A detailed map of the São Sebastião Canal in São Paulo, Brazil. The map shows the canal's path through a narrow strait, with various islands and peninsulas on either side. Numerous small labels identify specific locations and landmarks. The map is rendered in a high-contrast, black-and-white style, typical of a photocopy or a technical drawing.

# Modelagem e Simulação Numérica de Derrames de Óleo no Canal de São Sebastião, SP.

RENATO FERNANDES CANTÃO

IMECC - UNICAMP

1998

Apoio: CNPq - #139116/96



UNIDADE	BC
N.º CHAMADA:	
V. E.	
VALOR BC/	354,21
PROD.	395,98
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PRECIO	2811,00
DATA	09/10/98
N.º CPD	

CM-00117014-5

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

Cantão, Renato Fernandes

C166m Modelagem e simulação numérica de derrames de óleo no Canal de São Sebastião, SP / Renato Fernandes Cantão -- Campinas, [S.P. :s.n.], 1998.

Orientador : João Frederico da Costa Azevedo Meyer

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

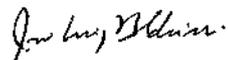
1. Poluição marinha por óleo. 2. Petróleo. 3. Óleo - Indústria. I. Meyer, João Frederico da Costa Azevedo. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

**Dissertação de Mestrado defendida e aprovada em 10 de agosto de 1998  
pela Banca Examinadora composta pelos Profs. Drs.**



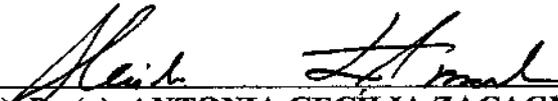
---

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



---

**Prof (a). Dr (a). JOSÉ LUIZ BOLDRINI**



---

**Prof (a). Dr (a). ANTONIA CECÍLIA ZACAGNINI AMARAL**

## Agradecimentos

Uma dissertação de Mestrado é usualmente um tipo de trabalho com uma “narrativa” bastante séria, até mesmo pesada. Portanto, reservo-me o direito de usar a seção de agradecimentos para tecer os comentários que julgo relevantes de forma leve e descontraída.

Um número absurdo de pessoas contribuiu das mais variadas formas pra execução deste trabalho (algumas até nem sabem disso!). Pra começar, o Joni, meu orientador. Aliás, mais do que isso. Considero o Joni meu amigo (uma palavra de uso muito restrito, diga-se de passagem). Trabalhar com ele é sempre bom pois há uma união perfeita da cobrança do chefe, da ajuda do amigo e da paciência de um excelente educador. Ao Joni meu muito obrigado (formiga, pardal e puxa-saco ...)!

Depois vem minha família, é claro. Ao meu pai - o “seu” Cantão - (*in memoriam*), que sempre quis que eu fosse aviador, mas que no fundo sabia que eu já tinha encontrado minhas asas. Minha mãezinha “dona” Thereza, que sempre que vê meus trabalhos, pergunta se estou aprendendo egípcio, mas mesmo assim acha tudo lindo. Minhas irmãs Lia e Vera, que são mais do que irmãs. O apoio delas é - e sempre será - muito bem vindo (embora elas não desconfiem muito disso!). Meus sobrinhos Daniel e Lucas que, no mínimo, me ensinaram o significado da palavra *backup*. A Luiza, minha companheira de coração e profissão, pelo amor e carinho, pelas dicas, pela paciência enorme todas as vezes que eu dizia “tenho de trabalhar” e pela sabedoria que me fez entender de novo algumas coisas sobre a vida que frequentemente esquecemos.

Aos amigos, Akiles (parte gráfica, milhões de palpites e computadores), Biloti & Débora (um computador e alguns milhares de palpites apenas), o Gal (nenhum palpite, mas ainda assim é um bom amigo), Benevides e Ítalo (pela suprema paciência de distribuir as velocidades no Canal). Aos velhos amigos que deram sua contribuição (ou amizade!) de algum modo: Marchetti, Akira e Marcus. Ao pessoal das Secretarias do IMECC, de Matemática Aplicada e de Pós-Graduação (valeu galera!). Aos meus alunos do curso de Estatística, que sempre me mostram como temos coisas a aprender.

Ao Senhor. Se Ele existe mesmo, então fez um ótimo trabalho escondendo tantos segredos por tanto tempo na nossa frente.

Finalmente, ao CNPq pelo apoio financeiro à realização deste trabalho.

*“Os provérbios são sempre  
chavões, até você experimentar  
a verdade contida neles”.*

*Aldous Huxley  
Escritor inglês*

*“Todas as coisas já foram ditas,  
mas como ninguém escuta  
é preciso sempre recomeçar”.*

*André Gide  
Escritor francês*

# Sumário

<b>1. INTRODUÇÃO</b>	<b>5</b>
<b>2. O MODELO MATEMÁTICO</b>	<b>8</b>
2.1 OS FENÔMENOS EM QUESTÃO	8
2.2 A EQUAÇÃO	9
2.3 O MODELO COMPLETO	11
2.4 SIMPLIFICAÇÕES QUE TORNAM O MODELO TRATÁVEL	12
<b>3. FORMULAÇÃO VARIACIONAL - EXISTÊNCIA E UNICIDADE</b>	<b>14</b>
3.1 A FORMULAÇÃO VARIACIONAL	14
3.2 EXISTÊNCIA E UNICIDADE DE SOLUÇÕES	15
<b>4. CONSTRUINDO A SOLUÇÃO</b>	<b>16</b>
4.1 GALERKIN E ELEMENTOS FINITOS	16
4.2 DISCRETIZANDO O DOMÍNIO ESPACIAL	20
<b>5. RESULTADOS, ENFIM!</b>	<b>20</b>
5.1 AS SIMULAÇÕES SOBRE O CANAL	20
5.1.1 CENÁRIO 1 - ROMPIMENTO DE OLEODUTO	22
5.1.2 CENÁRIO 2 - RESSUSPENSÃO DE ÓLEO I	24
5.1.3 CENÁRIO 3 - RESSUSPENSÃO DE ÓLEO II	26
5.1.4 CENÁRIO 4 - DERRAME NO TERMINAL DTCS I	28
5.1.5 CENÁRIO 5 - DERRAME NO TERMINAL DTCS II	30
5.1.6 CENÁRIO 6 - O CENÁRIO 2 REVISITADO	31
<b>6. CONCLUSÕES</b>	<b>33</b>
<b>APÊNDICE A - EXISTÊNCIA E UNICIDADE</b>	<b>35</b>
<b>APÊNDICE B - OSCILAÇÕES NUMÉRICAS E SUPG</b>	<b>36</b>
<b>APÊNDICE C - ASPECTOS COMPUTACIONAIS</b>	<b>38</b>
<b>REFERÊNCIAS</b>	<b>52</b>

## 1. Introdução

No litoral norte do estado de São Paulo está localizado o Canal de São Sebastião, composto pela faixa litorânea que vai da Ponta do Arpoar (norte) à praia de Toque-Toque Pequeno (sul) em sua porção continental. A outra parte do Canal consiste na Ilha de São Sebastião, comumente chamada Ilhabela.

Lá foi inaugurado em 1969 (PETROBRÁS, 1992 - deste trabalho também foram extraídas as fotos constantes desta seção), o outrora chamado TEBAR - Terminal Almirante Barroso, atualmente DTCS - Dutos e Terminais do Centro e Sul. O DTCS foi construído e instalado pela Petrobrás - Petróleo Brasileiro S. A. - para suprir as deficiências do porto de Santos, usado até então para carga e descarga de petróleo e derivados. São Sebastião foi a escolha lógica, por possuir águas relativamente calmas e permitir o atracamento de navios de grande calado (até 23 metros).

O DTCS cresceu em importância e atividade, à medida que novas refinarias e terminais foram sendo construídos. Atualmente ele está ligado por oleodutos a refinarias em Paulínia (SP), Capuava (SP), Cubatão (SP) e no Vale do Paraíba e será futuramente ligado a Brasília, sendo responsável pelo transporte e armazenamento de aproximadamente 55% do petróleo brasileiro, aqui produzido ou não.

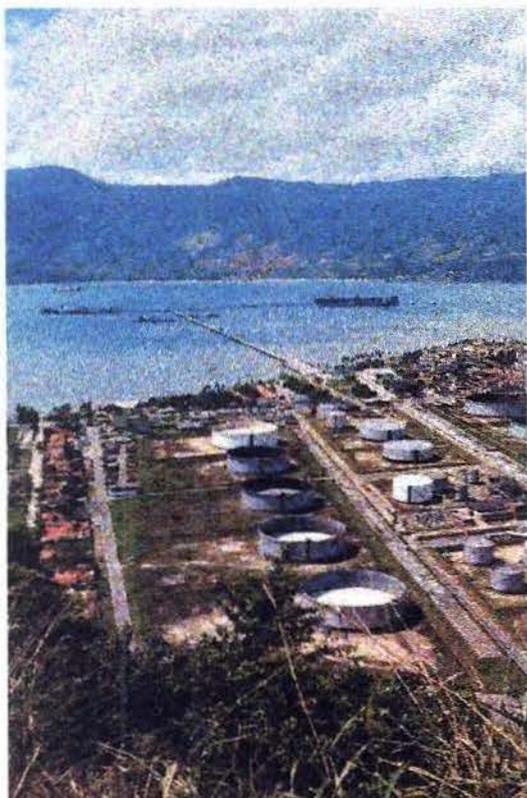


Figura 1 - DTCS em São Sebastião.

Como resultado desse volume de movimentação, foram registrados e catalogados pela CETESB - Companhia de Tecnologia de Saneamento Ambiental - cerca de 350 acidentes envolvendo petróleo nos últimos 24 anos, a maioria deles mal catalogada e normalmente detectada quando já era tarde demais (POFFO *et alii*, 1997). Esses acidentes abrangem uma vasta gama de situações, indo de pequenas manchas ditas “inofensivas” e sem origem aparente, até grandes vazamentos em oleodutos, com manchas de até 45 km<sup>2</sup>.

Esse tipo de ocorrência traz prejuízos imensos para a região como um todo. Dentre eles:

- O Canal é uma área de beleza natural incontestável, sendo o turismo uma das indústrias fortes na região. Derrames que porventura atinjam praias nas imediações de centros hoteleiros certamente desestimularão o turismo.
- Os poucos peixes que sobrevivem aos hidrocarbonetos tornam-se tóxicos e impróprios para o consumo, paralisando as atividades das comunidades caiçaras que sobrevivem da pesca.
- Mas é do ponto de vista ecológico que o fato assume suas devidas proporções

(na verdade, o “ponto de vista ecológico” engloba os itens anteriores, já que também somos parte do meio, fato que não raro esquecemos). O Canal é um patrimônio ecossistêmico imenso, tanto na água, na forma de microorganismos e animais de maior porte, como na areia das praias e rochas dos costões. Nessas comunidades a presença do óleo, e muitas vezes as tentativas de seu recolhimento e limpeza, são devastadoras. Para muitas espécies, o meio ambiente torna-se hostil, parte das populações é eliminada e parte dos sobreviventes ficam impedidos de se reproduzirem.

Infelizmente, apesar de tantos acidentes (e da absoluta certeza de que outros ocorreriam), não havia sido desenvolvida até recentemente uma política única de proteção e preservação ambiental, dado o grande número de agências e interesses envolvidos (o setor hoteleiro por exemplo, exige ações rápidas como o jateamento de rochas com areia, que frequentemente se traduzem em desastres maiores do que o derramamento em si).



Figura 2 - Atividades de limpeza.

Porém, desde 1994 um esforço conjunto na forma de um convênio vem sendo feito com a intenção de se criar um manual para emergências onde todos os procedimentos necessários no caso de derrames sejam organizados e padronizados. Esse documento pretende cobrir todas as atividades envolvidas com um derrame, desde a prevenção antes do acontecimento até a posterior tributação dos responsáveis (se identificados). Os órgãos envolvidos nesse convênio foram a Universidade Estadual de Campinas (UNICAMP) - através do IMECC e do IB - a Companhia Nacional de Petróleo (Petrobrás), a Companhia de Tecnologia de Saneamento Ambiental (CETESB) e a Universidade de São Paulo (USP) - com a participação do IO-USP.

Uma parte deste projeto se constituiu na transformação de modelos matemáticos convenientes em esquemas numéricos para predição qualitativa do movimento de manchas de óleo na região. A idéia básica é, dado um cenário - ou seja, a condição de derrame (aparecimento de mancha ou vazamento localizado), mais as condições de correntes no local - que uma simulação computacional descreva aproximadamente o comportamento evolutivo da mancha. Em suma, desejamos dispor de uma ferramenta para uso no local da ocorrência que ajude no direcionamento das ações emergenciais, otimizando assim o uso dos recursos disponíveis.

Dentre os modelos disponíveis, optamos por trabalhar com um sistema determinístico, no qual uma Equação Diferencial de Difusão-Advecção foi escolhida, dada a

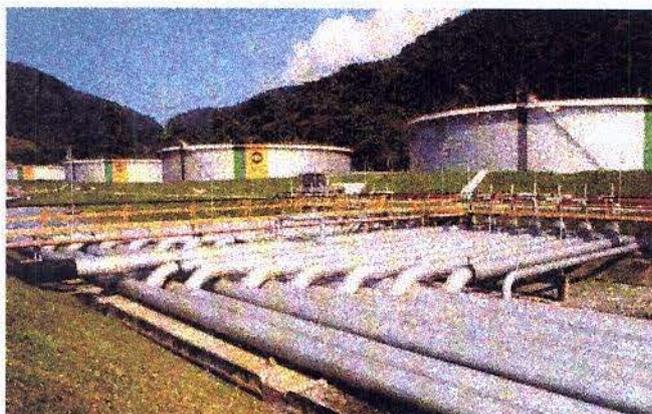


Figura 3 - Tanques de armazenamento.

fase em que o derrame seja estudado. Esta fase é chamada de fase difusiva-advectiva (FAY, 1969). Condições de fronteira convenientes para mar aberto e costa também são levadas em consideração. Esse modelo será numérico e computacionalmente trabalhado, de forma a fornecer informações úteis no rastreamento de manchas. Quando da redação deste trabalho a Petrobrás dispunha de um programa escrito em linguagem BASIC, denominado SIMOL, que considera a mancha

circular, com o seu centro movendo-se baseado em uma heurística acerca da hidrodinâmica do Canal, tratando-se portanto de um caso unidimensional de difusão radial. Neste trabalho iremos considerar a concentração de poluente em cada ponto, tratando a difusão e a advecção superficiais bidimensionalmente.

Nossa súite de simulação é denominada **Netuno** e compreende uma série de programas que vão desde um núcleo numérico até módulos gráficos para visualização de resultados e entrada de dados. Os programas foram todos escritos em C++ (compilador da *Borland*, versão 4.5) de maneira bastante genérica, tornando sua adaptação para outros locais relativamente exequível, bastando para tanto o fornecimento de mapas discretizados mais os perfis hidrodinâmicos da região que se deseja estudar. A aproximação espacial das soluções é feita pelo Método dos Elementos Finitos, com elementos triangulares de primeira e segunda ordem, com o posterior uso de SUPG (*Streamlined Upwind Petrov-Galerkin*). Na discretização temporal usa-se o clássico método de Crank-Nicolson. **Netuno** possui também um módulo de matrizes esparsas com alocação dinâmica de memória, o que permite a execução de simulações com dimensões dependentes apenas da memória - e da velocidade - do computador usado.

Na seção 2 introduziremos o modelo matemático genérico para o problema da fase de difusão-advecção e faremos algumas simplificações necessárias neste ponto. Na seção 3 apresenta-se a formulação variacional e um teorema de existência e unicidade de soluções fracas. Na seção 4 abordamos a discretização do modelo (Galerkin). Os resultados são apresentados e avaliados na seção 5. Temos ainda três apêndices, compreendendo a prova de existência e unicidade de soluções fracas, a adaptação da técnica de SUPG para o presente caso e alguns aspectos computacionais da implementação do programa. Finalmente apresentamos a conclusão e as referências usadas nesse trabalho.

## 2. O Modelo Matemático

### 2.1 Os Fenômenos em Questão

O primeiro passo de nosso trabalho é a obtenção de uma equação que descreva o comportamento do petróleo na água. O segundo passo é observar que a frase anterior está errada, ou é no mínimo pretensiosa. Isso porque uma equação nunca descreve inteiramente um fenômeno (daí o nome “modelo”), especialmente numa situação complexa como esta. Então, restringir-nos-emos a um conjunto de fenômenos básicos, mas que, esperamos, forneçam resultados qualitativamente razoáveis.

De acordo com o trabalho de FAY (1969), existem três fases principais no comportamento de uma mancha de óleo sobre a água:

**GRAVITACIONAL-INERCIAL** dura entre uma hora e duas. É a fase em que o óleo ainda é bastante denso e não está misturado com a água, assemelhando-se a um grande bloco de matéria sobre a água. É aí que o produto começará a se espalhar e a tomar o aspecto de um “tapete” sobre a superfície;

**DIFUSIVA-ADVECCIVA** é a fase de nosso interesse, durando até duas semanas. Aqui o óleo ainda tem densidade suficiente para manter-se razoavelmente coeso, mas continua se misturando com água e se degradando. É a fase usualmente considerada em derrames de óleo costeiros;

**TENSÃO SUPERFICIAL** tendo já perdido totalmente sua coesão molecular, o óleo agora só pode ser detectado através de meios químicos, já não sendo mais caracterizado como mancha, mas como mistura água-óleo.

Consultas a técnicos da CETESB, da Petrobrás e a registros dos derramamentos de óleo (*cf* POFFO *et alii*, 1997) nos convenceram da validade de um modelo bidimensional nas variáveis espaciais; isto porque na segunda fase do óleo a mancha tem em média 5 cm de espessura (TAVARES JR. – comunicação verbal, 1995), contra pelo menos alguns metros quadrados de área (às vezes quilômetros), tornando desnecessária a adoção de uma terceira coordenada.

A equação que obteremos nesse processo é conhecida como Equação de Difusão-Advecção, sendo já bastante tradicional. É interessante observar que são englobados vários fatores, alguns microscópicos, outros macroscópicos e alguns que poderiam ser encarados como externos.

