



Universidade Estadual de Campinas
Instituto de Matemática Estatística e Computação Científica
DEPARTAMENTO DE MATEMÁTICA APLICADA



Modelagem e simulação computacional de um problema tridimensional de difusão-advecção com uso de Navier-Stokes.

Andre Krindges

Doutorado em Matemática Aplicada

Orientador: **Prof. Dr. João Frederico da Costa Azevedo Meyer**

Campinas - SP

julho de 2011

Modelagem e simulação computacional de um problema tridimensional de difusão-advecção com uso de Navier-Stokes.

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por Andre Krindges e aprovada pela comissão julgadora.

Campinas, 8 de julho de 2011.



Prof. Dr. João Frederico da Costa Azevedo Meyer

Orientador

Banca Examinadora:

- 1 – Prof. Dr. João Frederico da Costa Azevedo Meyer
- 2 – Prof. Dr. Laécio Carvalho de Barros
- 3 – Prof^a. Dr^a. Véra Lucia da Rocha Lopes
- 4 – Prof^a. Dr^a. Rosana Sueli da Motta Jafelice
- 5 – Prof. Dr. Mauro Cirano

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

FICHA CATALOGRÁFICA ELABORADA POR ANA REGINA MACHADO – CRB8/5467
BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E
COMPUTAÇÃO CIENTÍFICA – UNICAMP

K897m	<p>Krindges, Andre, 1978- Modelagem e simulação computacional de um problema tridimensional de difusão-advecção com uso de Navier-Stokes / Andre Krindges. – Campinas, SP : [s.n.], 2011.</p> <p>Orientador: João Frederico da Costa Azevedo Meyer. Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.</p> <p>1. Ecologia - Matemática. 2. Método dos elementos finitos. 3. Biomatemática. 4. Navier-Stokes, Equações de. 5. Simulação (Computadores). I. Meyer, João Frederico da Costa Azevedo, 1947-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.</p>
-------	---

Informações para Biblioteca Digital

Título em inglês: Modeling and computer simulation of a three-dimensional problem of diffusion-advection using the Navier-Stokes equations

Palavras-chave em inglês:

Ecology - Mathematics

Finite element method

Biomathematics

Navier-Stokes equations

Computer simulation

Área de concentração: Matemática Aplicada

Titulação: Doutor em Matemática Aplicada

Banca examinadora:

João Frederico da Costa Azevedo Meyer [Orientador]

Laécio Carvalho de Barros

Véra Lucia da Rocha Lopes

Rosana Sueli da Motta Jafelice

Mauro Cirano

Data da defesa: 08-07-2011

Programa de Pós Graduação: Matemática Aplicada

Tese de Doutorado defendida em 08 de julho de 2011 e aprovada

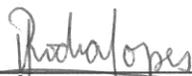
Pela Banca Examinadora composta pelos Profs. Drs.



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



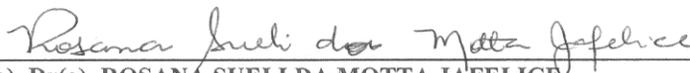
Prof(a). Dr(a). LAÉCIO CARVALHO DE BARROS



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



Prof(a). Dr(a). MAURO CIRANO



Prof(a). Dr(a). ROSANA SUELI DA MOTTA JAFELICE

Ao passado que **ainda** se faz presente:

Marta (*in memoriam*)

e ao futuro que **já** se faz presente:

Bento e Pedro.

Resumo

Um dos problemas enfrentados pelo grupo de Ecologia Matemática do IMECC da UNICAMP é o de trabalhar com difusão de uma pluma poluente com 3 variáveis espaciais, além da temporal. Esta tese não só aborda esta questão, propondo, inclusive um algoritmo computacional para esta situação, mas fá-lo resolvendo aproximadamente a Equação de Navier-Stokes num domínio irregular.

A primeira parte consiste na formulação do modelo matemático para o estudo de um sistema que inclui o campo de velocidades e o comportamento evolutivo de um material poluente. Na segunda parte, é feita a formulação variacional, são constituídas aproximações via o método de Galerkin para Elementos Finitos no espaço e Crank-Nicolson no tempo para a equação de difusão-advecção, e o método da projeção para a equação de Navier-Stokes.

Em seguida, faz-se a descrição do algoritmo, indicando dificuldades operacionais do ponto de vista de computação científica e apontando soluções. O domínio utilizado para o estudo de caso é o da represa do rio Manso, que, discretizada em três dimensões, foi tratado com o software livre GMSH. Finalmente, um código numérico em plataforma MATLAB foi executado e resultados são apresentados no texto. O programa e diversas considerações técnicas essenciais fazem parte dos anexos.

Palavras chave: Ecologia Matemática; Elementos Finitos; Biomatemática; Navier-Stokes; Simulação computacional.

Abstract

One of the challenges faced by the Mathematical Ecology group at the Mathematics Institute at Campinas State University is that of working with the diffusion of a pollutant plume in three spatial variables, besides time. This work not only addresses this issue by proposing an approximation strategy as well as a computer algorithm for this situation, but it also includes a three-dimensional numerical approximation for the Navier-Stokes equation in an irregular domain.

The first part consists in formulating the mathematical model for the study of a system that includes the velocity field and the evolutionary behavior of a polluting material. The second part begins with the variational formulation of the Navier-Stokes system, and approximations are undertaken via the Galerkin method for finite elements in space and Crank-Nicolson in time for both the advection-diffusion equation and the method of projection for the Navier-Stokes equations.

The algorithm is described, indicating operational difficulties in terms of scientific computing as well as the way in which these aforementioned difficulties are solved. The domain used for the case study is the Manso River reservoir, which, discretized in three dimensions, was treated with the free software GMSH. Finally, a numeric code in MATLAB environment was completed and results are presented in the text. The program and various essential technical considerations are part of the annexes.

Key words: Mathematical Ecology; finite elements; Biomathematics; Navier-Stokes equations, Computer simulation.

Sumário

Introdução	1
1 O problema: histórico e pretensões	3
1.1 Histórico: APM-Manso	3
1.2 Histórico: trabalhos anteriores	6
1.3 Descrição da proposta	8
2 A modelagem clássica do problema	11
2.1 Difusão-Advecção	11
2.2 Navier-Stokes	13
2.3 Considerações e adaptações do modelo	15
3 Formulações variacionais	16
3.1 Formulação variacional: Difusão-Advecção	16
3.2 Formulação variacional: Navier-Stokes	18
4 Discretizações	21
4.1 Discretizações espaciais	21
4.2 Discretizações temporais	27
4.3 Discretização do domínio	28
5 Experiências numéricas	33
5.1 Navier-Stokes	33
5.2 Cenários para poluição	36

Comentários e sugestões.	47
Anexo 1: Tratamento dos dados sobre a geometria e criação da malha	49
Anexo 2: Cálculos necessários para a implementação	54
Anexo 3: Recursos computacionais e códigos em MatLab [®] .	62
Referências bibliográficas	70

Lista de Figuras

1.1	Localização da Represa da APM-Manso. Fonte:[13].	5
1.2	Imagem do satélite Landsat5/TM (RGB 345).	6
1.3	Lago do Manso. Foto: Marcos Vergueiro-Secom-MT 25-08-2010.	7
1.4	Partes da fronteira: bordas do lago.	9
1.5	Exemplo da discretização espacial tridimensional - fundo e superfície de uma pequena parte do domínio do problema teste.	10
1.6	Imagem tridimensional do lago inteiro.	10
4.1	Transformação afim $T_e : \hat{k} \rightarrow k_e$	30
4.2	Imagem produzida pelo GMSH, ilustrando a malha.	31
4.3	Visão da malha produzida pelo GMSH, de baixo para cima.	32
4.4	Visão da malha na superfície do lago.	32
4.5	Recurso de transparência nos tetraedros.	32
5.1	Ampliação de um quadro próximo a entrada do rio Manso.	34
5.2	Campo de velocidades obtido das simulações.	35
5.3	Campo obtido das simulações com velocidades normalizadas.	35
5.4	Norma da diferença entre iterações.	37
5.5	Pontos analisados.	39
5.6	Linhas: iterações 1.000, 2.000, 3.000, 3.500 e 4.000. Colunas: cenários 1, 2 e 3.	41
5.7	Linhas: iterações 10.000, 50.000, 100.000, 200.000 e 600.000. Colunas: cenários 1, 2 e 3.	42
5.8	Linhas: níveis 0, 10, 20, 30 e 40m de profundidade. Colunas: cenários 1, 2 e 3.	43
5.9	Cenário 1: iterações de 350 a 2.450 ($t = 17,5$ horas a $t = 5.1$ dias).	44

5.10	Colunas de gráficos: iterações de 1 a 50.000 e de 1 a 600.000. Linhas de gráficos: cenários.	45
5.11	Gráficos comparativos de cada ponto nos 3 cenários.	46
A1.1	Curvas de nível da região do lago.	51
A1.2	Curvas de nível do lago: variação de cores indicando profundidade.	52
A1.3	Esquerda: curva original. Direita: depois de filtrada.	52
A1.4	Corte na malha numa região próxima à barragem.	53

Introdução

Nos tempos atuais, os avanços na compreensão dos fenômenos da natureza e a respectiva modelagem matemática, tem colocado o que se entende como matemática aplicada, numa posição de destaque no meio científico. Todas as áreas do conhecimento tem cada vez mais usado ferramentas matemáticas na compreensão de suas áreas de estudo. Todas essas mudanças juntamente com a explosão de recursos computacionais, têm contribuído para a composição do universo das simulações computacionais. Nessa perspectiva é que apresentamos este trabalho, fruto dessa evolução (de modelagem e de recursos computacionais para simulação) para contribuir com a área, hoje chamada de Ecologia Matemática.

Neste trabalho faremos a modelagem de um problema de dispersão de poluentes, na intenção de produzir material e método para avaliação qualitativa, com atenção não somente nesse problema, mas sim na categoria em que ele está. Durante a confecção, sempre levamos em consideração o caráter de generalização do estudo, mesmo estando num problema específico. O problema teste que abordamos foi a dispersão de poluentes no Lago de Manso em Cuiabá MT, lago esse, fruto do represamento do rio Manso para a construção de uma usina hidrelétrica. Na modelagem desse problema, enfatizamos as contribuições como sendo a combinação do tratamento tridimensional de uma geometria totalmente irregular com a aproximação do campo de velocidades (responsável pelo termo de transporte) através da equação de Navier-Stokes. Todo esse método foi submetido a simulações numéricas em computadores de alto desempenho que permitiram resultados muito valiosos para a análise de possíveis situações reais (cenários). Uma tese de doutorado deve se caracterizar por um aspecto relevante de originalidade. Neste caso, a maior ênfase está no tratamento algorítmico dos sistemas de Equações a Derivadas Parciais, bem como no uso integrado do software GMSH com o ambiente MATLAB no tratamento do sistema discreto com que foram aproximadas

as equações não lineares desses sistemas.

O trabalho é composto de 5 capítulos e 3 anexos. No primeiro capítulo relataremos a experiência do grupo de Ecologia Matemática da UNICAMP, bem como uma sequência de trabalhos que ao longo do tempo, foram contribuindo de forma a possibilitar a execução desta tese. Descreveremos também o nosso problema e algumas características da região que compreende o Lago de Manso. No segundo capítulo colocaremos a modelagem clássica do problema, que envolve a descrição da equação de Difusão-Advecção, da equação de Navier-Stokes bem como condições de fronteira específicas, que retratam a situação do Lago de Manso. No terceiro capítulo discutiremos as formulações variacionais introduzindo o método de Galerkin para elementos finitos e também o método da projeção de Chorin-Teman para Navier-Stokes. No quarto capítulo introduziremos as discretizações das duas equações espaciais e temporais e também do domínio. A discretização utilizada na equação de Difusão-Advecção será o esquema de diferenças finitas de Crank-Nicolson e para a equação de Navier-Stokes usaremos uma discretização de ordem 1. No quinto capítulo relataremos as simulações numéricas que foram realizadas a fim de validar o modelo e os algoritmos. Todos os algoritmos executados foram elaborados especificamente para este trabalho e estão em ambiente MatLab[®]. O anexo 1 relatará toda a complexidade de tratamento de dados sobre a geometria e construção da malha. O anexo 2 reunirá toda a manipulação algébrica proveniente do método de elementos finitos. Por fim, no anexo 3, teremos os códigos computacionais bem como a descrição dos recursos computacionais utilizados nas simulações.

Capítulo 1

O problema: histórico e pretensões

Neste primeiro capítulo descreveremos um breve histórico sobre os trabalhos que fundamentaram e possibilitaram a confecção deste e também sobre APM-Manso, que será nosso estudo de caso na construção da ferramenta para análise de problemas dessa categoria. Na sequência, descreveremos mais detalhadamente o problema da dispersão de poluentes no lago de Manso.

1.1 Histórico: APM-Manso

A usina hidrelétrica de Manso (UHE-Manso) foi idealizada a partir de uma enchente ocorrida no ano de 1975 que deixou desabrigadas as famílias ribeirinhas. Devido a grandes dúvidas sobre os reais impactos dessa obra no ecossistema local, a construção ficou paralisada e só no ano de 2000 é que foi concluída a construção. A geração de energia e também a necessidade de uma forma de controle do fluxo de água no período de cheia, é que caracterizaram a UHE-Manso, como uma Unidade de Aproveitamento Múltiplo (APM-Manso). O reservatório atinge uma área de $427km^2$ nos municípios de Cuiabá, Chapada dos Guimarães e Nova Brasilândia. Os dados técnicos da APM-Manso são ([9]):

- BARRAGEM:
 - Comprimento total: $3.680m$
 - Em concreto: $140m$
 - Em solo compactado: $3.120m$
 - Em enrocamento: $420m$

- RESERVATÓRIO:
 - Bacia hidrográfica: $9.365km^2$
 - Área inundada (NA Máx.Max): $427km^2$
 - Volume acumulado: $7,3 \times 10^9m^3$
 - Volume útil: $2,951 \times 10^6m^3$
 - NA máximo normal: $287m$
 - NA mínimo normal: $278m$
- TOMADA D'ÁGUA:
 - Tipo: gravidade
 - Comprimento: $45m$
 - Número de vãos (comportas): 04
 - Cota da soleira: $264,90m$
 - Número de condutos forçados: 04
 - Diâmetro interno; $5.200m$
- VERTEDOIRO:
 - Vazão: $2.990m^3/s$
 - Número de vãos: 03
 - Cota da soleira vertente: $276,25m$
 - Comprimento total: $50m$
 - Tipo comporta: segmento $(9,5 \times 13,5)$
- CASA DE FORÇA:
 - Tipo: abrigada
 - Dimensão: $116,80m \times 16,80m$
 - Número de unidades: 04
 - Potência instalada: $212MW$
 - Ponte rolante: $2 \times 650/150kN$
 - Gerador:
 - Potência nominal aparente: $55,5MVA$
 - Fator de potência: 0,95
 - Frequência: $60Hz$
 - Tensão nominal: $13,8kV + / - 5\%$
 - Potência máxima contínua: $62,5MVA$
 - Energia firme: $92MW/ano$
 - Turbinas:
 - Tipo: Francis de eixo vertical
 - Queda líquida nominal: $57,5m$
 - Potência nominal: $52,5MW$
 - Rotação: $180rpm$
 - Engolimento: $104,6m^3/s$
 - Transformadores: 5 (operação mais reserva)
 - Tipo - monofásico
 - Capacidade total em operação - $250MVA$
 - Relação de transformação - $13,8/230kV$
 - Fabricante - Toshiba

A Figura 1.1 ilustra a localização do lago do APM-Manso e também os principais rios que o alimentam. Esses rios são: Rio Manso, Rio Casca, Rio Palmeiras e Rio Quilombo. O lago possui uma intensa atividade agropecuária em parte de seu entorno. A Figura 1.4 nos mostra entre outras informações, essa região sob a forma de um traçado vermelho. Evidentemente a caracterização dessa região foi possível através de imagens de satélite, podendo assim ser apenas uma aproximação da realidade. Há também atividade agropecuária que está no entorno dos rios que abastecem o lago (a montante). Essas duas fontes de agentes poluentes serão fundamentais para a caracterização do modelo.

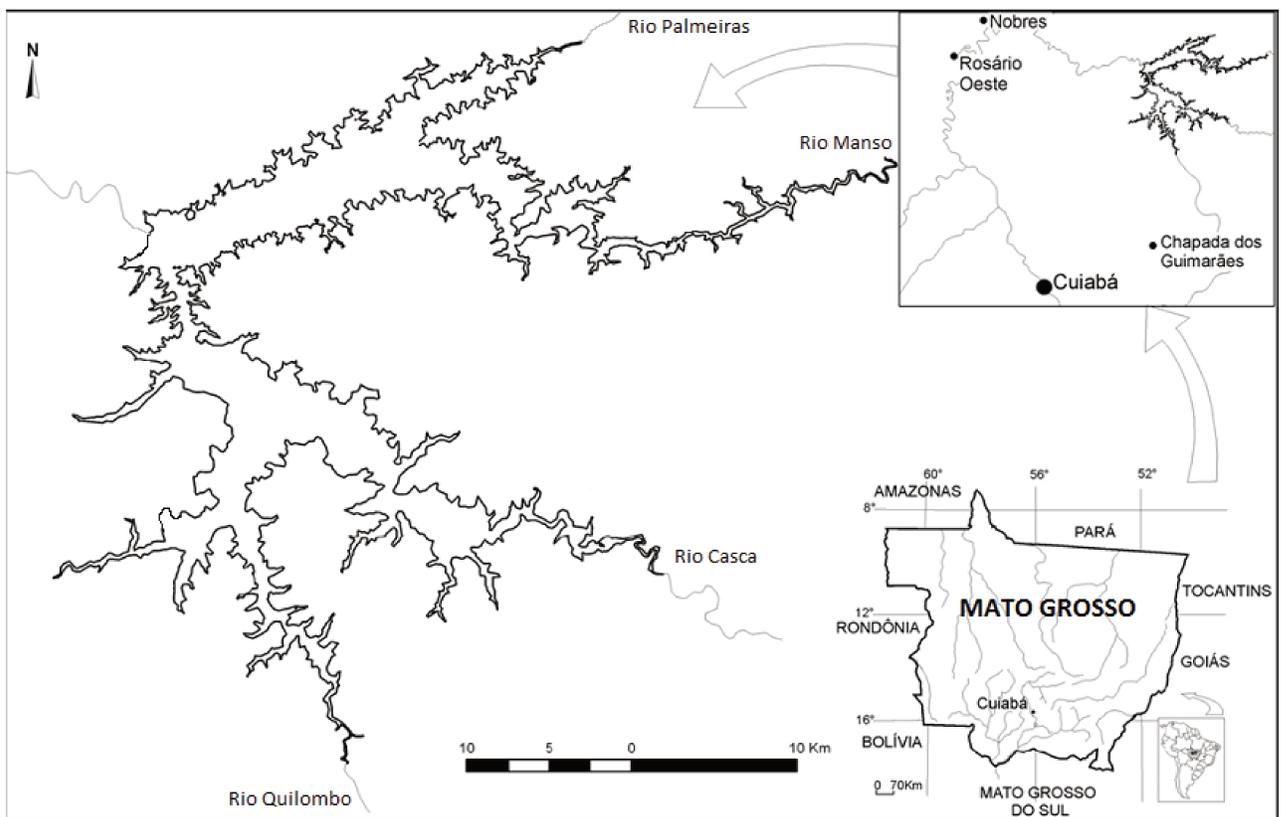


Figura 1.1: Localização da Represa da APM-Manso. Fonte:[13].

A Figura 1.2 mostra uma imagem do satélite Landsat5/TM (RGB 345). Nesta imagem podemos ver regiões onde há atividade agropecuária, fato que contribuirá na inserção de poluentes no lago sob a forma de *runoff*¹. Podemos ver a complexidade das curvas que

¹Escorrimento e infiltração de poluentes pela margem, deriva.

compõem o contorno do lago, o que contribuiu para a geração de grandes dificuldades na admissão do sólido que foi nosso domínio (região de estudo). A Figura 1.3 é uma fotografia aérea do lago de Manso que mostra uma região onde vemos a fronteira com mata.

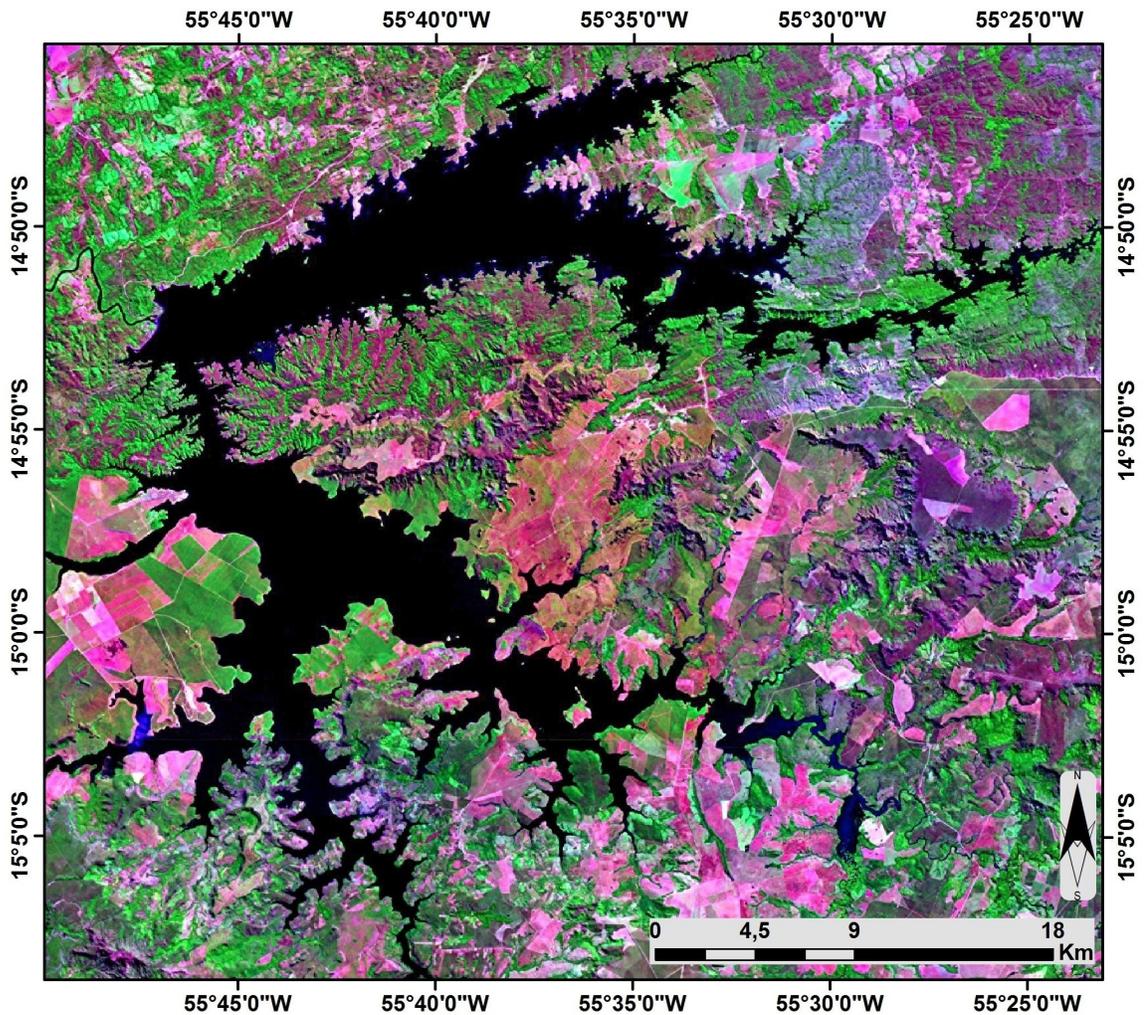


Figura 1.2: Imagem do satélite Landsat5/TM (RGB 345).

1.2 Histórico: trabalhos anteriores

O grupo de ecologia matemática da UNICAMP-Universidade Estadual de Campinas vem, desde a década de 70, produzindo diversos trabalhos (Teses, Dissertações, Iniciação científica e artigos) que contribuíram muito para o que temos hoje como modelagem matemática nessa



Figura 1.3: Lago do Manso. Foto: Marcos Vergueiro-Secom-MT 25-08-2010.

área. O trabalho que propomos é mais um passo na evolução da modelagem e simulação de fenômenos de dispersão de poluentes (uma das áreas de interesse do referido grupo) e está totalmente ancorado nas produções que antecederam e desenvolveram essa área que tem despertado a atenção de muitas outras áreas do conhecimento e de autoridades, no estudo de impactos ambientais.

A Tabela 1.1 traz um resumo dos trabalhos que contribuíram para a confecção deste. Esses trabalhos possuem algumas características em comum, por exemplo: problema alvo associado com poluição/contaminação, utilização do Método de Elementos Finitos na discretização espacial, água como meio em estudo, etc. A última linha dessa tabela contém nossa proposta de tese: dispersão de poluente na água, sendo o domínio um sólido irregular (lago de Manso, tridimensional) com modelagem do campo de velocidades com a equação de Navier-Stokes. Enfatizamos que o estudo do fenômeno da dispersão de poluentes usando um

domínio tridimensional e totalmente irregular, juntamente com a modelagem do campo de velocidades através da equação de Navier-Stokes, configuram a originalidade do trabalho.

