		Início	Término
Córrego dos Carvalhos	1.81	0.08	300	0	24
Córrego dos Coqueiros	2.64	0.08	300	0	24

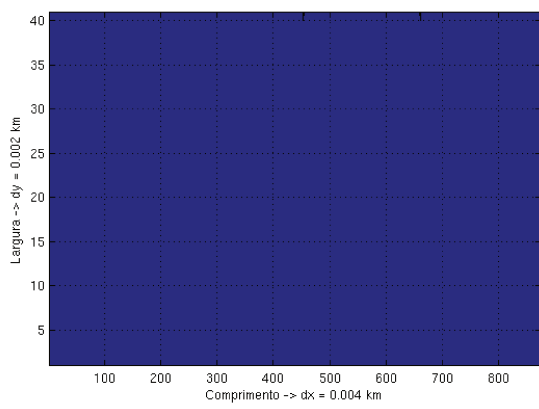
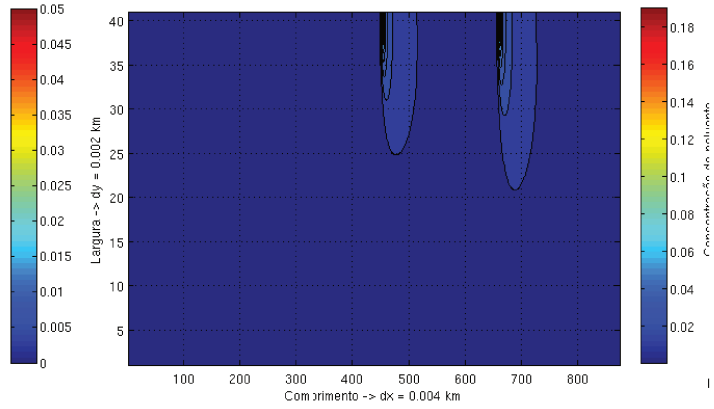
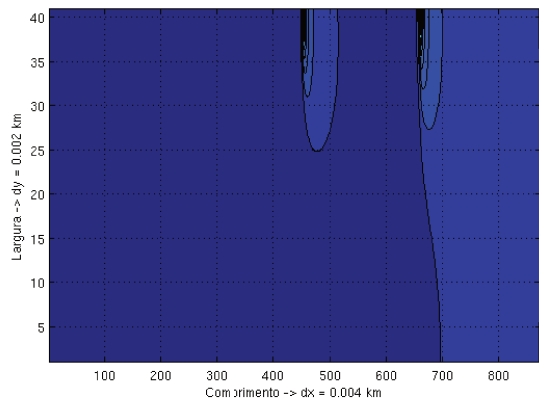
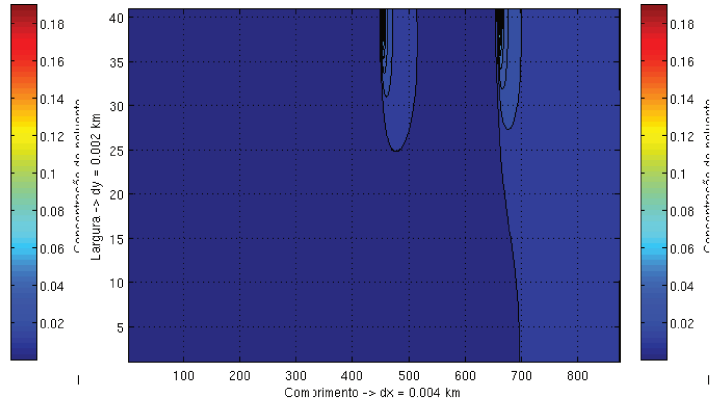
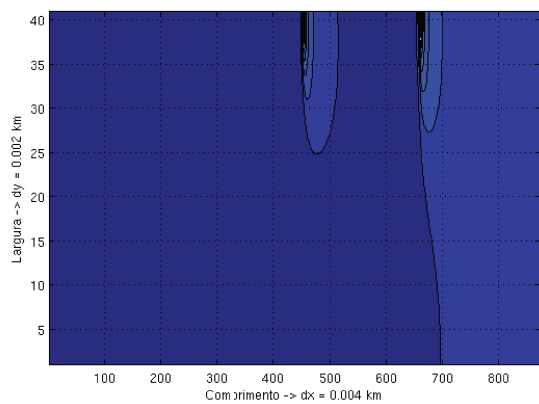
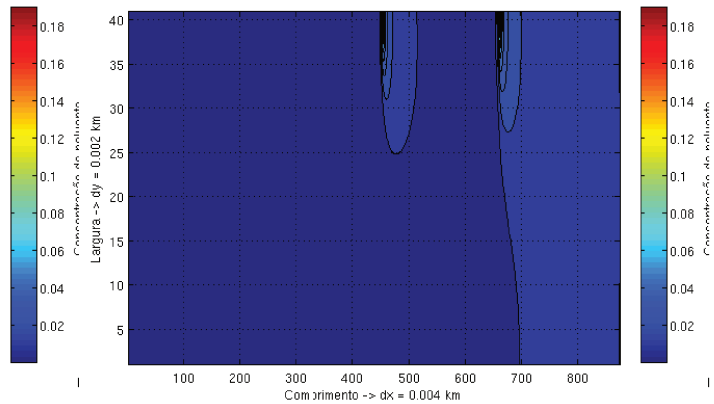
Tabela 3.5: Dados cenário 1

Com este cenário obtemos como saída as figuras (3.4), (3.5), (3.6), (3.7), (3.8), (3.9), (3.10) e (3.11).