Os fenômenos considerados basicamente no modelo são:

**DIFUSÃO** é levada em conta a “difusão efetiva” (MARCHUK, 1986) que considera a difusão molecular (microscópica) e a turbulenta (macroscópica). A primeira deriva do movimento browniano dos hidrocarbonetos e a segunda de um espalhamento macroscópico de tensão superficial.

**ADVECCÃO** é um movimento provocado por agentes externos (efetivamente as correntes marítimas locais).

**DEGRADAÇÃO** é também um fenômeno molecular, reunindo as alterações naturalmente sofridas pelos hidrocarbonetos ao longo do tempo, modificando suas

moléculas e, portanto, excluindo-os do cenário estudado. Na nossa abordagem, uma única degradação teórica englobará todas as que ocorrem usualmente, como fotodegradação, biodegradação, emulsificação, afundamento, etc. Finalmente, há as denominadas

**FONTES POLUENTES**, consideradas macroscopicamente: o produto é artificialmente (ou de forma forçada) introduzido no cenário. Serão usualmente denominadas apenas “fontes”.

## 2.2 A Equação

Definiremos a concentração de produto como a quantidade de matéria (petróleo, neste caso) existente em um determinado ponto do plano em determinado instante de tempo e a representaremos por  $u(x, y, t)$ . Em alguns casos, conforme a conveniência, usaremos a notação  $u(\mathbf{x}; t)$ , com  $\mathbf{x} = (x, y)$ , um ponto do plano.

A variação dessa concentração em um elemento de área em  $\mathbf{R}^2$  será dada pela caracterização dos fenômenos descritos na seção 2.1. Os microscópicos estão associados aos conceitos de fluxo (= movimento) e conservação e os macroscópicos mais diretamente ao de balanço de massa. Na figura abaixo temos um elemento de área com os fluxos envolvidos

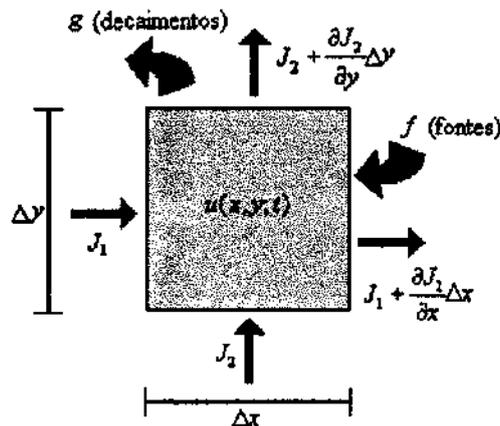


Figura 4: Elemento de área e fluxos de matéria a ele relacionados.

Intuitivamente, a concentração de produto deve aumentar com o que entra e diminuir com o que sai. Ou seja, a variação da concentração pode ser esquematizada como

$$\frac{\Delta u}{\Delta t} \Delta y \Delta x = \boxed{\text{Matéria que entra}} - \boxed{\text{Matéria que sai}}$$

Figura 5 - Esquema de conservação de fluxo.

ou, usando as expressões para os fluxos e tomando  $\Delta x, \Delta y \rightarrow 0$

$$\frac{d}{dt}u(x, y, t)dxdy = J_1 dy + J_2 dx - \left[ \left( J_1 + \frac{\partial J_1}{\partial x} dx \right) dy + \left( J_2 + \frac{\partial J_2}{\partial y} dy \right) dx \right] + f(x, y, t)dxdy - g(x, y, t)dxdy$$

onde  $f$  é a função relacionada com a fonte e  $g$  com o decaimento. Simplificando, obtemos

$$\begin{aligned} \frac{d}{dt}u(x, y, t)dxdy &= -\frac{\partial J_1}{\partial x}dxdy - \frac{\partial J_2}{\partial y}dxdy + f(x, y, t)dxdy - g(x, y, t)dxdy \\ &= -\text{div}(\mathbf{J})dxdy + f(x, y, t)dxdy - g(x, y, t)dxdy \end{aligned}$$

Finalmente, para obtermos o efeito sobre todo o domínio  $\Omega$ , integramos em todos os elementos de área e obtemos

$$\int_{\Omega} \frac{d}{dt}u(x, y, t)dA = \int_{\Omega} \frac{\partial}{\partial t}u(x, y, t)dA = \int_{\Omega} -\text{div}(\mathbf{J})dA + \int_{\Omega} f(x, y, t)dA - \int_{\Omega} g(x, y, t)dA$$

onde  $\mathbf{J}$  é chamado vetor de fluxo, com componentes  $\mathbf{J} = (J_1, J_2)$ . Como a equação acima deve valer para qualquer  $\Omega \subset \mathbb{R}^2$

$$\frac{\partial u}{\partial t} + \text{div}(\mathbf{J}) + g = f \quad (2.1)$$

Em (2.1) temos a forma mais geral de nossa equação. Para completar o modelo, temos de especificar o fluxo  $\mathbf{J}$  e o decaimento  $g$ .

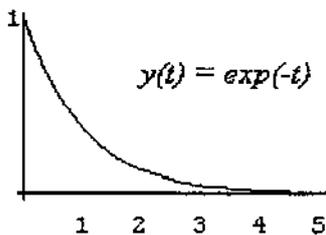


Figura 6 - Decaimento.

Assumiremos o decaimento diretamente proporcional à quantidade de matéria em estudo:  $\frac{dy}{dt} = -ky$ . É uma hipótese simples, baseada na solução  $y(t) = y_0 e^{-kt}$ , solução da EDO acima, e que apresenta um comportamento de decaimento ao longo do tempo. A figura 6 mostra um exemplo de função de decaimento, onde  $k = 1$  e  $y_0 = 1$ . Então tomaremos

$g(x, y, t) \equiv \sigma(x, y, t)u(x, y, t)$ , com  $\sigma(x, y, t) \geq 0$  como taxa de decaimento.

Adotaremos duas formas clássicas para o fluxo  $\mathbf{J}$ . A primeira é o tensor de difusão de Fick. Fick assume que há uma tendência da matéria preencher da maneira mais uniforme possível o espaço disponível (o que é uma hipótese muito razoável). Isso se traduz em um fluxo de matéria das regiões de maior concentração para as de menor, ou ainda um fluxo em alguma direção  $d$  que faça um ângulo obtuso com o gradiente da concentração ( $d^T \nabla u < 0$ ). Tomaremos então um vetor na direção do próprio gradiente da concentração, com sinal trocado ( $\mathbf{J}_1 = -\alpha(x, y, t) \nabla u$ ),  $\alpha(x, y, t) \geq 0$ . A função  $\alpha$  serve como parâmetro de intensidade. Quanto maior seu valor, maior a norma de  $\mathbf{J}_1$ , mais facilmente a matéria se desloca e por conseguinte maior o fluxo na direção de  $-\nabla u$ .

O outro fluxo refere-se à advecção, requerendo um agente de movimento ex-

terno. Seja então  $\mathbf{V} = (V_1(x, y, t), V_2(x, y, t))$  um campo vetorial de velocidades, com cada componente sendo função de  $x, y$  e  $t$ . Nesse caso a movimentação se dá na própria direção do campo vetorial e a quantidade movida é a própria concentração no ponto. Assim, teremos esse fluxo como  $\mathbf{J}_2 = \mathbf{V}u$ .

O fluxo resultante será  $\mathbf{J} = \mathbf{J}_1 + \mathbf{J}_2$  e incorporando-se o decaimento como exposto acima, a equação (2.1) torna-se

$$\frac{\partial u}{\partial t} + \text{div}(-\alpha \nabla u) + \text{div}(\mathbf{V}u) + \sigma u = f \quad (2.2)$$

denominada Equação de Difusão-Advecção.

### 2.3 O Modelo Completo

Juntando-se à (2.2) as condições de contorno, o modelo completo torna-se

$$\begin{aligned} \frac{\partial u}{\partial t} + \text{div}(-\alpha \nabla u) + \text{div}(\mathbf{V}u) + \sigma u &= f, \quad (x, y) \in \Omega \subset \mathbf{R}^2, \quad t \in (0, T] \\ u(x, y, 0) &= u_0(x, y), \quad (x, y) \in \Omega \text{ e} \\ -\alpha \frac{\partial u}{\partial \eta} + ku &= h(x, y, t), \quad (x, y) \in \partial\Omega, t \in (0, T] \text{ para} \end{aligned} \quad (2.3)$$

$$u, f: \Omega \times [0, T] \rightarrow \mathbf{R}, \quad h: \partial\Omega \times (0, T] \rightarrow \mathbf{R} \text{ e } u_0: \Omega \rightarrow \mathbf{R}$$

em que

- $u = u(x, y, t)$  é a concentração de óleo;
- $\alpha = \alpha(x, y, t)$  é uma função que descreve a difusibilidade do óleo no meio;
- $\mathbf{V}(x, y, t) = (V_1(x, y, t), V_2(x, y, t))$  é uma função vetorial que caracteriza em cada ponto, a cada instante, a intensidade e a direção das correntes induzidas pelo vento, circulação local e marés;
- $\sigma = \sigma(x, y, t)$  representa os vários tipos de decaimento do óleo no meio-ambiente;
- $k$  é uma constante que depende do tipo de óleo e da velocidade na fronteira;
- $f$  é uma (ou várias!) fonte poluente.

Na obtenção e a adequação de dados reside um dos problemas da modelagem de fenômenos biológicos/ecológicos. Alguns desses parâmetros podem ser obtidos, ainda que não facilmente. Outros, contudo, devem ser estimados de algum modo (inclusive de maneira muito geral), como as correntes presentes no canal, por exemplo. Neste caso podemos, no máximo, utilizarmos-nos de modelos de circulação - quando eles existem, ou de passadas caracterizações do comportamento hidrodinâmico do Canal (FURTADO, 1978).

Na melhor das hipóteses podemos fornecer uma previsão genérica e qualitativa de comportamento, mesmo quando calibramos o modelo com dados coletados de der-

rames passados. De qualquer maneira, os fenômenos estudados devem ser adequadamente modelados para que o modelo seja numericamente “tratável” em um equipamento acessível, como um PC Pentium. Situações menos simplificadas poderiam ser simuladas em estações de trabalho, computadores de grande porte ou ainda paralelamente através de redes ou máquinas dedicadas, mas estaríamos fugindo de nossa proposta, que é a simulação *in situ* em casos emergenciais.

## 2.4 Simplificações que tornam o modelo tratável

O modelo obtido ainda pode ser simplificado de modo a torná-lo tratável (o que não significa torná-lo simples nem eliminar sua força descritiva qualitativa). Como simplificações consideraremos:

$\alpha = \text{constante}$ . Embora seja mais razoável assumirmos  $\alpha$  dependente da própria solução  $u$ , além do tempo (MEYER, 1993), por períodos relativamente curtos a hipótese de  $\alpha$  constante é aceitável, especialmente na fase difusiva-advectiva. Esta escolha permite o uso de um sistema associado linear; sistemas não lineares (cf CUESTA *et alii*, 1990) exigiriam um tratamento numérico diferente do proposto aqui;

$V = V(x, y)$ , ou seja, o campo de correntes não varia com o tempo. Esta é uma expressão matemática de uma justificativa prática: até o momento não existem dados disponíveis e confiáveis acerca da circulação no Canal; portanto, para os ensaios numéricos aqui constantes, um campo único ao longo de cada a simulação será adotado - são os mencionados cenários. Diferentemente da simplificação acima, havendo dados, a variabilidade temporal de  $V$  pode ser acrescida ao esquema numérico de modo relativamente simples;

$\sigma = \text{constante}$ . Em um primeiro momento,  $\sigma$  também depende de  $u$ , mas como a evaporação acontece principalmente nas primeiras horas do derrame, nós consideraremos o decaimento constante ao longo da segunda fase. Alguns autores (MACKAY, 1980) usam uma equação de diferenças, ou como em MEYER *et alii*, (1992), uma equação diferencial ordinária acoplada, do tipo:

$$\frac{\partial \sigma}{\partial t} = \begin{cases} \kappa \frac{(1 - \sigma) e^{-12\sigma}}{u}, & \text{se } u > 0 \\ 0 & \text{se } u = 0 \end{cases}$$

Incorporando as hipóteses acima (sobretudo as viáveis) ao modelo (2.3), temos:

$$\frac{\partial u}{\partial t} - \alpha \Delta u + u \operatorname{div}(\mathbf{V}) + \mathbf{V} \cdot \nabla u + \sigma u = f, \quad (x, y) \in \Omega \subset \mathbf{R}^2, \quad t \in (0, T]$$

$$\begin{aligned} u(x, y; 0) &= u_0(x, y), & (x, y) \in \Omega \text{ e} \\ -\alpha \frac{\partial u}{\partial \eta} + ku &= h(x, y; t), & (x, y) \in \partial\Omega, t \in (0, T] \text{ para} \end{aligned} \quad (2.4)$$

$$u, f: \Omega \times [0, T] \rightarrow \mathbf{R}, \quad h: \partial\Omega \times (0, T] \rightarrow \mathbf{R} \text{ e } u_0: \Omega \rightarrow \mathbf{R}$$

com  $\Delta$  e  $\nabla$  representando respectivamente o Laplaciano e o gradiente em duas dimensões.



Figura 7 - Mapa do Canal de São Sebastião.

Na figura 7 apresentamos um mapa do Canal de São Sebastião, com destaque para o DTCS. Aquela parte da região que será considerada no estudo compreende as fronteiras continental e de mar aberto. À fronteira que denominaremos  $\Gamma_1$  corresponde a terra (a grosso modo, os “lados” do Canal) e à fronteira  $\Gamma_2$ , corresponde ao mar aberto (ou o que chamamos de “bocas do Canal”). Observemos que  $\Gamma_1 \cap \Gamma_2 = \emptyset$  e que  $\overline{\Gamma_1 \cup \Gamma_2} = \partial\Omega$

Assim, as condições de fronteira para nosso modelo tornam-se:

- $\frac{\partial u}{\partial \eta} \Big|_{\Gamma_1} = 0$ , ou seja, o óleo não “atravessa” a praia, o que é uma simplificação aceitável, e
- o fluxo através das fronteiras de mar aberto é proporcional à concentração de óleo, ou ainda:  $-\alpha \frac{\partial u}{\partial \eta} = ku$  em  $\Gamma_2$ , com  $k = C|\mathbf{V} \cdot \boldsymbol{\eta}|$ , onde figuram  $\boldsymbol{\eta}$  a normal unitária externa a  $\Gamma_2$  e  $C$ , uma constante de proporcionalidade.

### 3. Formulação Variacional - Existência e Unicidade

#### 3.1 A Formulação Variacional

A solução clássica de (2.4) depende fortemente do comportamento da fonte  $f$  e da condição inicial. Como  $f$  e  $u_0$  não precisam necessariamente ser suaves, mas regulares, adotaremos uma formulação fraca, ou variacional, em detrimento de um tratamento analítico direto.

Mas antes algumas notações úteis. De um modo geral, se  $\Omega \subset \mathbf{R}^n$  e  $\mathbf{x} \in \Omega$ , com  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , definimos:

1. O espaço das funções quadrado integráveis, no sentido de Lebesgue, sobre um domínio  $\Omega$  será denotado por  $L^2(\Omega)$ ; ou seja,

$$L^2(\Omega) \equiv \left\{ u : \Omega \rightarrow \mathbf{R} / \int_{\Omega} [u(\mathbf{x})]^2 d\mu < +\infty \right\}.$$

O produto interno e a norma em  $L^2(\Omega)$  serão respectivamente:

$$(u|v)_{L^2(\Omega)} = \int_{\Omega} u(\mathbf{x})v(\mathbf{x})d\mu \qquad \|u\|_{L^2(\Omega)} = \left[ \int_{\Omega} |u(\mathbf{x})|^2 d\mu \right]^{\frac{1}{2}} = (u|u)_{L^2(\Omega)}^{\frac{1}{2}}$$

para  $u, v \in L^2(\Omega)$ .

2. O espaço de Sobolev  $H^1(\Omega)$  é o espaço das funções quadrado integráveis com derivadas também quadrado integráveis:

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

Seus produto interno e norma são:

$$(u|v)_{H^1(\Omega)} = (u|v)_{L^2(\Omega)} + (\nabla u | \nabla v)_{L^2(\Omega)} \qquad \|u\|_{H^1(\Omega)}^2 = \|u\|_{L^2(\Omega)}^2 + \|\nabla u\|_{L^2(\Omega)}^2$$

para  $u, v \in H^1(\Omega)$ , onde  $(\nabla u | \nabla v)_{L^2(\Omega)} = \sum_{k=1}^n \int_{\Omega} \frac{\partial u}{\partial x_k} \frac{\partial v}{\partial x_k} d\mu$ .

Definamos então o subespaço  $S \equiv H^1[(0, T); \Omega]$  das funções que pertencem a  $H^1(\Omega)$  para  $\forall t \in (0, T)$ . Por conveniência de notação definiremos um produto interno em  $S$ . Assim, dadas  $u, v \in S$ , definimos

- o produto interno no interior de  $\Omega$ :  $(u | v)_\Omega = \int_\Omega u(x, y)v(x, y)d\mu$  ou, no caso especial:  $(\nabla u | \nabla v)_\Omega = \int_\Omega \nabla u(x, y) \cdot \nabla v(x, y)d\mu$
- o produto interno ao longo da fronteira:  $\langle u | v \rangle_{\partial\Omega} = \int_{\partial\Omega} u(x, y)v(x, y)d\gamma$

Com essa notação, o problema variacional associado a (2.4) torna-se: encontrar  $u \in S$ , tal que

$$\left( \frac{\partial u}{\partial t} | v \right)_\Omega - \underbrace{\alpha(\Delta u | v)_\Omega}_{(*)} + \underbrace{(\text{div}(\mathbf{V}u) | v)_\Omega}_{(**)} + \sigma(u | v)_\Omega = (f | v)_\Omega$$

para  $\forall v \in S$ .