Autor	Dimensão	Tipo do domínio	Meio	Campo de velocidades
Diomar Mistro, [17]	\mathbb{R}^2	retângulo	água	constante
Leidy Wolmuth, [11]	\mathbb{R}^2	mapa	água	constante
Mateus Bernardes, [14]	\mathbb{R}^2	mapa	água	constante por partes
Renato Cantão, [16]	\mathbb{R}^2	mapa	água	interpolação de dados
Geraldo Diniz, [1]	\mathbb{R}^2	regular	água e ar	perfil parabólico
Rosane Oliveira, [15]	\mathbb{R}^2	mapa	água	Stokes
Júlio Saavedra, [12]	\mathbb{R}^3	paralelepípedo	água	Stokes
Nelson Inforzato, [2]	\mathbb{R}^3	paralelepípedo	água e ar	Stokes
Andre Krindges	\mathbb{R}^3	mapa 3D	água	Navier-Stokes

Tabela 1.1: Histórico dos trabalhos realizados

1.3 Descrição da proposta

Baseado na experiência obtida dos trabalhos anteriores propomos, então, o estudo da dispersão de poluentes com as condições já descritas na seção anterior. Reservamos o Anexo 1 deste trabalho para a descrição dos detalhes da admissão do sólido geométrico que nos servirá de domínio. A fronteira desse domínio (conjunto de faces do sólido) foi dividida tendo em vista características físicas locais. Essas divisões são necessárias para que a modelagem contemple as particularidades da realidade e também para prever uma diversidade de situações possíveis. A Figura 1.6, ilustra a batimetria do lago inteiro, o que permite uma visualização de forma tridimensional. As Figuras 1.4 e 1.5 mostram essas divisões: a primeira retrata a superfície do lago e as diferentes situações que consideraremos no modelo e a segunda mostra uma visão tridimensional de um pedaço do lago, próxima à barragem, mostrando as fronteiras entendidas como superfície e fundo do lago. A notação usada nessas duas figuras tem o seguinte significado²:

²embora apareça na descrição, não colocamos nas figuras a fronteira Γ_4

- Γ_1 = parte da fronteira composta por mata
- Γ_2 = parte da fronteira que determina a superfície do lago
- Γ_3 = parte da fronteira que determina o fundo do lago
- Γ_4 = parte da fronteira onde há a saída de poluente com o vertedouro na barragem
- Γ_5 = parte da fronteira onde ocorre *runoff*
- Γ_6 = parte da fronteira onde há a entrada de poluente com o rio Palmeira
- Γ_7 = parte da fronteira onde há a entrada de poluente com o rio Manso
- Γ_8 = parte da fronteira onde há a entrada de poluente com o rio Casca
- Γ_9 = parte da fronteira onde há a entrada de poluente com o rio Quilombo
- Γ_{10} = parte da fronteira composta por rocha ou concreto

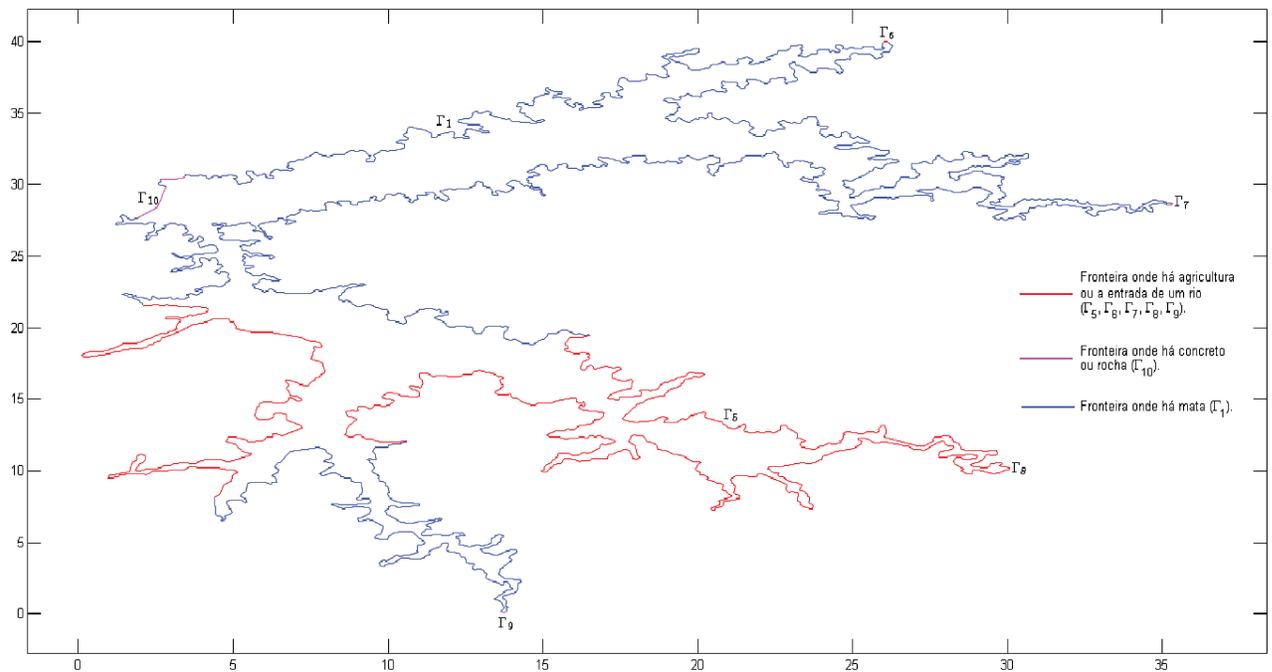


Figura 1.4: Partes da fronteira: bordas do lago.

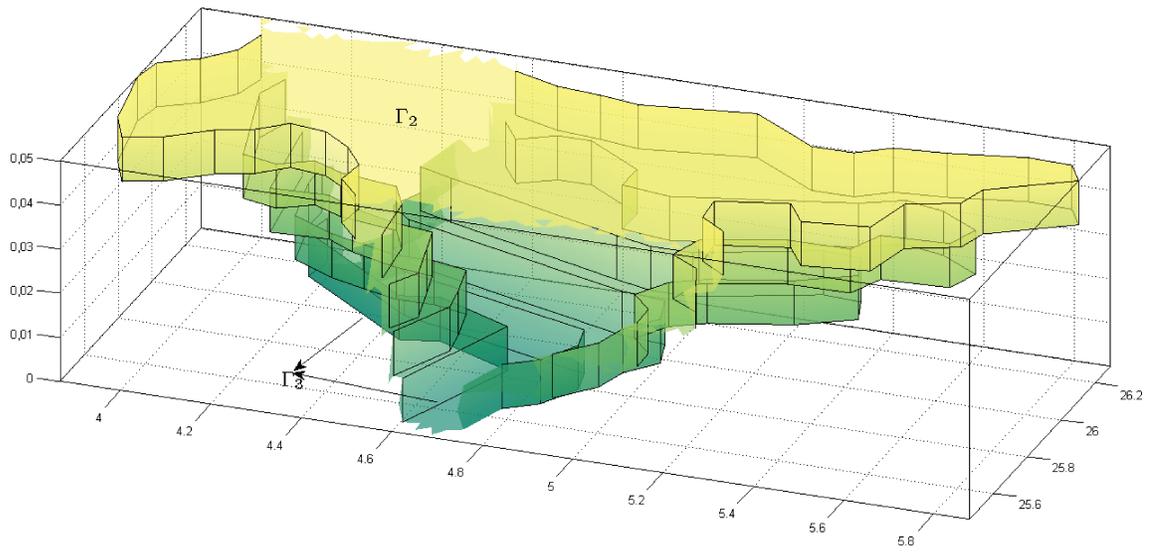


Figura 1.5: Exemplo da discretização espacial tridimensional - fundo e superfície de uma pequena parte do domínio do problema teste.

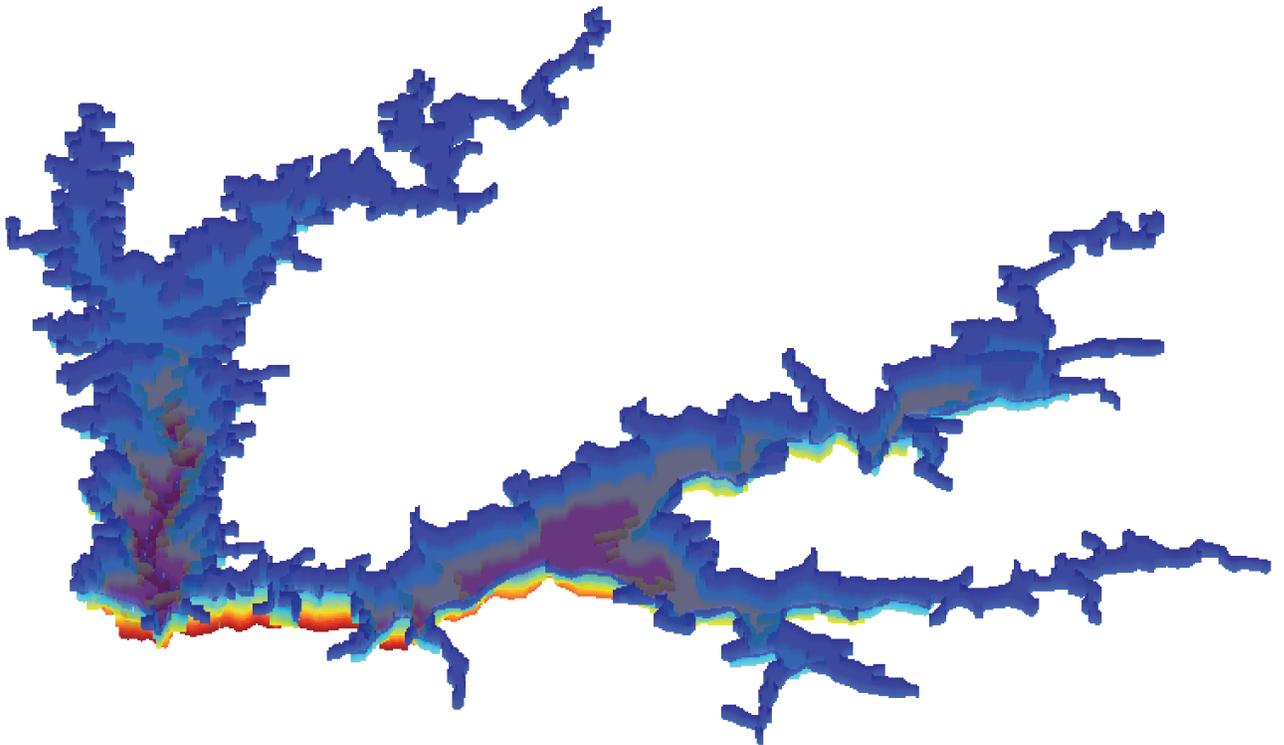


Figura 1.6: Imagem tridimensional do lago inteiro.

Capítulo 2

A modelagem clássica do problema

Neste capítulo descreveremos a modelagem do problema teste que é objeto de estudo deste trabalho. No final do capítulo anterior apresentamos considerações sobre a continuação dos trabalhos que tratam da dispersão de poluentes em meios aquáticos. Colocaremos agora as equações que regem a modelagem do problema na ótica desta continuação. Nesta modelagem consideraremos fenômenos de difusão efetiva (ou dispersão), de transporte advectivo, de decaimento e também a inserção de poluição na região estudada (fonte). O termo de transporte receberá atenção especial, devido ao fato de ser dependente de um campo de velocidades, que por sua vez, será modelado através da equação de Navier-Stokes.

2.1 Difusão-Advecção

Nesta seção abordaremos a equação de advecção-difusão, bem como suas particularidades na modelagem do nosso problema. Chamaremos de $u(t, x, y, z)$ a concentração de poluente no meio em estudo no ponto (x, y, z) e no instante t . Por fim, um modelo geral para o problema teste de nosso estudo, é:

$$\frac{\partial u}{\partial t} = \{\text{difusão}\} - \{\text{transporte}\} - \{\text{decaimento}\} + \{\text{fonte}\},$$

sendo,

$$\begin{aligned} \{\text{difusão}\} &= \nabla \cdot (\alpha_u \nabla u) \quad (\text{cf. Okubo})[3] \\ \{\text{transporte}\} &= \nabla \cdot (\vec{v}u) \quad (\text{cf. Edelstein-Keshet})[4] \\ \{\text{decaimento}\} &= \sigma_u u \quad (\text{cf. Marchuk})[5] \end{aligned}$$

Assim, a equação que modela o fenômeno da dispersão de um determinado poluente, num domínio tridimensional, é:

$$\frac{\partial u}{\partial t} = \nabla \cdot (\alpha_u \nabla u) - \nabla \cdot (\vec{v}u) - \sigma_u u + f, \quad (x, y, z) \in \Omega \subset \mathbb{R}^3, \quad t \in (0, T] \subset \mathbb{R}, \quad (2.1)$$

sendo:

- $\alpha_u = \alpha_u(t, x, y, z)$ o coeficiente de difusibilidade efetiva no meio;
- $\vec{v} = \vec{v}(t, x, y, z) = (v_1(t, x, y, z), v_2(t, x, y, z), v_3(t, x, y, z))$ o campo de velocidades no meio;
- $\sigma_u = \sigma_u(t, x, y, z)$ o coeficiente de decaimento total no meio;
- $f = f(t, x, y, z)$ é o termo fonte de poluição.

Consideramos a condição inicial: $u(0, x, y, z) = u_0(x, y, z)$, $(x, y, z) \in \Omega$.

A Figura 1.4 mostra a linha que demarca a superfície do lago. Como a legenda dessa figura mostra, temos parte vermelha, parte azul e parte magenta, representando respectivamente, regiões onde há agricultura, mata e rocha ou concreto. Considerando essas fronteiras, adotaremos as seguintes condições de contorno:

- **Ingresso de poluente:** Admitimos que a entrada de poluentes se dá através dos rios e também pela fronteira onde há agricultura (*runoff*). Utilizaremos uma condição do tipo Von Neumann não homogênea:

$$-\alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_i} = \theta_i(t, x, y, z) \quad \text{sendo } \theta_i \text{ uma função dada, } i = 5, 6, 7, 8, 9.$$

- **Perda de poluente:** Admitimos que existe uma perda de poluentes para o solo (fundo do lago), para a superfície, na margem onde há mata e no vertedouro da represa. Consideramos que essas perdas são proporcionais à quantidade presente na respectiva fronteira. Essa situação é modelada usando uma condição do tipo Robin:

$$-\alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_i} = k_i u \quad \text{sendo } k_i \text{ uma constante dada e que gradua essa perda, } i = 1, 2, 3, 4.$$

- **Fronteira sem perda:** Na região da fronteira onde há concreto ou formação rochosa, admitimos que não há perda nem ingresso de poluente. Para modelar essa situação,

consideramos uma condição de fronteira do tipo Von Neumann homogênea:

$$-\alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_{10}} = 0.$$

Podemos por fim, colocar:

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} = \nabla \cdot (\alpha_u \nabla u) - \nabla \cdot (\vec{v}u) - \sigma_u u + f, \quad (x, y, z) \in \Omega \subset \mathbb{R}^3, \quad t \in (0, T] \subset \mathbb{R}, \\ \nabla \cdot \vec{v} = 0 \\ -\alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_1} = k_1 u, \quad t \in (0, T]; \\ -\alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_2} = k_2 u, \quad t \in (0, T]; \\ -\alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_3} = k_3 u, \quad t \in (0, T]; \\ -\alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_4} = k_4 u, \quad t \in (0, T]; \\ \alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_5} = \theta_5(t, x, y, z), \quad t \in (0, T]; \\ \alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_6} = \theta_6(t, x, y, z), \quad t \in (0, T]; \\ \alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_7} = \theta_7(t, x, y, z), \quad t \in (0, T]; \\ \alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_8} = \theta_8(t, x, y, z), \quad t \in (0, T]; \\ \alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_9} = \theta_9(t, x, y, z), \quad t \in (0, T]; \\ \alpha \frac{\partial u}{\partial \eta} \Big|_{\Gamma_{10}} = 0, \quad t \in (0, T]; \\ u(x, 0) = u^0(x), \quad \forall x \in \Omega, \end{array} \right. \quad (2.2)$$

2.2 Navier-Stokes

O termo advectivo da equação (2.1) ($\nabla \cdot (\vec{v}u)$) contém um campo vetorial \vec{v} que descreve o campo de velocidades do meio em questão, no nosso caso a água. Esse campo contribui no transporte das partículas de poluição ao longo do domínio estudado. Como discutido no capítulo 1, colocamos como umas das contribuições inéditas deste trabalho, a modelagem desse campo de velocidades com a equação de Navier-Stokes, cuja versão para um fluido newtoniano e incompressível, como o caso da água, é:

$$\left\{ \begin{array}{l} \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} + \nabla p - \nu \Delta \vec{v} = \vec{f}, \quad (x, y, z) \in \Omega \subset \mathbb{R}^3, \quad t \in (0, T] \subset \mathbb{R}, \\ \nabla \cdot \vec{v} = 0, \end{array} \right. \quad (2.3)$$

juntamente com condições de contorno e inicial compatíveis. Sendo ν a viscosidade cinemática do fluido, p a pressão e $\vec{\mathbf{f}}$ um campo de forças externo (por exemplo gravidade).

Consideraremos condições de contorno específicas, tendo em vista as particularidades do nosso problema. São elas:

- **Entrada d'água:** A partir do conhecimento da velocidade da água dos rios na entrada do lago, prescreveremos uma condição de Dirichlet não homogênea:

$$\vec{v}|_{\Gamma_i} = \vec{g}_i(t, x, y, z), \quad \text{sendo } \vec{g}_i \text{ uma função dada, } (x, y, z) \in \Omega, \quad t \in (0, T], \quad i = 6, 7, 8, 9.$$

- **Saída d'água:** Para deixar livre a velocidade da água no vertedouro, usaremos uma condição de Von Neumann homogênea :

$$\frac{\partial v_1}{\partial \eta} \Big|_{\Gamma_4} = \frac{\partial v_2}{\partial \eta} \Big|_{\Gamma_4} = \frac{\partial v_3}{\partial \eta} \Big|_{\Gamma_4} = 0, \quad \text{sendo } \vec{v} = (v_1, v_2, v_3).$$

- **Fundo do lago:** Devido às dimensões do lago, consideraremos que a velocidade da água na margem e fundo são nulas. Para modelar esse fato, usaremos uma condição de Dirichlet homogênea:

$$\vec{v}|_{\Gamma_i} = (0, 0, 0), \quad i = 1, 3, 5, 10.$$

- **Superfície do lago:** Tendo em vista que o vetor velocidade num ponto (x, y, z) na superfície do lago é perpendicular a esta superfície, consideraremos a seguinte condição

de contorno:

$$\frac{\partial v_1}{\partial \eta} \Big|_{\Gamma_2} = \frac{\partial v_2}{\partial \eta} \Big|_{\Gamma_2} = 0 \quad \text{e} \quad v_3|_{\Gamma_2} = 0.$$

Assim como foi feito para a equação de advecção-difusão, podemos resumir:

$$\left\{ \begin{array}{l} \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} + \nabla p - \nu \Delta \vec{v} = \vec{\mathbf{f}}, \quad (x, y, z) \in \Omega \subset \mathbb{R}^3, \quad t \in (0, T] \subset \mathbb{R}, \\ \nabla \cdot \vec{v} = 0, \\ \vec{v}|_{\Gamma_i} = \vec{g}_i(t, x, y, z), \quad i = 6, 7, 8, 9. \\ \frac{\partial v_1}{\partial \eta} \Big|_{\Gamma_4} = \frac{\partial v_2}{\partial \eta} \Big|_{\Gamma_4} = \frac{\partial v_3}{\partial \eta} \Big|_{\Gamma_4} = 0 \\ \vec{v}|_{\Gamma_i} = (0, 0, 0), \quad i = 1, 3, 5, 10. \\ \frac{\partial v_1}{\partial \eta} \Big|_{\Gamma_2} = \frac{\partial v_2}{\partial \eta} \Big|_{\Gamma_2} = 0 \quad \text{e} \quad v_3|_{\Gamma_2} = 0. \\ v(x, 0) = v^0(x), \quad \forall x \in \Omega, \end{array} \right. \quad (2.4)$$

2.3 Considerações e adaptações do modelo

Nesta seção descreveremos algumas considerações e simplificações que precisaram ser feitas para se construir este primeiro modelo. Consideraremos as condições de fronteira independentes do tempo, mesmo tendo conhecimento de que a sazonalidade tem considerável influência. Os parâmetros α e σ foram considerados independentes de espaço e tempo: aqui poderíamos considerar uma variação temporal em σ , pois podemos ter agentes poluidores cujo decaimento pode também ter características sazonais. A inserção de poluentes no domínio (lago) foi modelada através de condições de fronteira. Desta forma temos o termo fonte da equação (2.2), considerado nulo. As condições iniciais para a dispersão de poluente e campo de velocidades foram consideradas iguais a zero.

Capítulo 3

Formulações variacionais

O método que usaremos para resolver numericamente as equações de nosso modelo é o *Método de Galerkin via Elementos Finitos*. Este método utiliza a formulação variacional das equações o que permite posteriormente a discretização e assim chegarmos numa aproximação numérica da solução. Neste capítulo construiremos a formulação variacional da equação de Advecção-difusão e também da equação de Navier-Stokes.

3.1 Formulação variacional: Difusão-Advecção

Vamos de início estabelecer a notação que servirá para a formulação variacional no sentido de organizar e principalmente simplificar a equação resultante.

Considerando nosso domínio Ω , temos:

1. Em $L^2(\Omega)$, denotamos o produto interno e a respectiva norma por:

$$(u, w)_{L^2(\Omega)} = \int_{\Omega} u w d\mu \quad \text{e} \quad \|u\|_{L^2(\Omega)}^2 = (u, u)_{L^2(\Omega)}.$$

Consideramos também o produto interno sobre parte da fronteira de Ω :

$$\langle u, w \rangle_{L^2(\Gamma)} = \int_{\Gamma} u w d\partial\Omega \quad \text{e} \quad \|u\|_{L^2(\Gamma)}^2 = (u, u)_{L^2(\Gamma)},$$

sendo $\Gamma \subset \partial\Omega$.

2. Em $H^1(\Omega) \subset L^2(\Omega)$, espaço das funções em $L^2(\Omega)$ cujas derivadas de primeira ordem, no sentido fraco, também pertençam a $L^2(\Omega)$:

$$H^1(\Omega) = \left\{ u \in L^2(\Omega) \left/ \frac{\partial u}{\partial x_k} \in L^2(\Omega), \quad k = 1, 2, 3 \right. \right\},$$

com produto interno e norma correspondente:

$$(u, w)_{H^1(\Omega)} = (u, w)_{L^2(\Omega)} + (\nabla u \parallel \nabla w)_{L^2(\Omega)} \text{ e } \|u\|_{H^1(\Omega)}^2 = \|u\|_{L^2(\Omega)}^2 + \|\nabla u\|_{L^2(\Omega)}^2,$$

$$\text{sendo: } (\nabla u \parallel \nabla w)_{L^2(\Omega)} = \sum_{k=1}^3 \int_{\Omega} \frac{\partial u}{\partial x_k} \frac{\partial w}{\partial x_k} d\mu.$$

Dando sequência à construção da formulação variacional, multiplicamos a equação em (2.2) por $w \in H^1(\Omega)$ e integramos em relação às variáveis espaciais:

$$\int_{\Omega} \frac{\partial u}{\partial t} w d\mu - \int_{\Omega} \nabla \cdot (\alpha \nabla u) w d\mu + \int_{\Omega} \nabla \cdot (\vec{v} u) w d\mu + \int_{\Omega} \sigma u w d\mu = \int_{\Omega} f w d\mu, \quad \forall w \in H^1(\Omega).$$

Usando que $\nabla \cdot (\vec{v} u) = (\nabla \cdot \vec{v}) u + \vec{v} \cdot \nabla u = \vec{v} \cdot \nabla u$, temos:

$$\int_{\Omega} \frac{\partial u}{\partial t} w d\mu - \alpha \int_{\Omega} \nabla \cdot (\nabla u) w d\mu + \int_{\Omega} (\vec{v} \cdot \nabla u) w d\mu + \sigma \int_{\Omega} u w d\mu = \int_{\Omega} f w d\mu, \quad \forall w \in H^1(\Omega). \quad (3.1)$$

O teorema de Green nos permite escrever:

$$-\alpha \int_{\Omega} \nabla \cdot (\nabla u) w d\mu = \alpha \int_{\Omega} (\nabla u \cdot \nabla w) d\mu - \alpha \int_{\partial\Omega} \frac{\partial u}{\partial \eta} w d\partial\Omega,$$

sendo η o vetor normal unitário exterior à Ω .

