



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



Modelagem matemática e simulação computacional da presença de materiais impactantes tóxicos em casos de dinâmica populacional com competição inter e intra-específica

Marcos Marreiro Salvatierra

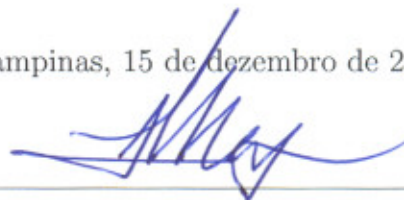
Mestrado em Matemática Aplicada - Campinas - SP

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

Modelagem matemática e simulação computacional da presença de materiais impactantes tóxicos em casos de dinâmica populacional com competição inter e intra-específica

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por **Marcos Marreiro Salvatierra** e aprovada pela comissão julgadora.

Campinas, 15 de dezembro de 2005.



Prof. Dr. João Frederico da C. A. Meyer
Orientador

Banca examinadora:

Prof. Dr. João Frederico da Costa Azevedo Meyer (IMECC/UNICAMP)

Prof. Dr. Laécio Carvalho de Barros (IMECC/UNICAMP)

Profa. Dra. Renata Cristina Sossae (UNISAL)

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

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

Bibliotecário: Maria Júlia Milani Rodrigues – CRB8a / 2116

Salvatierra, Marcos Marreiro

Sa38m Modelagem matemática e simulação computacional da presença de materiais impactantes tóxicos em casos de dinâmica populacional com competição inter e intra-específica / Marcos Marreiro Salvatierra -- Campinas, [S.P. :s.n.], 2005.

Orientador : João Frederico da Costa Azevedo Meyer

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

1. Equações diferenciais parciais. 2. Método dos elementos finitos. 3. Ecologia. I. Meyer, João Frederico da Costa Azevedo. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Título em inglês: Mathematical modeling and computational simulation of the presence of toxic impactant materials in cases of populational dynamics with inter- and intra-specific competition.

Palavras-chave em inglês (Keywords): 1. Partial differential equations. 2. Finite element method. 3. Ecology.

Área de concentração: Ecologia Matemática

Titulação: Mestre em Matemática Aplicada

Banca examinadora: Prof. Dr. João Frederico da Costa Azevedo Meyer (IMECC-UNICAMP)
Prof. Dr. Laécio Carvalho de Barros (IMECC-UNICAMP)
Profa. Dra. Renata Cristina Sossae (UNISAL)

Data da defesa: 15-12-2005

Dissertação de Mestrado defendida em 15 de dezembro de 2005 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



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



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



Prof. (a). Dr (a). RENATA CRISTINA SOSSAE

A Deus e à minha família.

Agradecimentos

Ao mestre Joni pela paciência e disposição em sempre me motivar e ajudar.

Aos professores da banca examinadora pelas críticas e sugestões feitas visando o aperfeiçoamento deste trabalho.

Aos professores que ministraram disciplinas ao longo do curso, pela contribuição no meu amadurecimento intelectual e pessoal.

Aos funcionários do instituto, pela eficácia nos trabalhos de regulamentação e burocracia.

Aos meus amigos e colegas de estudo e lazer, pelo companheirismo ao longo desta jornada.

Enfim, a todos que direta ou indiretamente contribuíram para a conclusão deste trabalho, meus sinceros agradecimentos.

Resumo

A proposta deste trabalho é criar um modelo para descrever computacionalmente o convívio entre duas espécies competidoras com características de migração na presença de um material impactante tóxico. As equações a serem utilizadas deverão incluir os fenômenos de dispersão populacional, processos migratórios, dinâmicas populacionais densidade-dependentes e efeitos tóxicos de um material impactante evoluindo no meio, provocando um decaimento proporcional. Recorrendo a um instrumental consagrado, embora com desenvolvimento relativamente recente, será usado um sistema clássico do tipo Lotka-Volterra (conseqüentemente não-linear) combinado a Equações Diferenciais Parciais de Dispersão-Migração. O primeiro passo é a formulação variacional discretizada deste sistema visando o uso de Elementos Finitos combinados a um método de Crank-Nicolson. Em segundo lugar, virá a formulação de um algoritmo (conjuntamente com sua programação em ambiente MATLAB) que aproxima as soluções discretas relativas a cada população em cada ponto e ao longo do intervalo de tempo considerado nas simulações. Por fim, serão obtidas saídas gráficas úteis dos pontos de vista quantitativo e qualitativo para uso em conjunto com especialistas de áreas de ecologia e meio ambiente na avaliação e na calibração de modelos e programas, bem como no estudo de estratégias de preservação, impacto e recuperação de ambientes.