No termo (\*) aplicaremos o Teorema de Green (integração por partes) para obtermos

$$-\alpha(\Delta u | v)_\Omega = \alpha(\nabla u | \nabla v)_\Omega - \alpha \left\langle \frac{\partial u}{\partial \eta} | v \right\rangle_{\partial\Omega} = \alpha(\nabla u | \nabla v)_\Omega + k \langle u | v \rangle_{\Gamma_2}$$

dadas as condições de fronteira assumidas e a linearidade do produto interno. Expandindo-se ainda o termo (\*\*), obtemos

$$\left( \frac{\partial u}{\partial t} | v \right)_\Omega + \alpha(\nabla u | \nabla v)_\Omega + (u \text{div} \mathbf{V} | v)_\Omega + (\mathbf{V} \cdot \nabla u | v)_\Omega + \sigma(u | v)_\Omega + k \langle u | v \rangle_{\Gamma_2} = (f | v)_\Omega$$

ou, de forma menos compacta

$$\begin{aligned} & \left( \frac{\partial u}{\partial t} | v \right)_\Omega + \alpha(\nabla u | \nabla v)_\Omega + \left( u \frac{\partial V_1}{\partial x} | v \right)_\Omega + \left( u \frac{\partial V_2}{\partial y} | v \right)_\Omega + \\ & + \left( V_1 \frac{\partial u}{\partial x} | v \right)_\Omega + \left( V_2 \frac{\partial u}{\partial y} | v \right)_\Omega + \sigma(u | v)_\Omega + k \langle u | v \rangle_{\Gamma_2} = (f | v)_\Omega \end{aligned} \quad (3.1)$$

para  $\forall v \in S$ .

### 3.2 Existência e unicidade de soluções

Para podermos prosseguir com a construção de aproximações para (3.1), precisamos garantir a existência e a unicidade da solução. Para tanto, faremos uso do Teorema de Lions (LIONS, 1961 - teorema 1.1, cap. IV).

**TEOREMA DE LIONS:** Seja uma família de formas sesquilineares  $a(t; u, v)$  sobre  $S$ , com  $t \in [0, T]$ . Se as hipóteses seguintes forem satisfeitas

1. Para  $\forall u, v \in \mathbf{S}$  a função  $t \rightarrow a(t; u, v)$  é mensurável;
  2. A forma  $a$  é contínua, ou seja,  $|a(t; u, v)| \leq M \|u\|_{H^1} \|v\|_{H^1}$ ;
  3. A forma  $a$  satisfaz a seguinte desigualdade: existe um  $\lambda > 0$  tal que  $a(t; v, v) + \lambda \|v\|_{L^2}^2 \geq \delta \|v\|_{H^1}^2$ ;
  4. O operador linear  $(f|v)_\Omega$  é contínuo em  $v$  e
  5.  $u_0, f \in L^2(\Omega)$ ,
- então o problema de encontrar  $u \in L^2[(-\infty, T), \mathbf{S}]$  com  $u \equiv 0$  para  $t < 0$  e

$$\left( \frac{\partial u}{\partial t} | v \right)_\Omega + a(t; u(t), v) = (f|v)_\Omega$$

tem solução única.

No Apêndice A o problema (3.1) é colocado nas condições do Teorema de Lions.

## 4. Construindo a solução

### 4.1 Galerkin e Elementos Finitos

Estamos no ponto apropriado para a apresentação do modelo discretizado com o qual se irá definir o método numérico que julgamos conveniente para a aproximação de soluções de nosso problema. Já obtivemos a formulação variacional e temos uma informação importante: o problema, na forma em que se encontra, possui solução que é única. O que precisamos agora, já que não podemos descrever analiticamente essa solução, é aproximá-la. Com esse propósito introduzimos o Método de Galerkin.

Seja  $S_h \subset \mathbf{S}$  um espaço de dimensão  $N < +\infty$  e com base  $\beta = \{\varphi_1, \varphi_2, \dots, \varphi_N\}$ . É importante ressaltar que cada  $\varphi_j, j = 1, \dots, N$  é função de  $x$  e  $y$ , já que  $\Omega \subset \mathbf{R}^2$ . Como a aproximação da solução de (3.1) será construída nesse espaço, então, separando-se as variáveis, optamos por

$$u(x, y, t) \equiv u_h(x, y, t) = \sum_{j=1}^N c_j(t) \varphi_j(x, y) \quad (4.1)$$

A idéia central por trás de (4.1) remete-nos a um método muito tradicional de resolução analítica de equações diferenciais parciais chamado separação de variáveis. Neste último assumimos nossa solução como um produto de funções independentes nas variáveis temporal e espaciais. No método de Galerkin também temos essa separação, não só da solução, mas também do espaço em que buscamos nossa solução (algo muito similar a tentar decompor  $\mathbf{S}$  como soma direta de subespaços convenientes - só que aqui a decomposição não é uma soma direta).

Usando a aproximação (4.1) acima com a formulação variacional, obtemos

$$\begin{aligned} & \left( \frac{\partial u_h}{\partial t} | v \right)_\Omega + \alpha (\nabla u_h | \nabla v)_\Omega + (u_h \operatorname{div} \mathbf{V} | v)_\Omega + \\ & + (\mathbf{V} \cdot \nabla u_h | v)_\Omega + \sigma (u_h | v)_\Omega + k \langle u_h | v \rangle_{\Gamma_2} = (f | v)_\Omega \end{aligned}$$

que nos leva a

$$\begin{aligned} & \left( \sum_j \frac{dc_j}{dt} \varphi_j | v \right)_\Omega + \alpha \left( \sum_j c_j \nabla \varphi_j | \nabla v \right)_\Omega + \left( \sum_j c_j \varphi_j \operatorname{div} \mathbf{V} | v \right)_\Omega + \left( \sum_j c_j \mathbf{V} \cdot \nabla \varphi_j | v \right)_\Omega + \\ & + \sigma \left( \sum_j c_j \varphi_j | v \right)_\Omega + k \left\langle \sum_j c_j \varphi_j | v \right\rangle_{\Gamma_2} = (f | v)_\Omega \end{aligned} \quad (4.2)$$

para  $\forall v \in \mathbf{S}_h$ . Como os coeficientes  $c_j = c_j(t)$  não dependem de  $x$  e  $y$ , podem sair dos produtos internos, que são definidos apenas nas variáveis espaciais. Assim:

$$\begin{aligned} & \sum_j \frac{dc_j}{dt} (\varphi_j | v)_\Omega + \alpha \sum_j c_j (\nabla \varphi_j | \nabla v)_\Omega + \sum_j c_j (\varphi_j \operatorname{div} \mathbf{V} | v)_\Omega + \\ & + \sum_j c_j (\mathbf{V} \cdot \nabla \varphi_j | v)_\Omega + \sigma \sum_j c_j (\varphi_j | v)_\Omega + k \sum_j c_j \langle \varphi_j | v \rangle_{\Gamma_2} = (f | v)_\Omega. \end{aligned} \quad (4.3)$$

Ora, escrever a expressão (4.3) para  $\forall v \in \mathbf{S}_h$  equivale a escrevê-la para os elementos de  $\beta$ . Portanto:

$$\begin{aligned} & \sum_j \frac{dc_j}{dt} (\varphi_j | \varphi_i)_\Omega + \alpha \sum_j c_j (\nabla \varphi_j | \nabla \varphi_i)_\Omega + \sum_j c_j (\varphi_j \operatorname{div} \mathbf{V} | \varphi_i)_\Omega + \\ & + \sum_j c_j (\mathbf{V} \cdot \nabla \varphi_j | \varphi_i)_\Omega + \sigma \sum_j c_j (\varphi_j | \varphi_i)_\Omega + k \sum_j c_j \langle \varphi_j | \varphi_i \rangle_{\Gamma_2} = (f | \varphi_i)_\Omega \end{aligned} \quad (4.4)$$

para  $i = 1, \dots, N$ .

Conforme descrito na seção (2.4), há um campo vetorial de forças relacionado ao termo advectivo e representado por  $\mathbf{V}(x, y) = (V_1(x, y), V_2(x, y))$ . Temos pouca informação a cerca da hidrodinâmica do Canal (correntes e ventos); então baseamos-nos no trabalho de FURTADO (1978) e atribuímos valores pontuais<sup>1</sup> de velocidades ao longo do Canal. Esses valores serão interpolados utilizando-se como funções de interpolação as próprias funções de base de  $\mathbf{S}_h$ . Desse modo

$$V_1(x, y) = \sum_i V_1^i \varphi_i(x, y) \quad \text{e} \quad V_2(x, y) = \sum_m V_2^m \varphi_m(x, y)$$

com  $V_1^l, V_2^m \in \mathbf{R}$ ,  $l, m = 1, \dots, N$ .

Substituindo-se as expressões acima em (4.4) obtemos

<sup>1</sup> Os pontos aonde foram assumidas velocidades serão explicitados mais adiante.

$$\begin{aligned}
& \sum_j \frac{dc_j}{dt} (\varphi_j | \varphi_i)_\Omega + \alpha \sum_j c_j (\nabla \varphi_j | | \nabla \varphi_i)_\Omega + \sigma \sum_j c_j (\varphi_j | \varphi_i)_\Omega + k \sum_j c_j \langle \varphi_j | \varphi_i \rangle_{\Gamma_2} + \\
& + \sum_j c_j \left( \varphi_j \sum_l V_l^1 \frac{\partial \varphi_l}{\partial x} | \varphi_i \right)_\Omega + \sum_j c_j \left( \varphi_j \sum_m V_m^2 \frac{\partial \varphi_m}{\partial y} | \varphi_i \right)_\Omega + \\
& + \sum_j c_j \left( \frac{\partial \varphi_j}{\partial x} \sum_l V_l^1 \varphi_l | \varphi_i \right)_\Omega + \sum_j c_j \left( \frac{\partial \varphi_j}{\partial y} \sum_m V_m^2 \varphi_m | \varphi_i \right)_\Omega = (f | \varphi_i)_\Omega.
\end{aligned} \tag{4.5}$$

O sistema acima pode ser escrito como

$$\mathbf{A}(\varphi_j, \varphi_i) \dot{\mathbf{c}}(t) = \mathbf{B}(\varphi_j, \varphi_i) \mathbf{c}(t) + \mathbf{d}(f, \varphi_i)$$

caracterizando-o como um sistema de equações diferenciais ordinárias, determinado pelas matrizes  $\mathbf{A}$  e  $\mathbf{B}$ , que por sua vez dependem exclusivamente de nossa escolha das funções de base.

As aproximações temporais usadas na resolução de (4.5) serão dadas pelo Método de Crank-Nicolson<sup>2</sup>, de onde obtemos:

(i) a derivada temporal

$$\frac{dc_j^{n+1/2}}{dt} \cong \frac{c_j^{n+1} - c_j^n}{\Delta t},$$

onde o índice  $n$  representa o passo no tempo e

(ii) para os coeficientes  $c$  e o termo fonte

$$c_j^{n+1/2} = \frac{c_j^{n+1} + c_j^n}{2} \quad \text{e} \quad f_j^{n+1/2} = \frac{f_j^{n+1} + f_j^n}{2}$$

Após uma série de manipulações algébricas e juntando-se as aproximações acima a (4.5), obtemos finalmente

<sup>2</sup> Que consiste em tomarmos a aproximação em um ponto intermediário no tempo  $(n+1/2)$  como a média entre dois tempos subsequentes.

$$\begin{aligned}
& \sum_j c_j^{n+1} \left\{ \left[ 1 + \sigma \frac{\Delta t}{2} \right] (\varphi_j | \varphi_i)_\Omega + \alpha \frac{\Delta t}{2} (\nabla \varphi_j | | \nabla \varphi_i)_\Omega + k \frac{\Delta t}{2} \langle \varphi_j | \varphi_i \rangle_{\Gamma_2} + \right. \\
& \quad \left. + \frac{\Delta t}{2} \left[ \sum_i V_1^i \left( \varphi_j \frac{\partial \varphi_i}{\partial x} | \varphi_i \right)_\Omega + \sum_m V_2^m \left( \varphi_j \frac{\partial \varphi_m}{\partial y} | \varphi_i \right)_\Omega \right] + \right. \\
& \quad \left. + \frac{\Delta t}{2} \left[ \sum_i V_1^i \left( \frac{\partial \varphi_j}{\partial x} \varphi_i | \varphi_i \right)_\Omega + \sum_m V_2^m \left( \frac{\partial \varphi_j}{\partial y} \varphi_m | \varphi_i \right)_\Omega \right] \right\} = \\
& \sum_j c_j^n \left\{ \left[ 1 - \sigma \frac{\Delta t}{2} \right] (\varphi_j | \varphi_i)_\Omega - \alpha \frac{\Delta t}{2} (\nabla \varphi_j | | \nabla \varphi_i)_\Omega - k \frac{\Delta t}{2} \langle \varphi_j | \varphi_i \rangle_{\Gamma_2} - \right. \\
& \quad \left. - \frac{\Delta t}{2} \left[ \sum_i V_1^i \left( \varphi_j \frac{\partial \varphi_i}{\partial x} | \varphi_i \right)_\Omega + \sum_m V_2^m \left( \varphi_j \frac{\partial \varphi_m}{\partial y} | \varphi_i \right)_\Omega \right] - \right. \\
& \quad \left. - \frac{\Delta t}{2} \left[ \sum_i V_1^i \left( \frac{\partial \varphi_j}{\partial x} \varphi_i | \varphi_i \right)_\Omega + \sum_m V_2^m \left( \frac{\partial \varphi_j}{\partial y} \varphi_m | \varphi_i \right)_\Omega \right] \right\} + \Delta t (f_j^{n+1/2} | \varphi_i)_\Omega
\end{aligned}$$

ou, em forma matricial

$$\mathbf{A} \mathbf{c}^{n+1} = \mathbf{B} \mathbf{c}^n + \mathbf{d} \quad (4.6)$$

onde

$$\begin{aligned}
a_{ij} = & \left[ 1 + \sigma \frac{\Delta t}{2} \right] (\varphi_j | \varphi_i)_\Omega + \alpha \frac{\Delta t}{2} (\nabla \varphi_j | | \nabla \varphi_i)_\Omega + k \frac{\Delta t}{2} \langle \varphi_j | \varphi_i \rangle_{\Gamma_2} + \\
& + \frac{\Delta t}{2} \left[ \sum_i V_1^i \left( \varphi_j \frac{\partial \varphi_i}{\partial x} | \varphi_i \right)_\Omega + \sum_m V_2^m \left( \varphi_j \frac{\partial \varphi_m}{\partial y} | \varphi_i \right)_\Omega \right] + \\
& + \frac{\Delta t}{2} \left[ \sum_i V_1^i \left( \frac{\partial \varphi_j}{\partial x} \varphi_i | \varphi_i \right)_\Omega + \sum_m V_2^m \left( \frac{\partial \varphi_j}{\partial y} \varphi_m | \varphi_i \right)_\Omega \right],
\end{aligned}$$

$$\begin{aligned}
b_{ij} = & \left[ 1 - \sigma \frac{\Delta t}{2} \right] (\varphi_j | \varphi_i)_\Omega - \alpha \frac{\Delta t}{2} (\nabla \varphi_j | | \nabla \varphi_i)_\Omega - k \frac{\Delta t}{2} \langle \varphi_j | \varphi_i \rangle_{\Gamma_2} - \\
& - \frac{\Delta t}{2} \left[ \sum_i V_1^i \left( \varphi_j \frac{\partial \varphi_i}{\partial x} | \varphi_i \right)_\Omega + \sum_m V_2^m \left( \varphi_j \frac{\partial \varphi_m}{\partial y} | \varphi_i \right)_\Omega \right] - \\
& - \frac{\Delta t}{2} \left[ \sum_i V_1^i \left( \frac{\partial \varphi_j}{\partial x} \varphi_i | \varphi_i \right)_\Omega + \sum_m V_2^m \left( \frac{\partial \varphi_j}{\partial y} \varphi_m | \varphi_i \right)_\Omega \right] e
\end{aligned}$$

$$d_i = \Delta t (f_j^{n+1/2} | \varphi_i)_\Omega,$$

tudo com  $i = 1, \dots, N$  e para cada  $n$ .

O sistema (4.6) será resolvido iterativamente no tempo, a partir da condição inicial  $\mathbf{c}^{(0)}$ , pontualmente fornecida.

## 4.2 Discretizando o Domínio Espacial

A discretização do Canal se dará através de elementos finitos triangulares. Nos primeiros testes, em que a representação do Canal era precária (apenas um retângulo, por exemplo), a triangulação era feita manualmente. Com a necessidade de maior precisão e

de maior semelhança com o formato geográfico do Canal, optamos por usar o programa *PDEase*. Esse *software* é especificamente desenhado para a resolução de equações diferenciais parciais e possui embutido um gerador de malhas triangulares. Abaixo apresentamos duas malhas já usadas para testes. A maior delas engloba tanto a porção continental quanto toda a ilha e um entorno, por sugestão da CETESB; a menor mostra só parte do continente e parte da Ilha, por sugestão da Petrobrás. Escolhemos posteriormente trabalhar apenas com esta última, dado que a

maioria dos derrames acontece nessa região, raramente atingindo outras partes da ilha. Outro motivo é o da limitação dos sistemas. No caso do entorno, o Canal “comporta” apenas dois ou três elementos em sua parte mais estreita, reduzindo a qualidade da saída do programa.

O vértice de cada triângulo (ou elemento) possui um número global único associado à suas coordenadas espaciais. Nesses vértices foram distribuídas as velocidades de acordo com o perfil traçado no trabalho de FURTADO, (1978). Essa distribuição de velocidades ficou a cargo de dois alunos de Iniciação Científica, DE OLIVEIRA, André Luis Benevides e PAIVA, Ítalo Amaral.

## 5. Resultados, enfim!

Nas próximas páginas apresentaremos alguns resultados de simulações executadas com o programa usando o domínio padrão já apresentado.

### 5.1 As Simulações sobre o Canal

Na figura abaixo estão indicados, sobre o mapa do Canal de São Sebastião, alguns pontos de interesse nas simulações. A relação desses pontos encontra-se na tabela na seqüência.