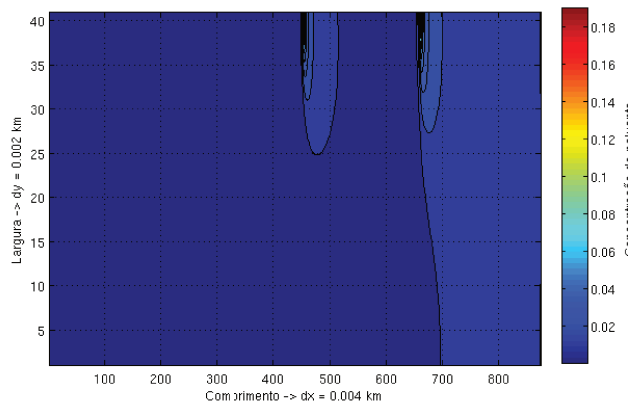
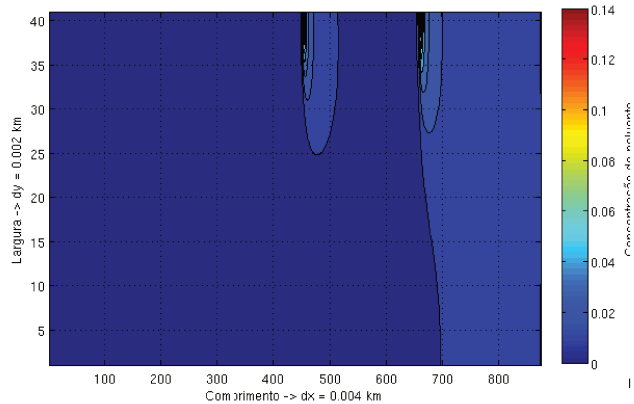
Nesta situação temos, após cerca de hora e meia uma situação estável à qual o sistema tende assintoticamente. Observamos que a poluição rapidamente atingiu a margem direita do rio também.

3.4.2 Cenário 2: Somente um vazamento accidental de óleo oriundo de equipamentos da CSN

Neste cenário a emissão de óleos (residenciais e industriais) foi considerada nula para os Córregos dos Carvalhos e dos Coqueiro. Propomos neste cenário emissão de poluentes oriundos da CSN

Figura 3.4: Estado com $t = 0$ horaFigura 3.5: Estado com $t = 0.5$ horaFigura 3.6: Estado com $t = 1$ horaFigura 3.7: Estado com $t = 2$ horasFigura 3.8: Estado com $t = 4$ horasFigura 3.9: Estado com $t = 8$ horas

devido a um acidente, conforme tabela (3.6).

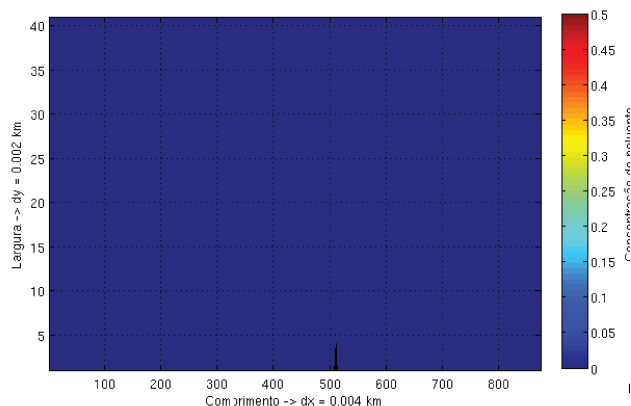
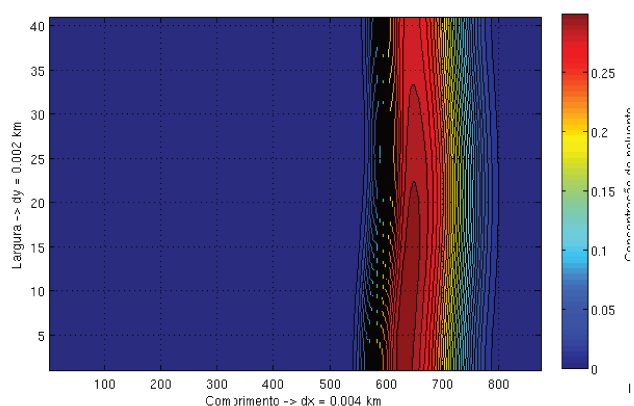
Figura 3.10: Estado com $t = 16$ horasFigura 3.11: Estado com $t = 24$ horas

Nossa proposta, com este cenário, foi a de possibilitar a análise do estado desse trecho do Rio Paraíba do Sul levando em consideração um acidente com vazamento de óleos oriundos dos equipamentos da CSN.

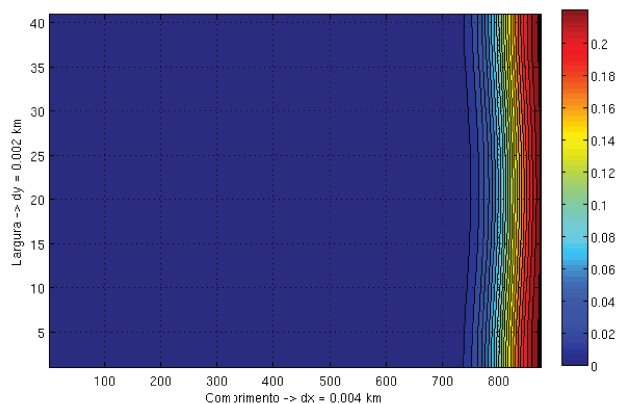
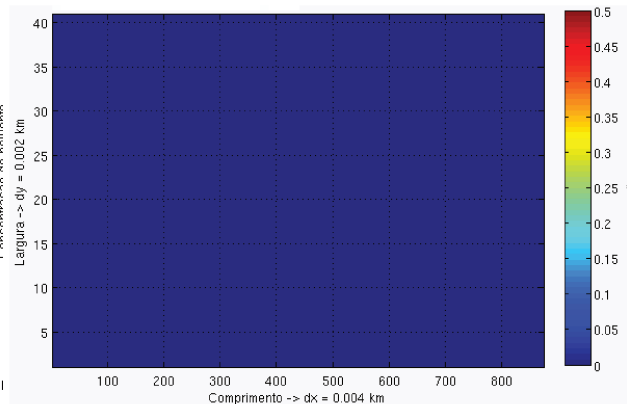
Ponto	Coordenadas (km)		Vazão	Período de Vazamento (h)	
	X	Y	(vol /h)	Início	Término
Emissão Principal	2.04	0	12000	0	0.3

Tabela 3.6: Dados cenário 2

Com este cenário obtemos como saída as figuras (3.12), (3.13), (3.14) e (3.15).