Tendo em vista que $\partial\Omega = \bigcup_{j=1}^{10} \Gamma_j$ e as condições de fronteira em (2.2), chegamos na expressão:

$$-\alpha \int_{\partial\Omega} \frac{\partial u}{\partial \eta} w d\partial\Omega = \sum_{m=1}^4 k_m \int_{\Gamma_m} u w d\partial\Omega - \sum_{n=5}^9 \int_{\Gamma_n} \theta_n w d\partial\Omega$$

Usando essas expressões em (3.1), chegamos a:

$$\begin{aligned} \int_{\Omega} \frac{\partial u}{\partial t} w d\mu + \alpha \int_{\Omega} (\nabla u \cdot \nabla w) d\mu + \sum_{m=1}^4 k_m \int_{\Gamma_m} u w d\partial\Omega + \int_{\Omega} (\vec{v} \cdot \nabla u) w d\mu + \sigma \int_{\Omega} u w d\mu = \int_{\Omega} f w d\mu + \\ + \sum_{n=5}^9 \int_{\Gamma_n} \theta_n w d\partial\Omega, \quad \forall w \in H^1(\Omega). \end{aligned}$$

Usando a notação de produto interno introduzida no início desta seção, temos:

$$\begin{aligned} \left(\frac{\partial u}{\partial t}, w \right)_{L^2(\Omega)} + \alpha (\nabla u \parallel \nabla w)_{L^2(\Omega)} + \sum_{m=1}^4 k_m \langle u, w \rangle_{L^2(\Gamma_m)} + ((\vec{v} \cdot \nabla u), w)_{L^2(\Omega)} + \sigma(u, w)_{L^2(\Omega)} = \\ = (f, w)_{L^2(\Omega)} + \sum_{n=5}^9 \langle \theta_n, w \rangle_{L^2(\Gamma_n)}, \quad \forall w \in H^1(\Omega). \end{aligned} \quad (3.2)$$

Esta é então, a formulação variacional da equação (2.2). Assim, procuramos uma solução $u \in \mathcal{V} = \left\{ u \in L^2[(0, T), H^1(\Omega)] \middle/ \frac{\partial u}{\partial t} \in L^2(\Omega), \quad \forall t \in [0, T] \right\}$ que satisfaça (3.2).

Em sua tese de doutorado, Inforzato ([2]) garante existência e unicidade de solução da equação (3.2), usando resultados teóricos de [10].

3.2 Formulação variacional: Navier-Stokes

Esta seção é dedicada à construção da formulação variacional para a equação de Navier-Stokes. Usaremos em nosso trabalho o método das projeções cujo esquema original é creditado a Chorin[6] e a Temam[7]. Este método tem sido muito utilizado na aproximação numérica de dinâmica de fluidos, principalmente pelo fato de permitir o desacoplamento das incógnitas velocidade e pressão da equação.

Embora dediquemos o capítulo 4 para as discussões sobre discretização, faremos aqui uma discretização temporal de (2.3), devido à construção do método de Chorin-Temam. Para um dado passo de tempo Δt e sendo v^0 uma condição inicial, uma discretização semi-implícita de (2.3) consiste em aproximar $v(\cdot, n\Delta t)$ e $p(\cdot, n\Delta t)$ para $n = 1, 2, \dots$ pela solução (v^n, p^n) de:

$$\begin{cases} \frac{v^n - v^{n-1}}{\Delta t} - \nu \Delta v^n + (v^{n-1} \cdot \nabla) v^n + \nabla p^n = \mathbf{f} \\ \nabla \cdot v^n = 0 \end{cases} \quad (3.3)$$

O algoritmo em sua formulação original, é apresentado da seguinte forma: Seja v^0 uma condição inicial. Para $n = 1, 3, \dots$, calcular v^n e p_*^n por:

$$\begin{cases} \frac{v^n - v_*^n}{\Delta t} + \nabla p_*^n = 0 \\ \nabla \cdot v^n = 0 \end{cases}, \quad (3.4)$$

sendo v_*^n solução de:

$$\frac{v_*^n - v_*^{n-1}}{\Delta t} - \nu \Delta v_*^n + (v_*^{n-1} \cdot \nabla) v_*^n = \mathbf{f}. \quad (3.5)$$

Aplicando o operador divergência nos dois membros da primeira equação de (3.4) e usando a segunda, temos:

$$\Delta p_*^n = \frac{\nabla \cdot v_*^n}{\Delta t}. \quad (3.6)$$

Assumindo que $(v^n - v_*^n) \cdot \eta = 0$ em $\partial\Omega$, aplica-se uma condição de Von Neumann $\partial p_*^n / \partial \eta = 0$.

Ficamos por fim, com o algoritmo:

$$\begin{cases} \frac{v_*^n - v_*^{n-1}}{\Delta t} - \nu \Delta v_*^n + (v_*^{n-1} \cdot \nabla) v_*^n = \mathbf{f}. \\ \Delta p_*^n = \frac{\nabla \cdot v_*^n}{\Delta t} \\ v^n = v_*^n - \Delta t \nabla p_*^n. \end{cases} \quad (3.7)$$

Tendo em vista as características e condições de fronteira do nosso problema, chegamos na formulação variacional para a equação (3.7):

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{v_*^n - v_*^{n-1}}{\Delta t} \mathbf{w} d\mu + \nu \int_{\Omega} (\nabla v_*^n : \nabla \mathbf{w}) d\mu + \int_{\Omega} (v_*^{n-1} \cdot \nabla) v_*^n \cdot \mathbf{w} d\mu = \int_{\Omega} \mathbf{f} \cdot \mathbf{w} d\mu \\ \forall \mathbf{w} \in (H_{\Gamma_{xy}}^1(\Omega))^2 \times H_{\Gamma_z}^1(\Omega), \\ \int_{\Omega} \nabla p_*^n \cdot \nabla q d\mu = \int_{\Omega} \frac{\nabla \cdot v_*^n}{\Delta t} q d\mu, \quad \forall q \in H^1(\Omega) \cap L_0^2(\Omega), \\ \int_{\Omega} v^n \cdot \mathbf{w} d\mu = \int_{\Omega} v_*^n \cdot \mathbf{w} d\mu - \Delta t \int_{\Omega} \nabla p_*^n \cdot \mathbf{w} d\mu, \quad \forall \mathbf{w} \in (H_{\Gamma_{xy}}^1(\Omega))^2 \times H_{\Gamma_z}^1(\Omega), \end{array} \right. \quad (3.8)$$

sendo $H_{\Gamma_{xy}}^1(\Omega) = \left\{ \mathbf{w} \in H^1(\Omega) \mid \mathbf{w}|_{\Gamma_{xy}} = 0 \right\}$, $H_{\Gamma_z}^1(\Omega) = \left\{ \mathbf{w} \in H^1(\Omega) \mid \mathbf{w}|_{\Gamma_z} = 0 \right\}$,

com $\Gamma_{xy} = \bigcup_i \Gamma_i$, $i = 1, 3, 5, 6, 7, 8, 9, 10$ e $\Gamma_z = \bigcup_j \Gamma_j$, $j = 1, 2, 3, 5, 6, 7, 8, 9, 10$.

Para completar essa formulação variacional, precisamos estabelecer condições de fronteira para v_*^n . Usamos as mesmas condições que v^n seguindo sugestão dada por Goldberg-Ruas [8]. Nesse artigo encontramos outras sugestões para condições em v_*^n , bem como um pós tratamento da pressão em cada passo do tempo, mas não consideramos essa variação, em virtude de não utilizarmos a pressão real p^n e sim somente p_*^n , que é uma pressão artificial não tendo significado físico. A existência e unicidade de solução em (3.8) são estabelecidas no mesmo artigo. Uma semelhante explanação e utilização do método de Chorin-Temam é feita em [19], porém bidimensional.

Capítulo 4

Discretizações

Depois de construídas as formulações variacionais ((3.2) e (3.8)), devemos utilizar algum método numérico adequado para aproximar a solução. O método que utilizaremos é o *Método de Galerkin* o qual necessita que se introduza uma discretização espacial. Para a discretização temporal usaremos o esquema de Crank-Nicolson¹ na equação de advecção-difusão e uma discretização semi implícita² para Navier-Stokes. Neste capítulo construiremos essas discretizações e discutiremos detalhes específicos do problema teste usado neste trabalho.

4.1 Discretizações espaciais

Nesta seção introduziremos as discretizações espaciais das formulações variacionais (3.2) e (3.8), iniciando pela primeira.

Seja $U^{(h)}$ um subespaço de $H^1(\Omega)$ de dimensão finita gerado pela base $\mathcal{B} = \{\varphi_1, \varphi_2, \dots, \varphi_{n_h}\}$. Assim, a solução de (3.2) no subespaço $U^{(h)}$ pode ser escrita como combinação linear dos elementos da base \mathcal{B} :

$$u^{(h)} = \sum_{j=1}^{n_h} u_j(t) \varphi_j(x, y, z). \quad (4.1)$$

¹Esquema de discretização por diferenças finitas muito usado por ser incondicionalmente estável e de ordem 2

²Essa discretização já foi introduzida na construção do método das projeções no capítulo anterior

Substituindo $u^{(h)}$ na formulação variacional (3.2), e tomando $w \in U^{(h)}$, temos³:

$$\begin{aligned} \left(\frac{\partial u^{(h)}}{\partial t}, w \right)_{\Omega} + \alpha (\nabla u^{(h)} \parallel \nabla w)_{\Omega} + \sum_{m=1}^4 k_m \langle u^{(h)}, w \rangle_{\Gamma_m} + ((\vec{v} \cdot \nabla u^{(h)}), w)_{\Omega} + \\ + \sigma (u^{(h)}, w)_{\Omega} = (f, w)_{\Omega} + \sum_{n=5}^9 \langle \theta_n, w \rangle_{\Gamma_n}, \quad \forall w \in U^{(h)}, \end{aligned} \quad (4.2)$$

ou seja:

$$\begin{aligned} \left(\sum_{j=1}^{n_h} \frac{du_j}{dt} \varphi_j, w \right)_{\Omega} + \alpha \left(\sum_{j=1}^{n_h} u_j \nabla \varphi_j \parallel \nabla w \right)_{\Omega} + \sum_{m=1}^4 k_m \left\langle \sum_{j=1}^{n_h} u_j \varphi_j, w \right\rangle_{\Gamma_m} + \\ + \left(\sum_{j=1}^{n_h} u_j (\vec{v} \cdot \nabla \varphi_j), w \right)_{\Omega} + \sigma \left(\sum_{j=1}^{n_h} u_j \varphi_j, w \right)_{\Omega} = (f, w)_{\Omega} + \sum_{n=5}^9 \langle \theta_n, w \rangle_{\Gamma_n}, \quad \forall w \in U^{(h)}, \end{aligned}$$

As funções $u_j(t)$ não dependem de (x, y, z) , por isso podemos tirá-las do produto interno.

Com isso temos:

$$\begin{aligned} \sum_{j=1}^{n_h} \frac{du_j}{dt} (\varphi_j, w)_{\Omega} + \alpha \sum_{j=1}^{n_h} u_j (\nabla \varphi_j \parallel \nabla w)_{\Omega} + \sum_{m=1}^4 k_m \sum_{j=1}^{n_h} u_j \langle \varphi_j, w \rangle_{\Gamma_m} + \sum_{j=1}^{n_h} u_j (\vec{v} \cdot \nabla \varphi_j, w)_{\Omega} + \\ + \sigma \sum_{j=1}^{n_h} u_j (\varphi_j, w)_{\Omega} = (f, w)_{\Omega} + \sum_{n=5}^9 \langle \theta_n, w \rangle_{\Gamma_n}, \quad \forall w \in U^{(h)}. \end{aligned}$$

Sendo essa equação válida $\forall w \in U^{(h)}$, temos que é suficiente avaliá-la para os elementos da base de $U^{(h)}$. Portanto:

$$\begin{aligned} \sum_{j=1}^{n_h} \frac{du_j}{dt} (\varphi_j, \varphi_i)_{\Omega} + \alpha \sum_{j=1}^{n_h} u_j (\nabla \varphi_j \parallel \nabla \varphi_i)_{\Omega} + \sum_{m=1}^4 k_m \sum_{j=1}^{n_h} u_j \langle \varphi_j, \varphi_i \rangle_{\Gamma_m} + \sum_{j=1}^{n_h} u_j (\vec{v} \cdot \nabla \varphi_j, \varphi_i)_{\Omega} + \\ + \sigma \sum_{j=1}^{n_h} u_j (\varphi_j, \varphi_i)_{\Omega} = (f, \varphi_i)_{\Omega} + \sum_{n=5}^9 \langle \theta_n, \varphi_i \rangle_{\Gamma_n}, \quad \forall \varphi_i \in \mathcal{B}. \end{aligned}$$

Fazendo $u_h = (u_1(t), u_2(t), \dots, u_{n_h}(t))^T$, podemos escrever essa sentença sob o forma de um sistema de equações diferenciais ordinárias de primeira ordem:

$$M \dot{u}_h(t) + N u_h(t) + W u_h(t) = F, \quad (4.3)$$

³Para simplificar a notação dos produtos internos, usaremos Ω e Γ no lugar de $L^2(\Omega)$ e $L^2(\Gamma)$

sendo as matrizes $M = (m_{ij})_{n_h \times n_h}$, $N = (n_{ij})_{n_h \times n_h}$, $W = (w_{ij})_{n_h \times n_h}$ e o vetor $F = (f_i)_{n_h \times 1}$ dados por:

$$\begin{aligned}
m_{ij} &= (\varphi_j, \varphi_i)_\Omega \\
n_{ij} &= \alpha (\nabla \varphi_j \parallel \nabla \varphi_i)_\Omega + \sigma (\varphi_j, \varphi_i)_\Omega + \sum_{m=1}^4 k_m \langle \varphi_j, \varphi_i \rangle_{\Gamma_m} \\
w_{ij} &= (\vec{v} \cdot \nabla \varphi_j, \varphi_i)_\Omega \\
f_i &= (f, \varphi_i)_\Omega + \sum_{n=5}^9 \langle \theta_n, \varphi_i \rangle_{\Gamma_n}
\end{aligned} \tag{4.4}$$

A matriz W incorpora o termo de transporte que por sua vez é dependente de um campo de velocidades e será calculado usando a equação de Navier-Stokes e por essa razão ela é separada, tendo em vista que precisará, depois da discretização, ser calculada a cada passo no tempo.

Passamos agora aos detalhes relativos à discretização de (3.8). Sejam $V_{xy}^{(h)}$, $V_z^{(h)}$ e $V_p^{(h)}$ respectivamente, subespaços de dimensão finita de $H_{\Gamma_{xy}}^1(\Omega)$, $H_{\Gamma_z}^1(\Omega)$ e $H_{\Gamma_4}^1(\Omega)$ gerados pelas bases: $\mathcal{B}_{xy} = \{\varphi_1^{xy}, \varphi_2^{xy}, \dots, \varphi_{n_h}^{xy}\}$, $\mathcal{B}_z = \{\varphi_1^z, \varphi_2^z, \dots, \varphi_{n_h}^z\}$ e $\mathcal{B}_p = \{\varphi_1^p, \varphi_2^p, \dots, \varphi_{n_h}^p\}$ (sendo $H_{\Gamma_4}^1(\Omega) = \{q \in H^1(\Omega) / q|_{\Gamma_4} = 0\}$). Seja também $V_{\widehat{xy}}^{(h)}$ um subespaço de dimensão finita de $H^1(\Omega)$, gerado pela base $\mathcal{B}_{\widehat{xy}} = \{\widehat{\varphi}_1^{\widehat{xy}}, \widehat{\varphi}_2^{\widehat{xy}}, \dots, \widehat{\varphi}_{n_h}^{\widehat{xy}}\}$. Podemos escrever v^n e v_*^n no espaço $V_{\widehat{xy}}^{(h)} \times V_{\widehat{xy}}^{(h)} \times V_z^{(h)}$ e também p_*^n no espaço $V_p^{(h)}$ da seguinte forma:

$$\begin{aligned}
v^n &= (v_1^n, v_2^n, v_3^n) = \left(\sum_{i=1}^{n_h} v_{1i}^n \widehat{\varphi}_i^{\widehat{xy}}(x, y, z), \sum_{j=1}^{n_h} v_{2j}^n \widehat{\varphi}_j^{\widehat{xy}}(x, y, z), \sum_{l=1}^{n_h} v_{3l}^n \varphi_l^z(x, y, z) \right) \\
v_*^n &= (v_{1*}^n, v_{2*}^n, v_{3*}^n) = \left(\sum_{i=1}^{n_h} v_{1*i}^n \widehat{\varphi}_i^{\widehat{xy}}(x, y, z), \sum_{j=1}^{n_h} v_{2*j}^n \widehat{\varphi}_j^{\widehat{xy}}(x, y, z), \sum_{l=1}^{n_h} v_{3*l}^n \varphi_l^z(x, y, z) \right) \\
p_*^n &= \left(\sum_{i=1}^{n_h} p_{*i}^n \varphi_i^z(x, y, z) \right)
\end{aligned} \tag{4.5}$$

Vamos colocar (3.8) na forma escalar para facilitar as substituições que irão se seguir:

$$\left\{ \begin{array}{l}
 \int_{\Omega} \frac{v_{1*}^n - v_1^{n-1}}{\Delta t} w_1 d\mu + \int_{\Omega} (v^{n-1} \cdot \nabla v_{1*}^n) w_1 d\mu + \nu \int_{\Omega} (\nabla v_{1*}^n \cdot \nabla w_1) d\mu = 0 \quad \forall w_1 \in V_{xy}^{(h)}, \\
 \int_{\Omega} \frac{v_{2*}^n - v_2^{n-1}}{\Delta t} w_2 d\mu + \int_{\Omega} (v^{n-1} \cdot \nabla v_{2*}^n) w_2 d\mu + \nu \int_{\Omega} (\nabla v_{2*}^n \cdot \nabla w_2) d\mu = 0 \quad \forall w_2 \in V_{xy}^{(h)}, \\
 \int_{\Omega} \frac{v_{3*}^n - v_3^{n-1}}{\Delta t} w_3 d\mu + \int_{\Omega} (v^{n-1} \cdot \nabla v_{3*}^n) w_3 d\mu + \nu \int_{\Omega} (\nabla v_{3*}^n \cdot \nabla w_3) d\mu = 0 \quad \forall w_3 \in V_z^{(h)}, \\
 \\
 \int_{\Omega} \nabla p_*^n \cdot \nabla q d\mu = -\frac{1}{\Delta t} \int_{\Omega} (\nabla \cdot v_*^n) q d\mu \\
 \\
 \int_{\Omega} v_1^n w_1 d\mu = \int_{\Omega} v_{1*}^n w_1 d\mu - \Delta t \int_{\Omega} \frac{\partial p_*^n}{\partial x} w_1 d\mu \\
 \\
 \int_{\Omega} v_2^n w_2 d\mu = \int_{\Omega} v_{2*}^n w_2 d\mu - \Delta t \int_{\Omega} \frac{\partial p_*^n}{\partial y} w_2 d\mu \\
 \\
 \int_{\Omega} v_3^n w_3 d\mu = \int_{\Omega} v_{3*}^n w_3 d\mu - \Delta t \int_{\Omega} \frac{\partial p_*^n}{\partial z} w_3 d\mu
 \end{array} \right. \tag{4.6}$$

Substituindo (4.5) em (4.6) e usando a notação de produto interno, temos:

$$\left\{ \begin{array}{l}
\left\{ \begin{array}{l}
\frac{1}{\Delta t} \left(\sum_{j=1}^{n_h^{\widehat{xy}}} v_{1*j}^n \widehat{\varphi}_j^{xy}, w_1 \right)_{\Omega} + \left(v^{n-1} \cdot \nabla \left(\sum_{j=1}^{n_h^{\widehat{xy}}} v_{1*j}^n \widehat{\varphi}_j^{xy} \right), w_1 \right)_{\Omega} \\
+ \nu \left(\nabla \left(\sum_{j=1}^{n_h^{\widehat{xy}}} v_{1*j}^n \widehat{\varphi}_j^{xy} \right) \|\nabla w_1 \right)_{\Omega} = \frac{1}{\Delta t} (v_1^{n-1}, w_1)_{\Omega} \quad \forall w_1 \in V_{xy}^{(h)}, \\
\frac{1}{\Delta t} \left(\sum_{j=1}^{n_h^{\widehat{xy}}} v_{2*j}^n \widehat{\varphi}_j^{xy}, w_2 \right)_{\Omega} + \left(v^{n-1} \cdot \nabla \left(\sum_{j=1}^{n_h^{\widehat{xy}}} v_{2*j}^n \widehat{\varphi}_j^{xy} \right), w_2 \right)_{\Omega} \\
+ \nu \left(\nabla \left(\sum_{j=1}^{n_h^{\widehat{xy}}} v_{2*j}^n \widehat{\varphi}_j^{xy} \right) \|\nabla w_2 \right)_{\Omega} = \frac{1}{\Delta t} (v_2^{n-1}, w_2)_{\Omega} \quad \forall w_2 \in V_{xy}^{(h)}, \\
\frac{1}{\Delta t} \left(\sum_{j=1}^{n_h^z} v_{3*j}^n \varphi_j^z, w_3 \right)_{\Omega} + \left(v^{n-1} \cdot \nabla \left(\sum_{j=1}^{n_h^z} v_{3*j}^n \varphi_j^z \right), w_3 \right)_{\Omega} \\
+ \nu \left(\nabla \left(\sum_{j=1}^{n_h^z} v_{3*j}^n \varphi_j^z \right) \|\nabla w_3 \right)_{\Omega} = \frac{1}{\Delta t} (v_3^{n-1}, w_3)_{\Omega} \quad \forall w_3 \in V_z^{(h)},
\end{array} \right. \quad (4.7) \\
\left(\nabla \left(\sum_{j=1}^{n_h^p} p_{*j}^n \varphi_j^p \right) \|\nabla q \right)_{\Omega} = -\frac{1}{\Delta t} (\nabla \cdot v_*^n, q)_{\Omega} \\
\left\{ \begin{array}{l}
\left(\sum_{j=1}^{n_h^{\widehat{xy}}} v_{1j}^n \widehat{\varphi}_j^{xy}, w_1 \right)_{\Omega} = (v_{1*}^n, w_1)_{\Omega} - \Delta t \left(\frac{\partial p_*^n}{\partial x}, w_1 \right)_{\Omega} \\
\left(\sum_{j=1}^{n_h^{\widehat{xy}}} v_{2j}^n \widehat{\varphi}_j^{xy}, w_2 \right)_{\Omega} = (v_{2*}^n, w_2)_{\Omega} - \Delta t \left(\frac{\partial p_*^n}{\partial y}, w_2 \right)_{\Omega} \\
\left(\sum_{j=1}^{n_h^z} v_{3j}^n \varphi_j^z, w_3 \right)_{\Omega} = (v_{3*}^n, w_3)_{\Omega} - \Delta t \left(\frac{\partial p_*^n}{\partial z}, w_3 \right)_{\Omega}
\end{array} \right.
\end{array} \right.$$

Podemos ainda reorganizar as equações e substituir w_1 , w_2 , w_3 e q pelos elementos das bases respectivas, de forma a ficar:

$$\left\{ \begin{array}{l}
 \left\{ \begin{array}{l}
 \sum_{j=1}^{n_h^{\widehat{xy}}} v_{1*j}^n \left(\frac{1}{\Delta t} (\widehat{\varphi}_j^{xy}, \varphi_i^{xy})_{\Omega} + (v^{n-1} \cdot \nabla \widehat{\varphi}_j^{xy}, \varphi_i^{xy})_{\Omega} + \nu (\nabla \widehat{\varphi}_j^{xy} \parallel \nabla \varphi_i^{xy})_{\Omega} \right) = \\
 \qquad \qquad \qquad = \frac{1}{\Delta t} (v_1^{n-1}, \varphi_i^{xy})_{\Omega} \quad \forall \varphi_i^{xy} \in V_{xy}^{(h)}, \\
 \sum_{j=1}^{n_h^{\widehat{xy}}} v_{2*j}^n \left(\frac{1}{\Delta t} (\widehat{\varphi}_j^{xy}, \varphi_i^{xy})_{\Omega} + (v^{n-1} \cdot \nabla \widehat{\varphi}_j^{xy}, \varphi_i^{xy})_{\Omega} + \nu (\nabla \widehat{\varphi}_j^{xy} \parallel \nabla \varphi_i^{xy})_{\Omega} \right) = \\
 \qquad \qquad \qquad = \frac{1}{\Delta t} (v_2^{n-1}, \varphi_i^{xy})_{\Omega} \quad \forall \varphi_i^{xy} \in V_{xy}^{(h)}, \\
 \sum_{j=1}^{n_h^z} v_{3*j}^n \left(\frac{1}{\Delta t} (\varphi_j^z, \varphi_i^z)_{\Omega} + (v^{n-1} \cdot \nabla \varphi_j^z, \varphi_i^z)_{\Omega} + \nu (\nabla \varphi_j^z \parallel \nabla \varphi_i^z)_{\Omega} \right) = \\
 \qquad \qquad \qquad = \frac{1}{\Delta t} (v_3^{n-1}, \varphi_i^z)_{\Omega} \quad \forall \varphi_i^z \in V_z^{(h)},
 \end{array} \right. \\
 \\
 \sum_{j=1}^{n_h^p} p_{*j}^n (\nabla \varphi_j^p \parallel \nabla \varphi_i^p)_{\Omega} = -\frac{1}{\Delta t} (\nabla \cdot v_*^n, \varphi_i^p)_{\Omega} \\
 \\
 \left\{ \begin{array}{l}
 \sum_{j=1}^{n_h^{\widehat{xy}}} v_{1j}^n (\widehat{\varphi}_j^{xy}, \varphi_i^{xy})_{\Omega} = (v_{1*}^n, \varphi_i^{xy})_{\Omega} - \Delta t \left(\frac{\partial p_*^n}{\partial x}, \varphi_i^{xy} \right)_{\Omega} \\
 \sum_{j=1}^{n_h^{\widehat{xy}}} v_{2j}^n (\widehat{\varphi}_j^{xy}, \varphi_i^{xy})_{\Omega} = (v_{2*}^n, \varphi_i^{xy})_{\Omega} - \Delta t \left(\frac{\partial p_*^n}{\partial y}, \varphi_i^{xy} \right)_{\Omega} \\
 \sum_{j=1}^{n_h^z} v_{3j}^n (\varphi_j^z, \varphi_i^z)_{\Omega} = (v_{3*}^n, \varphi_i^z)_{\Omega} - \Delta t \left(\frac{\partial p_*^n}{\partial z}, \varphi_i^z \right)_{\Omega}
 \end{array} \right.
 \end{array} \right. \quad (4.8)$$