Abstract

The purpose of this work is to create a model to computationally describe the co-existence of two competing species with migration features in the presence of a toxic impactant material. The equations must include the phenomena of populational dispersion, migratory processes, density-dependent populational dynamics and toxic effects of the evolutive presence of an impactant material developing in the environment, generating a proportional decrease in both populations. Resorting to well-established, although relatively recent, mathematical instruments, a Lotka-Volterra type (and consequently non-linear) system, including characteristics of a Migration-Dispersion PDE. The first step is the discrete variational formulation of this system aiming for the use of the Finite Element Method together with a Crank-Nicolson Method. Second, the formulation of an algorithm (together with a programme in MATLAB environment) that approximates the relative discrete solutions to each population in each point and along of the time interval considered in the simulations. Lastly, useful graphics will be obtained of the quantitative and qualitative viewpoints for use with specialists of the fields of ecology and environment and in the evaluation and calibration of models and programmes.

Sumário

Introdução	1
1 A dispersão de um poluente no meio	5
1.1 O modelo matemático	5
1.2 Simplificações do modelo	6
1.3 Formulação variacional	7
1.4 Aproximação da solução	8
1.5 Implementação computacional	10
2 Resultados das simulações – Parte I	13
3 O convívio entre duas espécies competidoras sob efeito do poluente	18
3.1 O modelo matemático	19
3.2 Simplificações do modelo	20
3.3 Formulação variacional	21
3.4 Aproximação da solução	22
3.5 Implementação computacional	25
4 Resultados das simulações – Parte II	27
Conclusões	32
Apêndice	34
Bibliografia	51

Lista de Figuras

1	Imagem da área central dos Esteros de Iberá. No centro, à direita, a lagoa Iberá. Fonte: site do projeto <i>The sustainable management of wetland resources</i>	2
2	Duas espécies-chave do ecossistema da lagoa Iberá	3
3	Plantação de arroz próxima à margem da lagoa Iberá	3
1.1	Discretização da lagoa Iberá através de elementos finitos triangulares de segunda ordem usada nas simulações: 6853 nós e 3297 elementos	12
2.1	Determinação das fronteiras	13
2.2	O campo de velocidades da lagoa	14
2.3	Distribuição inicial da concentração do poluente na lagoa	15
2.4	Distribuição da concentração do poluente na lagoa após 1000 iterações . . .	15
2.5	Distribuição da concentração do poluente na lagoa após 2000 iterações . . .	16
2.6	Distribuição final da concentração do poluente na lagoa (após 3000 iterações)	16
2.7	Localização dos nós 6259, 3638 e 369 e o acompanhamento do comportamento evolutivo do poluente nestes pontos	17
3.1	Diagrama ilustrativo da cadeia trófica do ecossistema da lagoa Iberá (site do projeto <i>The sustainable management of wetland resources</i>)	20
4.1	Distribuição das densidades populacionais das duas espécies na lagoa após 1000 iterações	28
4.2	Distribuição das densidades populacionais das duas espécies na lagoa após 2000 iterações	28
4.3	Distribuição final da densidade populacional da espécie 1 (<i>chajá</i>) na lagoa (após 3000 iterações)	29
4.4	Visualização 3D da distribuição final da densidade populacional da espécie 1 (<i>chajá</i>) na lagoa (após 3000 iterações)	30

4.5	Visualização 3D da distribuição final da densidade populacional da espécie 2 (jacaré) na lagoa (após 3000 iterações)	30
4.6	Localização do nós 6259, 3638 e 369 e o acompanhamento do comportamento evolutivo das densidades populacionais das duas espécies nestes pontos	31
4.7	Triângulo genérico	34
4.8	Triângulo padrão	34