Figura 3.12: Estado com $t = 0$ horaFigura 3.13: Estado com $t = 0,5$ hora

Observamos que a poluição rapidamente atingiu a margem esquerda do rio formando uma faixa densa que se deslocou no sentido do escoamento. Em menos de duas horas esta faixa havia desaparecido do domínio em estudo.

Figura 3.14: Estado com $t = 1$ horaFigura 3.15: Estado com $t = 2$ horas

3.4.3 Cenário 3: Vazamento constante de óleo nos Córregos dos Coqueiros e dos Carvalhos e um vazamento acidental de óleo, oriundo de equipamentos da CSN

Neste cenário a emissão de óleos (residenciais e industriais) foi considerada constante para os Córregos dos Carvalhos e dos Coqueiro. Propomos neste cenário emissão de poluentes oriundos da CSN devido a um acidente, conforme tabela (3.7).

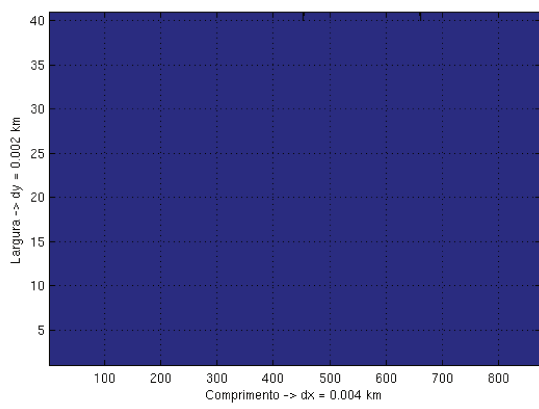
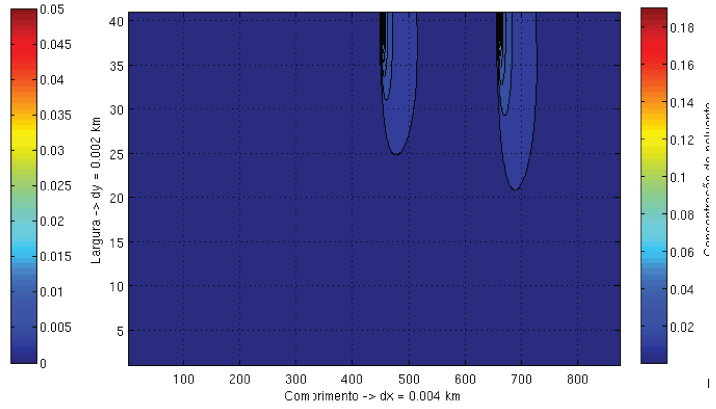
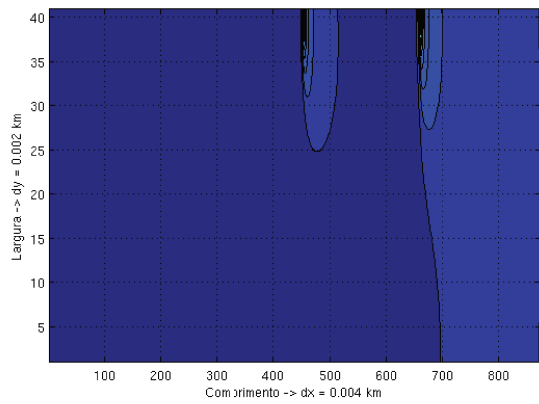
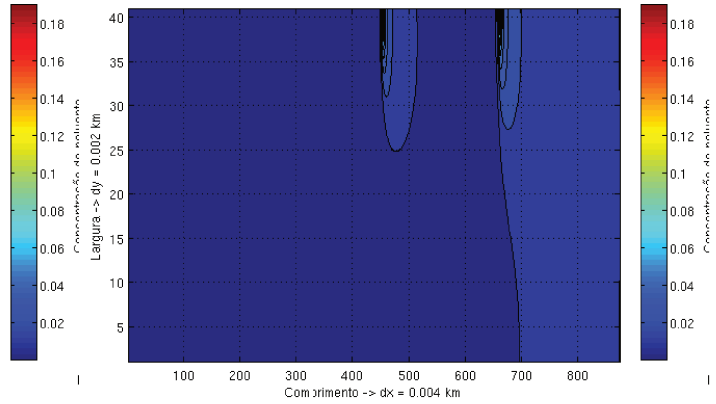
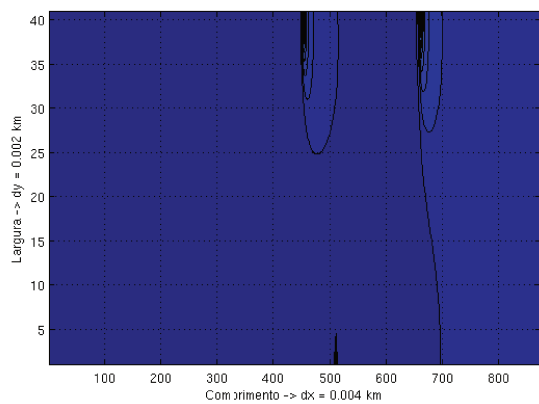
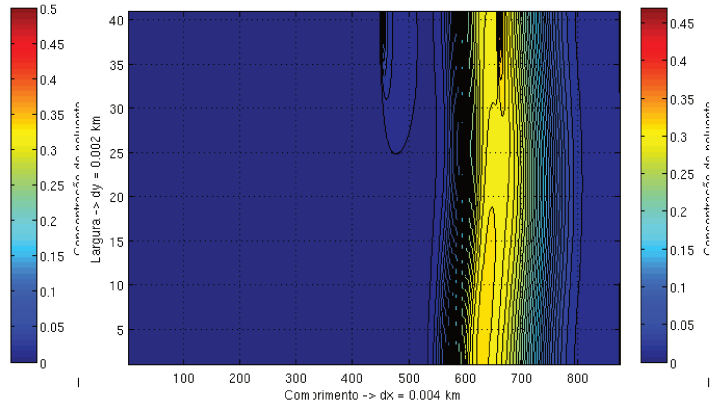
Nossa proposta, com este cenário, foi a de possibilitar a análise do estado desse trecho do Rio Paraíba do Sul levando em consideração um acidente com vazamento de óleos oriundos dos equipamentos da CSN em conjunto com emissão de óleos (residenciais e industriais) dos córregos dos Carvalhos e dos Coqueiros, ou seja, mesclamos o cenário 1 com o cenário 2.