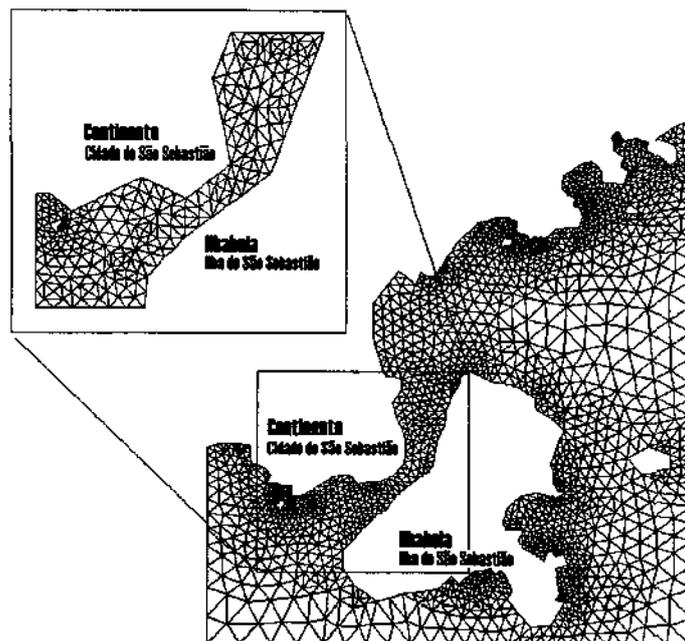


Figura 8 - Discretizações do Canal.

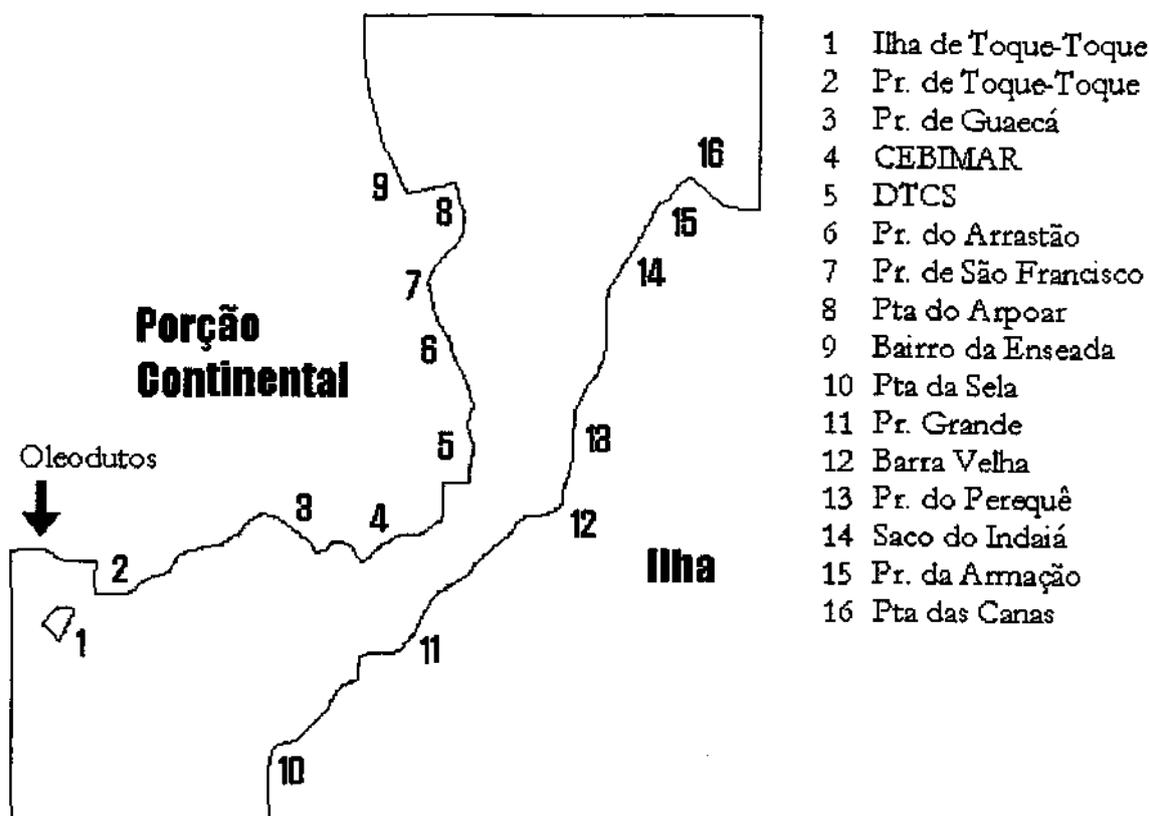


Figura 9 - Canal de São Sebastião com pontos chave.

No.	Localidade	Nó	Comentário <sup>3</sup>
1	Ilha de Toque-Toque	62	Ponto ao norte da Ilha
2	Ponta de Toque-Toque	96	Sul da porção continental
3	Praia de Guaecá	312	Sul da porção continental
-	Oleodutos	22	Oleodutos da Petrobrás
4	CEBIMAR	407	Centro de Biologia Marinha Centro da porção continental
5	DTCS	563	Dutos e Terminais do Centro e Sul Terminal Petrolífero da Petrobrás Centro da porção continental
6	Praia do Arrastão	488	Centro da porção continental
7	Praia de São Francisco	469	Norte da porção continental
8	Ponta do Arpoar	566	Norte da porção continental
9	Bairro da Enseada	438	Norte da porção continental
10	Ponta da Sela	273	Sul de Ilhabela
11	Praia Grande	379	Sul de Ilhabela
12	Barra Velha	763	Centro de Ilhabela
13	Praia do Perequê	777	Centro de Ilhabela
14	Saco do Indaiá	841	Norte de Ilhabela

<sup>3</sup> Não necessariamente de acordo com classificação adotada pela CETESB.

15	Praia da Armação	874	Norte de Ilhabela
16	Ponta das Canas	895	Norte de Ilhabela

Tabela 1 - Pontos de interesse no Canal de São Sebastião.

Estas localidades foram escolhidas por serem pontos normalmente afetados em casos de derrames, ou por uma questão de uniformidade na distribuição dos pontos.

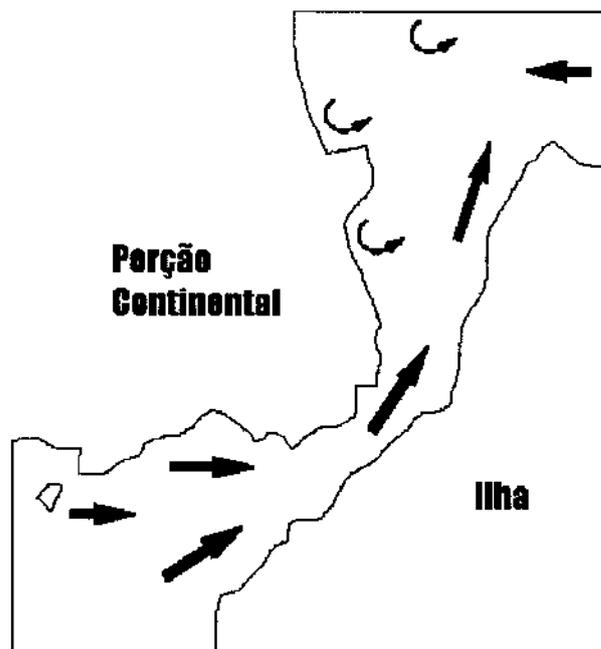


Figura 10 - Representação do campo de velocidades SUL-NORTE no Canal.

A figura 10 ilustra o campo base de velocidades, que denominaremos de SUL-NORTE, pois as correntes predominantes têm sentido de sul para norte. Em algumas simulações é adotado um campo oposto ao de base, denominado NORTE-SUL, obtido do primeiro simplesmente por inversão das componentes dos vetores componentes do campo (esse tipo de comportamento, de norte para sul, é mais raro, mas normalmente é o que provoca desastres de maior extensão - POFFO, 1995 - comunicação verbal). Finalmente adotaremos um terceiro campo de velocidade que tenta reproduzir, na medida do possível, o de FURTADO (1978).

### 5.1.1 Cenário 1 - rompimento de oleoduto

Felizmente o número de ocorrências de derrame de óleo por rompimento de oleoduto é inversamente proporcional a sua gravidade. O rompimento de um oleoduto configura uma das piores situações de derrame de óleo, pelo menos na região do Canal de São Sebastião. Apesar das válvulas de segurança, parte do óleo que estava subindo em direção às refinarias volta e vaza pelo local do rompimento. Os oleodutos que levam o petróleo do DTCS às refinarias localizam-se na parte sul do continente (cf figura 9), à entrada do Canal.

Os dados do cenário 1 são:

Parâmetro	Valor	Parâmetro	Valor
Difusão ( $\alpha$ )	0.005	Iterações	7200
Decaimento ( $\sigma$ )	0.001	Corrente	Sul-norte
Permeabilidade ( $k$ )	0.1	Condição inicial	Ausente
$\Delta t$	0.1	Fonte poluente	Oleoduto Pontos: 22, 37, 50 Intensidade = 4
Elementos	Primeira ordem	Erro por negativos	0.036 %
Upwind	Não	Pico de concentra-	158.1881

	ção
Tempo de simulação 4m10s (Pentium 133 MHz)	

Tabela 2 - Dados do cenário 1.

O termo advectivo em (2.4) tende a gerar oscilações quando tratado numericamente em equações desse tipo (BROOKS *et alii*, 1982 e CHRISTIE *et alii*, 1976). Uma das medidas da qualidade do método é então o tamanho do pico negativo em relação ao pico positivo da solução. Essa medida é aqui fornecida em forma de porcentagem, sob o nome de erro por negativos.

Os resultados da simulação foram:



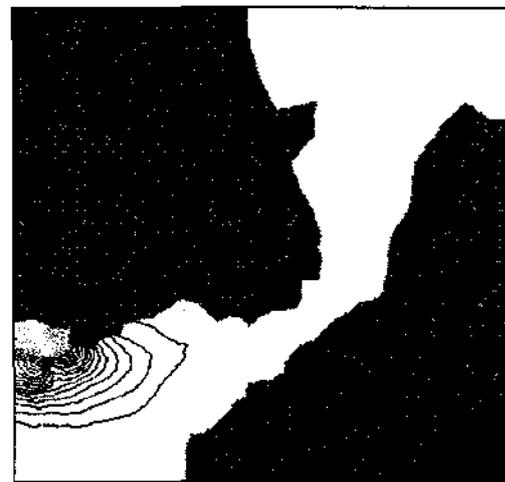
300 iterações



2400 iterações

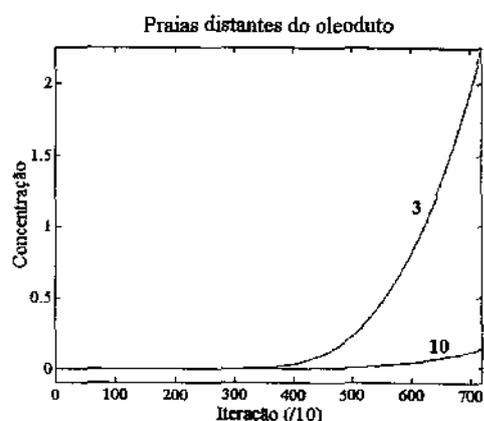
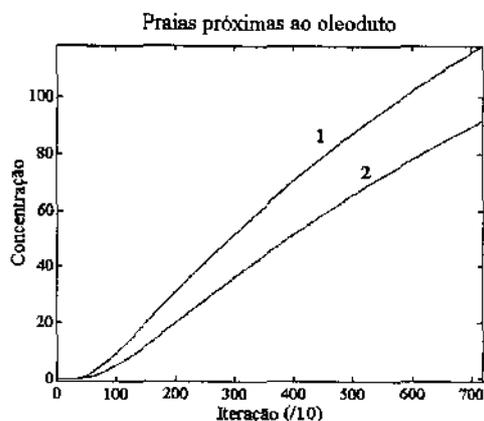


4800 iterações



7200 iterações

Aqui os pontos mais afetados foram a Ilha e a Praia de Toque-Toque. Com os parâmetros usados, o óleo praticamente não afeta os outros pontos de interesse. Uma explicação para este comportamento é a presença da própria Ilha de Toque-Toque, que bloqueia parcialmente as correntes que vêm de sul. A figura abaixo mostra a concentração em alguns desses pontos (de acordo com a tabela de praias previamente fornecida).



### 5.1.2 Cenário 2 – Mancha sem origem determinada I

Em alguns casos manchas sem origem conhecida simplesmente aparecem em determinados pontos do Canal. Uma possível razão para o aparecimento destas manchas é a simples falta de observação. Alguns navios lavam seus tanques clandestinamente, durante a madrugada. O óleo lavado às vezes é detectado horas depois, distante do local de origem e portanto classificado como “sem origem aparente”.

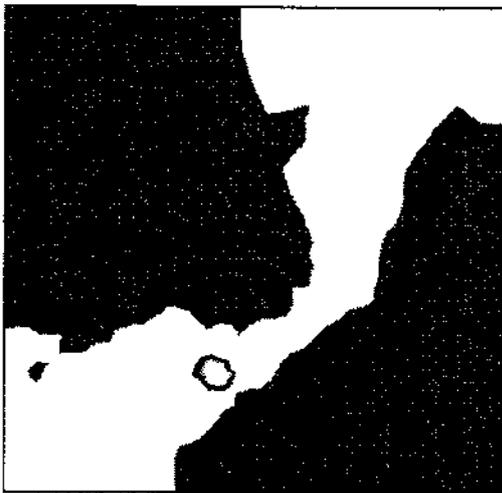
Outra diferença básica deste cenário com o anterior é o uso de Elementos Finitos de Segunda Ordem.

Os dados do cenário 2 são:

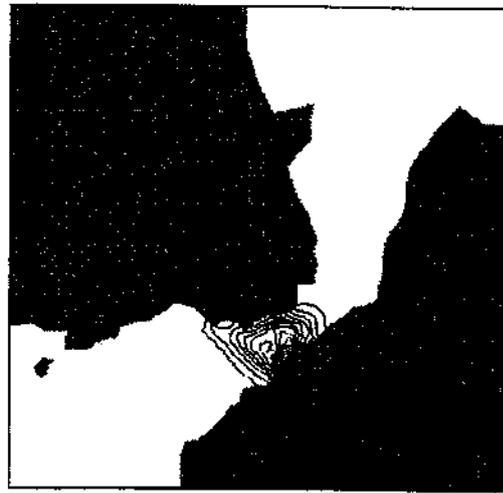
Parâmetro	Valor	Parâmetro	Valor
Difusão ( $\alpha$ )	0.003	Iterações	4800
Decaimento ( $\rho$ )	0.0001	Corrente	Sul-norte
Permeabilidade ( $k$ )	0.1	Condição inicial	Mancha sem origem aparente, nos pontos: 323, 331, 335, 342, 350, 355 e 361 Qtde = 600
$\Delta t$	0.1	Fonte poluente	Ausente
Elementos	Segunda ordem	Erro por negativos	0.74 %
Upwind	Não	Pico de concentração	600
Tempo de simulação (Pentium 133 MHz)	4m08s		

Tabela 3 - Dados do cenário 2.

Os resultados da simulação podem ser apreciados na seqüência.



**Condição Inicial**



**1500 iterações**

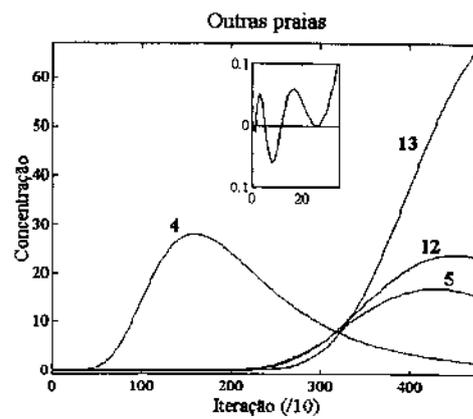
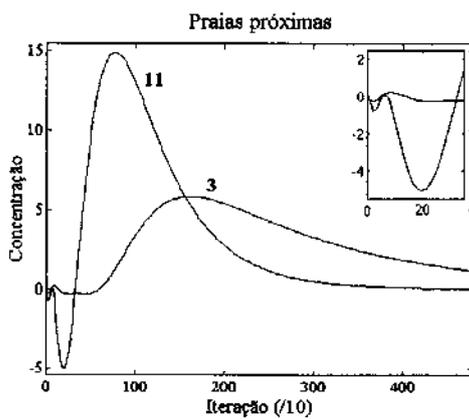


**3000 iterações**



**4800 iterações**

Nas praias obtivemos os seguintes resultados:



Dois fenômenos valem a pena ser observados aqui. Na curva 4 (CEBIMAR) percebemos claramente o efeito difusivo-advectivo. Antes de decorridas 1550 iterações,

a concentração cresce rapidamente, pois estão somados os efeitos difusivo e advectivo; após as 1550 iterações a queda na concentração é mais lenta, pois os efeitos tendem a se anular (parte do óleo que é levado embora por advecção acaba voltando por difusão).

Outro fenômeno interessante diz respeito às curvas 12 (Barra Velha) e 13 (Perequê). Apesar da proximidade geográfica - são praias vizinhas - a concentração em Perequê é bem maior do que em Barra Velha. A explicação disso reside na distribuição de correntes, que coloca em Barra Velha um fluxo na direção da Praia do Perequê. Portanto, a invés de se acumular em Barra Velha o óleo se dirige a Perequê.

No destaque, as oscilações próximas a  $t = 0$ , resultantes das discretizações adotadas.

### 5.1.3 Cenário 3 – Mancha sem origem determinada II

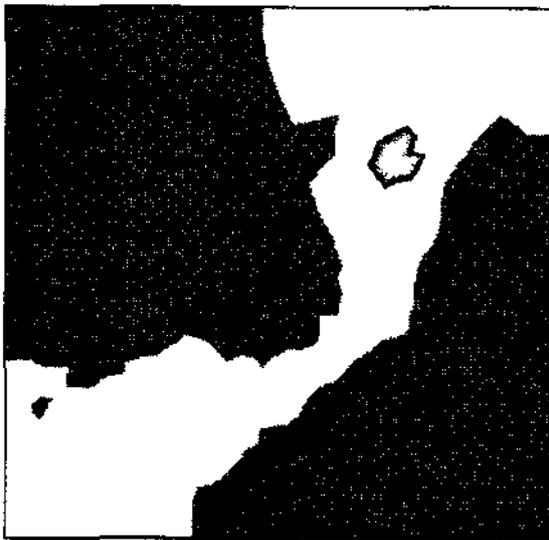
Trata-se do mesmo caso do cenário 2, mas desta vez a mancha é colocada na parte norte do Canal, com a corrente no sentido norte-sul.