Podemos expressar as equações em (4.8) sob a forma da seguinte cadeia de sistemas lineares:

$$\left\{ \begin{array}{l}
 \left\{ \begin{array}{l}
 (A_1 + B_1^{n-1}) v_{1*}^n = b_{1*}^{n-1} \\
 (A_2 + B_2^{n-1}) v_{2*}^n = b_{2*}^{n-1} \\
 (A_3 + B_3^{n-1}) v_{3*}^n = b_{3*}^{n-1}
 \end{array} \right. \\
 Cp_*^n = c^n \\
 \left\{ \begin{array}{l}
 D_1 v_1^n = b_1^n \\
 D_2 v_2^n = b_2^n \\
 D_3 v_3^n = b_3^n,
 \end{array} \right.
 \end{array} \right. \quad (4.9)$$

sendo:

$$\begin{aligned}
(A_1)_{ij} &= \frac{1}{\Delta t} \left(\varphi_j^{\widehat{xy}}, \varphi_i^{xy} \right)_\Omega + \nu \left(\nabla \varphi_j^{\widehat{xy}} \parallel \nabla \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times n_h^{\widehat{xy}} \\
(A_2)_{ij} &= \frac{1}{\Delta t} \left(\varphi_j^{\widehat{xy}}, \varphi_i^{xy} \right)_\Omega + \nu \left(\nabla \varphi_j^{\widehat{xy}} \parallel \nabla \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times n_h^{\widehat{xy}} \\
(A_3)_{ij} &= \frac{1}{\Delta t} \left(\varphi_j^z, \varphi_i^z \right)_\Omega + \nu \left(\nabla \varphi_j^z \parallel \nabla \varphi_i^z \right)_\Omega, \quad n_h^z \times n_h^z \\
(B_1^{n-1})_{ij} &= \left(v^{n-1} \cdot \nabla \varphi_j^{\widehat{xy}}, \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times n_h^{\widehat{xy}} \\
(B_2^{n-1})_{ij} &= \left(v^{n-1} \cdot \nabla \varphi_j^{\widehat{xy}}, \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times n_h^{\widehat{xy}} \\
(B_3^{n-1})_{ij} &= \left(v^{n-1} \cdot \nabla \varphi_j^z, \varphi_i^z \right)_\Omega, \quad n_h^z \times n_h^z \\
(b_{1*}^{n-1})_i &= \frac{1}{\Delta t} \left(v_1^{n-1}, \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times 1 \\
(b_{2*}^{n-1})_i &= \frac{1}{\Delta t} \left(v_2^{n-1}, \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times 1 \\
(b_{3*}^{n-1})_i &= \frac{1}{\Delta t} \left(v_3^{n-1}, \varphi_i^z \right)_\Omega, \quad n_h^z \times 1 \\
(C)_{ij} &= \left(\nabla \varphi_j^p \parallel \nabla \varphi_i^p \right)_\Omega, \quad n_h^p \times n_h^p \\
(c^n)_i &= -\frac{1}{\Delta t} \left(\nabla \cdot v_*^n, \varphi_i^p \right)_\Omega, \quad n_h^p \times 1 \\
(D_1)_{ij} &= \left(\varphi_j^{\widehat{xy}}, \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times n_h^{\widehat{xy}} \\
(D_2)_{ij} &= \left(\varphi_j^{\widehat{xy}}, \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times n_h^{\widehat{xy}} \\
(D_3)_{ij} &= \left(\varphi_j^z, \varphi_i^z \right)_\Omega, \quad n_h^z \times n_h^z \\
(b_1^n)_i &= \left(v_{1*}^n, \varphi_i^{xy} \right)_\Omega - \Delta t \left(\frac{\partial p_*^n}{\partial x}, \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times 1 \\
(b_2^n)_i &= \left(v_{2*}^n, \varphi_i^{xy} \right)_\Omega - \Delta t \left(\frac{\partial p_*^n}{\partial y}, \varphi_i^{xy} \right)_\Omega, \quad n_h^{xy} \times 1. \\
(b_3^n)_i &= \left(v_{3*}^n, \varphi_i^z \right)_\Omega - \Delta t \left(\frac{\partial p_*^n}{\partial z}, \varphi_i^z \right)_\Omega, \quad n_h^z \times 1
\end{aligned}$$

A criação de duas matrizes A e B , se deve ao fato de necessitarmos atualizar B em cada passo no tempo.

4.2 Discretizações temporais

Nesta seção descreveremos as discretizações relativas ao tempo. Para a equação de Navier-Stokes a discretização do tempo já foi feita na construção do método das projeções (discretização semi-implícita). A equação de advecção-difusão terá o tempo discretizado através do esquema de diferenças finitas conhecido pelo nome de Crank-Nicolson. Esse método gera um

esquema incondicionalmente estável e de ordem 2 e por estas razões tem sido muito utilizado. Para construir este esquema usamos diferença centrada no tempo $t^{n+1/2}$, obtendo as seguintes aproximações:

$$\left. \frac{du_j}{dt} \right|_{t^{n+1/2}} = \frac{u_j^{n+1} - u_j^n}{\Delta t}, \quad u_j^{n+1/2} \approx \frac{u_j^{n+1} + u_j^n}{2} \quad \text{e} \quad W^{n+1/2} = \frac{W^{n+1} + W^n}{2}.$$

Sendo assim, temos que a solução do sistema de equações diferenciais (4.3) pode ser aproximada pela solução do sistema seguinte:

$$\begin{aligned} M \frac{u_h^{n+1} - u_h^n}{\Delta t} + N \frac{u_h^{n+1} + u_h^n}{2} + \frac{W^{n+1} + W^n}{2} \frac{u_h^{n+1} + u_h^n}{2} &= F^{n+1/2} \\ \left(M + \frac{\Delta t}{2} N + \frac{\Delta t}{2} \left(\frac{W^{n+1} + W^n}{2} \right) \right) u_h^{n+1} &= \left(M - \frac{\Delta t}{2} N - \frac{\Delta t}{2} \left(\frac{W^{n+1} + W^n}{2} \right) \right) u_h^n + \Delta t F^{n+1/2} \end{aligned} \quad (4.10)$$

4.3 Discretização do domínio

Para a resolução do sistema linear que consta em (4.10), e também dos sistemas lineares que serão produzidos a partir de (4.8), necessitamos que se faça uma malha sobre o domínio Ω e também uma adequada escolha das funções que compõem as bases dos espaços $U^{(h)}$, $V_{xy}^{(h)}$, $V_z^{(h)}$, $V_p^{(h)}$ e $V_{\widehat{xy}}^{(h)}$ que darão as aproximações espaciais para a solução via elementos finitos. Existem muitos poliedros que podem ser usados na discretização de um domínio. Dentre esses escolhemos o tetraedro tendo em vista a adaptação deste com a complexidade do nosso domínio.

Denotemos com $\{k_e\}_{e=1}^{nte}$ uma família finita de nte tetraedros k_e , dois a dois disjuntos ou tendo como interseção somente uma face, uma aresta ou um vértice e tal que:

$$\bar{\Omega} = \bigcup_{e=1}^{nte} k_e,$$

e, associado a esta malha, definimos o parâmetro h dado por

$$h = \max_{e=1:nte} \{diam(k_e)\}$$

Dentre as inúmeras escolhas para $U^{(h)}$, escolhemos o espaço das funções polinomiais de três variáveis de grau igual a 1 definidas em k_e , ou seja:

$$P_1(k_e) = \left\{ \mathbf{p} : k_e \rightarrow \mathbb{R} \quad / \quad \mathbf{p}(x, y, z) = a_1 x + a_2 y + a_3 z + a_4, \quad a_1, a_2, a_3, a_4 \in \mathbb{R} \right\}$$

Podemos assim definir $U^{(h)}$, $V_{xy}^{(h)}$, $V_z^{(h)}$, $V_p^{(h)}$ e $V_{\widehat{xy}}^{(h)}$ como sendo:

$$\begin{aligned} U^{(h)} &= \left\{ \varphi \in C^0(\bar{\Omega}) / \varphi|_{k_e} \in P_1(k_e), \forall k_e \in \{k_e\}_{e=1}^{nte} \right\} \\ V_{xy}^{(h)} &= \left\{ \varphi \in C^0(\bar{\Omega}) / \varphi|_{k_e} \in P_1(k_e), \forall k_e \in \{k_e\}_{e=1}^{nte}, \varphi|_{\Gamma_{xy}} = 0 \right\} \\ V_z^{(h)} &= \left\{ \varphi \in C^0(\bar{\Omega}) / \varphi|_{k_e} \in P_1(k_e), \forall k_e \in \{k_e\}_{e=1}^{nte}, \varphi|_{\Gamma_z} = 0 \right\} \\ V_p^{(h)} &= \left\{ \varphi \in C^0(\bar{\Omega}) / \varphi|_{k_e} \in P_1(k_e), \forall k_e \in \{k_e\}_{e=1}^{nte}, \varphi|_{\Gamma_4} = 0 \right\} \\ V_{\widehat{xy}}^{(h)} &= \left\{ \varphi \in C^0(\bar{\Omega}) / \varphi|_{k_e} \in P_1(k_e), \forall k_e \in \{k_e\}_{e=1}^{nte} \right\} \end{aligned}$$

Tomaremos como base desses espaços, funções φ_i , tais que:

$$\varphi_i(b_j) = \delta_{ij}, \quad i, j = 1 : ntn,$$

sendo δ_{ij} o *delta de Kronecker*, ntn o número total de nós e b_j o j -ésimo nó da malha.

Vamos definir o tetraedro cujos vértices são $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 0)$ e $(0, 0, 1)$, como sendo de referência: \hat{k} .

Dessa forma, as funções φ 's restritas a esse elemento de referência, terão a forma:

$$\begin{aligned} \varphi_1(\xi, \eta, \zeta) &= \xi \\ \varphi_2(\xi, \eta, \zeta) &= \eta \\ \varphi_3(\xi, \eta, \zeta) &= 1 - \xi - \eta - \zeta \\ \varphi_4(\xi, \eta, \zeta) &= \zeta \end{aligned}$$

A técnica usada na implementação do método de elementos finitos requer que se defina uma transformação afim $T_e : \hat{k} \rightarrow k_e$ que relaciona um elemento referência \hat{k} , a um elemento genérico k_e . Na Figura 4.1 ilustramos essa transformação T_e . Com todas essas informações, podemos definir:

$$T_e(\xi, \eta, \zeta) = \begin{pmatrix} x_1 - x_3 & x_2 - x_3 & x_4 - x_3 \\ y_1 - y_3 & y_2 - y_3 & y_4 - y_3 \\ z_1 - z_3 & z_2 - z_3 & z_4 - z_3 \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} + \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix} \quad (4.11)$$

A construção da malha usada no trabalho, foi feita com o software livre GMSH⁴ usando dados obtidos do projeto da APM Manso. Devido a grande complexidade, desde o tratamento

⁴<http://geuz.org/gmsh/>, consultada em 20/04/2011

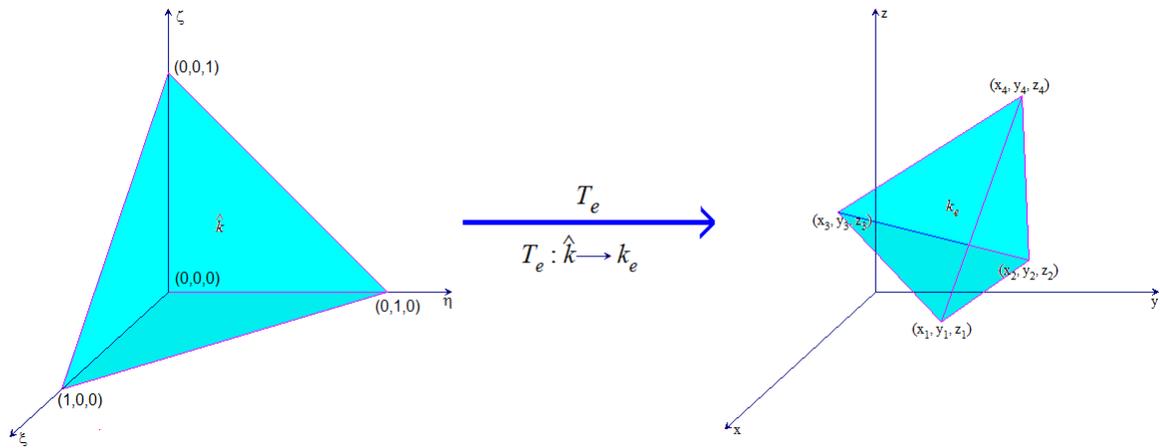


Figura 4.1: Transformação afim $T_e : \hat{k} \rightarrow k_e$.

dos dados do projeto e geração do sólido até a construção da malha, decidimos detalhar todo esse procedimento no Anexo 1. Como resultado do que foi aí feito, temos uma malha que está ilustrada nas Figuras 4.2, 4.3, 4.4 e 4.5. Na Figura 4.3 temos a visão de baixo para cima da parte mais profunda do lago. A intenção desta imagem é de mostrar as camadas inferiores da malha. As Figuras 4.4 e 4.5 referem-se aproximadamente ao mesmo local, tendo como diferença somente o recurso de transparência das faces dos tetraedros.

A malha gerada possui os seguintes valores e características:

- $ntn = 724.008$, número total de nós;
- $nf = 442.613$, número de nós de fronteira;
- $ni = 281.395$, número de nós internos;
- $nfe = 2.949.031$, número de tetraedros;

Uma particularidade muito útil do GMSH é a organização dos pontos: no arquivo de saída, primeiro são colocados os pontos de fronteira e depois os internos. Essa organização facilita na implementação do método de elementos finitos. Sendo assim, os primeiros 442.613 pontos são de fronteira e do ponto 442.614 até o ponto 724.008 são pontos internos ao domínio.

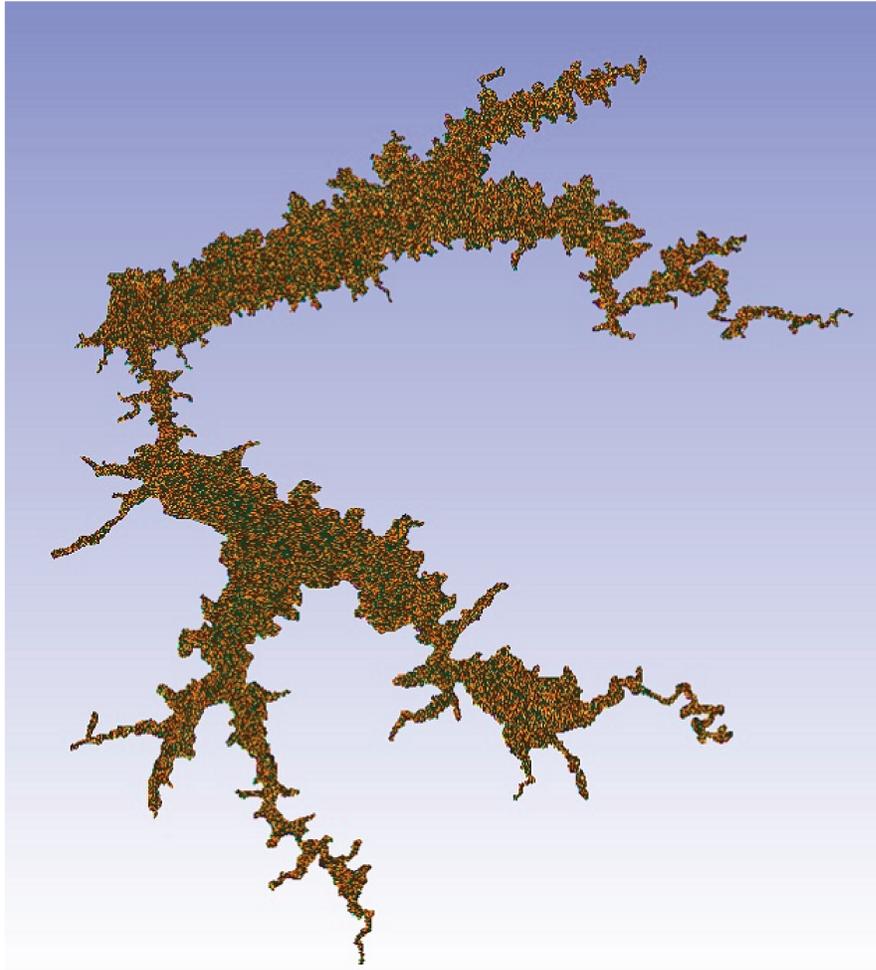


Figura 4.2: Imagem produzida pelo GMSH, ilustrando a malha.

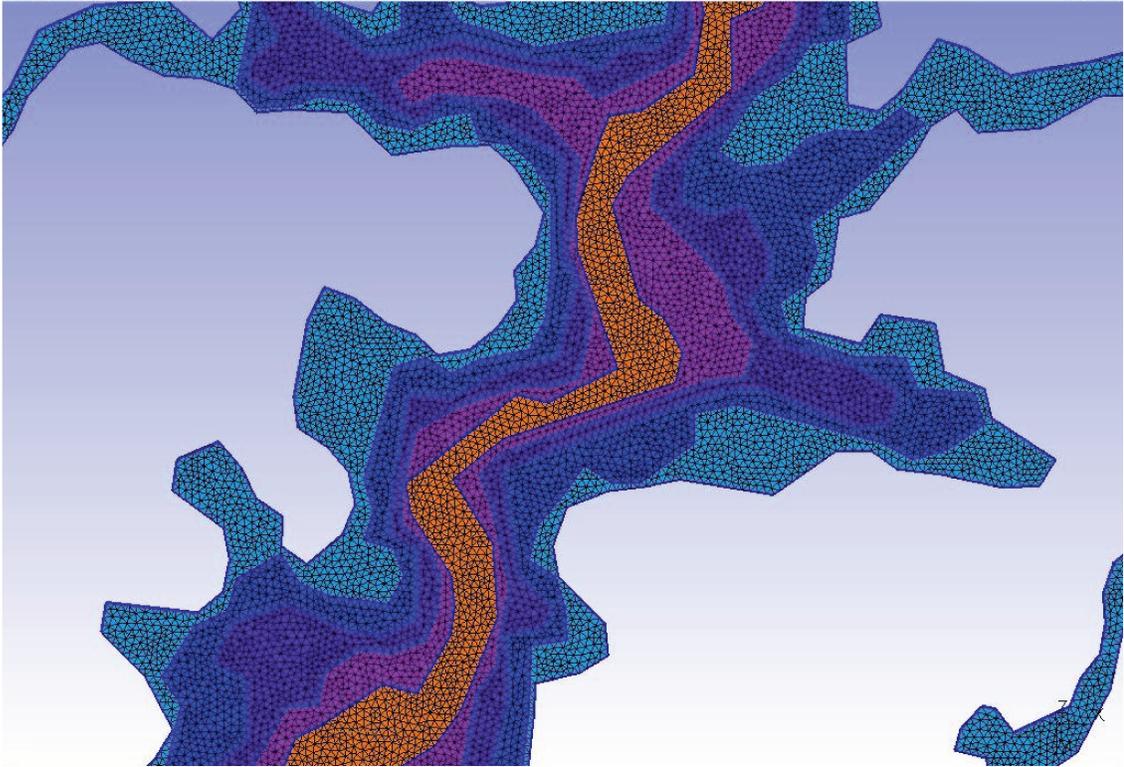


Figura 4.3: Visão da malha produzida pelo GMSH, de baixo para cima.

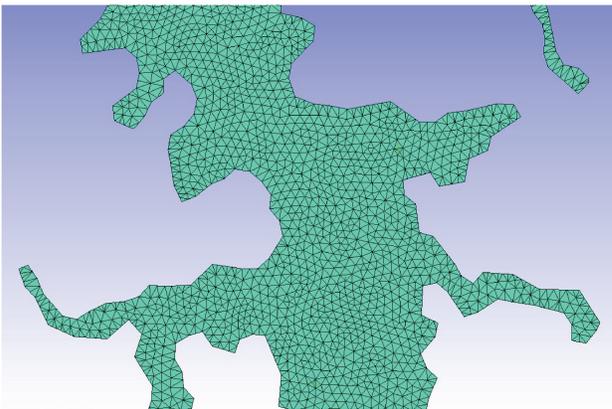


Figura 4.4: Visão da malha na superfície do lago.

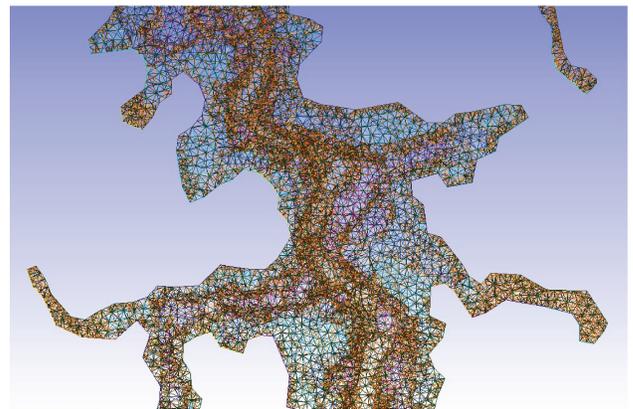


Figura 4.5: Recurso de transparência nos tetraedros.

Capítulo 5

Experiências numéricas

Neste capítulo descreveremos os resultados numéricos obtidos através da simulação de alguns cenários que julgamos contribuir para a análise do problema e dos algoritmos. Primeiro relataremos alguns resultados sobre a equação de Navier-Stokes e depois os cenários para a equação de Difusão-Advecção. Todas as simulações levaram em conta parâmetros obtidos na literatura geral ou específica da região da APM-Manso, ou seja, o projeto desta tese não envolveu a coleta de dados ou qualquer outro trabalho de campo. Assim, daremos grande enfoque aos resultados qualitativos que são de grande importância na validação do modelo e dos algoritmos.

5.1 Navier-Stokes

As simulações computacionais para a equação de Navier-Stokes, na prática ocorreram junto com as da equação de Difusão-Advecção, tendo em vista a necessidade de cálculo da aproximação do campo de velocidades a cada passo no tempo, para se calcular a aproximação do perfil de poluição neste passo. Devido ao grande esforço computacional, não pudemos realizar a simulação de vários cenários também para a equação de Navier-Stokes e com isso, decidimos utilizar um mesmo cenário de aproximação de Navier-Stokes nos cenários da Difusão-Advecção, ressaltando que mesmo sendo o mesmo campo, ele sempre foi calculado para cada cenário, em razão da dificuldade de armazenamento de tamanho volume de dados. Depois de vários testes de cenários com número de Reynolds variando de 100 até $2,77 \times 10^8$,

escolhemos o cenário com número de Reynolds igual a 100. A escolha deste levou em conta os resultados numéricos comparados com alguns dados sobre circulação na represa de Manso, pelo regime turbulento que queríamos evitar e principalmente por causa da estabilidade numérica da equação de Difusão-Advecção, cujos detalhes serão discutidos na próxima seção. Com a equação de Navier-Stokes, avançamos até a iteração número 4.000 onde podemos observar, além da estabilização da solução, também a estabilidade do algoritmo.

A Figura 5.1 ilustra uma ampliação de um quadro ($28, 9 \leq x \leq 31$ e $27 \leq y \leq 30, 2$) onde se observa o campo de velocidades na iteração 1350 numa região de transição rio-represa. Na parte esquerda temos o lago todo, mas contendo a representação de apenas 1% das setas que compõem o campo¹. Na parte direita temos a ampliação com todos as setas do referido quadro.

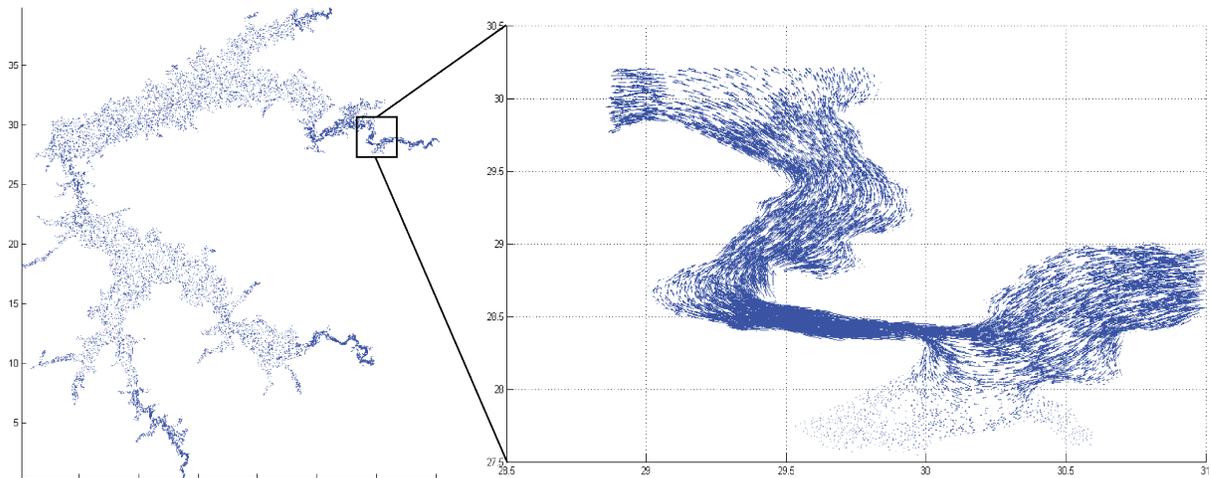


Figura 5.1: Ampliação de um quadro próximo a entrada do rio Manso.