Ponto	Coordenadas (km)		Vazão (vol /h)	Período de Vazamento (h)	
	X	Y		Início	Término
Córrego dos Carvalhos	1.81	0.08	300	0	24
Córrego dos Coqueiros	2.64	0.08	300	0	24
Emissão Principal	2.04	0	12000	4	4.3

Tabela 3.7: Dados cenário 3

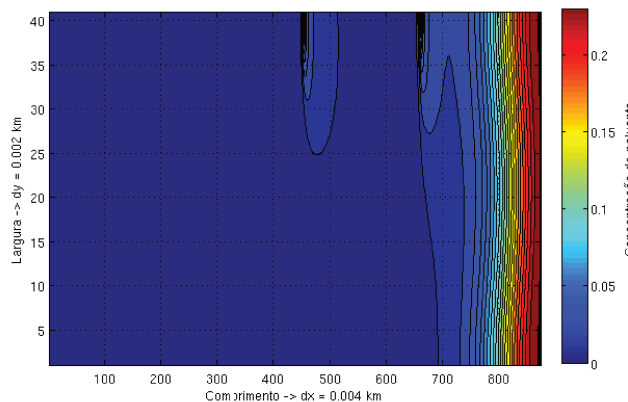
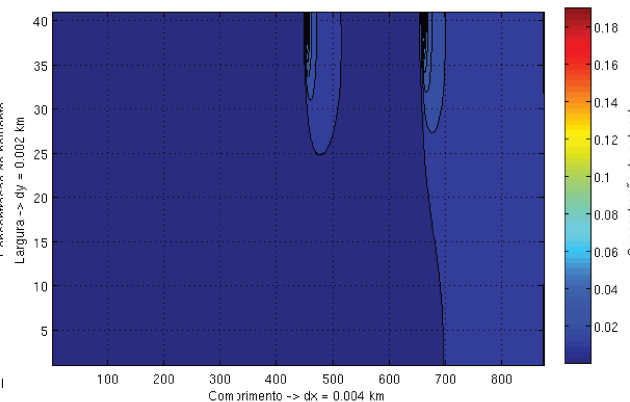
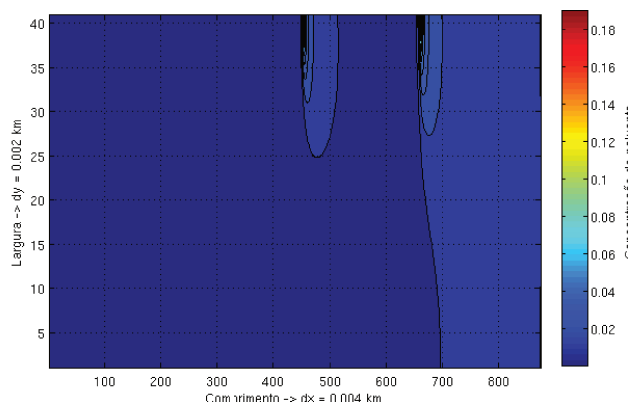
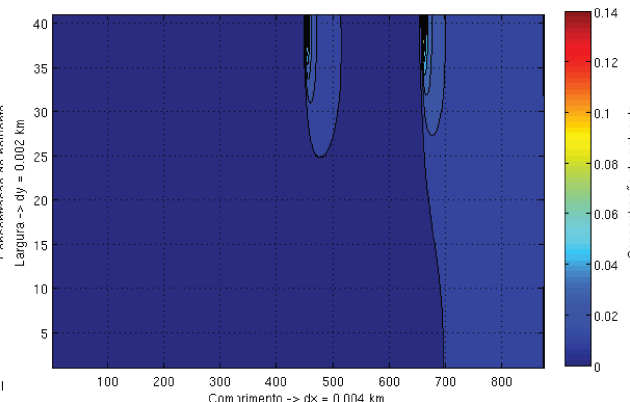
Com este cenário obtemos como saída as figuras (3.16), (3.17), (3.18), (3.19), (3.20), (3.21), (3.22), (3.23), (3.24) e (3.25).

Como na simulação realizada para cenário 1, apesar do derrame acidental, temos um quadro de estabilidade assintótico.