Os dados do cenário 3 são:

Parâmetro	Valor	Parâmetro	Valor
Difusão ( $\alpha$ )	0.001	Iterações	4800
Decaimento ( $\sigma$ )	0.0001	Corrente	Norte-sul
Permeabilidade ( $k$ )	0.1	Condição inicial	Mancha sem origem aparente, nos pontos: 697, 709, 716, 736, 746, 750 e 766 Qtde = 600
$\Delta t$	0.1	Fonte poluente	Ausente
Elementos	Segunda ordem	Erro por negativos	0.25 %
Upwind	Não	Pico de concentração	600
Tempo de simulação (Pentium 133 MHz)	6m36s		

Tabela 4 - Dados do cenário 3.

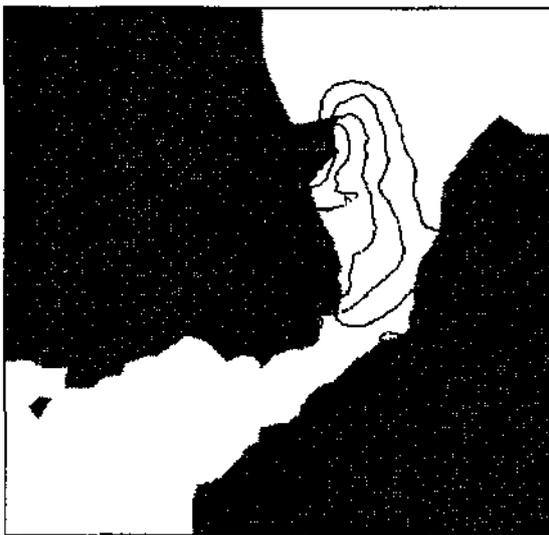
Resultado da simulação com o cenário 3:



**Condição Inicial**



**1500 iterações**

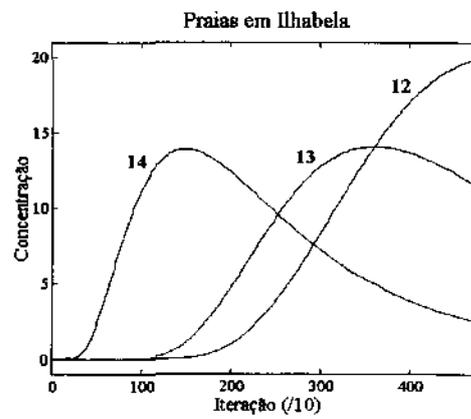
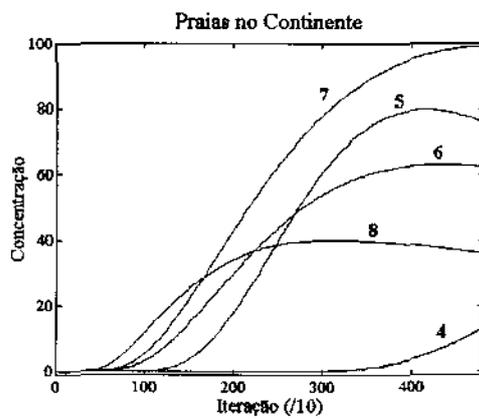


**3000 iterações**



**4800 iterações**

Nas praias do continente e de Ilhabela obtivemos as concentrações:



Podemos extrair algumas conclusões muito interessantes da comparação entre os cenários 2 e 3. Apesar da diferença na localização da mancha inicial, a situação simulada é muito semelhante (presença de mancha inicial, porém sem fonte). Observe que no caso de correntes Sul-Norte (cenário 2), o óleo vai “raspando” uma série de praias em Ilhabela, ocasionando um acúmulo final de produto menor por praia do que no cenário 3. Neste último, o óleo fica ensacado<sup>4</sup>, dificultando seu espalhamento por outras praias e resultando assim numa concentração maior de poluente.

#### 5.1.4 Cenário 4 - Derrame no Terminal DTCS I

Aqui apresentamos uma simulação que leva em consideração uma presença inicial de óleo, mais uma fonte poluente. É o caso de derramamentos provenientes de navios, quando há demora na detecção do problema. Forma-se uma mancha nos primeiros momentos do derramamento, que continuará vazando até que as providências necessárias sejam tomadas.

Os dados do cenário 4 são:

Parâmetro	Valor	Parâmetro	Valor
Difusão ( $\alpha$ )	0.001	Iterações	7200
Decaimento ( $\sigma$ )	0.0005	Corrente	Sul-norte
Permeabilidade ( $k$ )	0.01	Condição inicial	Mancha proveniente de navio atracado. Pontos: 562, 596 e 600 Qtde = 200
$\Delta t$	0.1	Fonte poluente	Vazamento no casco do navio. Intensidade = 2
Elementos	Primeira ordem	Erro por negativos	0.64 %
Upwind	Não	Pico de concentração	200
Tempo de simulação (Pentium 133 MHz)	4m09s		

Tabela 5 - Dados do cenário 4.

Como resultados obtivemos:

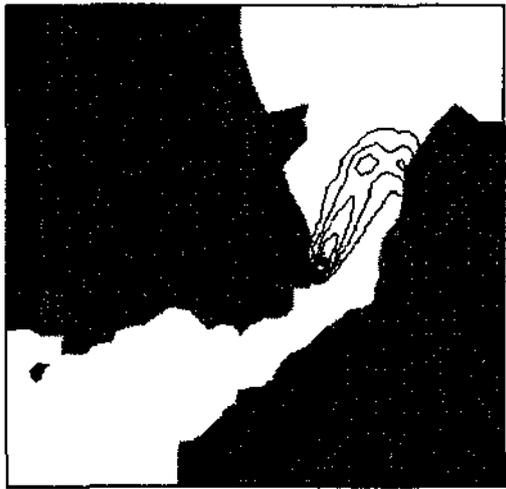
<sup>4</sup> O óleo fica preso em determinada formação geográfica.



**Condição Inicial**



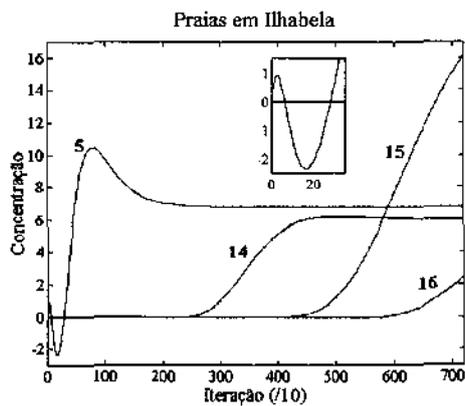
**2400 iterações**



**4800 iterações**



**7200 iterações**



Uma observação muito importante a ser feita é a de que o ponto onde foi feita a tomada de valores não coincide exatamente com o ponto onde foram colocadas a condição inicial e a fonte. Assim, temos um pico de concentração, que depois estabiliza mais ou menos a partir de 1000 iterações.

### 5.1.5 Cenário 5 - Derrame no Terminal DTCS II

O cenário 5 muda em relação ao 4 apenas por inverter o sentido das correntes.

Os dados deste cenário são:

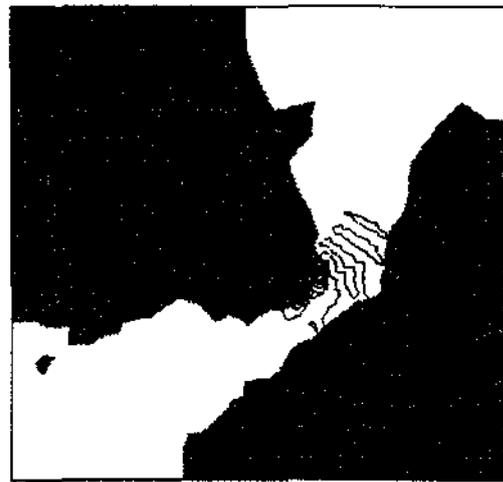
Parâmetro	Valor	Parâmetro	Valor
Difusão ( $\alpha$ )	0.001	Iterações	7200
Decaimento ( $\sigma$ )	0.0005	Corrente	Norte-sul
Permeabilidade ( $k$ )	0.01	Condição inicial	Mancha proveniente de navio atracado. Pontos: 562, 596 e 600 Qtde = 200
$\Delta t$	0.1	Fonte poluente	Vazamento no casco do navio. Intensidade = 2
Elementos	Primeira ordem	Erro por negativos	2.56 %
Upwind	Não	Pico de concentração	486.5955
Tempo de simulação (Pentium 133 MHz)	4m17s		

Tabela 6 - Dados do cenário 5.

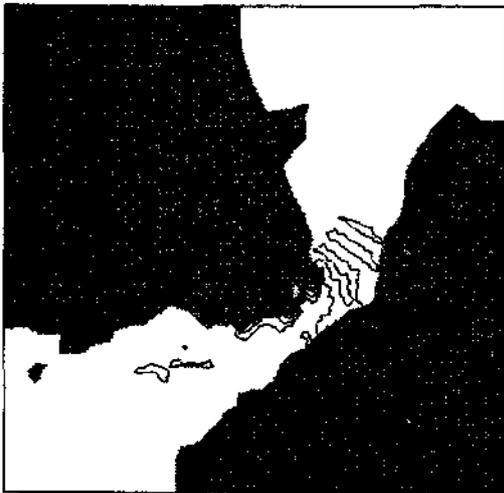
Resultados:



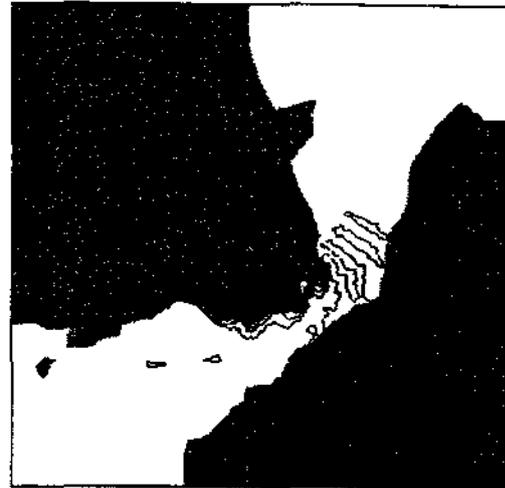
Condição Inicial



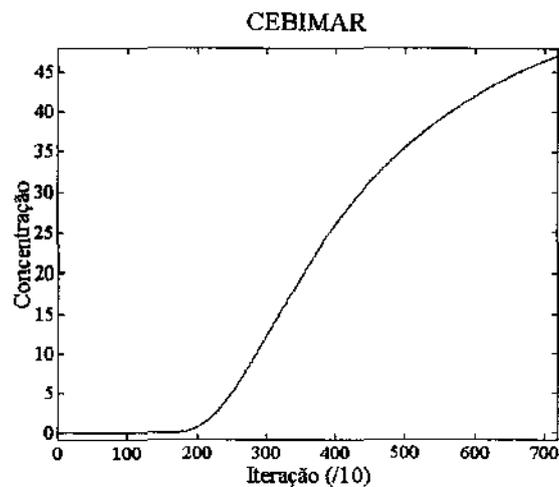
2400 iterações



4800 iterações



7200 iterações



Aqui aparece um acúmulo de óleo maciço em locais próximos ao terminal, que estão no caminho do fluxo, como o CEBIMAR por exemplo.

### 5.1.6 Cenário 6 - O Cenário 2 revisitado

O cenário 2 é ideal para a aplicação do Método SUPG (Apêndice B). Impomos a condição de divergência nula no campo de velocidades e não há fontes poluentes. Além dos gráficos com as soluções, apresentamos a comparação entre a solução obtida em um ponto específico com e sem SUPG.

É interessante salientar que nesse caso são necessárias muito menos iterações para que a mancha se desloque distâncias semelhantes às dos outros cenários. Isso pode ser explicado pela ausência do termo divergente, que está diretamente associado com dissipação de energia. Em outras palavras, energia que nos cenários anteriores era dissipada no próprio campo de velocidade é aqui aproveitada no próprio movimento do óleo.

Os dados do cenário 6 são:

Parâmetro	Valor	Parâmetro	Valor
Difusão ( $\alpha$ )	0.00001	Iterações	600
Decaimento ( $\sigma$ )	0.0001	Corrente	Furtado (1978)
Permeabilidade ( $k$ )	0.1	Condição inicial	Mancha sem origem aparente, nos pontos: 323, 331, 335, 342, 350, 355 e 361 Qtde = 600
$\Delta t$	0.1	Fonte poluente	Ausente
Elementos	Primeira ordem	Erro por negativos	10.16 %
Upwind	Sim	Pico de concentração	600
Tempo de simulação (Pentium 200 MHz)	29s		

Tabela 7 - Dados do cenário 6.

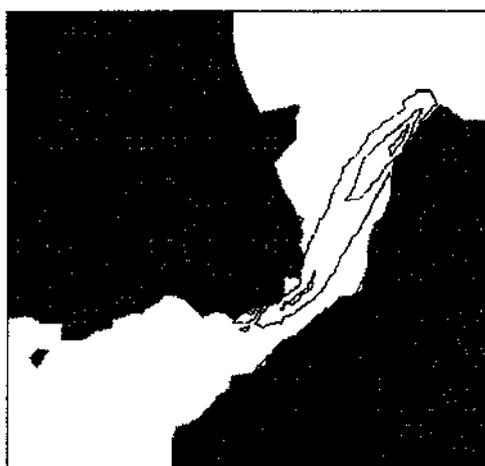
Os resultados obtidos foram:



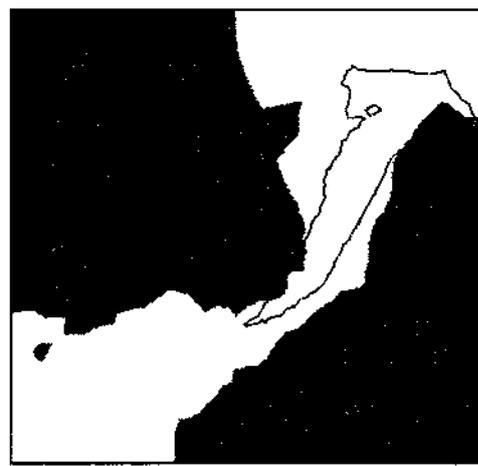
Condição Inicial



90 iterações



240 iterações

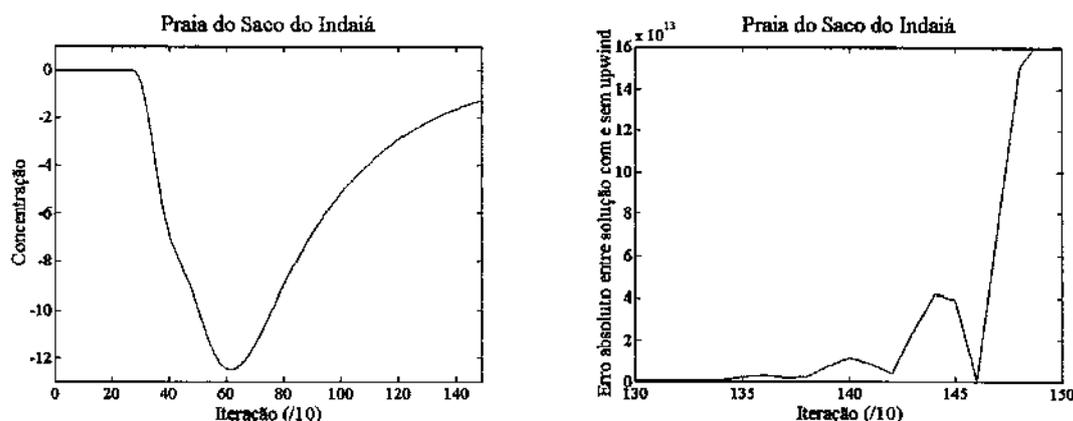


450 iterações

Devemos fazer agora uma análise cuidadosa deste resultado, devida à presença de alguns resultados denominados de erros por negativos de grande magnitude (e.g.  $\approx 10\%$  no cenário 6).

A ideia básica por trás do *upwinding* é a de controlar oscilações explosivas na solução. Essas oscilações aparecem sempre que o limiar do número de Péclet é violado (cf BROOKS & HUGHES, 1982 ou HEINRICH *et alii*, 1977), ou, correspondentemente, quando o efeito advectivo suplanta o difusivo.

Porém um efeito indesejado da introdução dos termos de *upwind* é o aumento nas oscilações numéricas próximo de  $t = 0$  devidas às discretizações adotadas. Esse fenômeno pode ser claramente observado nos gráficos abaixo, onde ilustramos a solução obtida com *upwinding* (mostrando a oscilação negativa, porém controlada) e à direita o erro absoluto entre as soluções com e sem *upwinding*:



Nestes casos, pode-se confiar apenas em uma predição qualitativa e nunca quantitativa. As oscilações próximas de  $t = 0$  acabam sendo amortecidas e o comportamento final da solução acaba por cumprir seu papel.

## 6. Conclusões

Uma das coisas que realmente esperávamos obter do convênio citado na introdução eram os dados necessários para a calibração do modelo. O leitor atento deve ter percebido que em nenhum momento foi apresentada uma comparação com um derrame real, documentado pela CETESB. Há uma resposta simples para isso, através de uma pergunta: como criar o cenário se não temos dados confiáveis de correntes, ventos, difusão e local do derrame? Se há uma grande falha em todo este trabalho, ei-la.

Queremos crer que *Netuno* apresente-se como uma ferramenta futuramente valiosa, mas por enquanto isso é somente uma esperança. O programa se comportou bem nos testes, é robusto, os resultados obtidos situaram-se dentro do esperado, e sim, ainda há coisas a serem melhoradas. Porém, será que uma ferramenta acadêmica como esta irá continuar a se comportar de maneira satisfatória quando confrontada com situações reais ou de emergências? Infelizmente até o fechamento do presente trabalho a resposta é “ainda não sabemos”.

Por outro lado, do ponto de vista didático ousamos dizer que o sucesso foi total. Erramos muito. Aprendemos muito. E também deixamos um material que, cremos, poderá ser muito bem aproveitado por outros, ampliando, melhorando, errando e aprendendo de novo.