Lista de Tabelas

2.1	Parâmetros utilizados na simulação da dispersão de um poluente na lagoa .	13
4.1	Parâmetros utilizados na simulação do convívio entre duas espécies competidoras sob efeito do poluente	27

Introdução

Em situações já registradas e analisadas dos efeitos intermediários e permanentes de grupos de populações coexistentes em determinados ecossistemas, tem-se, muitas vezes, a prevalência final de uma delas em vez do retorno à convivência entre ambas. Este resultado depende de muitos fatores, mas, de início, os principais destes são o tipo de competição e a rapidez de recuperação de uma ou algumas das espécies envolvidas (ver SOSSAE, 2003, PREGNOLATTO, 2003).

Riscos ambientais e dificuldades de controle de experiências *in situ* sugerem fortemente que ensaios computacionais prévios sustentados em modelagem matemática de cuidada formulação sejam usados em primeiras aproximações, evitando os efeitos potencialmente letais e tóxicos de muitos possíveis casos (COSNER e outros, 2000). Estes casos de experiências a serem evitadas são os de simulação (mesmo controlada) de efeitos tóxicos sobre populações em determinado convívio. Estas simulações envolvem um risco nem sempre muito bem calculado: são casos em que é preferível a simulação computacional apoiada em modelagens matemáticas e discretizações confiáveis.

Em várias situações, as dimensões de corpos aquáticos introduzem dificuldades de estudos, de análises e de realização de experimentos em quantidades que justificam a adoção de simulações computacionais nas primeiras aproximações, reduzindo drasticamente as necessidades de experimento no campo, com seus efeitos a curto, médio e longo prazos. Em conjunto com esta dificuldade, existe a necessidade de se prevenir o meio ambiente de riscos reais de efeitos tóxicos provenientes de atividades antrópicas, sejam elas ligadas a projetos agroindustriais, sejam resultantes de explorações, sejam elas pela presença de comunidades e seus efluentes normais. Embora já haja um volume razoavelmente desenvolvido de estudos em termos da presença evolutiva de poluentes (*cf* MISTRO, 1992, BERNARDES, 1998, CANTÃO, 1998, DINIZ, 2003, OLIVEIRA, 2003, SOSSAE, 2003, VÁSQUEZ, 2005, por exemplo), há menos resultados efetivos em termos da modelagem matemática dos efeitos evolutivos desse tipo de material impactante sobre populações que interagem ao mesmo tempo em que são atingidas pelos ditos efeitos.

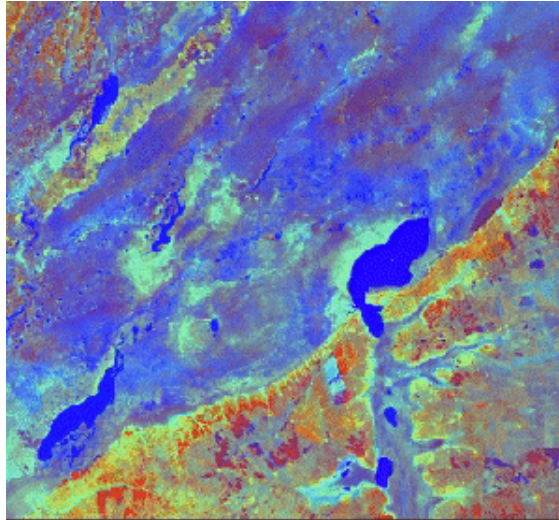


Figura 1: Imagem da área central dos Esteros de Iberá. No centro, à direita, a lagoa Iberá. Fonte: site do projeto *The sustainable management of wetland resources*

Este trabalho visa, então, recorrer a sistemas não-lineares de Equações Diferenciais Parciais que modelem tanto a dispersão populacional com efeitos outros como a migração, por exemplo, em conjunto com dinâmicas populacionais de tipo logístico genérico incorporando, também, as ações interespecíficas de tipo clássico, ou seja, na linha de Lotka-Volterra (ver, por exemplo, MEYER e SOSSAE, 1999).