Figura 3.16: Estado com $t = 0$ horaFigura 3.17: Estado com $t = 0,5$ horaFigura 3.18: Estado com $t = 1$ horaFigura 3.19: Estado com $t = 2$ horasFigura 3.20: Estado com $t = 4$ horasFigura 3.21: Estado com $t = 4,5$ horas

3.5 Análise dos Resultados

Os gráficos das simulações dos três cenários revelam resultados compatíveis com o comportamento espaço-temporal esperado para manchas de óleos quando influenciadas por fenômenos de difusão,

Figura 3.22: Estado com $t = 5$ horasFigura 3.23: Estado com $t = 6$ horasFigura 3.24: Estado com $t = 12$ horasFigura 3.25: Estado com $t = 24$ horas

dispersão sob determinadas condições de fronteira e seus respectivos parâmetros.

Com as simulações realizadas com o primeiro cenário mostramos que é possível, mesmo em pequenas quantidades, determinar o comportamento da mancha de óleo, que é persistente, e que pode causar danos ambientais.

Já nas simulações do segundo e terceiro cenários mostramos que podemos determinar o comportamento da mancha de óleo causada por um acidente ambiental.

Com esses resultados, organizações governamentais e/ou não governamentais poderão tomar medidas de prevenção e/ou tratamento a fim de minimizar os efeitos negativos causados ao meio ambiente por emissão de poluentes.

Capítulo 4

Conclusão

Inicialmente apresentamos um problema real e atual de emissão de poluição, de uma importante região do estado do Rio de Janeiro, que pode ser modelado matematicamente com equação diferencial parcial e aproximado pelo método numérico de diferenças finitas em discretizações espacial e temporal.

Durante nosso estudo percebemos que há relativamente pouca literatura brasileira que trata do método de diferenças finitas em situações como esta. Levando esse fato em consideração resolvemos apresentar o Método de Diferenças Finitas nas variáveis espaciais e o Método de Crank-Nicolson na aproximação temporal a fim de construir uma nova referência instrumental para este problema.

Com modelo matemático discretizado desenvolvemos a implementação computacional, que, inicialmente, foi utilizada para simular emissão de poluentes (óleos) em um trecho do Rio Paraíba do Sul, porém, com o desenvolvimento da GUI¹ e a construção de aplicativo executável percebemos que havíamos desenvolvido um programa computacional que poderá ser utilizado para simular emissão de diversos tipos de poluentes e não, necessariamente, neste trecho do Rio Paraíba do Sul, pois, pela GUI, o usuário, mesmo que sem conhecimentos da linguagem utilizada no MatLab, poderá construir seus próprios cenários.

¹ GUI - Graphical User Interface (Interface gráfica do usuário)

Portanto concluímos que o “grande produto” deste trabalho foi a construção de um programa computacional, interligando a matemática com problema da vida real. Destacamos algumas das utilidades que poderão ser dadas a esse programa:

1. Para definir leis ambientais;
2. Por órgãos ambientais a fim de definirem padrões de emissão de poluição, procedimentos de tratamento a serem tomados em caso de acidentes;
3. Dimensionar o acidente ambiental;
4. Prevenir que estações de tratamento de água captem água contaminada para consumo humano;
5. Por escolas de Engenharia Ambiental a fim de simular acidentes; e, finalmente
6. Como o modelo adotado e o algoritmo comportam diferentes geometrias e perfis de velocidade, trabalhos como este podem (e devem!), mediante mudanças convenientes, prestar valioso serviço de avaliação de políticas públicas e estratégias de atendimento emergencial a acidentes com derrame de poluentes no rio, bem como no estabelecimento de planejamento a longo prazo em termos de preservação e proteção ambientais.

Referências Bibliográficas

- [1] ALEGRIA, Suely Aparecida Furlan. **Dispersão de poluentes em um trecho do Rio Coxipó: Modelagem e Aproximação Numérica**. 2006. Dissertação (Mestre) - Instituto de Ciências Exatas e da Terra, Universidade Federal de Mato Grosso, Cuiabá, 2006.
- [2] AGÊNCIA NACIONAL DE ÁGUAS. **Bacia do Rio Paraíba do Sul** - Regiões Hidrográficas. Disponível em: <http://www.ana.gov.br/bibliotecavirtual/arquivos/Bacia_Rio_Paraiba_do-Sul_RegioesHidrograficas.pdf>. Acesso em: 07 de set. 2009.
- [3] ARAÚJO, Marcus Vinicius Faria.; VEIGA, Fernando César Vieira. **Estudo do Comportamento Depurativo do Rio do Paraíba do Sul no Trecho Receptor de Poluentes Líquidos da CSN** In: SIMPÓSIO SOBRE CONTROLE AMBIENTAL NA SIDERURGIA, 4., 1989, Espírito Santo. p. J1-J36
- [4] ASSOCIAÇÃO PRÓ-GESTÃO DAS ÁGUAS DA BACIA HIDROGRÁFICA DO RIO PARAÍBA DO SUL - AGEVAP. **Plano de Recursos Hídricos da Bacia do Rio Paraíba do Sul**. Rio de Janeiro: Fundação COPPETEC, (2007). 147 p. (Relatório Contratual R-10, PSR-012-R0).
- [5] BORTOLOSSI, Humberto José. ***Cálculo diferencial a várias variáveis***: Uma introdução à teoria de otimização; (2002) ed. Loyola, (Rio de Janeiro), ISBN 85-15-02442-X
- [6] BRASIL. Conselho Nacional do Meio Ambiente. Resolução nº 357, de 17 de março de 2005. **Dispõe sobre a classificação dos corpos de água e diretrizes ambientais para o seu enquadramento, bem como estabelece as condições e padrões de lançamento de efluentes, e dá outras providências.**, Brasília, DF, Diário Oficial, p. 58-63, 18 de março de 2005.
- [7] BURDEN, Richard L., FAIRES, J. Douglas. ***Análise Numérica***; (2008) ed. Cengage Learning, (São Paulo), ISBN 978-85-221-0601-1
- [8] CARNAHAN, Brice, LUTHER, H. A., WILKES, James O.. ***Applied Numerical Methods***; (1969) ed. John Wiley & Sons, (New York), SBN 471-13507-0