As Figuras 5.2 e 5.3 ilustram o campo de velocidades no trecho do lago em que $3.85 \leq x \leq 6$ e $25.5 \leq y \leq 26.25$ na iteração número 1350. A diferença entre elas é somente a norma dos vetores: a primeira representa o campo de velocidades real enquanto que na segunda, normalizamos todos os vetores velocidade objetivando a visualização da direção do campo nos locais onde a circulação é muito baixa, por exemplo nas pequenas baías formadas.

¹Mesmo nos computadores de alto desempenho que tivemos acesso, não foi possível a plotagem do total de setas.

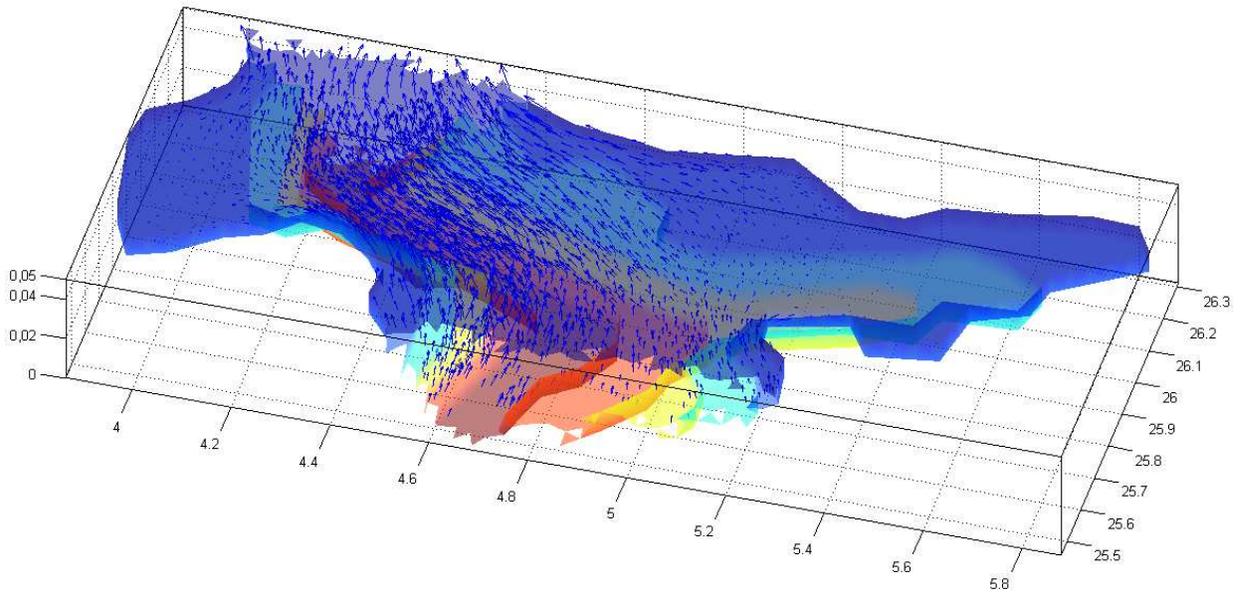


Figura 5.2: Campo de velocidades obtido das simulações.

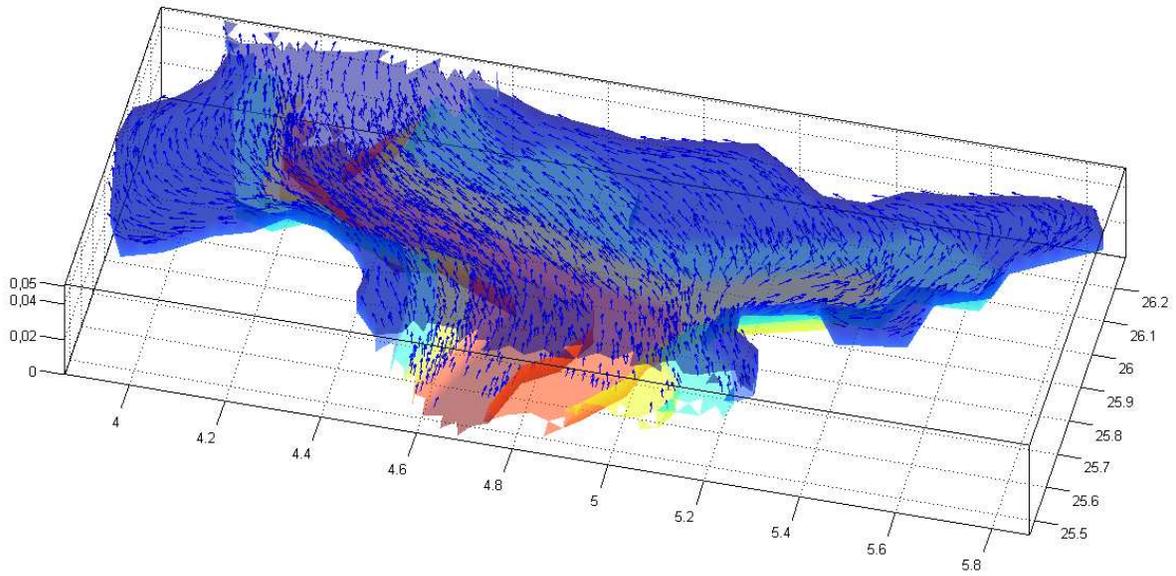


Figura 5.3: Campo obtido das simulações com velocidades normalizadas.

5.2 Cenários para poluição

Nesta seção descreveremos as simulações feitas em computador, de 3 cenários escolhidos de forma a retratar possíveis situações reais. Na Tabela 5.1 temos um resumo dos valores utilizados nas 3 situações. O coeficiente de difusão α foi mantido constante nas três situações.

Fazendo também uma leitura transversal nessa tabela, vemos que, em resumo, no cenário 1 temos um *runoff* pequeno e um fluxo razoavelmente grande de entrada de poluentes através dos rios. Na absorção nas margens com mata (k_1), e no fundo do lago (k_3), temos valores grandes, se comparados com os demais cenários. Temos também uma grande degradação, expressa pelo coeficiente σ .

No segundo cenário diminuimos a degradação, diminuimos a saída de poluente pelas margens com mata e fundo e também a entrada pelos rios e por *runoff*, resumindo: menos entrada e também menos saída.

Por fim o terceiro cenário traz uma situação onde a entrada de poluição pelos rios é mínima, o *runoff* é o mesmo do cenário 2, ou seja, mediano e a perda para margens com mata igual à zero e para o fundo, muito pequena. Temos também uma degradação bem menor do que nos demais cenários.

A etapa de simulações foi dividida em duas fases. Na primeira fizemos 4.000 iterações, o que equivale ao tempo real de 8,3 dias. Nesta fase procedemos a simulação do campo de velocidades juntamente com o perfil de poluição. Na segunda fase fixamos o campo de velocidades dado pela iteração número 4.000 e avançamos com a equação de Difusão-Advecção até obtermos a estabilização da solução. Para os cenários 1 e 2 fomos até a iteração número 600.000 (3,5 anos). Já no cenário 3 avançamos até a iteração número 2.000.000 (11,6 anos) para obter essa estabilização. Para efeito de comparação a Figura 5.4 mostra um gráfico com a norma 2 da diferença entre duas iterações até a iteração número 100.000 onde já temos essa norma na ordem 10^{-5} . Vale ressaltar que o vetor solução tem 724.008 entradas, portanto a norma dessa diferença é relativamente pequena.

Uma condição suficiente (mas não necessária) para a estabilidade numérica é o controle do número de Péclet:

$$Péclet_{x_i} = \frac{v_i \cdot \Delta x_i}{\alpha} \leq 2, \quad i = 1, 2, 3,$$

sendo v_i a componente da velocidade na direção x_i , Δx_i o passo da malha na direção x_i e α o coeficiente de difusão. No nosso problema obtivemos 0,3% dos elementos com número de Péclet maior que 2, nas componentes x e y , mas não em z . Mesmo com esse problema conseguimos obter estabilidade. A escolha do número de Reynolds na seção anterior, também levou isso em consideração, pois com outros valores, obtínhamos um campo de velocidades que fazia com que o número de Péclet ultrapassasse 2 numa grande quantidade de elementos, gerando problemas numéricos que nós observamos na prática.

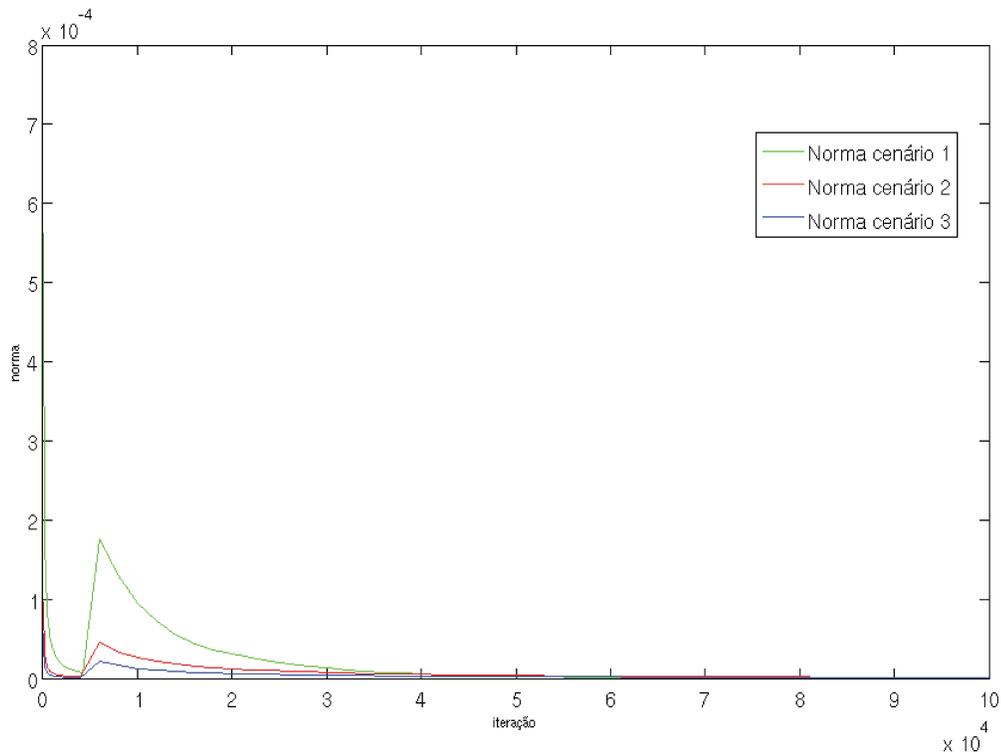


Figura 5.4: Norma da diferença entre iterações.

Para observarmos a variação da concentração de poluição ao longo do tempo, escolhemos alguns pontos. O número associado a cada ponto é o número do nó da malha. A Figura 5.5, mostra a localização dos pontos.

- nó 442.267: próximo à barragem, 50m de profundidade;
- nó 689.631: no meio do braço da represa formado pelos rios Manso e Palmeiras, 13m de profundidade;

- nó 716.869: no meio do braço da represa formado pelos rios Casca e Quilombo, 28m de profundidade;
- nó 392.283: próximo à entrada do rio Quilombo na represa, 30m de profundidade;
- nó 216.734: próximo à entrada do rio Casca na represa, superfície;
- nó 537.315: próximo à entrada do rio Manso na represa, 5m de profundidade;
- nó 369.379: próximo à entrada do rio Palmeiras na represa, 20m de profundidade.

Coeficiente	Valores cenário 1	Valores cenário 2	Valores cenário 3	Unidade
Δt	0,05	0,05	0,05	h
Re	100	100	100	-
α	0,01	0,01	0,01	km^2/h
σ	10^{-4}	10^{-5}	10^{-6}	$1/h$
k_1	10^{-5}	10^{-6}	0	km/h
k_2	10^{-6}	10^{-6}	10^{-6}	km/h
k_3	10^{-5}	10^{-6}	10^{-8}	km/h
k_4	10^{-3}	10^{-3}	10^{-3}	km/h
θ_5	5×10^{-6}	5×10^{-5}	5×10^{-5}	$(ppm)(km/h)$
θ_6	10^{-4}	10^{-5}	10^{-8}	$(ppm)(km/h)$
θ_7	10^{-4}	10^{-5}	10^{-8}	$(ppm)(km/h)$
θ_8	10^{-4}	10^{-5}	10^{-8}	$(ppm)(km/h)$
θ_9	10^{-4}	10^{-5}	10^{-8}	$(ppm)(km/h)$
$\ \vec{g}\ $	1	1	1	km/h

Tabela 5.1: valores de coeficientes nos 3 cenários

As Figuras 5.6 e 5.7 retratam a dispersão de poluente na superfície do lago. A Figura 5.6 mostra os 3 cenários em iterações que vão de 1.000 (2,08 dias) até 4.000 (8,3 dias) e a Figura 5.7 mostra da iteração 10.000 (20,8 dias) até a 600.000 (3,5 anos). Mantivemos um mesmo mapa de cores dentro dos gráficos de cada figura, visando poder ter comparação

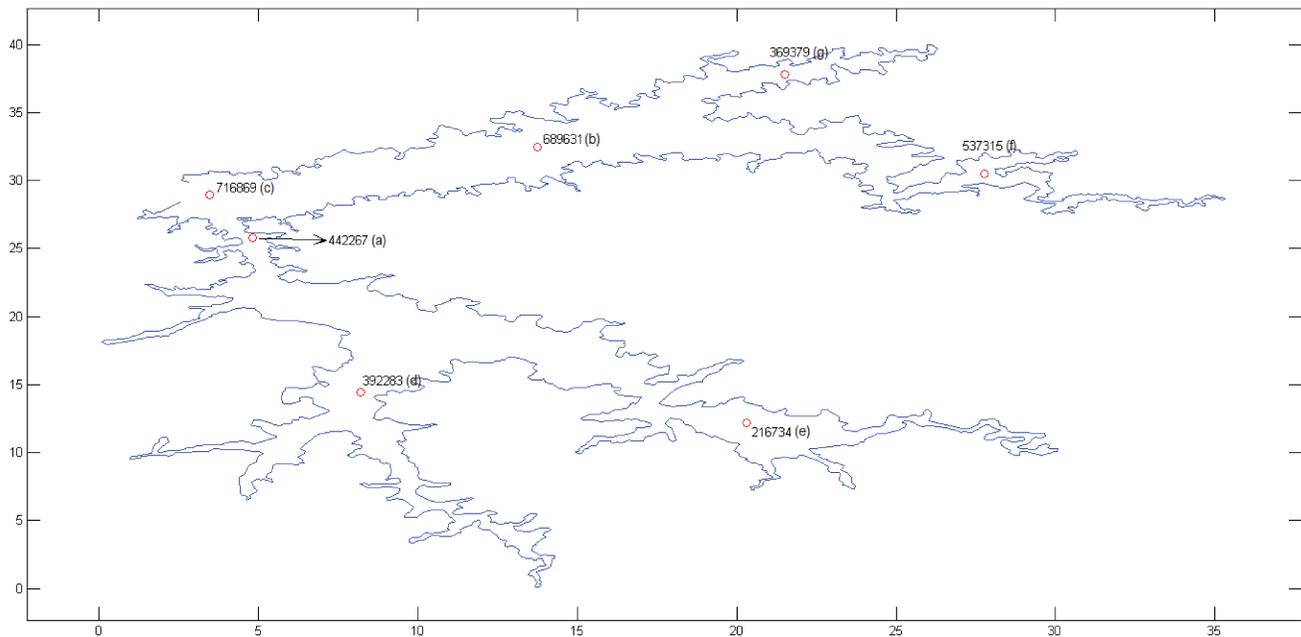


Figura 5.5: Pontos analisados.

quantitativa entre os cenários. Nestas figuras podemos ver as diferenças de cenário impostas pela variações de parâmetros. Observa-se por exemplo no cenário 3 a grande influência do aumento do *runoff* e diminuição da poluição chegando pelos rios.

A Figura 5.8 refere-se à região do lago de coordenadas cartesianas tais que $3,85 \leq x \leq 6$ e $25,5 \leq y \leq 26,25$. Nesses gráficos consideramos a iteração 1.000 e colocamos cortes no lago para mostrar os níveis mais profundos do lago. Nessa figura também consideramos um mesmo mapa de cores entre os cenários e profundidades, a fim de permitir comparações quantitativas. Já a Figura 5.9 mostra a mesma região, mas desta vez considerando a evolução no tempo e sem a pré-definição de um mapa de cores, em virtude da impossibilidade de se observar as mudanças, dada a grande variação de concentração ao longo desse tempo, o que faz com que não tenhamos diferenciação de cores de modo a ilustrar tal evolução. Consideramos nesta figura somente o cenário 1, fato justificado pela semelhança com os demais cenários, tendo em vista o local escolhido (canal próximo à barragem).

Como ilustra a Figura 5.5, escolhemos 7 pontos distribuídos pelo lago para analisarmos a evolução temporal da concentração de poluente. Esses pontos foram tomados tentando considerar as regiões de entrada dos rios, região intermediária do lago e também as proximi-

dades da saída pelo vertedouro. Os resultados dessa análise, estão nas Figuras 5.10 e 5.11. A Figura 5.10 mostra em suas linhas os cenários 1, 2 e 3 e em suas colunas as iterações de 1 a 50.000 e de 1 a 600.000, com exceção da última, que ao invés de 600.000, vai até 1 milhão. Observa-se por exemplo, nessa figura, primeira linha e primeira coluna, que o ponto f é o primeiro a receber poluição (dentre os 7) e o ponto c é o último, comportamento esse, totalmente coerente com a posição deles no lago. Já a Figura 5.11 mostra um comparativo entre os cenários em cada ponto, ou seja, cada gráfico contém 3 curvas representando os 3 cenários num mesmo ponto. Aqui também, uma análise detalhada dos gráficos evidencia as características de cada cenário. Por exemplo a intensidade de poluição e a ordem com que cada ponto passa a receber o poluente, tudo isso em perfeita concordância com o pré-definido e esperado em cada cenário.

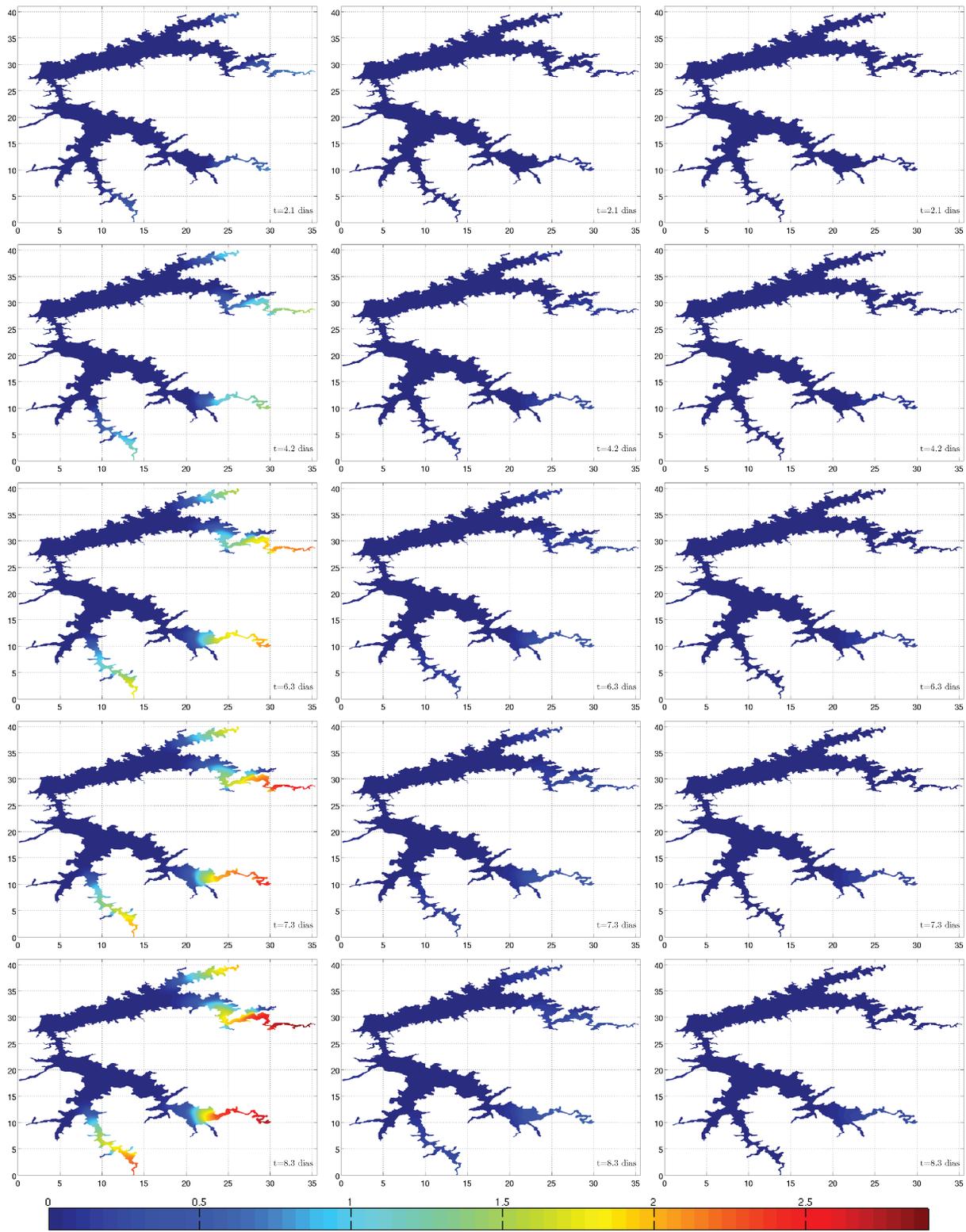


Figura 5.6: Linhas: iterações 1.000, 2.000, 3.000, 3.500 e 4.000. Colunas: cenários 1, 2 e 3.

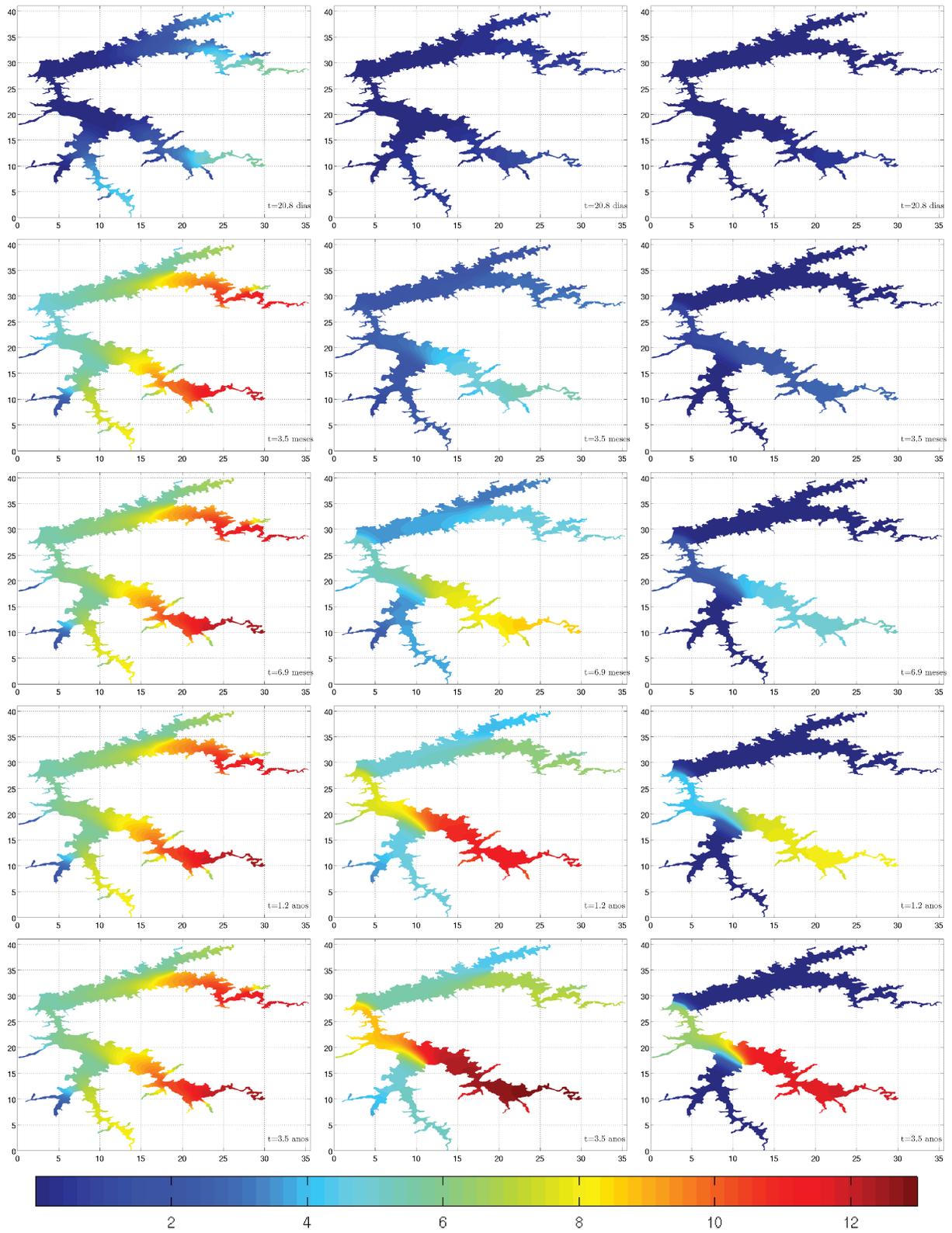


Figura 5.7: Linhas: iterações 10.000, 50.000, 100.000, 200.000 e 600.000. Colunas: cenários 1, 2 e 3.