As equações dos sistemas propostos permitem a inclusão de efeitos tóxicos letais variáveis de um poluente no meio, poluente este cuja presença evolui no tempo em função não apenas de difusão efetiva (no sentido de MARCHUK, 1986, ou de OKUBO, 1980), mas também do transporte advectivo – no caso que se deseja estudar, este transporte é devido à correnteza em corpos aquáticos, e, além disso, as populações estudadas, em função de ambientes de competição intra-específica, podem sofrer efeitos permanentes, visto que os equilíbrios de convivência competitiva são, reconhecidamente, equilíbrios de frágil estabilidade – uma modelagem que reúne idéias já clássicas em termos de convívio entre espécies (EDELSTEIN-KESHET, 1988, MURRAY, 1989).

O cenário em que serão feitas as simulações deste trabalho é a lagoa Iberá, situada no nordeste da Argentina. O grupo de Biomatemática do IMECC-UNICAMP, em conjunto com outros centros de pesquisa como os da Università degli Studi di Siena (Itália), da Universidad del Salvador (Argentina), da Universidad Nacional del Centro de la Provincia de Buenos Aires (Argentina), da Universidad de Cádiz (Espanha), da Universidade de Aveiro (Portugal), da University of York (Reino Unido) e da Universidade Federal do Rio

Grande do Sul (Brasil), já vem desenvolvendo estudos sobre esta região, obtendo resultados significativos. Na definição da cadeia trófica deste meio (MOMO e GANTES, 1999), duas espécies-chave – o “chajá” (*Chauna torquata*) e o jacaré (*Caiman yacare*) – estão no topo, competindo entre si por presas de pequeno porte (basicamente peixes e rãs). O material impactante tóxico que será considerado nas simulações é proveniente das culturas de arroz presentes nas regiões vizinhas da lagoa. Os agrotóxicos são as principais fontes de poluição naquele meio.



Figura 2: Duas espécies-chave do ecossistema da lagoa Iberá

No capítulo 1 apresenta-se um modelo para a dispersão de um poluente no meio. Resultados de simulações computacionais são mostrados no capítulo 2 para se observar a sua evolução no decorrer do tempo, prevendo-se assim as regiões onde as populações das duas espécies-chave serão afetadas por ele, bem como as tendências dos efeitos posteriores.



Figura 3: Plantação de arroz próxima à margem da lagoa Iberá

No capítulo 3 propõe-se um modelo que descreve o convívio entre as duas espécies-chave incluindo os efeitos tóxicos do material impactante evoluindo no meio, provocando

um decaimento proporcional.

No capítulo 4 são apresentados os resultados das simulações computacionais feitas com base na modelagem proposta no capítulo 3, seguidos de análises e conclusões.

Capítulo 1

A dispersão de um poluente no meio

Modelos matemáticos para dispersão de poluentes em diferentes meios já estão sendo bastante desenvolvidos e difundidos atualmente. Sobre a lagoa Iberá, o meio onde serão feitas as aplicações do presente estudo, BERNARDES (1998) obteve primeiras aproximações para a simulação do comportamento evolutivo de poluentes aquáticos de superfície, poluentes estes cuja introdução na lagoa se deve à presença das culturas de soja e arroz nas suas vizinhanças e é dada através do uso de pesticidas e repelentes para aves e da queima de combustíveis e solventes à base de petróleo, entre outros.

Sobre a mesma região, DINIZ (2003) também tratou da dispersão de poluentes mas desta vez dando enfoque ao sistema ar-água.

Neste capítulo será feita uma modelagem matemática para a dispersão de um poluente na lagoa, proveniente de plantações de arroz numa situação tal como mostrou a figura 3.

1.1 O modelo matemático

A equação que descreve esse tipo de modelagem é a já clássica Equação de Difusão-Advecção. Neste modelo, são considerados basicamente os seguintes fenômenos:

- A difusão, ou seja, espalhamento natural do poluente em contato com o meio. A dinâmica difusiva resulta de movimentos aleatórios micro e macroscópicos. Neste trabalho será considerada a difusão efetiva, no sentido de MARCHUK (1986), em que são levadas em conta a difusão molecular, que deriva do movimento browniano das moléculas, e a difusão turbulenta, derivada do espalhamento macroscópico de tensão superficial;
- A advecção, ou seja, o movimento provocado pela movimentação do próprio meio

(correntezas);

- O decaimento, fenômeno que reúne alterações sofridas pelas moléculas das substâncias componentes do produto tóxico ao longo do tempo, ocasionando perda de massa e consequentemente sua exclusão do meio;
- Fontes poluentes, introduzindo o material impactante no meio.