- [9] COMITÊ PARA INTEGRAÇÃO DA BACIA HIDROGRÁFICA DO RIO PARAÍBA DO SUL - CEIVAP. **Bacia Hidrográfica do Rio Paraíba do Sul - Histórico da região**. Disponível em: <http://ceivap.org.br/bacia_1.1.php>. Acesso em: 02 set. 2009.
- [10] COMITÊ PARA INTEGRAÇÃO DA BACIA HIDROGRÁFICA DO RIO PARAÍBA DO SUL - CEIVAP. **Bacia do Rio Paraíba do Sul - Dados Geoambientais**. Disponível em: <http://ceivap.org.br/bacia_1.2.php>. Acesso em: 02 set. 2009.
- [11] FORTUNA, Armando de Oliveira. **Técnicas Computacionais para Dinâmica dos Fluidos** Conceitos Básicos e Aplicações; (2000) ed. Editora da Universidade de São Paulo, (São Paulo), ISBN 85-314-0526-2
- [12] GOOGLE Earth. Version 5.0.11733.9347: Google Inc., 2009 Disponível em: <<http://earth.google.com.br/>>, Acesso em: 27 ago. de 2009.
- [13] GOVERNO do RJ interdita unidade da CSN após novo vazamento. **Globo.com Notícias**, Rio de Janeiro, 06 de ago. 2009. Disponível em: <<http://g1.globo.com/Noticias/Rio/0,,MUL1257868-5606,00.html>>. Acesso em: 07 de set. 2009.
- [14] INFORZATO, Nelson Fernando. **Dispersão de poluentes num sistema ar-água: modelagem matemática, aproximação numérica e simulação computacional**. 2008. Dissertação (Doutorado) - Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, Campinas, 2008.
- [15] LIMA, Elon Lages. **Curso de Análise** Volume 2; 10^a edição (2008) ed. IMPA, (Rio de Janeiro), ISBN 978-85-244-0049-0, pp 151-159
- [16] NOVAIS, Amélia, CUNHA, Maria C. C.. **Métodos Numéricos para Equações Diferenciais Parciais**; (2003) ed. SBMAC, (São Paulo)
- [17] OLIVEIRA, Rosane Ferreira de. **O Comportamento evolutivo de uma mancha de óleo na Baía de Ilha Grande, RJ: modelagem, análise numérica e simulações**. 2003. Dissertação (Doutorado) - Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, Campinas, 2003.
- [18] RIO PARAÍBA: multa pode chegar a R\$ 10 milhões. **O Globo**, Rio de Janeiro, 20 de nov. 2008. Disponível em: <http://oglobo.globo.com/rio/mat/2008/11/20/rio_paraiba-multa_pode_chegar_r_10_milhoes-586491417.asp>. Acesso em: 02 de set. 2009.
- [19] VÁSQUEZ, Júlio Cesar Saavedra. **Comportamento Evolutivo de Descarga de Água de Produção Decorrente de Atividade Offshore: Tratamento Numérico e Simulação Computacional**. 2005. Dissertação (Doutorado) - Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, Campinas, 2005.

Apêndice

Código fonte

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Aplicacao desenvolvida por: %
% Jose Ricardo Ferreira de Almeida %
% Sob orientacoes dos professores: %
% Dr. Joao Frederico da Costa Azevedo Meyer %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Esta aplicacao e parte da dissertacao de mestrado %
% apresentada no IMECC/Unicamp %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function varargout = simulacao(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @simulacao_OpeningFcn, ...
                  'gui_OutputFcn', @simulacao_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function simular(handles)
    % %% - Dados do problema - %% %
    format long;
```

```

% Constante de difusibilidade: km^2/h
alfa = str2double(get(handles.txtAlfa,'String'));
% Constante de decaimento: 1/h^-1
delta = str2double(get(handles.txtDelta,'String'));
% Constante de encalhe nas margens direita e esquerda
k1 = str2double(get(handles.txtK1,'String'));
% Constante de poluicao segue jusante
k2 = str2double(get(handles.txtK2,'String'));

%% - Dados do domínio - %%
% Comprimento: km
C = str2double(get(handles.txtC,'String'));
% Largura: km
L = str2double(get(handles.txtL,'String'));
% Velocidade central km/h
vz = str2double(get(handles.txtVz,'String'));
% Tempo total a ser simulado: h
T = str2double(get(handles.txtT,'String'));

%% - Dados da discretizacao - %%
dx = 0.004;
dy = 0.002;
dt = 1 / 3600;
nx = C/dx;      % quantidade de divisoes no eixo x
ny = L/dy;      % quantidade de divisoes no eixo y
nny = ny + 1;
ni = nx * nny;  % numero de nos

nPeclet = vz * (dx / alfa);

if (nPeclet < 2)
    %% - Calculos auxiliares - %%
    adtx = alfa / (2 * dx * dx);
    adty = alfa / (2 * dy * dy);
    deltadt = delta * dt / 2;
    pv = dt / (4 * dx);

    % Diagonal principal esquerda
    dpe = 1 + 2 * dt * (adtx + adty) + deltadt;
    % Diagonal principal direita
    dpd = 1 - 2 * dt * (adtx + adty) - deltadt;

    dsipe = - adty * dt;      % Diagonal secundaria inferior perto esquerda
    dsspe = - adty * dt;      % Diagonal secundaria superior perto esquerda
    dsipd =  adty * dt;      % Diagonal secundaria inferior perto esquerda
    dsspd =  adty * dt;      % Diagonal secundaria superior perto esquerda

```

```

kk1 = k1 * dt / dy;
kk2 = k2 * dt / (2 * alfa * dx);