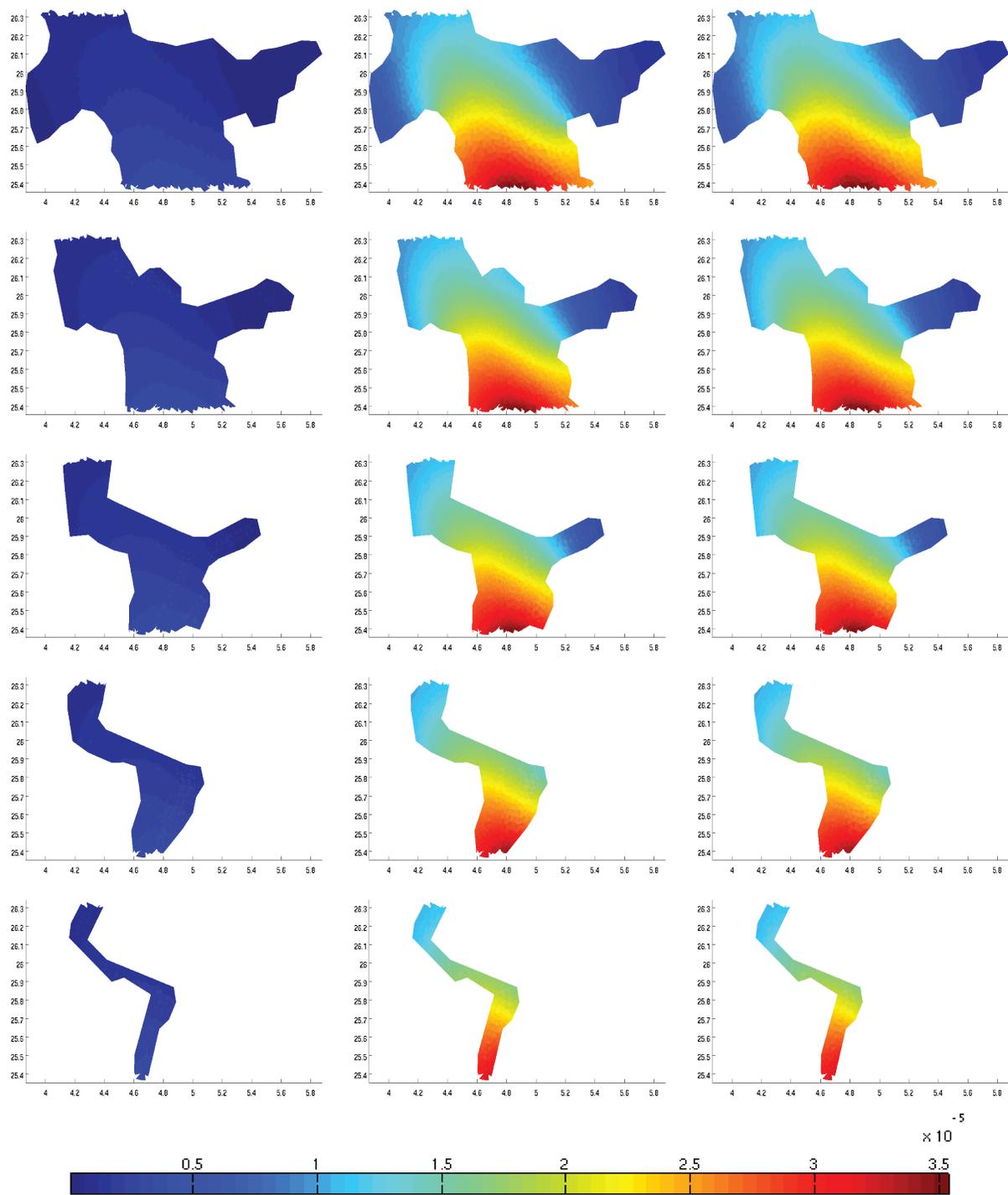


Figura 5.8: Linhas: níveis 0, 10, 20, 30 e 40m de profundidade. Colunas: cenários 1, 2 e 3.

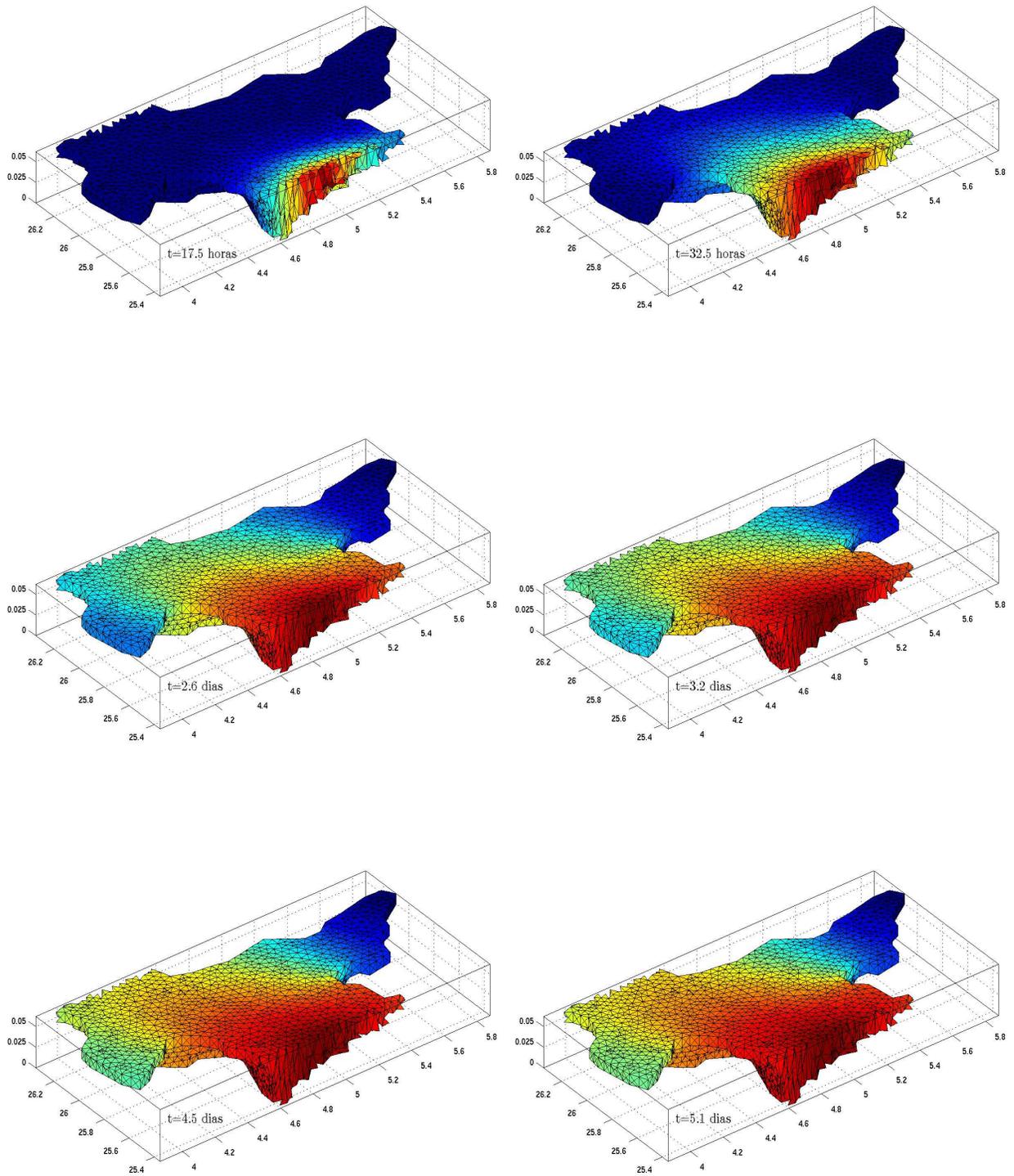


Figura 5.9: Cenário 1: iterações de 350 a 2.450 ($t = 17,5$ horas a $t = 5.1$ dias).

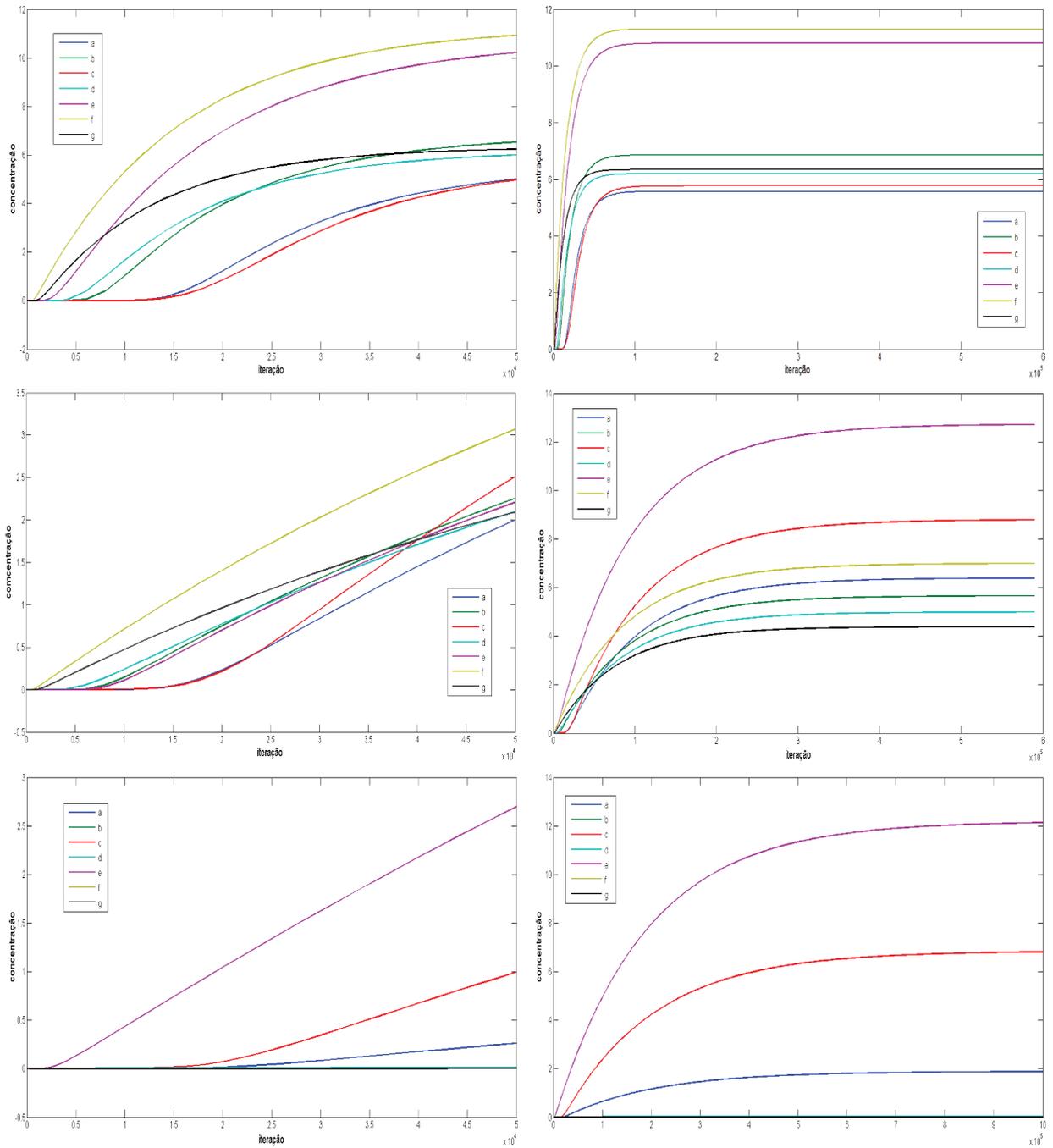


Figura 5.10: Colunas de gráficos: iterações de 1 a 50.000 e de 1 a 600.000. Linhas de gráficos: cenários.

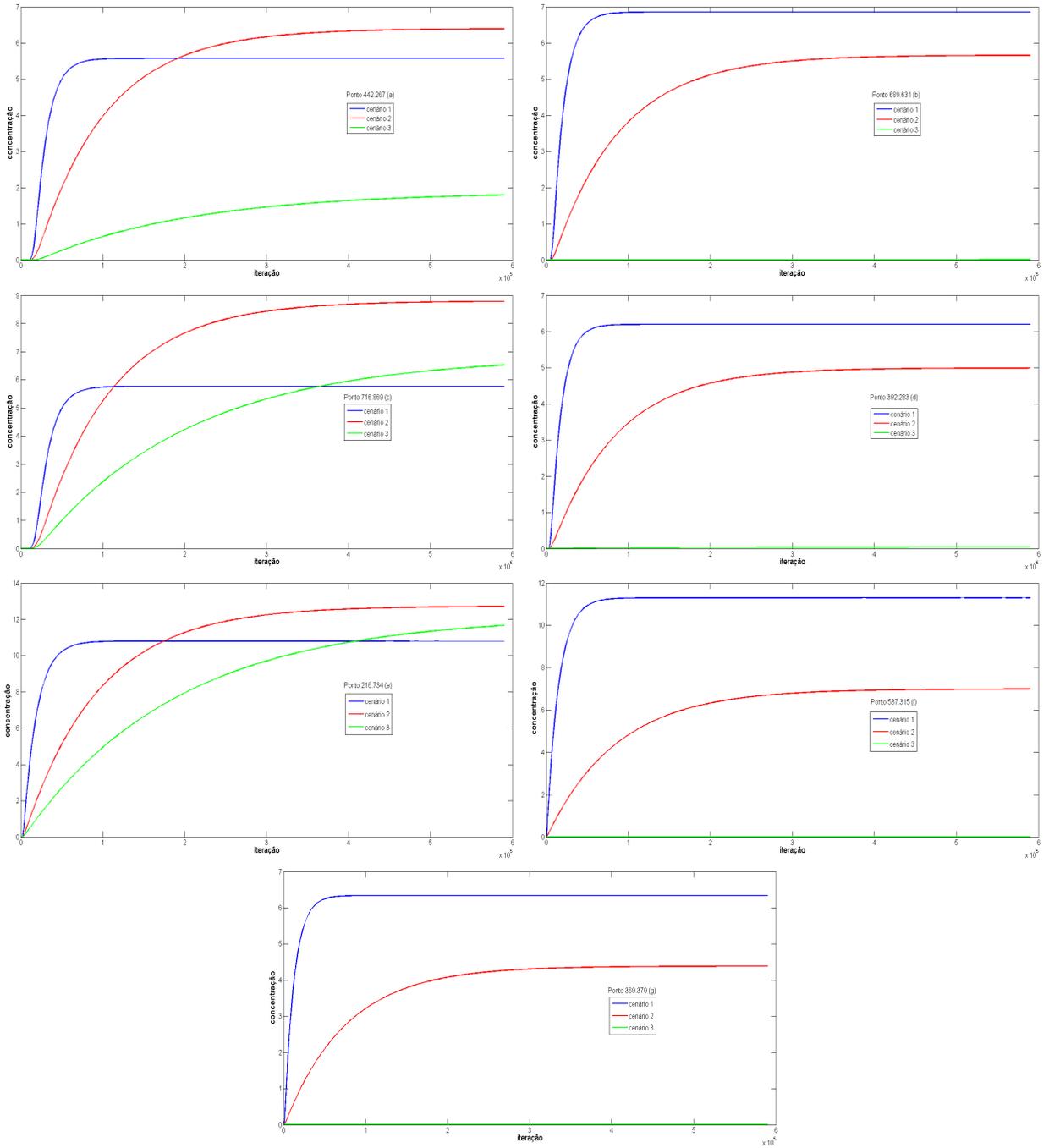


Figura 5.11: Gráficos comparativos de cada ponto nos 3 cenários.

Comentários e sugestões.

Reservamos essa parte da tese para relatar algumas conclusões a que chegamos depois de concluído o trabalho.

De maneira geral podemos afirmar que a modelagem, discretização e a simulação do problema geraram resultados muito coerentes com todas as hipóteses feitas. Como mostram os resultados do capítulo 5, os algoritmos se mostraram robustos pois evidenciaram nos gráficos as particularidades de cada cenário, bem como estabilidade, pois fomos até a iteração 600.000 nos cenários 1 e 2 e até 2.000.000 no cenário 3. Para se chegar na estabilidade, inúmeras horas de processamento foram usadas em testes dos mais variados tipos. Vários foram os problemas que exigiram abordagem diferenciada, sempre motivados pelo grande porte do problema. Podemos citar, entre outros problemas, limitações de linguagem (MatLab), que em certo momento nos causou dúvida quanto à possibilidade de simulação nessa linguagem e também quanto a escassez de recursos computacionais à altura do volume de dados manipulados.

Queremos também, evidenciar a característica generalista de toda a construção. Como foi relatado desde a introdução, afirmamos aqui que devido ao cuidado em deixar cada etapa do modo mais geral possível, estamos prontos, em nível de algoritmo para tratar de uma gama de problemas dessa categoria, o que faz parte do objetivo dessa tese.

Como sugestão para melhoria dos resultados podemos incorporar uma melhor ordem de aproximação nos elementos finitos, ou o ideal, utilizar técnicas de Galerkin Descontínuo, aumentando o grau polinomial da aproximação somente onde a malha é inevitavelmente grosseira, aumentando a precisão, sem um aumento do custo computacional, como seria no convencional. A utilização de SUPG (Streamline Upwind Petrov-Galerkin) para que se possa ter mais liberdade com o campo de velocidades (número de Péclet grande) sem que isso reflita em instabilidades numéricas.

Como melhoria do modelo, sugerimos a incorporação da influência do vento na superfície do lago, incorporando o meio aéreo como feito em [1] e [2]. Condições de fronteira com variações temporais também dariam mais realismo ao modelo, tendo em vista as particularidades da região.

Anexo 1: Tratamento dos dados sobre a geometria e criação da malha

Neste anexo colocaremos os dados sobre a construção da geometria que serviu como domínio de nosso problema. Inicialmente solicitamos junto a FURNAS dados da batimetria do lago, mas tivemos informação que ainda não havia sido realizada tal medição depois do lago cheio. Pedimos então informações do projeto da usina, onde consta toda a medição feita no campo antes do enchimento (fase de projeto) e que revelam as curvas de nível. Nosso contato em FURNAS foi o engenheiro Ademar de Brito que nos disponibilizou não só os dados de curvas, mas sim o projeto total da usina. O projeto foi concebido usando o software ArcView[®], do qual não tínhamos nem licença nem conhecimento técnico para a extração dos dados que nos interessavam. Com a ajuda do professor Dr. Álvaro Penteado Crósta do Instituto de Geociências da UNICAMP, através de seu orientando de doutorado Marcos A.R.Vasconcelos, é que pudemos ter acesso às informações que nos auxiliariam na digitalização de nosso domínio.

A Figura A1.1 mostra as curvas de nível feitas no local próximo de onde viria a ser a barragem. Esses dados foram obtidos no projeto da usina e serviram para a demarcação da região alagável. Já a Figura A1.2 mostra somente as 11 curvas de nível que demarcam o lago. As cotas são de 275m (alaranjada), 270m, 265m, 260m, 255m, 250m, 245m, 240m, 235m, 230m e 225m (magenta) em relação ao nível do mar e estão georeferenciados segundo Córrego Alegre. Devido à complexidade de se trabalhar com um sólido que contivesse todas essas curvas de nível, decidimos aproximar, usando 6 das 11 disponíveis, ficando apenas com níveis com diferença de 10m.

O software livre GMSH que foi utilizado na confecção da malha, a faz levando em consi-

deração, entre outras coisas, a densidade de pontos da curvas que compõem as fronteiras do sólido. Por isso, era extremamente necessário refinar a quantidade de pontos, pois somente na cota $275m$ temos 58.175 pontos, quantidade esta suficiente para extrapolar os limites do software ou, se não, gerar uma malha absurda e desnecessariamente refinada. Tal refinamento foi realizado em ambiente MatLab[®] e composto pela implementação de 3 filtros:

- O primeiro deles retira pontos que estão alinhados observando somente a vizinhança local;
- O segundo retira também pontos alinhados tendo em vista a curvatura da fronteira;
- O terceiro filtro retira pontos que demarquem bicos muito fechados, retirando da curva pequenas reentrâncias que não interferem no macro comportamento do fenômeno estudado.

A Figura A1.3 mostra a curva da cota $275m$, ou seja, da superfície do lago, antes (58.175 pontos) e depois (3826 pontos) da passagem desses filtros. Realizamos muitos testes calibrando esses filtros a fim de gerar uma curva próxima da original com a menor quantidade de pontos possível. As demais curvas também foram submetidas às mesmas intervenções, com redução de pontos e fidelidade próximas do que ocorreu com a curva da superfície.

Para que o GMSH assimile o sólido para fazer a malha, é necessário que o alimentemos com um arquivo .txt contendo informações sobre os pontos, sobre linhas que ligam esses pontos, sobre as superfícies geradas por essas linhas, tudo numa sintaxe específica do programa. Fizemos essa tarefa em ambiente MatLab gerando assim o arquivo de entrada para o GMSH.

O GMSH possui diversos algoritmos para a confecção da malha, bem como parâmetros em cada um desses. Depois de reconhecido o sólido tridimensional pelo GMSH, iniciamos alguns testes nas malhas produzidas, alterando os algoritmos e também parâmetros. A intenção era produzir uma malha de qualidade, tendo em vista a complexidade do sólido. Esses testes envolveram quantidade de elementos que interceptam cada nó, regularidade do elemento e tamanho do elemento. Chegamos por fim na seguinte malha:

- $n_{tn} = 724.008$, número total de nós;
- $n_f = 442.613$, número de nós de fronteira;

- $n_i = 281.395$, número de nós internos;
- $n_{te} = 2.949.031$, número de tetraedros;
- Algoritmo usado para a geração da malha 2D: MeshAdapt+Delaunay;
- Algoritmo usado para a geração da malha 3D: Tetgen+Delaunay.

Teoricamente, seria possível produzir uma malha melhor, mais regular e homogênea, mas os computadores com os quais tivemos contato (descrição no Anexo 3) não suportaram tal confecção, devido à falta de memória (32GB de memória ram) por conta das altas exigências impostas aos parâmetros.

As Figuras 4.2, 4.3, 4.4 e 4.5 mostram, como já descrito no capítulo 4, algumas imagens da malha produzida. A Figura A1.4 mostra recorte nessa malha apenas na região próxima à barragem. Todas essas imagens foram obtidas a partir do GMSH.

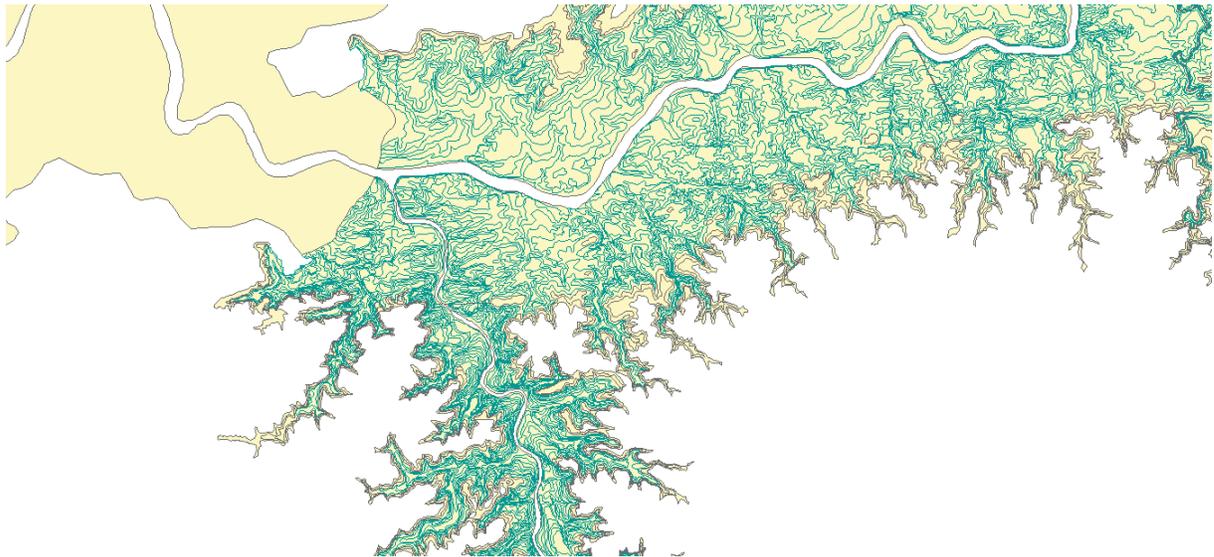


Figura A1.1: Curvas de nível da região do lago.