Nestas situações, a equação que descreve tais fenômenos para a concentração do poluente, $C = C(x, y; t)$ com $(x, y) \in \Omega \subset \mathbb{R}^2$ e $t \in (0, T]$, é dada por

$$\frac{\partial C}{\partial t} - \operatorname{div}(\alpha_c \nabla C) + \operatorname{div}(\mathbb{V}C) + \sigma_c C = f \quad (1.1)$$

onde

- $\alpha_c = \alpha_c(x, y; t)$ representa a difusibilidade do poluente no meio;
- $\mathbb{V} = (V_1(x, y; t), V_2(x, y; t))$ representa o campo de velocidades da circulação local, com $\operatorname{div}(\mathbb{V}) = 0$;
- $\sigma_c = \sigma_c(x, y; t)$ representa o decaimento do poluente no meio;
- $f = f(x, y; t)$ representa as fontes poluentes.

Convém observar que o gradiente e o divergente são considerados como operadores somente sobre as variáveis espaciais, e que as condições inicial e de contorno serão explicitadas mais adiante.

1.2 Simplificações do modelo

Conforme já foi dito antes, a proposta de aplicação deste modelo é a simulação do comportamento evolutivo de um material impactante tóxico proveniente das culturas de arroz próximos à lagoa Iberá. O uso de agrotóxicos nas plantações é feito em períodos sazonais, e então considerando o intervalo do domínio temporal $(0, T]$ contido entre uma e outra época de lavoura, é razoável admitir $f = 0$.

O campo de velocidades \mathbb{V} será considerado independente do tempo, não levando em conta, assim, perturbações locais (no tempo) que possa haver na circulação da água, sejam elas causadas por chuvas, rajadas de vento, etc., mas descrevendo globalmente o fenômeno advectivo.

Os parâmetros α_c e σ_c podem até (por algum tempo) depender fortemente da concentração C , porém, devido ao intervalo de tempo que será considerado nas simulações e à proposta de tratamento numérico para este modelo (um sistema associado linear), é aceitável usar valores médios para α_c e σ_c , tornando-os constantes.

Finalmente, completando a equação (1.1) com as condições de contorno* $C|_{\Gamma_0} = 0$ e $\frac{\partial C}{\partial \eta}|_{\Gamma_1} = 0$, o modelo torna-se

$$\left\{ \begin{array}{l} \frac{\partial C}{\partial t} - \alpha_c \Delta C + \mathbb{V} \cdot \nabla C + \sigma_c C = 0, \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in (0, T] \\ C(x, y; 0) = C_0(x, y), \quad \forall (x, y) \in \Omega \\ C|_{\Gamma_0} = 0, \quad \forall t \in (0, T] \\ \frac{\partial C}{\partial \eta}|_{\Gamma_1} = 0, \quad \forall t \in (0, T] \end{array} \right. \quad (1.2)$$

1.3 Formulação variacional

Devido às dificuldades teóricas e práticas na busca de uma solução do problema (1.2), o modelo será descrito na sua formulação fraca, no sentido de distribuições, pois assim pode-se exigir menos regularidade da solução. Para isto, definem-se os produtos internos em $L^2(\Omega)$:

- $(f|g) = \int_{\Omega} f(x, y)g(x, y)ds$
- $(\nabla f||\nabla g) = \int_{\Omega} \nabla f(x, y) \cdot \nabla g(x, y)ds$

Assim, o problema torna-se encontrar

$$C \in \mathcal{S} = \{C \in L^2((0, T), H^1(\Omega)); C = 0 \text{ em } \Gamma_0 \text{ e } \frac{\partial C}{\partial t} \in L^2(\Omega) \forall t \in (0, T]\}$$

tal que

$$\left(\frac{\partial C}{\partial t} | \nu \right) - \alpha_c (\Delta C | \nu) + (\mathbb{V} \cdot \nabla C | \nu) + \sigma_c (C | \nu) = 0 \quad \forall \nu \in \mathcal{V} \quad (1.3)$$

*As partes da fronteira Γ_1 e Γ_2 serão explicitadas no capítulo 2.

onde $\mathcal{V} = \{\nu \in H^1(\Omega); \nu = 0 \text{ em } \Gamma_0\}$.