% %% - Preencher matrizes - %% %
me = sparse(ni, ni);
md = sparse(ni, ni);
% %% - Condicao inicial - %% %
uPassado = zeros(ni,1);
% %% - Preencher diagonal principal e secundarias- %% %
% diagonal principal
for i=1:ni
    me(i,i) = dpe;
    md(i,i) = dpd;
end
% diagonal secundaria
for i=1:ni-1
    me(i+1,i) = dsipe;
    me(i,i+1) = dsspe;
    md(i+1,i) = dsipd;
    md(i,i+1) = dsspd;
end

% %% - Preparar calculos para diagonais secundarias distantes - %% %
h = L/2;
axh = 1 / (h*h);
for j=1:nny
    y(j) = -h + (j-1)*dy;
    v(j) = vz * (1 - y(j)*y(j)*axh);
end
for i=1:nx
    for j=1:nny
        ind = (i-1)*nny + j;
        vi(ind) = v(j);
    end
end
% %% - Preencher diagonais secundarias distantes - %% %
for i=1:ni-nny
    me(i+nny,i) = -adtx * dt - vi(i) * pv;
    me(i,i+nny) = -adtx * dt + vi(i) * pv;
    md(i+nny,i) = adtx * dt + vi(i) * pv;
    md(i,i+nny) = adtx * dt - vi(i) * pv;
end
% %% - Corrigir matrizes me e md borda superior - %% %
for k=1: nx -1
    i = k * nny;
    me(i,i+1) = 0;
    me(i,i-1) = -2 * adty * dt;

```

```

        md(i,i+1) = 0;
        md(i,i-1) = 2 * adty * dt ;
    end
    me(ni,ni-1) = -2 * adty * dt;
    md(ni,ni-1) = 2 * adty * dt ;
    % %% - Corrigir matrizes me e md borda inferior - %% %
    for k=2: nx
        i = (k-1) * nny + 1;
        me(i,i+1) = -2 * adty * dt;
        me(i,i-1) = 0;
        md(i,i+1) = 2 * adty * dt;
        md(i,i-1) = 0;
    end
    me(1,2) = -2*adty*dt;
    md(1,2) = 2*adty*dt;
    % %% - Tratamento da diagonal principal da borda sup. e inf. - %% %
    % Diagonal principal esquerda
    dpe = 1 + 2 * dt * (adtx + adty) + deltadt + kk1;
    % Diagonal principal direita
    dpd = 1 - 2 * dt * (adtx + adty) - deltadt - kk1;
    for k=1:nx
        i = k*nny;
        me(i,i) = dpe;
        md(i,i) = dpd;
        i = (k-1)*nny + 1;
        me(i,i) = dpe;
        md(i,i) = dpd;
    end
    % %% - Tratamento da borda lateral direita - %% %
    % Diagonal principal esquerda
    dpe = 1 + 2 * dt * (adtx + adty) + deltadt + 2*alfa*kk2;
    % Diagonal principal direita
    dpd = 1 - 2 * dt * (adtx + adty) - deltadt - 2*alfa*kk2;

    for k=1:nny
        i = (nx -1) * nny + k;
        me(i,i) = dpe - vi(i) * kk2 * dx;
        md(i,i) = dpd + vi(i) * kk2 * dx;

        me(i,i-nny) = -2 * adtx * dt;
        md(i,i-nny) = 2 * adtx * dt;
    end

    % %% - Fontes Poluidoras - %% %
    fp = zeros(ni,1);
    fonte = zeros(4,4);

```

```

x1 = str2double(get(handles.txtX1,'String'));
y1 = str2double(get(handles.txtY1,'String'));
fonte(1,1) = floor(nx * x1 / C)*nny + floor(nny * y1 / L);
if (y1 == 0)
    fonte(1,1) = fonte(1,1) + 1;
end
fonte(1,2) = str2double(get(handles.txtV1,'String'));
fonte(1,3) = str2double(get(handles.txtI1,'String'));
fonte(1,4) = str2double(get(handles.txtF1,'String'));

x2 = str2double(get(handles.txtX2,'String'));
y2 = str2double(get(handles.txtY2,'String'));
fonte(2,1) = floor(nx * x2 / C)*nny + floor(nny * y2 / L);
if (y2 == 0)
    fonte(2,1) = fonte(2,1) + 1;
end
fonte(2,2) = str2double(get(handles.txtV2,'String'));
fonte(2,3) = str2double(get(handles.txtI2,'String'));
fonte(2,4) = str2double(get(handles.txtF2,'String'));

x3 = str2double(get(handles.txtX3,'String'));
y3 = str2double(get(handles.txtY3,'String'));
fonte(3,1) = floor(nx * x3 / C)*nny + floor(nny * y3 / L);
if (y3 == 0)
    fonte(3,1) = fonte(3,1) + 1;
end
fonte(3,2) = str2double(get(handles.txtV3,'String'));
fonte(3,3) = str2double(get(handles.txtI3,'String'));
fonte(3,4) = str2double(get(handles.txtF3,'String'));

x4 = str2double(get(handles.txtX4,'String'));
y4 = str2double(get(handles.txtY4,'String'));
fonte(4,1) = floor(nx * x4 / C)*nny + floor(nny * y4 / L);
if (y4 == 0)
    fonte(4,1) = fonte(4,1) + 1;
end
fonte(4,2) = str2double(get(handles.txtV4,'String'));
fonte(4,3) = str2double(get(handles.txtI4,'String'));
fonte(4,4) = str2double(get(handles.txtF4,'String'));

% %% - Repeticao Temporal - %% %
[lm um] = lu(me);
visu = zeros(nny,nx);
contorno = handles.grafContorno;
superficie = handles.grafSuperficie;
lblTempo = handles.txtTempoSimulacao;
intervalo = str2num(get(handles.txtIntervalo,'String'));

```