Alguns desafios técnicos ainda persistem:

- Amenizar ou - de modo otimista - eliminar as oscilações;
- Desenvolver e codificar outras técnicas e
- Realizar uma análise criticamente comparativa entre diferentes métodos;
- Conseguir dados confiáveis e utilizáveis que sirvam para “calibrar” modelo e esquemas numéricos;
- Implementar versões paralelas;
- Incluir a variação (aliás realista) de certos parâmetros.

Além de tudo isto, há a robustez do programa: com sua utilização outros domínios, outros cenários, outros poluentes podem ter seu estudo beneficiado pelo uso deste *software* desenvolvido.

## Apêndice A - Existência e Unicidade

A fim de colocarmos nossa formulação variacional (3.1) nas condições do Teorema de Lions, definiremos:

$$\begin{aligned}
 a(t; u, v) = & \alpha(\nabla u | \nabla v)_{\Omega} + \left( u \frac{\partial V_1}{\partial x} | v \right)_{\Omega} + \left( u \frac{\partial V_2}{\partial y} | v \right)_{\Omega} + \\
 & + \left( V_1 \frac{\partial u}{\partial x} | v \right)_{\Omega} + \left( V_2 \frac{\partial u}{\partial y} | v \right)_{\Omega} + \sigma(u|v)_{\Omega} + k(u|v)_{\Gamma_2}
 \end{aligned} \tag{A.1}$$

Verifiquemos então as hipóteses:

**MENSURABILIDADE DE  $a$ :** a mensurabilidade de  $a$  é prontamente garantida de sua própria definição.

**CONTINUIDADE DE  $a$ :** faremos a verificação termo a termo:

1. Usando a desigualdade de Cauchy-Schwartz, temos

$$\left| \alpha(\nabla u | \nabla v)_{\Omega} + \sigma(u|v)_{\Omega} \right| \leq \max\{\alpha, \sigma\} \left| (\nabla u | \nabla v)_{\Omega} + (u|v)_{\Omega} \right| \leq M_1 \|u\|_{H^1} \|v\|_{H^1}$$

onde  $M_1 = \max\{\alpha, \sigma\}$ . Note-se que  $(\nabla u | \nabla v)_{\Omega} + (u|v)_{\Omega}$  é exatamente o produto interno em  $H^1$ .

2. Assumindo que  $\mathbf{V} \in H^1(\Omega) \times H^1(\Omega)$ , com  $V_1, V_2$  limitadas em  $\Omega$ , podemos escrever

$$\begin{aligned}
 |(\mathbf{V} \cdot \nabla u | v)_{\Omega}| & \leq \left| \int_{\Omega} V_1 \frac{\partial u}{\partial x} v \, d\mu \right| + \left| \int_{\Omega} V_2 \frac{\partial u}{\partial y} v \, d\mu \right| \leq \\
 & \leq \supess_{\Omega} \{ |V_1| \} \left| \int_{\Omega} \frac{\partial u}{\partial x} v \, d\mu \right| + \supess_{\Omega} \{ |V_2| \} \left| \int_{\Omega} \frac{\partial u}{\partial y} v \, d\mu \right| \leq \\
 & \leq \max \left\{ \supess_{\Omega} \{ |V_1| \}, \supess_{\Omega} \{ |V_2| \} \right\} \|u\|_{H^1} \|v\|_{H^1} = M_2 \|u\|_{H^1} \|v\|_{H^1}
 \end{aligned}$$

com  $M_2 = \max \left\{ \supess_{\Omega} \{ |V_1| \}, \supess_{\Omega} \{ |V_2| \} \right\}$ .

3. De forma análoga à 2 acima, obtemos a desigualdade:

$$|(u \operatorname{div}(\mathbf{V}) | v)_{\Omega}| \leq \max \left\{ \supess_{\Omega} \left\{ \left| \frac{\partial V_1}{\partial x} \right| \right\}, \supess_{\Omega} \left\{ \left| \frac{\partial V_2}{\partial y} \right| \right\} \right\} \|u\|_{H^1} \|v\|_{H^1} = M_3 \|u\|_{H^1} \|v\|_{H^1}$$

com  $M_3 = \max \left\{ \supess_{\Omega} \left\{ \left| \frac{\partial V_1}{\partial x} \right| \right\}, \supess_{\Omega} \left\{ \left| \frac{\partial V_2}{\partial y} \right| \right\} \right\}$ .

4. Devido à imersão contínua de  $H^1$  em  $L^2$  vale

$$|k \langle u | v \rangle_{\partial\Omega}| \leq k \|u\|_{L^2(\partial\Omega)} \|v\|_{L^2(\partial\Omega)} \leq k \|u\|_{L^2} \|v\|_{L^2} \leq k \|u\|_{H^1} \|v\|_{H^1} = M_4 \|u\|_{H^1} \|v\|_{H^1}$$

Juntando-se 1, 2, 3 e 4 temos

$$|a(t; u, v)| \leq (M_1 + M_2 + M_3 + M_4) \|u\|_{H^1} \|v\|_{H^1} = M \|u\|_{H^1} \|v\|_{H^1}$$

e garantimos a continuidade de  $a$ .

**QUASI-COERCIVIDADE:** prontamente obtida da desigualdade de Gårding (BRENNER & SCOTT, 1994), que relaciona operadores elípticos gerais com o tipo de coercividade exigida.

**CONTINUIDADE DO TERMO DO LADO DIREITO** é imediato ver que para  $\forall v \in S$

$$|(f|v)_{\Omega}| \leq \|f\|_{L^2} \|v\|_{L^2} \leq \|f\|_{L^2} \|v\|_{H^1}$$

Como a condição 5 impõe que  $f \in L^2$ , temos garantida a continuidade de  $(f|v)_{\Omega}$ .

**FONTE E CONDIÇÃO INICIAL:** a definição do problema proposto garante que  $u_0, f \in L^2(\Omega)$ .

Portanto, com todas as condições do teorema satisfeitas, garantimos a existência e unicidade da solução fraca de (3.1).

## Apêndice B - Oscilações Numéricas e SUPG

A aproximação de soluções de problemas de Difusão-Advecção começa a se tornar delicada à medida que o coeficiente de difusão  $\alpha$  tende a zero ou, de maneira equivalente, o módulo das velocidades em  $\mathbf{V}$  começam a aumentar (comportamento advectivo-dominante). Se o chamado número de Péclet local (a ser definido posteriormente)  $\gamma^e$  viola a condição  $\gamma^e < 1$ , aparecem oscilações na solução (HEINRICH *et alii*, 1977). Uma das possíveis soluções para esse caso são as chamadas técnicas de *upwinding*.

A idéia é conceitualmente simples: no método de Galerkin funções de base e de teste pertencem ao mesmo espaço. Já o *upwinding* se utiliza de funções teste diferentes, adaptadas ao caso advectivo-dominante. No caso unidimensional essas funções podem ser construídas de modo a obtermos valores nodais exatos da solução, mas esse fato não se generaliza para duas ou mais dimensões.

A técnica específica usada neste trabalho, e aqui brevemente apresentada, denomina-se *Streamline Upwind/Petrov-Galerkin (SUPG)* e foi introduzida por BROOKS & HUGHES em 1982, num trabalho já considerado clássico na área.

## O Método SUPG

Algumas preliminares se fazem necessárias antes da introdução do método pro-

priamente dito.

Como mencionado na seção 2, a fronteira de  $\Omega$  está dividida em  $\Gamma_1$  e  $\Gamma_2$ , de modo que  $\overline{\Gamma_1 \cup \Gamma_2} = \Gamma$  e  $\Gamma_1 \cap \Gamma_2 = \emptyset$ . A cada elemento da discretização será associado um subdomínio  $\Omega_e$ ,  $e = 1, 2, \dots, NEL$  (número total de elementos), com fronteira  $\Gamma_e$  de modo que:

$$\bigcup_e \overline{\Omega_e} = \Omega \quad \text{e} \quad \bigcap_e \Omega_e = \emptyset.$$

Finalmente, definimos a “fronteira interior”:  $\Gamma_{\text{int}} = \bigcup_e \Gamma_e - \Gamma$  (em outras palavras, os lados dos elementos que não pertencem à fronteira do domínio).

De acordo com os autores acima citados, a nova função teste compreende dois termos: a função teste original ( $v$ ), mais uma correção na direção do fluxo:

$$w = v + \tau \mathbf{V} \cdot \nabla v \quad (\text{B.1})$$

onde o parâmetro  $\tau$  controla o grau de *upwinding* (vê-se que  $\tau = 0$  leva exatamente ao Galerkin tradicional).

Repetindo-se a formulação variacional (3.1) com essa nova função teste obtemos:

$$\sum_e \int_{\Omega_e} \left( \frac{\partial u}{\partial t} + \mathbf{V} \cdot \nabla u + \alpha \Delta u - f \right) w d\Omega + \alpha \int_{\Gamma_{\text{int}}} [\eta \cdot \nabla u] w d\Gamma + k \int_{\Gamma_2} uv d\Gamma, \quad (\text{B.2})$$

onde  $\Gamma_{\text{int}}$  representa as fronteiras inter-elementos e com

$$[\eta \cdot \nabla u](\mathbf{x}, t) = \eta(\mathbf{x}) \cdot \{ \nabla u(\mathbf{x}^+, t) - \nabla u(\mathbf{x}^-, t) \}$$

(a figura a seguir ilustra as quantidades envolvidas).

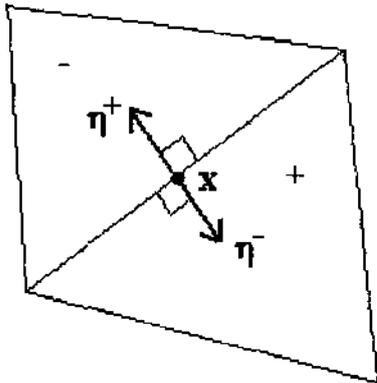


Figura 11 - Normais em  $x$  na fronteira interna.

No presente caso duas ressalvas devem ser feitas: na forma com está, o SUPG só produz bons resultados (i) se o campo de velocidade for conservativo ( $\text{div}(\mathbf{V}) = 0$ ) e (ii) se não houver termos fonte muito intensos. Portanto, ocorrências de derramamento em situações de correntes marítimas mais fortes só serão simuladas com a hipótese do campo conservativo e

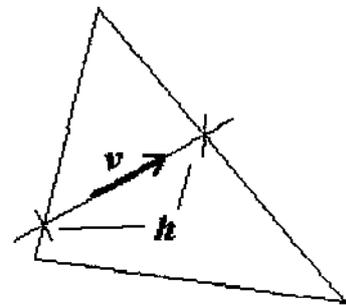


Figura 12 - Comprimento específico.

sem fontes poluentes intensas.

Computacionalmente falando devemos apenas calcular os novos produtos internos e o parâmetro  $\tau$ . Os produtos internos são inteiramente análogos ao da formulação anterior, sem *upwinding*. Já o cálculo de  $\tau$  exige um certo cuidado. Diversas “receitas” são fornecidas na literatura, pois esse parâmetro depende do denominado “dimensão característica do elemento na direção do fluxo”. Ou seja, dada uma velocidade sobre

um elemento, devemos determinar um comprimento dentro do elemento na direção dessa velocidade (cf figura 12). Usamos a “receita” fornecida em CODINA, (1992) e que vem fornecendo bons resultados. O algoritmo<sup>5</sup> para cálculo de  $\tau$  é:

- i. Calcule a média aritmética das velocidades no elemento; chame de  $\mathbf{V}^e$ ;
- ii. Use a transformação (C.1) e calcule velocidades equivalentes às reais, no elemento padrão;
- iii. Calcule a média das velocidades obtidas em (ii); chame de  $\mathbf{V}_N^e$ ;
- iv. Obtenha o comprimento específico  $h_e$ :  $h_e = h_N \frac{\|\mathbf{V}^e\|}{\|\mathbf{V}_N^e\|}$ . A moral da história é uma regra de três entre as velocidades e os comprimentos específicos. Para elementos triangulares usa-se frequentemente  $h_N = 0.7$  (aproximadamente a menor das alturas do triângulo padrão);
- v. Obtenha o número de Péclet local:  $\gamma^e = \frac{\|\mathbf{V}^e\|}{2\alpha} h_e$ ;
- vi. Calcule o parâmetro de *upwind*:  $\tau^e = \frac{\varepsilon^e h^e}{2\|\mathbf{V}^e\|}$ , onde  $\varepsilon^e = \coth(\gamma^e) - \frac{1}{\gamma^e}$ .

## Apêndice C - Aspectos computacionais

Nas seções anteriores preparamos o problema 2.4 para ser computacionalmente tratado. Isto significa, a grosso modo, programar o método de Galerkin. Mas uma olhada mais atenta nos revela uma série de problemas embutidos num contexto maior: discretização do domínio, resolução de sistemas esparsos, armazenamento das soluções e visualização, só pra citar alguns.

### Um pouco de história ...

Como de costume num projeto desse tipo, as primeiras versões de **Netuno** eram muito simples, bem como os resultados com ele obtidos. Os domínios eram retangulares e a triangulação era gerada pelo próprio programa. Havia uma limitação séria de tempo/memória, em sua maior parte por não utilizarmos nenhum esquema de armazenamento esparsos (limitando-nos a no máximo 100 nós e/ou 150 elementos) e em menor escala por deficiências do *hardware* disponível à época.

Mas com o desenvolvimento do projeto, as necessidades foram aumentando, conjuntamente com as dificuldades. Num primeiro momento generalizamos um pouco mais o tratamento de malhas do programa e passamos a trabalhar com malhas em domínios com um formato mais semelhante ao do formato geográfico do Canal; mas ainda assim eram malhas de dimensões reduzidas por causa dos problemas de armazenamento. Quando houve a decisão de realmente fazermos as simulações “sobre o mapa do Canal”, a necessidade de matrizes esparsas não pôde mais ser adiada. Esse foi

<sup>5</sup> Lembrando que  $\tau$  é calculado para cada elemento.

um dos passos importantes do projeto: a criação de estruturas especializadas de armazenamento, e ao mesmo tempo matematicamente apropriadas.

Em dado momento tínhamos todas essas estruturas prontas, mas não era mais viável gerar triangulações razoáveis artesanalmente. A quantidade de elementos e nós era muito grande, além da tendência natural de distribuí-los de maneira cartesiana, e não seguindo os contornos naturais do Canal. Aqui entra em cena o *PDEase*, “talhado” especificamente para resolução de equações diferenciais parciais em domínios bidimensionais. Entre as capacidades do programa está a de gerar malhas de elementos triangulares de primeira e segunda ordens, com algum controle sobre o processo (ressalva: este não é exatamente o produto final ou o objetivo básico do programa, mas de qualquer jeito, funcionou!).

Nossas simulações “primordiais” eram de caráter muito simplificado e podiam ser facilmente visualizadas com um software com capacidades gráficas como o *Mathematica*. Porém, quando adquirimos a capacidade de ampliarmos o domínio, limitados apenas pela memória e velocidade da máquina, essa capacidade de visualização começou a se degenerar. Era muita informação mal representada. Então criamos uma nova forma de visualização de resultados, que foi implementada computacionalmente com a ajuda de D’AFONSECA, Luis Alberto. São os gráficos das simulações já apresentadas neste trabalho.

Finalmente, desejávamos simular um Canal que não tivesse apenas velocidades iguais em todos os pontos, mas sim com um perfil hidrodinâmico que se assemelhasse com os dados que tínhamos a mão na época, baseado no já mencionado trabalho de FURTADO, (1978). E aqui entrou o também já citado trabalho de DE OLIVEIRA, André Luis Benevides e PAIVA, Ítalo Amaral.

Enfim, ao longo das próximas subseções trataremos de todos estes “sub-problemas”.

## Esparsidade

As primeiras versões de *Netuno* trabalhavam com classes de matrizes cheias, levando em conta todos os zeros. Mas não é difícil ver que as matrizes envolvidas no processo são paradigmaticamente esparsas. As bases de elementos finitos usadas têm suporte compacto definido a partir de um nó base e nos nós adjacentes. Portanto os produtos internos entre  $\varphi_i$  e  $\varphi_j$  só serão não nulos se (i)  $i = j$  ou (ii)  $i$  for adjacente a  $j$ . Para o tipo de triangulação fornecida pelo *PDEase* estimamos uma média de 8 elementos não nulos por linha, independentemente do número de elementos ou vértices. A título de exemplo, nosso domínio padrão de testes possui 936 nós, resultando numa matriz com um total de  $936 \times 936 = 876096$  elementos sendo apenas  $8 \times 936 = 7488$  não nulos (em média), correspondendo a menos de 9% da matriz.

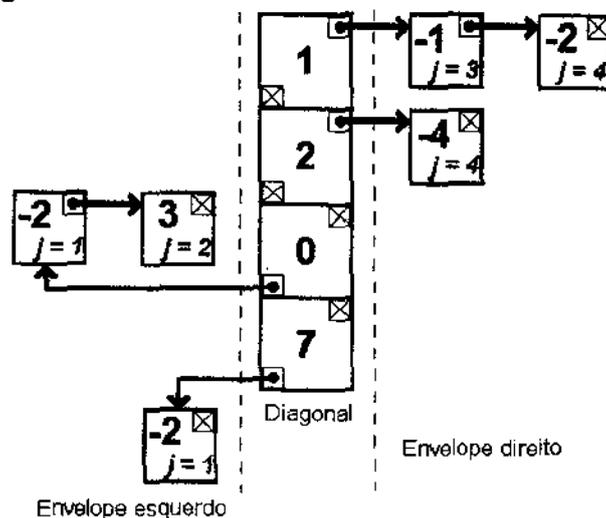
O esquema de esparsidade que adotamos é simples e de autoria própria. Nele levamos em conta não só o armazenamento em si, mas também a facilidade de resolução de sistemas lineares por decomposição LU e o produto matriz/vetor.