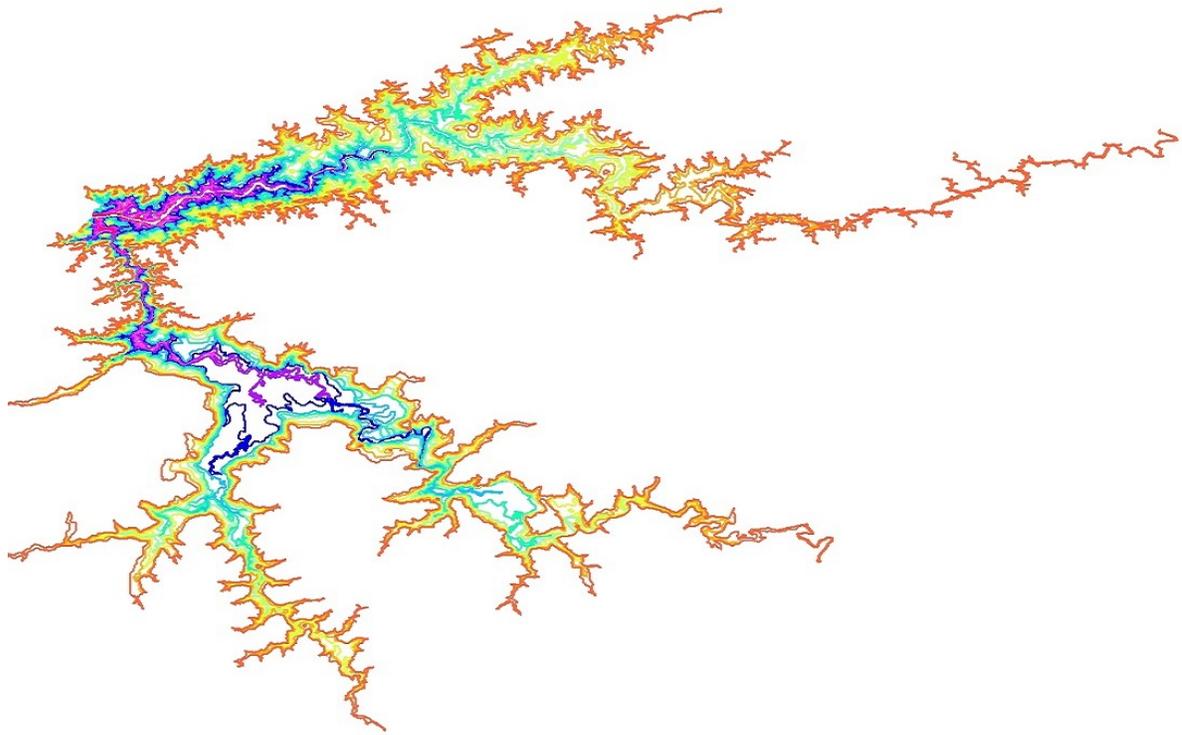


Figura A1.2: Curvas de nível do lago: variação de cores indicando profundidade.

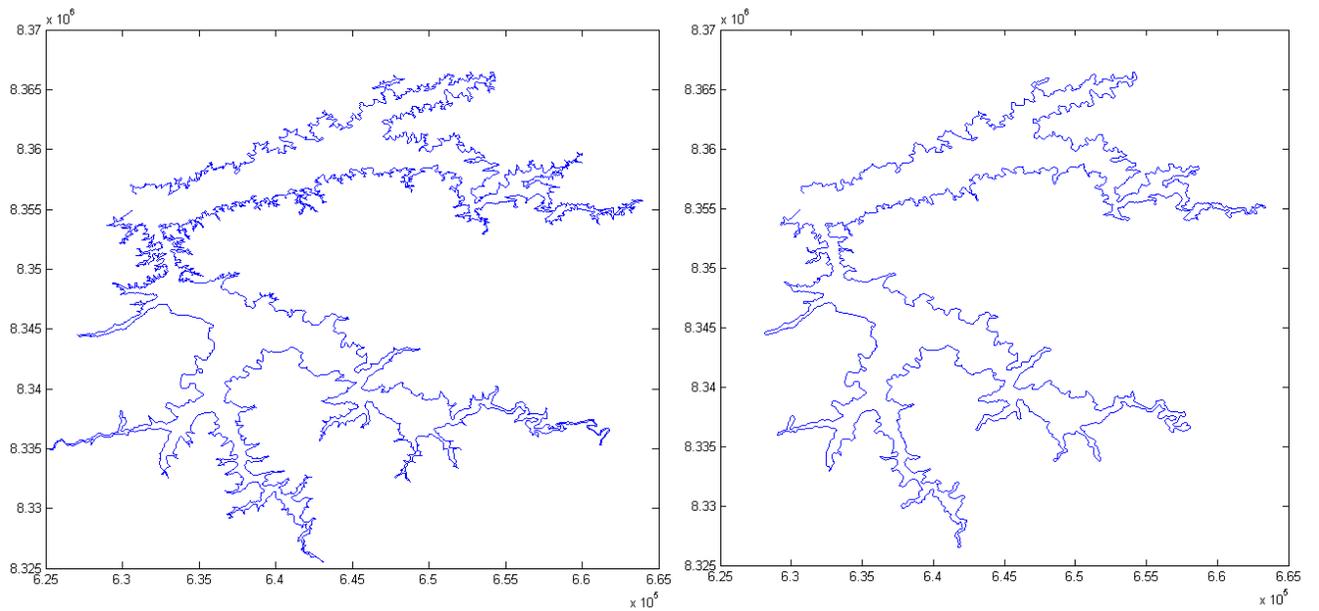


Figura A1.3: Esquerda: curva original.

Direita: depois de filtrada.

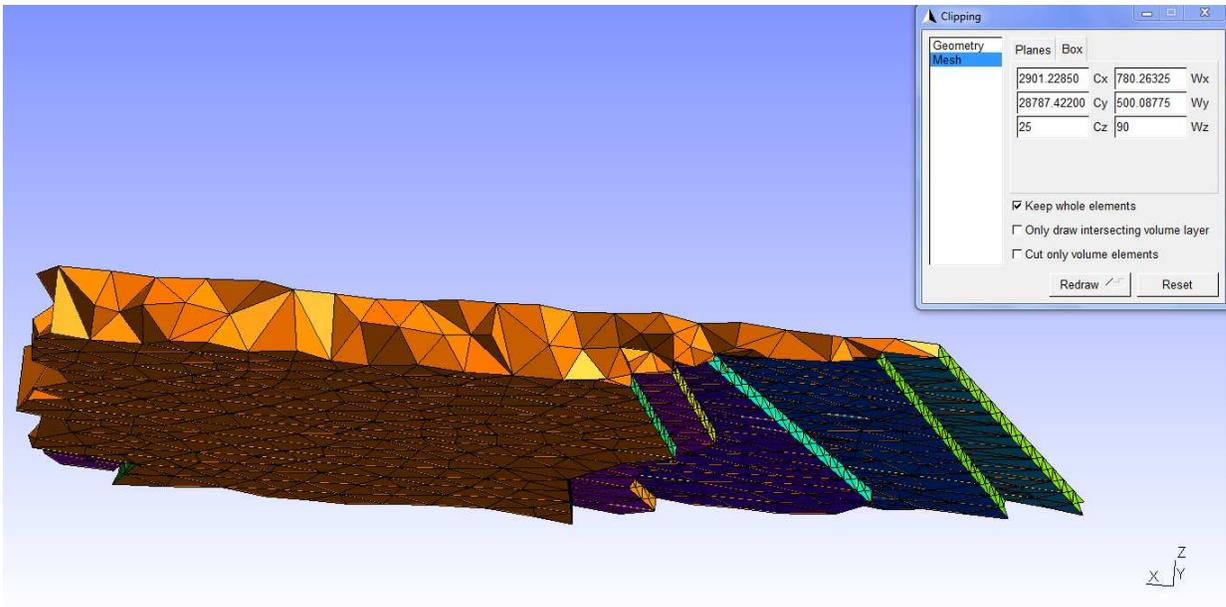


Figura A1.4: Corte na malha numa região próxima à barragem.

Anexo 2: Cálculos necessários para a implementação

Transformação e integrais sobre elementos e elementos de fronteira

Nesta seção colocaremos as manipulações algébricas que foram usadas para a técnica de elementos finitos. Iniciamos com a transformação e integrais relativas aos elementos (tetraedros) e na sequência temos a transformação e as integrais sobre os elementos de fronteira (triângulos).

φ 's sobre o elemento de referencia \hat{k} :

$$\varphi_1(\xi, \eta, \zeta) = \xi$$

$$\varphi_2(\xi, \eta, \zeta) = \eta$$

$$\varphi_3(\xi, \eta, \zeta) = 1 - \xi - \eta - \zeta$$

$$\varphi_4(\xi, \eta, \zeta) = \zeta$$

Transformação afim $T_e : \hat{k} \rightarrow k_e$:

$$(1, 0, 0) \rightarrow (x_1, y_1, z_1)$$

$$(0, 1, 0) \rightarrow (x_2, y_2, z_2)$$

$$(0, 0, 0) \rightarrow (x_3, y_3, z_3)$$

$$(0, 0, 1) \rightarrow (x_4, y_4, z_4).$$

Assim:

$$T_e(\xi, \eta, \zeta) = \begin{pmatrix} x_1 - x_3 & x_2 - x_3 & x_4 - x_3 \\ y_1 - y_3 & y_2 - y_3 & y_4 - y_3 \\ z_1 - z_3 & z_2 - z_3 & z_4 - z_3 \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} + \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix},$$

ou seja:

$$x(\xi, \eta, \zeta) = x_3 + (x_1 - x_3)\xi + (x_2 - x_3)\eta + (x_4 - x_3)\zeta$$

$$y(\xi, \eta, \zeta) = y_3 + (y_1 - y_3)\xi + (y_2 - y_3)\eta + (y_4 - y_3)\zeta$$

$$z(\xi, \eta, \zeta) = z_3 + (z_1 - z_3)\xi + (z_2 - z_3)\eta + (z_4 - z_3)\zeta$$

Jacobiana de T_e :

$$JT_e = \begin{pmatrix} x_1 - x_3 & x_2 - x_3 & x_4 - x_3 \\ y_1 - y_3 & y_2 - y_3 & y_4 - y_3 \\ z_1 - z_3 & z_2 - z_3 & z_4 - z_3 \end{pmatrix}$$

Sabemos que:

$$\varphi_i(\xi, \eta, \zeta) = \varphi_i(x(\xi, \eta, \zeta), y(\xi, \eta, \zeta), z(\xi, \eta, \zeta))$$

$$\frac{\partial \varphi_i}{\partial \xi} = \frac{\partial \varphi_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial \varphi_i}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial \varphi_i}{\partial z} \frac{\partial z}{\partial \xi}$$

$$\frac{\partial \varphi_i}{\partial \eta} = \frac{\partial \varphi_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial \varphi_i}{\partial y} \frac{\partial y}{\partial \eta} + \frac{\partial \varphi_i}{\partial z} \frac{\partial z}{\partial \eta}$$

$$\frac{\partial \varphi_i}{\partial \zeta} = \frac{\partial \varphi_i}{\partial x} \frac{\partial x}{\partial \zeta} + \frac{\partial \varphi_i}{\partial y} \frac{\partial y}{\partial \zeta} + \frac{\partial \varphi_i}{\partial z} \frac{\partial z}{\partial \zeta}$$

Substituindo as derivadas, temos:

$$\frac{\partial \varphi_i}{\partial \xi} = \frac{\partial \varphi_i}{\partial x} (x_1 - x_3) + \frac{\partial \varphi_i}{\partial y} (y_1 - y_3) + \frac{\partial \varphi_i}{\partial z} (z_1 - z_3)$$

$$\frac{\partial \varphi_i}{\partial \eta} = \frac{\partial \varphi_i}{\partial x} (x_2 - x_3) + \frac{\partial \varphi_i}{\partial y} (y_2 - y_3) + \frac{\partial \varphi_i}{\partial z} (z_2 - z_3)$$

$$\frac{\partial \varphi_i}{\partial \zeta} = \frac{\partial \varphi_i}{\partial x} (x_4 - x_3) + \frac{\partial \varphi_i}{\partial y} (y_4 - y_3) + \frac{\partial \varphi_i}{\partial z} (z_4 - z_3)$$

Podemos escrever sob a forma matricial, chegando em:

$$\begin{pmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{pmatrix} = (JT_e)^T \begin{pmatrix} \frac{\partial \varphi_i}{\partial x} \\ \frac{\partial \varphi_i}{\partial y} \\ \frac{\partial \varphi_i}{\partial z} \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{\partial \varphi_i}{\partial x} \\ \frac{\partial \varphi_i}{\partial y} \\ \frac{\partial \varphi_i}{\partial z} \end{pmatrix} = T \begin{pmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{pmatrix}$$

Sendo $T = ((JT_e)^T)^{-1}$. Usando essa relação, podemos escrever:

$$\begin{aligned} \frac{\partial \varphi_i}{\partial x} &= \begin{pmatrix} T_{11} & T_{12} & T_{13} \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{pmatrix} = T(1, :) \cdot \begin{pmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{pmatrix} \\ \frac{\partial \varphi_i}{\partial y} &= \begin{pmatrix} T_{21} & T_{22} & T_{23} \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{pmatrix} = T(2, :) \cdot \begin{pmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{pmatrix} \\ \frac{\partial \varphi_i}{\partial z} &= \begin{pmatrix} T_{31} & T_{32} & T_{33} \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{pmatrix} = T(3, :) \cdot \begin{pmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{pmatrix}, \end{aligned}$$

sendo $T(1, :)$, $T(2, :)$ e $T(3, :)$, a primeira, a segunda e a terceira linha de T , respectivamente.

O que fizemos até aqui foi escrever as derivadas das funções de base globais como uma combinação linear das funções de base locais. Agora podemos escrever as integrais sobre os elementos globais em função da integral sobre o elemento de referência. Os cálculos estão na sequência, divididos em casos:

Primeiro caso:

$$\begin{aligned} (\varphi_j, \varphi_i)_{k_e} &= \iiint_{k_e} \varphi_j \varphi_i dz dy dx = \iiint_{\hat{k}} \varphi_j \varphi_i |\det JT_e| d\zeta d\eta d\xi = |\det JT_e| \iiint_{\hat{k}} \varphi_j \varphi_i d\zeta d\eta d\xi = \\ &= |\det JT_e| (\varphi_j, \varphi_i)_{\hat{k}} \end{aligned}$$

Segundo caso:

$$\left(\frac{\partial \varphi_j}{\partial x}, \frac{\partial \varphi_i}{\partial x} \right)_{k_e} = \iiint_{k_e} \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} dz dy dx =$$

$$\begin{aligned}
\left(\frac{\partial\varphi_j}{\partial y}, \frac{\partial\varphi_i}{\partial y}\right)_k &= |\det JT_e| T(2, :) \cdot Mr \cdot T(2, :)' \\
\left(\frac{\partial\varphi_j}{\partial y}, \frac{\partial\varphi_i}{\partial z}\right)_k &= |\det JT_e| T(2, :) \cdot Mr \cdot T(3, :)' \\
\left(\frac{\partial\varphi_j}{\partial z}, \frac{\partial\varphi_i}{\partial x}\right)_k &= |\det JT_e| T(3, :) \cdot Mr \cdot T(1, :)' \\
\left(\frac{\partial\varphi_j}{\partial z}, \frac{\partial\varphi_i}{\partial y}\right)_k &= |\det JT_e| T(3, :) \cdot Mr \cdot T(2, :)' \\
\left(\frac{\partial\varphi_j}{\partial z}, \frac{\partial\varphi_i}{\partial z}\right)_k &= |\det JT_e| T(3, :) \cdot Mr \cdot T(3, :)'
\end{aligned}$$

O termo $(\nabla\varphi_j \parallel \nabla\varphi_i)_{k_e}$, usando essa notação, ficará:

$$\begin{aligned}
(\nabla\varphi_j \parallel \nabla\varphi_i)_{k_e} &= \left(\frac{\partial\varphi_j}{\partial x}, \frac{\partial\varphi_i}{\partial x}\right)_{k_e} + \left(\frac{\partial\varphi_j}{\partial y}, \frac{\partial\varphi_i}{\partial y}\right)_{k_e} + \left(\frac{\partial\varphi_j}{\partial z}, \frac{\partial\varphi_i}{\partial z}\right)_{k_e} = \\
&= |\det JT_e| \cdot [T(1, :) \cdot Mr \cdot T(1, :)' + T(2, :) \cdot Mr \cdot T(2, :)' + T(3, :) \cdot Mr \cdot T(3, :)']
\end{aligned}$$

Terceiro caso:

$$\begin{aligned}
\left(\frac{\partial\varphi_j}{\partial x}, \varphi_i\right)_{k_e} &= \iiint_{k_e} \frac{\partial\varphi_j}{\partial x} \varphi_i dz dy dx = \iiint_{\hat{k}} \left(T_{11} \frac{\partial\varphi_j}{\partial\xi} + T_{12} \frac{\partial\varphi_j}{\partial\eta} + T_{13} \frac{\partial\varphi_j}{\partial\zeta}\right) \varphi_i |\det JT_e| d\zeta d\eta d\xi = \\
&= |\det JT_e| \iiint_{\hat{k}} \left(T_{11} \frac{\partial\varphi_j}{\partial\xi} + T_{12} \frac{\partial\varphi_j}{\partial\eta} + T_{13} \frac{\partial\varphi_j}{\partial\zeta}\right) \varphi_i d\zeta d\eta d\xi = \\
&= |\det JT_e| \left[T_{11} \left(\frac{\partial\varphi_j}{\partial\xi}, \varphi_i\right)_{\hat{k}} + T_{12} \left(\frac{\partial\varphi_j}{\partial\eta}, \varphi_i\right)_{\hat{k}} + T_{13} \left(\frac{\partial\varphi_j}{\partial\zeta}, \varphi_i\right)_{\hat{k}} \right] = \\
&= |\det JT_e| \begin{pmatrix} T_{11} & T_{12} & T_{13} \end{pmatrix} \begin{pmatrix} \left(\frac{\partial\varphi_j}{\partial\xi}, \varphi_i\right)_{\hat{k}} \\ \left(\frac{\partial\varphi_j}{\partial\eta}, \varphi_i\right)_{\hat{k}} \\ \left(\frac{\partial\varphi_j}{\partial\zeta}, \varphi_i\right)_{\hat{k}} \end{pmatrix} = |\det JT_e| T(1, :) \begin{pmatrix} \left(\frac{\partial\varphi_j}{\partial\xi}, \varphi_i\right)_{\hat{k}} \\ \left(\frac{\partial\varphi_j}{\partial\eta}, \varphi_i\right)_{\hat{k}} \\ \left(\frac{\partial\varphi_j}{\partial\zeta}, \varphi_i\right)_{\hat{k}} \end{pmatrix}
\end{aligned}$$

Portanto:

$$\begin{aligned}
\left(\frac{\partial\varphi_j}{\partial x}, \varphi_i\right)_{k_e} &= |\det JT_e| T(1, :) \begin{pmatrix} \left(\frac{\partial\varphi_j}{\partial\xi}, \varphi_i\right)_{\hat{k}} \\ \left(\frac{\partial\varphi_j}{\partial\eta}, \varphi_i\right)_{\hat{k}} \\ \left(\frac{\partial\varphi_j}{\partial\zeta}, \varphi_i\right)_{\hat{k}} \end{pmatrix} \\
\left(\frac{\partial\varphi_j}{\partial y}, \varphi_i\right)_{k_e} &= |\det JT_e| T(2, :) \begin{pmatrix} \left(\frac{\partial\varphi_j}{\partial\xi}, \varphi_i\right)_{\hat{k}} \\ \left(\frac{\partial\varphi_j}{\partial\eta}, \varphi_i\right)_{\hat{k}} \\ \left(\frac{\partial\varphi_j}{\partial\zeta}, \varphi_i\right)_{\hat{k}} \end{pmatrix}
\end{aligned}$$

$$\left(\frac{\partial \varphi_j}{\partial z}, \varphi_i \right)_{k_e} = |\det JT_e| T(3, :) \begin{pmatrix} \left(\frac{\partial \varphi_j}{\partial \xi}, \varphi_i \right)_{\hat{k}} \\ \left(\frac{\partial \varphi_j}{\partial \eta}, \varphi_i \right)_{\hat{k}} \\ \left(\frac{\partial \varphi_j}{\partial \zeta}, \varphi_i \right)_{\hat{k}} \end{pmatrix}$$

Vamos agora tratar das integrais sobre os elementos de fronteira (triângulos). Usaremos a mesma notação usada nos tetraedros, observando que trata-se de dimensão diferente.

φ 's sobre o elemento de referência \hat{k} :

$$\varphi_1(\xi, \eta) = 1 - \xi - \eta$$

$$\varphi_2(\xi, \eta) = \xi$$

$$\varphi_3(\xi, \eta) = \eta$$

Transformação afim: $T_e : \hat{k} \rightarrow k_e$

$$(0, 0) \rightarrow (x_1, y_1, z_1)$$

$$(1, 0) \rightarrow (x_2, y_2, z_2)$$

$$(0, 1) \rightarrow (x_3, y_3, z_3).$$

Assim:

$$T_e(\xi, \eta) = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \\ z_2 - z_1 & z_3 - z_1 \end{pmatrix} \begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} x_1 - \xi x_1 + \xi x_2 - \eta x_1 + \eta x_3 \\ y_1 - \xi y_1 + \xi y_2 - \eta y_1 + \eta y_3 \\ z_1 - \xi z_1 + \xi z_2 - \eta z_1 + \eta z_3 \end{pmatrix}$$

$$x(\xi, \eta) = x_1 - \xi x_1 + \xi x_2 - \eta x_1 + \eta x_3$$

$$y(\xi, \eta) = y_1 - \xi y_1 + \xi y_2 - \eta y_1 + \eta y_3$$

$$z(\xi, \eta) = z_1 - \xi z_1 + \xi z_2 - \eta z_1 + \eta z_3$$

$$\frac{\partial T_e}{\partial \xi} = \frac{\partial x}{\partial \xi} \vec{i} + \frac{\partial y}{\partial \xi} \vec{j} + \frac{\partial z}{\partial \xi} \vec{k} = (x_2 - x_1) \vec{i} + (y_2 - y_1) \vec{j} + (z_2 - z_1) \vec{k} = (x_2, y_2, z_2) - (x_1, y_1, z_1)$$

$$\frac{\partial T_e}{\partial \eta} = \frac{\partial x}{\partial \eta} \vec{i} + \frac{\partial y}{\partial \eta} \vec{j} + \frac{\partial z}{\partial \eta} \vec{k} = (x_3 - x_1) \vec{i} + (y_3 - y_1) \vec{j} + (z_3 - z_1) \vec{k} = (x_3, y_3, z_3) - (x_1, y_1, z_1)$$

$$\varphi_i(\xi, \eta) = \varphi_i(x(\xi, \eta), y(\xi, \eta), z(\xi, \eta))$$

Para os triângulos, teremos dois casos:

Primeiro caso

$$\begin{aligned} (\varphi_j, \varphi_i)_{k_e} &= \iint_{k_e} \varphi_j \varphi_i dy dx = \iint_{\hat{k}} \varphi_j(\xi, \eta) \varphi_i(\xi, \eta) \left\| \frac{\partial T_e}{\partial \xi} \times \frac{\partial T_e}{\partial \eta} \right\| d\eta d\xi = \\ &= \left\| \frac{\partial T_e}{\partial \xi} \times \frac{\partial T_e}{\partial \eta} \right\| \iint_{\hat{k}} \varphi_j \varphi_i d\eta d\xi = \left\| \frac{\partial T_e}{\partial \xi} \times \frac{\partial T_e}{\partial \eta} \right\| (\varphi_j, \varphi_i)_{\hat{k}} \end{aligned}$$

Segundo caso:

$$\begin{aligned} (1, \varphi_i)_{k_e} &= \iint_{k_e} \varphi_i dy dx = \iint_{\hat{k}} \varphi_i(\xi, \eta) \left\| \frac{\partial T_e}{\partial \xi} \times \frac{\partial T_e}{\partial \eta} \right\| d\eta d\xi = \left\| \frac{\partial T_e}{\partial \xi} \times \frac{\partial T_e}{\partial \eta} \right\| \iint_{\hat{k}} \varphi_i d\eta d\xi \\ &= \left\| \frac{\partial T_e}{\partial \xi} \times \frac{\partial T_e}{\partial \eta} \right\| (1, \varphi_i)_{\hat{k}} \end{aligned}$$

Submatrizes de rigidez

Na seção passada expressamos as integrais sobre os elementos e elementos de fronteira como combinação de integrais sobre o elemento de referência (tetraedro e triângulo de referência). Nesta seção faremos os cálculos dessas integrais sobre os elementos de referência, deixando-as armazenadas em matrizes, as chamadas *Submatrizes de rigidez*. Usaremos a mesma notação, ficando a diferença explícita na dimensão dessas matrizes.