```

nomeDoArquivo = get(handles.txtArquivo,'String');
niveisPoluicao = 0:0.01:0.5;
for it=0:dt:T
    if (it == fonte(1,3))
        fp(fonte(1,1),1) = fonte(1,2) * dt;
    end
    if (it == fonte(1,4))
        fp(fonte(1,1),1) = 0;
    end
    if (it == fonte(2,3))
        fp(fonte(2,1),1) = fonte(2,2) * dt;
    end
    if (it == fonte(2,4))
        fp(fonte(2,1),1) = 0;
    end
    if (it == fonte(3,3))
        fp(fonte(3,1),1) = fonte(3,2) * dt;
    end
    if (it == fonte(3,4))
        fp(fonte(3,1),1) = 0;
    end
    if (it == fonte(4,3))
        fp(fonte(4,1),1) = fonte(4,2) * dt;
    end
    if (it == fonte(4,4))
        fp(fonte(4,1),1) = 0;
    end

    y = lm \ (md * uPassado + fp);
    u = um \ y;
    uPassado = u;

    for j=2:nx
        for i=1:nny
            ind = (j-1)*nny + i;
            visu(i,j) = u(ind);
        end
    end

    if mod(it*3600,intervalo) == 0
        axes(contorno);
        contourf(visu,niveisPoluicao);
        %clabel(c,'FontSize',8,'BackgroundColor','w','Rotation',0);
        ylabel 'Largura -> dy = 0.002 km';
        xlabel 'Comprimento -> dx = 0.004 km';
        colorbar;
        grid on;
    end
end

```

```

        arquivo = strcat('graficos/', nomeDoArquivo, 'C', num2str(it*3600), '.png');
        winsize = [-40 -40 600 390];
        X=getframe(gca, winsize);
        imwrite(X.cdata, arquivo)

        axes(superficie);
        surf(visu);
        zlabel 'Concentrac o de poluente';
        ylabel 'Largura -> dy = 0.002 km';
        xlabel 'Comprimento -> dx = 0.004 km';
        axis([0 nx 0 nny 0 0.5]);
        colorbar;
        arquivo = strcat('graficos/', nomeDoArquivo, 'S', num2str(it*3600), '.png');
        winsize = [-48 -40 600 390];
        X=getframe(gca, winsize);
        imwrite(X.cdata, arquivo)
        set(lblTempo,'String', (it*60));
        pause(0.05);
    end
end
else
    error('Numero de Plecet deve ser menor do que dois',
'Simulacador JRFA', 'modal')
end

function simulacao_OpeningFcn(hObject, eventdata, handles, varargin)
    handles.output = hObject;
    guidata(hObject, handles);

function varargout = simulacao_OutputFcn(hObject, eventdata, handles)
    varargout{1} = handles.output;

function txtAlfa_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtDelta_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtVz_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```
        set(hObject,'BackgroundColor','white');
    end

function txtL_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtC_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtT_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtK1_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtK2_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function bttSimular_Callback(hObject, eventdata, handles)
    simular(handles)

function txtAlfa_Callback(hObject, eventdata, handles)

function txtDelta_Callback(hObject, eventdata, handles)

function txtK1_Callback(hObject, eventdata, handles)

function txtK2_Callback(hObject, eventdata, handles)

function txtVz_Callback(hObject, eventdata, handles)

function txtL_Callback(hObject, eventdata, handles)
```

```
function txtC_Callback(hObject, eventdata, handles)

function txtT_Callback(hObject, eventdata, handles)

function txtX1_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtY1_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtV1_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtI1_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtF1_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtX2_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtY2_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
```

```
end

function txtV2_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtI2_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtF2_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtX3_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtY3_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtV3_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtI3_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtF3_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
        set(hObject,'BackgroundColor','white');
    end

function txtX4_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtY4_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtV4_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtI4_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtF4_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtX1_Callback(hObject, eventdata, handles)

function txtY1_Callback(hObject, eventdata, handles)

function txtV1_Callback(hObject, eventdata, handles)

function txtI1_Callback(hObject, eventdata, handles)

function txtF1_Callback(hObject, eventdata, handles)

function txtX2_Callback(hObject, eventdata, handles)

function txtY2_Callback(hObject, eventdata, handles)
```

```
function txtV2_Callback(hObject, eventdata, handles)

function txtI2_Callback(hObject, eventdata, handles)

function txtF2_Callback(hObject, eventdata, handles)

function txtX3_Callback(hObject, eventdata, handles)

function txtY3_Callback(hObject, eventdata, handles)

function txtV3_Callback(hObject, eventdata, handles)

function txtI3_Callback(hObject, eventdata, handles)

function txtF3_Callback(hObject, eventdata, handles)

function txtX4_Callback(hObject, eventdata, handles)

function txtY4_Callback(hObject, eventdata, handles)

function txtV4_Callback(hObject, eventdata, handles)

function txtI4_Callback(hObject, eventdata, handles)

function txtF4_Callback(hObject, eventdata, handles)

function txtArquivo_Callback(hObject, eventdata, handles)

function txtArquivo_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function txtIntervalo_Callback(hObject, eventdata, handles)

function txtIntervalo_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end
```

Índice Remissivo

diferenças finitas, 12, 13

GUI - Graphical User Interface, 26

condições de contorno homogênea de Dirichlet
(1.2) e de Robin (1.3), 7

deploytool, 26

Equação Diferencial Parcial de Difusão-Advecção-
Reação, 6

equação do modelo discretizada, 17

fronteira a jusante do rio, 8

fronteira a montante do rio, 7

fronteira referente às margens direita e esquerda,
7

implementação algorítmica, 22

método de Crank-Nicolson, 15, 16

Núcleo de Péclet, 21

operador de diferença central , 14

perfil de velocidades, 8

perfil parabólico, 8

Polinômio de Taylor, 10

trecho em destaque será o utilizado para as si-
mulações, 5

trechos estaduais que compõem o vale do rio Paraíba
do Sul, 1