## Armazenamento

A única porção estática do armazenamento da matriz  $A$  (nossa matriz exemplo) é a diagonal. Trata-se de um vetor com  $n$  posições, com  $n$  sendo a dimensão da matriz (preocupamo-nos só com matrizes quadradas, mas a adaptação para matrizes retangulares é imediata). Em cada posição deste vetor, além do valor do elemento  $a_{ii}$ ,  $i = 1, \dots, n$ , temos duas listas simplesmente ligadas e ordenadas: uma aponta para o envelope esquerdo e outra para o envelope direito da matriz, contendo os elementos e o índice da coluna. Por exemplo, a matriz

$$A = \begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & 2 & 0 & -4 \\ -2 & 3 & 0 & 0 \\ -2 & 0 & 0 & 7 \end{pmatrix}$$

é armazenada da seguinte forma:



Esse tipo de armazenamento favorece operações que levem em conta separadamente as partes triangular superior e inferior da matriz, como por exemplo em *backward substitution* e *forward substitution*.

A rotina de inserção de elementos na matriz é relativamente simples. Seu algoritmo é:

Dados  $A$ ,  $i$ ,  $j$  e  $a_{ij}$

1. Determinar o envelope em que o elemento se encontra: se  $i > j$ , então está no envelope esquerdo; se  $i < j$ , no direito; caso contrário está na diagonal, necessitando apenas de atualização; encerrar.
2. Pesquisar a lista  $i$  correspondente ao envelope encontrado acima, determinando em que posição o elemento  $a_{ij}$  deve ser inserido; isto é feito percorrendo-se a lista e testando-se os índices  $j$ ;

3. Se o elemento **não** existir, criar a posição de memória e associar o valor correspondente. Fazer as ligações dos ponteiros da lista. encerrar;
4. Se o elemento **existir** e  $\alpha_{ij} \neq 0$ , substituir o valor, deixando a lista intacta. encerrar;
5. Se o elemento existir, mas  $\alpha_{ij} = 0$ , destruir a posição de memória (para manter a esparsidade). Encerrar.

A economia é facilmente computada. Um número em ponto flutuante com precisão simples ocupa tipicamente 4 bytes. No caso aqui descrito, multiplicando-se por  $4 \times 876096 = 3504384$  bytes = 3.34M bytes. No caso esparsa, gastamos 8 bytes por elemento fora da diagonal e 12 por elemento da diagonal, resultando num total de  $12 \times 936 + 7 \times 8 \times 936 = 63648$  bytes = 0.062 M bytes, uma economia de aproximadamente 98%!

### Decomposição LU

A decomposição LU da matriz **A** é feita do modo tradicional, a matriz **U** ficando na parte triangular superior e a matriz **L** na inferior. Não é feito pivoteamento (foram feitos alguns testes com pivoteamento, mas relação custo X benefício mostrou-se desvantajosa).

As facilidades na decomposição LU com o armazenamento proposto residem na própria "triangularidade" do esquema. Por exemplo, se estivermos na linha-pivô  $i$ , a informação sobre os elementos logo abaixo do pivô é instantaneamente obtida apenas percorrendo-se os primeiros elementos das listas ligadas de  $i + 1$  a  $n$ . Percebe-se que a economia de tempo reside no fato de atualizarmos apenas e tão somente os elementos necessários.

A atualização das linhas segue o esquema geral da decomposição LU para matrizes cheias. O esqueleto de seu algoritmo é:

Dada **A**, quadrada e de dimensão  $n$ , fazer

1. Para  $i = 1$  até  $n-1$ , fazer
  2. Para  $j = i + 1$  até  $n$ 
    3. Determinar se a linha  $j$  precisa de atualização percorrendo a  $j$ -ésima lista ligada correspondente ao envelope esquerdo.
    4. Se existir um elemento não-nulo abaixo do pivô, atualizar a linha  $j$ , percorrendo as lista na linha pivô e atualizar na linha  $j$  se necessário. Caso contrário, voltar a 2.
    5. A matriz **U** será armazenada na parte triangular superior de **A** e **L** na inferior, calculados da maneira usual. O pivô é calculado por:

$$\alpha_{ij} = \frac{a_{ij}}{a_{ii}}$$

6. Percorrer a  $i$ -ésima lista ligada do envelope direito. Se existir o elemento  $a_{ik}$  nesta lista, atualizar:

$$a_{jk} = a_{jk} - \alpha_{ij} \times a_{ik}$$

7. Voltar para 2.

8. Voltar para 1.

### Resolvendo sistemas triangulares

A matriz  $A$  foi decomposta com a intenção de resolvermos sistemas lineares. Tendo em mãos os fatores  $L$  e  $U$ , o problema torna-se bastante simples. Basta aplicarmos os procedimentos de *forward* e *backward substitution*. Aqui aparece a grande economia de tempo em relação às primeiras versões baseadas em matrizes cheias. Fazemos as operações exatamente com os elementos necessários. A fórmula para o *backward substitution*, por exemplo, é:

$$x_k = \frac{1}{u_{kk}} \left( b_k - \sum_{j=k+1}^n u_{kj} x_j \right).$$

Mas o somatório é incompleto por causa da esparsidade<sup>6</sup> de  $U$ , e esse fato é aproveitado no momento de percorrer as listas ligadas. Acompanhemos o algoritmo de *backward substitution* (o do *forward* é análogo).

Dada  $A$ , quadrada de dimensão  $n$ , fazer

1. Para  $i = n$  até 1, fazer
  2. SOMA = 0.
  3. Enquanto houver elementos na  $i$ -ésima lista à direita, fazer:
    4. SOMA = SOMA + (valor do elemento  $u_{ij}$ ) \*  $x_j$ , onde  $j$  é o índice da coluna fornecido pela própria estrutura de dados.
  5. Voltar para 3.
  6. Calcular  $x_i = (x_i - \text{SOMA}) / u_{ii}$ .
7. Voltar para 1.

### Produto matriz/vetor

Essa é a rotina mais simples: basta fazermos o produto pelos elementos não nulos de cada envelope da matriz, mais a diagonal.

Dada  $A$ , quadrada e  $x$ , vetor, ambos de dimensão  $n$ , fazer

1. Para  $i = 1$  até  $n$ , fazer
  2. SOMA = 0.
  3. Enquanto houver elementos na  $i$ -ésima lista à esquerda, fazer: SOMA = SOMA + (valor do elemento  $a_{ij}$ ) \*  $x_j$ , onde  $j$  é o índice da coluna fornecido pela própria estrutura de dados.
  4. Enquanto houver elementos na  $i$ -ésima lista à direita, fazer: SOMA = SOMA + (valor do elemento  $a_{ij}$ ) \*  $x_j$ .
  5. SOMA = SOMA + (diagonal  $a_{ii}$ ) \*  $x_i$ .
  6.  $y_i = \text{SOMA}$ .
7. Volta para 1.

---

<sup>6</sup> Na decomposição LU aqui apresentada, não entramos em questões de *fill-in* (preenchimento), mesmo porque o padrão de distribuição dos elementos na matriz dependerá da numeração dos nós produzida pelo *PDEase*.

O vetor  $y$  retorna o produto  $Ax$ .

### Distribuição dos elementos não nulos de $A$

O *PDEase* faz a numeração dos vértices “de fora pra dentro”, o que significa que ele começa a numerar a partir da fronteira em direção ao centro do domínio. No presente caso, em que o domínio é extenso e possui muitos elementos, nós adjacentes podem ter números não relacionados com sua proximidade geométrica. Esse tipo de numeração favorece o preenchimento da matriz durante o processo de decomposição, levando em alguns casos à quase total perda de esparsidade. Duas soluções são apropriadas para esse caso: algoritmos de decomposição com pivoteamento heurístico (“parentes” do *Minimum Degree*<sup>7</sup>) ou uma reordenação da numeração do domínio (neste caso são técnicas equivalentes, só mudando o ponto em que o problema é atacado). Optamos pela segunda, porém sem nenhuma heurística em especial. Os nós são renumerados a partir de uma ordem geométrica (ordena-se primeiro em  $x$  e depois em  $y$ ). Mesmo sem uma heurística especializada o aumento de performance foi significativo, o que indica uma diminuição do fenômeno de preenchimento.

### Discretização do domínio e generalidade

O intuito do programa era o de ser o mais genérico possível, significando que, fornecido o mapa da região, simulações poderiam ser aí construídas com relativa facilidade. Essa generalidade implica em duas coisas: (i) obtenção da discretização da região e (ii) classes especializadas para o tratamento dessa discretização por elementos finitos.

O item (i), como já mencionamos, foi resolvido através do uso adequado do *PDEase* (na verdade qualquer gerador de malhas triangulares disponível pode ser usado, devendo-se respeitar apenas o formato de arquivos do *Netuno*). O item (ii) é um tanto delicado, pois as malhas podem ser de primeira ou segunda ordem, de tamanhos variados, etc. Para o tratamento de todas essas características criamos classes especializadas para tratamento da malha, dos vértices, fronteiras, etc.

Todos os dados são armazenados em arquivo. O formato conveniente desses dados para o correto funcionamento do *Netuno* é apresentado nas seções subsequentes.

### A malha

As malhas de elementos finitos possuem sempre o mesmo formato (trecho da malha de teste de 936 nós):

```
(a) [Net]
(b) 1
(c) 3
(d) 1640
(e) 1 11 18 12
    2 18 34 19
    3 19 12 18
    4 12 19 13
    5 27 15 14
```

<sup>7</sup> O *Minimum Degree* consiste em uma reordenação dos elementos da matriz de modo a diminuir o preenchimento da Decomposição de Cholesky em matrizes esparsas, definidas-positivas.

onde vemos o (a) cabeçalho (toda porção do arquivo de dados tem um cabeçalho identificador), (b) a ordem dos elementos finitos (1 ou 2), (c) o número de vértices de cada triângulo (3 ou 6), (d) o total de elementos e (e) a descrição de cada elemento. Nessa descrição constam o número global do triângulo e os números globais de cada vértice (ou vértices e pontos médios, no caso de segunda ordem).

As únicas diferenças no caso de segunda ordem estão nos itens (b), (c) e (e). O item (b) recebe o valor 2, o item (c) o valor 6 e o item (e) terá sete valores ao invés de apenas quatro (os valores extras correspondem aos pontos médios).

Vale lembrar que a alocação de memória é dinâmica, estando essa malha limitada apenas pela capacidade de memória da máquina.

### Os vértices (ou nós)

O arquivo de vértices deve ser da forma

```
(a)      [Vertex]
(b)      2
(c)      936
(d)      1   0.6   0.8
          2   0.6   0.915608
          3   0.6   1.03122
          4   0.6   1.11452
          5   0.6   1.19783
          6   0.6   1.25986
          .
          .
          .
```

novamente composto por (a) um cabeçalho, (b) o número de coordenadas (é redundante, já que são sempre duas, mas está aí já prevendo futuras implementações 3D no espaço), (c) o número de vértices e (d) o número global e as coordenadas  $x$  e  $y$  de cada vértice. A descrição dos vértices não necessita da ordem dos elementos.

### A fronteira

A descrição da fronteira obedece o formato

```
(a)      [Bounds]
(b)      1
(c)      4
(d)      20
(e)      1   1   3   1   0
          2   3   7   1   0
          3   7  12   1   0
          4  65  78   0   1
          5  78  91   0   1
          .
          .
          .
```

novamente composto por (a) um cabeçalho, (b) a ordem dos elementos, (c) sempre igual a 4 (primeira ordem) ou 5 (segunda ordem), (d) o número de partes constituintes da fronteira e (e) o número dos dois vértices que compõe aquele pedaço da fronteira e as direções da normal externa à fronteira.

Em segunda ordem, adicionamos em (e) mais um vértice correspondente ao ponto médio dos lados dos triângulos que formam a fronteira.

### O domínio padrão

Apesar do grau de generalidade obtido no programa optamos por executar todos os testes em dois domínios padronizados, um retangular representando uma área geográfica de 100m x 40m, e outro com o formato do Canal de São Sebastião (figura 13).

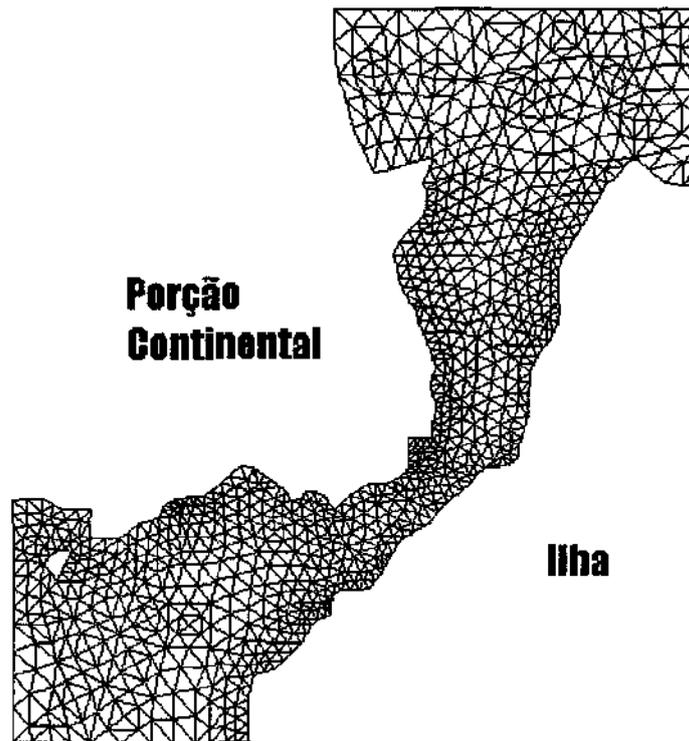


Figura 13 - Domínio padrão usado para testes.

### Cenários de simulação

**Netuno** está sendo desenvolvido para o trabalho *in-situ* no local do derrame e para a modelagem de situações mais comuns antes do acontecimento.

Aqui entra o conceito de cenário. Um cenário é todo o conjunto de elementos envolvidos em uma simulação particular. Cada cenário é composto por campos informativos, que não tem valor numérico (por exemplo, local de derrame, hora do acontecimento, etc.) e por campos de dados.

Os campos de dados são divididos em 7 partes: parâmetros do problema (difusão, decaimento), arquivos de saída, malha de elementos finitos, coordenadas dos nós, fronteiras, campo de velocidades (correntes e ventos) e fontes + condição inicial.

### Campos informativos

Nos campos informativos são relacionados dados relativos à ocorrência do derrame. Vêm nessa ordem: nome da operação, local da ocorrência, data e hora, nacionalidade do navio, a empresa responsável, a área do local afetado, velocidades máximas em  $x$  (de oeste a leste) e  $y$  (de sul a norte). Essas informações são apenas para efeito de comparação e não afetam de forma alguma as simulações.

### Parâmetros do problema

Basicamente são as constantes presentes no problema ou que ajudem a caracterizá-lo. Seguem a seqüência: ordem dos elementos usados (1 ou 2), coeficiente de difusão, decaimento, permeabilidade de fronteira ( $k$ ) e intervalo de tempo ( $\Delta t$ ).

### Arquivos de saída

Compreende dois nomes de arquivo, onde serão colocadas respectivamente a saída no padrão *Mathematica* e no padrão *Netuno*.

### Descrição do domínio

A descrição do domínio segue os formatos especificados na seção anterior.

### Correntes e ventos

O formato dos dados é absolutamente análogo ao dos nós:

```
(a) [DriveForce]
(b) 2
(c) 936
(d) 1 0.01 0.002
     2 0.01 0.002
     3 0.01 0.002
     4 0.01 0.002
     5 0.01 0.002
     .
     .
     .
```

Em (d) temos as componentes pontuais da resultante vento+corrente em cada nó,  $V_1$  e  $V_2$ . Note que a velocidade fica associada à coordenada do nó em que foi definida.

### Fontes e condição inicial

As eventuais fontes poluentes e a condição inicial são fornecidas pontualmente, nos vértices da discretização. Um arquivo exemplo é:

```
(a) [External]
(b) 2
(c) 672 0.1
     675 0.1
(d) [External]
```

(e)	3	
(f)	687	300.0
	679	300.0
	765	300.0

Em (a) está o cabeçalho (igual para fonte e condição inicial), em (b) o número de pontos em que estão definidos valores e (c) o número do vértice e o valor da fonte. Em (d), (e) e (f) os significados são os mesmos para a condição inicial.

### Finalmente, as iterações

Em (4.6) chegamos a uma forma para a aproximação das soluções de (2.4) que envolve a criação de duas matrizes, **A** e **B**, um vetor **d**, e multiplicação matriz/vetor e resolução de sistemas lineares em cada iteração no tempo. Detalharemos agora a construção dessas matrizes.

#### As sub-matrizes de rigidez

Cada elemento de **A** (ou **B**) corresponde a uma série de produtos internos entre as funções de base, ou suas derivadas. Como afirmamos anteriormente, cada  $\varphi_i$  global possui suporte compacto, abrangendo uns poucos elementos. Portanto, a primeira conclusão que extraímos desse fato é que os produtos internos podem ser calculados apenas no suporte das funções em questão. Melhor ainda: o suporte de cada  $\varphi_i$  consiste num número finito de triângulos muito bem caracterizados. Então nossa segunda conclusão é que podemos calcular as integrais localmente, vasculhando cada elemento da malha e efetuando as operações apropriadas (esse procedimento ficará mais claro na próxima seção).

A fim de utilizarmos as definições das funções de base locais criaremos uma transformação que leve do triângulo padrão ( $\hat{K}$ ), em coordenadas  $(\xi, \eta)$  para um triângulo real qualquer ( $K$ ), em coordenadas  $(x, y)$ . Essa transformação é:

$$\begin{aligned} x(\xi, \eta) &= x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta \\ y(\xi, \eta) &= y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta \end{aligned} \quad (C.1)$$

de onde verificamos que  $\varphi_i(\xi, \eta) = \varphi_i(x(\xi, \eta), y(\xi, \eta))$ , ou seja, ela leva funções globais em locais.