Começamos com as integrais sobre o tetraedro de referência:

$$\begin{aligned} ((\varphi_j, \varphi_i)_{\hat{k}})_{\hat{i}\hat{j}} &= \int_0^1 \int_0^{1-\xi} \int_0^{1-\eta-\xi} \begin{pmatrix} \xi \\ \eta \\ 1-\xi-\eta-\zeta \\ \zeta \end{pmatrix} \begin{pmatrix} \xi & \eta & 1-\eta-\zeta-\xi & \zeta \end{pmatrix} d\zeta d\eta d\xi = \\ &= \int_0^1 \int_0^{1-\xi} \int_0^{1-\eta-\xi} \begin{pmatrix} \xi^2 & \xi\eta & -\xi(\xi+\eta+\zeta-1) & \xi\zeta \\ \xi\eta & \eta^2 & -\eta(\xi+\eta+\zeta-1) & \eta\zeta \\ -\xi(\xi+\eta+\zeta-1) & -\eta(\xi+\eta+\zeta-1) & (\xi+\eta+\zeta-1)^2 & -\zeta(\xi+\eta+\zeta-1) \\ \xi\zeta & \eta\zeta & -\zeta(\xi+\eta+\zeta-1) & \zeta^2 \end{pmatrix} d\zeta d\eta d\xi \\ &= \begin{pmatrix} 1/60 & 1/120 & 1/120 & 1/120 \\ 1/120 & 1/60 & 1/120 & 1/120 \\ 1/120 & 1/120 & 1/60 & 1/120 \\ 1/120 & 1/120 & 1/120 & 1/60 \end{pmatrix} \end{aligned}$$

Da mesma forma, calculamos:

$$\left(\left(\frac{\partial \varphi_j}{\partial \xi}, \frac{\partial \varphi_i}{\partial \xi} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} = \begin{pmatrix} 1/6 & 0 & -1/6 & 0 \\ 0 & 0 & 0 & 0 \\ -1/6 & 0 & 1/6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \left(\left(\frac{\partial \varphi_j}{\partial \xi}, \frac{\partial \varphi_i}{\partial \eta} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/6 & 0 & -1/6 & 0 \\ -1/6 & 0 & 1/6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{aligned}
\left(\left(\frac{\partial \varphi_j}{\partial \xi}, \frac{\partial \varphi_i}{\partial \zeta} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/6 & 0 & 1/6 & 0 \\ 1/6 & 0 & -1/6 & 0 \end{pmatrix}, & \left(\left(\frac{\partial \varphi_j}{\partial \eta}, \frac{\partial \varphi_i}{\partial \xi} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \left(\left(\left(\frac{\partial \varphi_j}{\partial \xi}, \frac{\partial \varphi_i}{\partial \eta} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} \right)^T, \\
\left(\left(\frac{\partial \varphi_j}{\partial \eta}, \frac{\partial \varphi_i}{\partial \eta} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/6 & -1/6 & 0 \\ 0 & -1/6 & 1/6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & \left(\left(\frac{\partial \varphi_j}{\partial \eta}, \frac{\partial \varphi_i}{\partial \zeta} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1/6 & 1/6 & 0 \\ 0 & 1/6 & -1/6 & 0 \end{pmatrix}, \\
\left(\left(\frac{\partial \varphi_j}{\partial \zeta}, \frac{\partial \varphi_i}{\partial \xi} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \left(\left(\left(\frac{\partial \varphi_j}{\partial \xi}, \frac{\partial \varphi_i}{\partial \zeta} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} \right)^T, & \left(\left(\frac{\partial \varphi_j}{\partial \zeta}, \frac{\partial \varphi_i}{\partial \eta} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \left(\left(\left(\frac{\partial \varphi_j}{\partial \eta}, \frac{\partial \varphi_i}{\partial \zeta} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} \right)^T, \\
\left(\left(\frac{\partial \varphi_j}{\partial \zeta}, \frac{\partial \varphi_i}{\partial \zeta} \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/6 & -1/6 \\ 0 & 0 & -1/6 & 1/6 \end{pmatrix}, & \left(\left(\frac{\partial \varphi_j}{\partial \xi}, \varphi_i \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \begin{pmatrix} 1/24 & 0 & -1/24 & 0 \\ 1/24 & 0 & -1/24 & 0 \\ 1/24 & 0 & -1/24 & 0 \\ 1/24 & 0 & -1/24 & 0 \end{pmatrix}, \\
\left(\left(\frac{\partial \varphi_j}{\partial \eta}, \varphi_i \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \begin{pmatrix} 0 & 1/24 & -1/24 & 0 \\ 0 & 1/24 & -1/24 & 0 \\ 0 & 1/24 & -1/24 & 0 \\ 0 & 1/24 & -1/24 & 0 \end{pmatrix}, & \left(\left(\frac{\partial \varphi_j}{\partial \zeta}, \varphi_i \right)_{\hat{k}} \right)_{\hat{i}\hat{j}} &= \begin{pmatrix} 0 & 0 & -1/24 & 1/24 \\ 0 & 0 & -1/24 & 1/24 \\ 0 & 0 & -1/24 & 1/24 \\ 0 & 0 & -1/24 & 1/24 \end{pmatrix}.
\end{aligned}$$

Para o caso dos elementos de fronteira, temos:

$$\begin{aligned}
((\varphi_j, \varphi_i)_{\hat{k}})_{\hat{i}\hat{j}} &= \int_0^1 \int_0^{1-\xi} \begin{pmatrix} (\xi + \eta - 1)^2 & -\xi(\xi + \eta - 1) & -\eta(\xi + \eta - 1) \\ -\xi(\xi + \eta - 1) & \xi^2 & \xi\eta \\ -\eta(\xi + \eta - 1) & \xi\eta & \eta^2 \end{pmatrix} d\eta d\xi \\
&= \begin{pmatrix} 1/12 & 1/24 & 1/24 \\ 1/24 & 1/12 & 1/24 \\ 1/24 & 1/24 & 1/12 \end{pmatrix} \\
((1, \varphi_i)_{\hat{k}})_i &= \int_0^1 \int_0^{1-\xi} \begin{pmatrix} 1 - \xi - \eta \\ \xi \\ \eta \end{pmatrix} d\eta d\xi = \begin{pmatrix} 1/6 \\ 1/6 \\ 1/6 \end{pmatrix} \text{ Em todo esse desenvolvimeto algébrico}
\end{aligned}$$

e preparação para a linguagem computacional, usamos como referência [18], [16] e [2].

Anexo 3: Recursos computacionais e códigos em MatLab[®].

Este anexo está dividido em duas partes: uma primeira relatando alguns detalhes sobre a implementação e os recursos computacionais e uma segunda contendo os códigos utilizados nas simulações.

As primeiras simulações e validações de algoritmo foram feitas no CENAPAD-UNICAMP. Lá tivemos contato com máquinas antigas, mas mesmo assim muito melhores que os PCs que tínhamos até então. Eram computadores com 12GB de memória ram e processadores não informado. Essa fase durou 30 dias, tempo de acesso temporário. A impossibilidade de utilização das máquinas de grande porte foi devido à inexistência de MatLab para essa arquitetura (processadores Itanium). Depois disso, utilizamos uma workstation e um node do cluster do departamento de Física da UFMT, que tinham 16GB de ram e dois processadores Xeon quad-core, totalizando 8 núcleos físicos cada máquina. Essa fase foi de grande avanço no aperfeiçoamento dos algoritmos, nos possibilitando até o encontro com problemas de tempo computacional que nos fizeram lançar mão de técnicas de paralelização para atingirmos tempos tangíveis de processamento. Nessas máquinas, o tempo para se fazer a simulação das primeiras 4.000 iterações, foi de 336 hs (14 dias). Tivemos acesso também a um nó do cluster do Instituto de Computação da UNICAMP², composto de dois processadores AMD Opteron hexa-core e 12GB. Nesta máquina efetuamos a simulação de mais um cenário. Por fim, recebemos no Departamento de matemática da UFMT uma máquina com dois processadores

²agradecemos ao professor Dr. Rodolfo Jardim de Azevedo e seu orientando de doutorado Leonardo de Paula Rosa Piga pela ajuda e disponibilização de recursos computacionais, e também ao CNPq que custeou esse laboratório

Xeon Hexa-core, com 32GB de ram e que nos possibilitou a execução de cenários com um ganho de cerca de 35% em relação as anteriores, chegando nos 11 dias de processamento.

Os algoritmos que confeccionamos foram aprimorados para poderem ser paralelizados, fato este que possibilitou a execução pois, do contrário, seria impraticável. A necessidade de paralelização gerou uma experiência muito gratificante e importante para as experiências futuras.

Colocaremos agora os códigos computacionais que implementamos. Este primeiro programa é o principal e calcula o campo de velocidades e depois o perfil de poluição a cada passo no tempo. Este programa utiliza 3 subrotinas, que serão colocadas na sequência.

```

clear all
tic
load dominio_Manso.mat %Dados do domínio: matrizes coord, elem e elem_front
                    %coord (matriz contendo as cordenadas de cada);
                    %elem   (matriz contendo os nós de cada tetraedro;
                    %elem_front (matriz contendo os nós dos triângulos de fronteira)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Submatrizes de rigidez
fj_fi=[1/60 1/120 1/120 1/120;1/120 1/60 1/120 1/120;1/120 1/120 1/60 1/120;
        1/120 1/120 1/120 1/60];
dfjcsi_dficsi=[1/6 0 -1/6 0;0 0 0 0;-1/6 0 1/6 0;0 0 0 0];
dfjcsi_dfieta=[0 0 0 0;1/6 0 -1/6 0;-1/6 0 1/6 0;0 0 0 0];
dfjcsi_dfizeta=[0 0 0 0;0 0 0 0;-1/6 0 1/6 0;1/6 0 -1/6 0];
dfjeta_dficsi=dfjcsi_dfieta';
dfjeta_dfieta=[0 0 0 0;0 1/6 -1/6 0;0 -1/6 1/6 0;0 0 0 0];
dfjeta_dfizeta=[0 0 0 0;0 0 0 0;0 -1/6 1/6 0;0 1/6 -1/6 0];
dfjzeta_dficsi=dfjcsi_dfizeta';
dfjzeta_dfieta=dfjeta_dfizeta';
dfjzeta_dfizeta=[0 0 0 0;0 0 0 0;0 0 1/6 -1/6;0 0 -1/6 1/6];
dfjcsi_fi=[1/24 0 -1/24 0;1/24 0 -1/24 0;1/24 0 -1/24 0;1/24 0 -1/24 0];
dfjeta_fi=[0 1/24 -1/24 0;0 1/24 -1/24 0;0 1/24 -1/24 0;0 1/24 -1/24 0];
dfjzeta_fi=[0 0 -1/24 1/24;0 0 -1/24 1/24;0 0 -1/24 1/24;0 0 -1/24 1/24];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS (Cenário 2) %%%%%%%%%
n_ite=input('numero de iteracoes=');
dt=0.05;
Re=100;
alf=0.01;
sig=0.00001;
k1=1.0e-6; % mata
k2=1.0e-6; % superficie
k3=1.0e-6; %fundo do lago
k4=1.0e-3; %vertedouro
tet5=5.0e-5; %runoff

```

```

tet6=1.0e-5; %rio palmeira
tet7=1.0e-5; %rio manso
tet8=1.0e-5; %rio casca
tet9=1.0e-5; % rio quilombo
vel_rio=1; % km/h
nte=max(size(elem)); %numero total de elementos, tetraedros.
ntn=max(size(coord)); % numero total de nos
nf=442613;% numero de nos internos ao domínio
ni=ntn-nf;% numero de nos na fronteira do domínio
poluicao=zeros(ntn,1);
velocidade=zeros(ntn,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% geracao dos vetores incognitas
load superficie.mat % contém o conjunto de nós que estão na superfície
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% incognitas e cognitias da pressão
cog_pre=[15769 16154 16155 16540 48249 48250 48251 48252 48253 48254 ...
         49075 49076 49077 49078 49079 49080 87073 87074 87075 87076 ...
         87077 87078 87079 87080 87081 87082 87083 87084 87085 87086]';
incog_pre=[1:ntn]';
incog_pre(cog_pre)=[];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% incognitas e cognitias da v1
incog_v1=[nf+1:ntn cog_pre' superficie'];
incog_v1=sort(incog_v1);
incog_v1=incog_v1';
cog_v1=[1:ntn]';
cog_v1(incog_v1)=[];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% incognitas e cognitias da v2
incog_v2=incog_v1;
cog_v2=cog_v1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% incognitas e cognitias da v3
incog_v3=[cog_pre' nf+1:ntn];
incog_v3=sort(incog_v3);
incog_v3=incog_v3';
cog_v3=[1:ntn]';
cog_v3(incog_v3)=[];
clear superficie
%integrals de fronteira para difusão-advvecção
[F_front_1 F_front_2]=int_front_dif(coord,elem_front,bandeira,k1,k2,k3,k4,tet5,tet6,tet7,tet8,tet9);
%int_front_dif é uma subrorina para cálculo da integrais de fronteira.
load matrizes.mat %arquivo contendo os produtos internos já calculados
load front.mat %arquivo contendo condições de fronteira para Navier-Stokes
front=vel_rio*front;
sol_poll=zeros(ntn,1);
vel1=zeros(ntn,3);
f=0;
for t=1:n_ite

```

```

    Fis_ns=zeros(nte,4,4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% perfil de velocidade no tempo n atraves de Navier Stokes
    parfor k=1:nte % looping paralelizado com a possibilidade de escolha do número de processadores a usar.
        Fis_ns(k, :, :)=lupe_vel(coord,vel1,elem,dfjcsi_fi,dfjeta_fi,dfjzeta_fi,k); % subrotina
    end
    Fis_ns=Fis_ns(:);
    FE=sparse(III,JJJ,Fis_ns,ntn,ntn);
    clear Fis_ns
b1_est=(1/dt)*FA(incog_v1, :)*vel1(:,1)-((1/dt)*FA(incog_v1,cog_v1)+(1/Re)*FB(incog_v1,cog_v1)+ ...
FE(incog_v1,cog_v1))*vel1(cog_v1,1);
b2_est=(1/dt)*FA(incog_v2, :)*vel1(:,2)-((1/dt)*FA(incog_v2,cog_v2)+(1/Re)*FB(incog_v2,cog_v2)+ ...
FE(incog_v2,cog_v2))*vel1(cog_v2,2);
b3_est=(1/dt)*FA(incog_v3, :)*vel1(:,3)-((1/dt)*FA(incog_v3,cog_v3)+(1/Re)*FB(incog_v3,cog_v3)+ ...
FE(incog_v3,cog_v3))*vel1(cog_v3,3)-FA(incog_v3, :)*(f*ones(ntn,1));
AA1=(1/dt)*FA(incog_v1,incog_v1)+(1/Re)*FB(incog_v1,incog_v1)+FE(incog_v1,incog_v1);
AA2=(1/dt)*FA(incog_v2,incog_v2)+(1/Re)*FB(incog_v2,incog_v2)+FE(incog_v2,incog_v2);
AA3=(1/dt)*FA(incog_v3,incog_v3)+(1/Re)*FB(incog_v3,incog_v3)+FE(incog_v3,incog_v3);
vel_est1=AA1\b1_est;
vel_est2=AA2\b2_est;
vel_est3=AA3\b3_est;
vel_est=zeros(ntn,3);
vel_est(1:nf,1:3)=front;
vel_est(incog_v1,1)=vel_est1;
vel_est(incog_v2,2)=vel_est2;
vel_est(incog_v3,3)=vel_est3; %Cálculo da velocidade*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b_pre=-((1/dt)*FC(incog_pre, :)*vel_est(:,1)-(1/dt)*FD(incog_pre, :)*vel_est(:,2)- ...
(1/dt)*FF(incog_pre, :)*vel_est(:,3);
x_pre=FB(incog_pre,incog_pre)\b_pre;
pre=zeros(ntn,1);
pre(incog_pre)=x_pre; % cálculo da pressão
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b1=FA(incog_v1, :)*vel_est(:,1)-dt*FC(incog_v1, :)*pre;
b2=FA(incog_v2, :)*vel_est(:,2)-dt*FD(incog_v2, :)*pre;
b3=FA(incog_v3, :)*vel_est(:,3)-dt*FF(incog_v3, :)*pre;
vel_1=FA(incog_v1,incog_v1)\b1;
vel_2=FA(incog_v2,incog_v2)\b2;
vel_3=FA(incog_v3,incog_v3)\b3;
vel2=zeros(ntn,3);
vel2(1:nf,1:3)=front;
vel2(incog_v1,1)=vel_1;
vel2(incog_v2,2)=vel_2;
vel2(incog_v3,3)=vel_3;%cálculo da velocidade real.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% perfil de poluicao no tempo n atraves de difusão-Adveccão
    Fis_adv=zeros(nte,4,4);

```

```

    parfor k=1:nnte % looping paralelizado com a possibilidade de escolha do número de processadores a usar.
        Fis_adv(k, :, :) = lupe_pol(coord, vel1, vel2, elem, dfjcsi_fi, dfjeta_fi, dfjzeta_fi, k); %subrotina
    end
Fis_adv = Fis_adv(:);
W1 = sparse(III, JJJ, Fis_adv, ntn, ntn);
clear Fis_adv
bb = (FA - (dt/2) * (alf * FB + sig * FA + F_front_1) - (dt/2) * W1) * sol_pol1 + dt * F_front_2; % + dt * FA * f
AA = (FA + (dt/2) * (alf * FB + sig * FA + F_front_1) + (dt/2) * W1);
sol_pol2 = AA \ bb;
clear AA bb
sol_pol2 = full(sol_pol2);
sol_pol1 = sol_pol2;
vel1 = vel2;
if mod(t, 50) == 0 || t == 10
    t
    poluicao = [poluicao sol_pol2];
    velocidade = [velocidade vel2];
end
end
save iteracoes_C2.mat poluicao velocidade
toc

```

Subrotina 1: **int_front_dif**

Esta subrotina monta a matriz que contém as integrais de fronteira no cálculo do perfil de poluição no tempo atual.

```

function [F_front_1 F_front_2] = int_front_dif(coord, elem_front, bandeira, k1, k2, k3, k4, tet5, tet6, tet7, tet8, tet9)
% k1= mata
% k2= superficie
% k3=fundo do lago
% k4=vertedouro
% tet5=runoff
% tet6=rio palmeira
% tet7=rio manso
% tet8=rio casca
% tet9=rio quilombo
%
% informacao do vetor "bandeira"
% 1=mata
% 2=superficie
% 3=rocha ou concreto
% 4=plantacao, chácaras (runoff)
% 5=rio palmeira
% 6=rio manso
% 7=rio da casca

```

```

% 8=rio quilombo
% 9=vertedouro
% 10=fundo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Submatrizes de rigidez
fj_fi=[1/12 1/24 1/24;1/24 1/12 1/24;1/24 1/24 1/12];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ntef=max(size(elem_front)); %numero total de elementos de fronteira, triangulos.
ntn=max(size(coord)); % numero total de nos
nf=442613;
contador=1;
Fis_front=zeros(9*ntef,3);
F_front_2=zeros(ntn,1);
for k=1:ntef
    X1=coord(elem_front(k,1+1),:);
    X2=coord(elem_front(k,2+1),:);
    X3=coord(elem_front(k,3+1),:);
    jac=cross(X2-X1,X3-X1);
    jac=norm(jac);
    if bandeira(k)==1
        coef1=k1;
    elseif bandeira(k)==2
        coef1=k2;
    elseif bandeira(k)==9
        coef1=k4;
    elseif bandeira(k)==10
        coef1=k3;
    else
        coef1=0;
    end
    if bandeira(k)==4
        coef2=tet5;
    elseif bandeira(k)==5
        coef2=tet6;
    elseif bandeira(k)==6
        coef2=tet7;
    elseif bandeira(k)==7
        coef2=tet8;
    elseif bandeira(k)==8
        coef2=tet9;
    else
        coef2=0;
    end
    for il=1:3
        ig=elem_front(k,il+1);
        F_front_2(ig)=F_front_2(ig)+coef2*jac*(1/6);
    end
end

```

```

        for jl=1:3
            jg=elem_front(k,jl+1);
            Fis_front(contador,:)=ig jg coef1*jac*fj_fi(il,jl)];
            contador=contador+1;
        end
    end
end
F_front_1=sparse(Fis_front(:,1),Fis_front(:,2),Fis_front(:,3),ntn,ntn);
end

```

Subrotina 2: **lupe_vel**

Esta subrotina monta a matriz que depende da velocidade no passo anterior e por isso, precisa ser atualizada em cada passo. Essa matriz é usada no cálculo da velocidade para o próximo passo.

```

function [NNN] = lupe_vel(coord,vel1,elem,dfjcsi_fi,dfjeta_fi,dfjzeta_fi,k)
    TT=[coord(elem(k,1),:)-coord(elem(k,3),:);
        coord(elem(k,2),:)-coord(elem(k,3),:);
        coord(elem(k,4),:)-coord(elem(k,3),:)];
    T=inv(TT);
    detJ=det(TT);
    vel_media_k1=(1/4)*(vel1(elem(k,1),:)+vel1(elem(k,2),:)+vel1(elem(k,3),:)+vel1(elem(k,4),:));
    NNN=zeros(4,4);
    for il=1:4
        for jl=1:4
            dd=[dfjcsi_fi(il,jl);dfjeta_fi(il,jl);dfjzeta_fi(il,jl)];
            NNN(il,jl)=(abs(detJ))*vel_media_k1*[T(1,:)*dd;T(2,:)*dd;T(3,:)*dd];
        end
    end
end
end

```

Subrotina 3: **lupe_pol**

Esta subrotina monta a matriz que depende da velocidade no passo atual e por isso, precisa ser atualizada em cada passo. Essa matriz é usada no cálculo do perfil de poluição no passo atual.

```

function [AAA] = lupe_pol(coord,vel1,vel2,elem,dfjcsi_fi,dfjeta_fi,dfjzeta_fi,k)
    TT=[coord(elem(k,1),:)-coord(elem(k,3),:);
        coord(elem(k,2),:)-coord(elem(k,3),:);
        coord(elem(k,4),:)-coord(elem(k,3),:)];
    T=inv(TT);

```

```

detJ=det(TT);
vel_media_k1=(1/4)*(vel1(elem(k,1),:)+vel1(elem(k,2),:)+vel1(elem(k,3),:)+vel1(elem(k,4),:));
vel_media_k2=(1/4)*(vel2(elem(k,1),:)+vel2(elem(k,2),:)+vel2(elem(k,3),:)+vel2(elem(k,4),:));
AAA=zeros(4,4);
for il=1:4
    for jl=1:4
        dd=[dfjcsi_fi(il,jl);dfjeta_fi(il,jl);dfjzeta_fi(il,jl)];
        AAA(il,jl)=0.5*(abs(detJ))*(vel_media_k1+vel_media_k2)*[T(1,:)*dd;T(2,:)*dd;T(3,:)*dd];
    end
end
end
end

```

Referências Bibliográficas

- [1] G. L. Diniz, **Dispersão de Poluentes num Sistema Ar-Água: Modelagem, Aproximação e Aplicações**, (Tese de Doutorado), Imecc-Unicamp. Campinas (2003).
- [2] N. F. Inforzato, **Dispersão de Poluentes num Sistema ar-água: modelagem matemática, aproximação numérica e simulação computacional**, (Tese de Doutorado), Imecc-Unicamp. Campinas (2008).
- [3] A. Okubo, **Diffusion and Ecological Problems: Mathematical Models**. Springer-Verlag, New York (1980).
- [4] L. Edelstein-Keshet, **Mathematical Models in Biology**. Random-House, New York (1988).
- [5] G. I. Marchuk, **Mathematical Models in Environmental Problems**. Studies in Mathematical and its Applications, 16(1986).
- [6] A. Chorin, **A numerical method for solving incompressible viscous problems**, J. Comp. Physics, Vol. 2, pp 12-16, (1967).
- [7] R. Temam, **Navier-Stokes equations**, North-Holland, Amsterdam, (1977).
- [8] D. Goldberg; V. Ruas, **A numerical study of projection algorithms in the finite element simulation of three-dimensional viscous incompressible flow**. International journal for numerical methods in fluids, 30:233-256 (1999).
- [9] http://www.furnas.com.br/hotsites/sistemapurnas/usina_hidr_manso.asp, acesso em 09/05/2011.

- [10] J. L. Lions, **Equations Differentielles Operationelles et Problèmes aux Limites**. Springer-Verlag, (1961).
- [11] L. Wolmuth, **Modelagem e simulações do comportamento evolutivo de poluentes em corpos aquáticos de grande extensão: o caso da represa do rio Manso**, (Dissertação de mestrado), Imecc-Unicamp. Campinas (2009).
- [12] J. C. S. Vásquez, **Comportamento Evolutivo de Descarga de Água de Produção Decorrente de Atividade Offshore: Tratamento Numérico e Simulação Computacional**, (Tese de Doutorado), Imecc-Unicamp. Campinas (2005).
- [13] M. A. da Silva, P. Zeilhofer, E. S. dos Santos, A. L. M. Ribeiro, R. D. Miyazaki, M. A. dos Santos, **Mapeamento de habitats de *Haemagogus janthinomys* e *Haemagogus leucocelaenus* na área de influência da APM-Manso/MT com técnicas de geoprocessamento**, XVIII Congresso Mundial de Epidemiologia e VII Congresso Brasileiro de Epidemiologia em 2008.
- [14] M. Bernardes, **Poluição de Corpos Aquáticos de Baixa Circulação**, (Dissertação de mestrado), Imecc-Unicamp. Campinas (1998).
- [15] R. F. de Oliveira, **O Comportamento Evolutivo de uma Mancha de Óleo na Baía de Ilha Grande, RJ: Modelagem, Análise Numérica e Simulações**. (Tese de Doutorado), Imecc-Unicamp. Campinas (2003).
- [16] R. F. Cantão, **Modelagem e Simulação Numérica de Derrames de Óleo no Canal de São Sebastião, SP**, (Dissertação de mestrado), Imecc-Unicamp. Campinas (1998).
- [17] D. C. Mistro, **O Problema da Poluição em Rios por Mercúrio Metálico: Modelagem e Simulação Numérica**, (Dissertação de mestrado), Imecc-Unicamp. Campinas (1992).
- [18] H. Kardestuncer; D. H. Norrie, Finite Element Handbook. Can. Journal Aquat. Science, 53(1996), 2554-2566.

- [19] P. R. Trales, **Resolução das equações de Navier-Stokes pelo método das projeções via Elementos Finitos com aceleração multigrid.** (Tese de Doutorado), Departamento de Informática - Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro (1995).