O operador laplaciano que figura no segundo termo desta última equação parece introduzir uma incoerência na escolha do espaço \mathcal{S} . Para contornar essa situação, aplica-se o Teorema de Green neste termo. Então, fazendo uma expansão do terceiro termo e usando as condições de contorno, tem-se

$$\left(\frac{\partial C}{\partial t}|\nu\right) + \alpha_c(\nabla C||\nabla\nu) + \left(V_1\frac{\partial C}{\partial x}|\nu\right) + \left(V_2\frac{\partial C}{\partial y}|\nu\right) + \sigma_c(C|\nu) = 0 \quad \forall \nu \in \mathcal{V} \quad (1.4)$$

Resultados sobre existência e unicidade de solução para este tipo de problema podem ser encontrados nos trabalhos de MISTRO (1992), BERNARDES (1998) e CANTÃO (1998), entre outros.

1.4 Aproximação da solução

Na tarefa da construção de uma solução aproximada da equação (1.4), serão trabalhadas primeiramente as variáveis espaciais, pelo método de Galerkin. Para isto, seja $\mathcal{V}_h \subset \mathcal{V}$ um subespaço de dimensão N e seja $\mathcal{B} = \{\varphi_1, \varphi_2, \dots, \varphi_N\}$ uma base para esse subespaço. Visando construir uma aproximação da solução nesse espaço, separam-se as suas variáveis obtendo

$$C(x, y; t) \cong C_h(x, y; t) = \sum_{j=1}^N c_j(t)\varphi_j(x, y)$$

$$\frac{\partial C_h}{\partial t} = \sum_{j=1}^N \frac{dc_j}{dt}\varphi_j, \quad \frac{\partial C_h}{\partial x} = \sum_{j=1}^N c_j \frac{\partial \varphi_j}{\partial x} \text{ e } \frac{\partial C_h}{\partial y} = \sum_{j=1}^N c_j \frac{\partial \varphi_j}{\partial y}$$

Usando estas aproximações na formulação variacional (1.4) tem-se

$$\left(\sum_{j=1}^N \frac{dc_j}{dt}\varphi_j|\nu_h\right) + \alpha_c\left(\nabla\sum_{j=1}^N c_j\varphi_j||\nabla\nu_h\right) + \left(V_1\sum_{j=1}^N c_j\frac{\partial\varphi_j}{\partial x}|\nu_h\right) + \left(V_2\sum_{j=1}^N c_j\frac{\partial\varphi_j}{\partial y}|\nu_h\right) + \sigma_c\left(\sum_{j=1}^N c_j\varphi_j|\nu_h\right) = 0 \quad \forall \nu_h \in \mathcal{V}_h$$

Como esta expressão deve valer para todo $\nu_h \in \mathcal{V}_h$, basta escrevê-la para todo elemento de \mathcal{B} . Então, retirando os coeficientes c_j dos produtos internos, pois eles não dependem das variáveis espaciais, tem-se

$$\begin{aligned}
& \sum_{j=1}^N \frac{dc_j}{dt}(\varphi_j|\varphi_i) + \alpha_c \sum_{j=1}^N c_j(\nabla\varphi_j||\nabla\varphi_i) + \sum_{j=1}^N c_j \left(V_1 \frac{\partial\varphi_j}{\partial x} | \varphi_i \right) + \sum_{j=1}^N c_j \left(V_2 \frac{\partial\varphi_j}{\partial y} | \varphi_i \right) + \\
& + \sigma_c \sum_{j=1}^N c_j(\varphi_j|\varphi_i) = 0 \quad \forall i = 1, \dots, N
\end{aligned} \tag{1.5}$$