Antes de prosseguirmos, é conveniente expressarmos as derivadas das funções globais em função das derivadas de funções locais. Para tanto, da regra da cadeia sabemos que:

$$\frac{\partial \varphi_i}{\partial \xi} = \frac{\partial \varphi_i}{\partial x} \frac{dx}{d\xi} + \frac{\partial \varphi_i}{\partial y} \frac{dy}{d\xi} \quad \text{e} \quad \frac{\partial \varphi_i}{\partial \eta} = \frac{\partial \varphi_i}{\partial x} \frac{dx}{d\eta} + \frac{\partial \varphi_i}{\partial y} \frac{dy}{d\eta}$$

Resolvendo-se o sistema acima para  $\frac{\partial \varphi_i}{\partial x}$  e  $\frac{\partial \varphi_i}{\partial y}$  obtemos as seguintes relações:

$$\frac{\partial \varphi_i}{\partial x} = \frac{1}{|\det \mathbf{J}|} \left\{ \frac{\partial \varphi_i}{\partial \xi} \frac{dy}{d\eta} - \frac{\partial \varphi_i}{\partial \eta} \frac{d\eta}{d\xi} \right\} \quad (\text{C.2})$$

$$\frac{\partial \varphi_i}{\partial y} = \frac{1}{|\det \mathbf{J}|} \left\{ \frac{\partial \varphi_i}{\partial \eta} \frac{dx}{d\xi} - \frac{\partial \varphi_i}{\partial \xi} \frac{dx}{d\eta} \right\}$$

onde  $\mathbf{J}$  é a matriz jacobiana da transformação

$$\mathbf{J} = \begin{bmatrix} \frac{dx}{d\xi} & \frac{dx}{d\eta} \\ \frac{dy}{d\xi} & \frac{dy}{d\eta} \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix}$$

o que nos leva a  $|\det \mathbf{J}| = |(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)|$ , que é constante por elemento, dependendo apenas de suas coordenadas.

Explicitemos agora os produtos internos constantes de (4.5).

**Caso 1:**  $\iint_K \varphi_i \varphi_j \, dydx$

Usando a transformação acima, temos, diretamente do Cálculo:

$$\iint_K \varphi_i \varphi_j \, dydx = \iint_{\tilde{K}} \varphi_i \varphi_j |\det \mathbf{J}| \, d\eta d\xi = |\det \mathbf{J}| \underbrace{\iint_{\tilde{K}} \varphi_i \varphi_j \, d\eta d\xi}_{(*)}$$

**Caso 2:**  $\iint_K \nabla \varphi_i \cdot \nabla \varphi_j \, dydx$

Neste caso vamos abrir o produto interno dos gradientes e efetuar a substituição das derivadas (C.2)

$$\begin{aligned} \iint_K \nabla \varphi_i \cdot \nabla \varphi_j \, dydx &= \iint_K \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} + \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial y} \, dydx = \\ &= \iint_{\tilde{K}} \left\{ \frac{1}{|\det \mathbf{J}|} \left[ \frac{\partial \varphi_i}{\partial \xi} \frac{dy}{d\eta} - \frac{\partial \varphi_i}{\partial \eta} \frac{d\eta}{d\xi} \right] \frac{1}{|\det \mathbf{J}|} \left[ \frac{\partial \varphi_j}{\partial \xi} \frac{dy}{d\eta} - \frac{\partial \varphi_j}{\partial \eta} \frac{d\eta}{d\xi} \right] + \right. \\ &\quad \left. + \frac{1}{|\det \mathbf{J}|} \left[ \frac{\partial \varphi_i}{\partial \eta} \frac{dx}{d\xi} - \frac{\partial \varphi_i}{\partial \xi} \frac{dx}{d\eta} \right] \frac{1}{|\det \mathbf{J}|} \left[ \frac{\partial \varphi_j}{\partial \eta} \frac{dx}{d\xi} - \frac{\partial \varphi_j}{\partial \xi} \frac{dx}{d\eta} \right] \right\} |\det \mathbf{J}| \, d\eta d\xi \end{aligned}$$

que devidamente distribuída resulta em

$$\frac{1}{|\det \mathbf{J}|} \iint_{\tilde{K}} \left[ (y_3 - y_1)^2 + (x_3 - x_1)^2 \right] \frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_j}{\partial \xi} + \left[ (y_2 - y_1)^2 + (x_2 - x_1)^2 \right] \frac{\partial \varphi_i}{\partial \eta} \frac{\partial \varphi_j}{\partial \eta} -$$

$$- \left[ (y_3 - y_1)(y_2 - y_1) + (x_3 - x_1)(x_2 - x_1) \right] \left( \frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_j}{\partial \eta} + \frac{\partial \varphi_j}{\partial \xi} \frac{\partial \varphi_i}{\partial \eta} \right) d\eta d\xi$$

ou ainda

$$\iint_K \nabla \varphi_i \nabla \varphi_j dy dx =$$

$$= h_1 \underbrace{\iint_{\tilde{K}} \frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_j}{\partial \xi} d\eta d\xi}_{(*)} + h_2 \underbrace{\iint_{\tilde{K}} \frac{\partial \varphi_i}{\partial \eta} \frac{\partial \varphi_j}{\partial \eta} d\eta d\xi}_{(*)} - h_3 \underbrace{\iint_{\tilde{K}} \frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_j}{\partial \eta} + \frac{\partial \varphi_j}{\partial \xi} \frac{\partial \varphi_i}{\partial \eta} d\eta d\xi}_{(*)}$$

com

$$h_1 = \frac{1}{|\det \mathbf{J}|} \left[ (y_3 - y_1)^2 + (x_3 - x_1)^2 \right],$$

$$h_2 = \frac{1}{|\det \mathbf{J}|} \left[ (y_2 - y_1)^2 + (x_2 - x_1)^2 \right] \text{ e}$$

$$h_3 = \frac{1}{|\det \mathbf{J}|} \left[ (y_3 - y_1)(y_2 - y_1) + (x_3 - x_1)(x_2 - x_1) \right].$$

**Caso 3a:**  $\iint_K \frac{\partial \varphi_m}{\partial x} \varphi_i \varphi_j dy dx$

A partir de um procedimento análogo ao do Caso 2, escrevemos

$$\iint_K \frac{\partial \varphi_m}{\partial x} \varphi_i \varphi_j dy dx = \iint_{\tilde{K}} \frac{1}{|\det \mathbf{J}|} \left[ \frac{\partial \varphi_{\tilde{m}}}{\partial \xi} \frac{dy}{d\eta} - \frac{\partial \varphi_{\tilde{m}}}{\partial \eta} \frac{dy}{d\xi} \right] \varphi_i \varphi_j |\det \mathbf{J}| d\eta d\xi =$$

$$= (y_3 - y_1) \underbrace{\iint_{\tilde{K}} \frac{\partial \varphi_{\tilde{m}}}{\partial \xi} \varphi_i \varphi_j d\eta d\xi}_{(*)} - (y_2 - y_1) \underbrace{\iint_{\tilde{K}} \frac{\partial \varphi_{\tilde{m}}}{\partial \eta} \varphi_i \varphi_j d\eta d\xi}_{(*)}$$

**Caso 3b:**  $\iint_K \frac{\partial \varphi_m}{\partial y} \varphi_i \varphi_j dy dx$

Totalmente análogo ao Caso 3a:

$$\iint_K \frac{\partial \varphi_m}{\partial y} \varphi_i \varphi_j dy dx = \iint_{\tilde{K}} \frac{1}{|\det \mathbf{J}|} \left[ \frac{\partial \varphi_{\tilde{m}}}{\partial \eta} \frac{dx}{d\xi} - \frac{\partial \varphi_{\tilde{m}}}{\partial \xi} \frac{dx}{d\eta} \right] \varphi_i \varphi_j |\det \mathbf{J}| d\eta d\xi =$$

$$= (x_2 - x_1) \underbrace{\iint_{\tilde{K}} \frac{\partial \varphi_{\tilde{m}}}{\partial \eta} \varphi_i \varphi_j d\eta d\xi}_{(*)} - (x_3 - x_1) \underbrace{\iint_{\tilde{K}} \frac{\partial \varphi_{\tilde{m}}}{\partial \xi} \varphi_i \varphi_j d\eta d\xi}_{(*)}$$

O que é realmente importante observar, é que escrevemos os produtos internos sobre um elemento real a partir de integrais sobre o elemento padrão. Assim, as únicas integrais que realmente necessitam ser calculadas, são as marcadas com (\*). Estas integrais foram previamente calculadas com a ajuda do *Mathematica*, e seus resultados armazenados em arquivo para uso pelo programa.

### Criação das matrizes A e B e do vetor d

Finalmente temos em mãos todos os elementos necessários para a criação das matrizes e vetores envolvidos na simulação.

Afim de simplificar um pouco a notação, introduziremos as matrizes:  $\mathbf{R}$ ,  $\mathbf{S}^\xi$ ,  $\mathbf{S}^\eta$ ,  $\mathbf{S}^{\xi\eta}$ ,  $\mathbf{T}$  e  $\mathbf{U}$ , onde:

$$\begin{aligned} r_{ij} &= \iint_{\hat{k}} \varphi_i \varphi_j \, d\eta d\xi & s_{ij}^\xi &= \iint_{\hat{k}} \frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_j}{\partial \xi} \, d\eta d\xi \\ s_{ij}^\eta &= \iint_{\hat{k}} \frac{\partial \varphi_i}{\partial \eta} \frac{\partial \varphi_j}{\partial \eta} \, d\eta d\xi & s_{ij}^{\xi\eta} &= \iint_{\hat{k}} \frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_j}{\partial \eta} + \frac{\partial \varphi_i}{\partial \eta} \frac{\partial \varphi_j}{\partial \xi} \, d\eta d\xi \\ t_{ij} &= \iint_{\hat{k}} \frac{\partial \varphi_m}{\partial \xi} \varphi_i \varphi_j \, d\eta d\xi & u_{ij} &= \int \frac{\partial \varphi_m}{\partial \eta} \varphi_i \varphi_j \, d\eta d\xi \end{aligned}$$

com  $i, j = 1, \dots, NNL$ , e  $NNL =$  número de nós locais.  $NNL = 3$  para elementos de primeira ordem e  $NNL = 6$  para os de segunda ordem.

Se denominarmos **MALHA** a matriz que representa a malha de elementos finitos, a rotina para preenchimento de **A** e **B** fica:

1. Para  $k = 1$  até  $NEL$  (= número de elementos), fazer:
2. Para  $i = 1$  até  $NNL$ , fazer:
3. Para  $j = 1$  até  $NNL$ , fazer:

$$4. \quad \begin{aligned} p_1 &= x_2 - x_1 & p_2 &= x_3 - x_1 \\ p_3 &= y_2 - y_1 & p_4 &= y_3 - y_1 \end{aligned} ;$$

$$5. \quad J = |\det \mathbf{J}|;$$

$$6. \quad i_{glob} = \mathbf{MALHA}(k, i) \quad j_{glob} = \mathbf{MALHA}(k, j);$$

$$7. \quad \begin{aligned} a_{i_{glob}j_{glob}} &= a_{i_{glob}j_{glob}} + \left(1 + \sigma \frac{\Delta t}{2}\right) J r_{ij} + \\ &+ \alpha \frac{\Delta t}{2} \frac{1}{J} \left[ (p_2^2 + p_4^2) s_{ij}^\xi + (p_2^2 + p_4^2) s_{ij}^\eta - (p_1 p_2 + p_3 p_4) s_{ij}^{\xi\eta} \right] ; \end{aligned}$$

$$8. \quad \begin{aligned} b_{i_{glob}j_{glob}} &= b_{i_{glob}j_{glob}} + \left(1 - \sigma \frac{\Delta t}{2}\right) J r_{ij} - \\ &- \alpha \frac{\Delta t}{2} \frac{1}{J} \left[ (p_2^2 + p_4^2) s_{ij}^\xi + (p_2^2 + p_4^2) s_{ij}^\eta - (p_1 p_2 + p_3 p_4) s_{ij}^{\xi\eta} \right] ; \end{aligned}$$

9.  $d_{i_{glob}} = d_{i_{glob}} + J f_{i_{glob}} r_{ij}$
10. Voltar a 3;
11. Para  $m = 1$  até  $NNL$ , fazer:
  12.  $m_{glob} = \text{MALHA}(k, m)$ ;
  13.  $a_{i_{glob}j_{glob}} = a_{i_{glob}j_{glob}} + V_{1,m_{glob}}(p_4 t_{ij} - p_3 u_{ij}) + V_{2,m_{glob}}(p_1 u_{ij} - p_2 t_{ij})$ ;
  14.  $a_{i_{glob}j_{glob}} = a_{i_{glob}j_{glob}} + V_{1,m_{glob}}(p_4 t_{im} - p_3 u_{im}) + V_{2,m_{glob}}(p_1 u_{im} - p_2 t_{im})$ ;
  15.  $b_{i_{glob}j_{glob}} = b_{i_{glob}j_{glob}} - V_{1,m_{glob}}(p_4 t_{ij} - p_3 u_{ij}) - V_{2,m_{glob}}(p_1 u_{ij} - p_2 t_{ij})$ ;
  16.  $b_{i_{glob}j_{glob}} = b_{i_{glob}j_{glob}} - V_{1,m_{glob}}(p_4 t_{im} - p_3 u_{im}) - V_{2,m_{glob}}(p_1 u_{im} - p_2 t_{im})$ ;
17. Voltar a 11;
18. Incluir os termos de fronteira correspondentes a 1a. ou 2a. ordem;
19. Voltar a 2;
20. Voltar a 1;

### Iterando ...

Estando **A**, **B** e **d** prontos, basta executar as iterações:

1. Decompor a matriz **A** em  $\mathbf{A} = \mathbf{LU}$ ;
2. Fazer  $\mathbf{u} = \mathbf{u}^{(0)}$
3. Para  $t = 1$  até o número total de iterações, fazer:
  4.  $\mathbf{u} = \mathbf{Bu} + \mathbf{d}$ ;
  5. Resolver o sistema  $\mathbf{Ly} = \mathbf{u}$ ; faça  $\mathbf{u} = \mathbf{y}$ ;
  6. Resolver o sistema  $\mathbf{Uy} = \mathbf{u}$ ; faça  $\mathbf{u} = \mathbf{y}$ ;
  7. Se conveniente, guardar a solução  $\mathbf{u}$ ;
8. Voltar ao passo 3;

## Referências

1. BRENNER, S. C. e SCOTT, L. R. *The Mathematical Theory of Finite Element Methods*, Springer-Verlag, 1994.
2. BROOKS, A. N e HUGHES, T. J. R. *Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations*. Computer Methods in Applied Mechanics and Engineering, North-Holland, vol. 32, pp. 199-259, 1982.
3. CHRISTIE, D. F., GRIFFITHS, D. F., MITCHELL, A. R. e ZIENKIEWICZ, O. C. *Finite Element Methods for Second Order Differential Equations with Significant First Derivatives*. International Journal for Numerical Methods in Engineering, North-Holland, vol. 10, pp. 1389-1396, 1976.
4. CODINA, R., OÑATE, E. e CERVERA, M. *The Intrinsic Time for the Streamline Upwind/Petrov-Galerkin Formulation Using Quadratic Elements*. Computer Methods in Applied Mechanics and Engineering, no. 94, pp. 239-262, 1992.
5. CUESTA, I., GRAU, F. X. e GIRALT, F. *Numerical Simulation of Oil Spills in a Generalized Domain*. Oil & Chemical Pollution, no. 7, pp. 143-159, 1990.
6. D'AFONSECA, L. A. - colaborador - Criação de bibliotecas especializadas baseadas em elementos finitos triangulares, para a animação das soluções geradas por **Netuno**. Essas bibliotecas foram usadas no módulo de visualização (**Allen**).
7. DINIZ, G. L. *A Mudança no Habitat de Populações de Peixes: de Rio a Represa - O Modelo Matemático*. Tese de Mestrado, IMECC - UNICAMP, 1994.
8. FAY, J. A. The Spread of Oil Slicks on a Calm Sea. Oil in the Sea, Plenum Press, pp. 53-63, 1969.
9. FURTADO, V. V. *Contribuição ao Estudo da Sedimentação Atual do Canal de São Sebastião, Estado de São Paulo*. Tese de Doutorado, IG - USP, 1978.
10. HEINRICH, J. C., HUYAKORN, P. S. e ZIENKIEWICZ, O. C. *An 'Upwind' Finite Element Scheme for Two-Dimensional Convective Transport Equation*. International Journal for Numerical Methods in Engineering, North-Holland, vol. 11, pp. 131-143, 1977.
11. KELLY, D. W., NAKAZAWA, S., ZIENKIEWICZ, O. C. e HEINRICH, J. C. *A Note on Upwinding and Anisotropic Balancing Dissipation in Finite Element Approximations to Convective Diffusion Problems*. International Journal for Numerical Methods in Engineering, North-Holland, vol. 15, pp. 1705-1711, 1980.
12. LIONS, J. L. *Equations Differentielles Operationelles*, Springer, 1961.

13. MACKAY, D., BUIST, I., MASCARENHAS e R.; PATERSON, S. *Oil Spill Processes and Models*, Report of Environmental Canada, Research and Development Division - Ottawa - Canadá, 1980.
14. MARCHUK, G. I. *Mathematical Models in Environmental Problems*. North Holland, 1986.
15. MEYER, J. F. C. A. *Derrames de Petróleo em Águas Costeiras: Modelagem Matemática e Simulação Numérica*, in IIIº Simpósio de Ecossistemas da Costa Brasileira, pp. 238-247 - Serra Negra - São Paulo, 1993.
16. MEYER, J. F. C. A., MONTE, M. S. & GUIMARÃES, A. F. *Modelagem e Simulação Numérica de Derrames de Petróleo em Águas Costeiras*, Biomatemática, vol. 2, pp. 156-172, 1992.
17. PAIVA, I. A., DE OLIVEIRA, A. L. B. - colaboradores - Distribuição de velocidades no mapa discretizado do Canal de São Sebastião, de acordo com o trabalho de FURTADO, supra-citado.
18. PETROBRÁS, Petróleo Brasileiro S. A. *Conheça o Terminal Almirante Barroso - TEBAR*, Serviço de Comunicação Social (SERCOM), 1992.
19. POFFO, I. R. F., NAKASAKI, A., AVENTURATO, H., MEYER, J. F. C. A. e CANTÃO, R. F. *Estudo dos Deslocamentos de Manchas de Óleo no Canal de São Sebastião - SP, Como Instrumento para Planejamento de Operações Emergenciais*, in VII Colacmar - Congresso Latinoamericano de Ciencias del Mar, Instituto Oceanográfico da USP - Santos - SP, 1997.