O campo de velocidades $\mathbb{V} = (V_1(x, y), V_2(x, y))$ do meio em questão foi objeto de estudo de CANTÃO e D'AFONSECA (1998). Seus valores pontuais aproximados foram obtidos através da resolução numérica da Equação de Stokes (ver, também, SAAVEDRA VÁSQUEZ)

$$-div(\nabla\mathbb{V}) + \nabla\mathcal{P} = g,$$

em que \mathcal{P} representa a pressão e g uma perturbação que pode ser nula. Interpolando esses valores através das funções da base \mathcal{B} tem-se

$$V_1(x, y) = \sum_{k=1}^N V_{1k} \varphi_k(x, y) \text{ e } V_2(x, y) = \sum_{m=1}^N V_{2m} \varphi_m(x, y).$$

Assim, o sistema (1.5) torna-se

$$\begin{aligned}
& \sum_{j=1}^N \frac{dc_j}{dt}(\varphi_j|\varphi_i) + \alpha_c \sum_{j=1}^N c_j(\nabla\varphi_j||\nabla\varphi_i) + \sum_{j=1}^N c_j \left(\sum_{k=1}^N V_{1k} \varphi_k \frac{\partial\varphi_j}{\partial x} | \varphi_i \right) + \\
& + \sum_{j=1}^N c_j \left(\sum_{m=1}^N V_{2m} \varphi_m \frac{\partial\varphi_j}{\partial y} | \varphi_i \right) + \sigma_c \sum_{j=1}^N c_j(\varphi_j|\varphi_i) = 0 \quad \forall i = 1, \dots, N
\end{aligned} \tag{1.6}$$

Agora o objetivo é encontrar uma aproximação temporal. Para este fim, será usado o método de Crank-Nicolson, de onde obtém-se

$$\frac{dc_j^{(n+1/2)}}{dt} \cong \frac{c_j^{(n+1)} - c_j^{(n)}}{\Delta t} \text{ e } c_j^{(n+1/2)} \cong \frac{c_j^{(n+1)} + c_j^{(n)}}{2}$$

onde Δt representa o passo no tempo. Usando-se essas discretizações no sistema (1.6) e arranjando-se os termos convenientemente, tem-se

$$\begin{aligned}
& \sum_{j=1}^N c_j^{(n+1)} \left\{ \left[1 + \sigma_c \frac{\Delta t}{2} \right] (\varphi_i | \varphi_j) + \frac{\Delta t}{2} \left[\alpha_c (\nabla \varphi_i | | \nabla \varphi_j) + \sum_{k=1}^N V_{1_k} \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \varphi_k \right) + \right. \right. \\
& \left. \left. + \sum_{m=1}^N V_{2_m} \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \varphi_m \right) \right] \right\} = \\
& = \sum_{j=1}^N c_j^{(n)} \left\{ \left[1 - \sigma_c \frac{\Delta t}{2} \right] (\varphi_i | \varphi_j) - \frac{\Delta t}{2} \left[\alpha_c (\nabla \varphi_i | | \nabla \varphi_j) + \sum_{k=1}^N V_{1_k} \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \varphi_k \right) + \right. \right. \\
& \left. \left. + \sum_{m=1}^N V_{2_m} \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \varphi_m \right) \right] \right\} \quad \forall i = 1, \dots, N
\end{aligned} \tag{1.7}$$

que na forma matricial escreve-se $\mathbf{Ac}^{(n+1)} = \mathbf{Bc}^{(n)}$, em que

$$\begin{aligned}
a_{ij} &= \left[1 + \sigma_c \frac{\Delta t}{2} \right] (\varphi_i | \varphi_j) + \frac{\Delta t}{2} \left[\alpha_c (\nabla \varphi_i | | \nabla \varphi_j) + \sum_{k=1}^N V_{1_k} \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \varphi_k \right) + \right. \\
& \left. + \sum_{m=1}^N V_{2_m} \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \varphi_m \right) \right] \\
b_{ij} &= \left[1 - \sigma_c \frac{\Delta t}{2} \right] (\varphi_i | \varphi_j) - \frac{\Delta t}{2} \left[\alpha_c (\nabla \varphi_i | | \nabla \varphi_j) + \sum_{k=1}^N V_{1_k} \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \varphi_k \right) + \right. \\
& \left. + \sum_{m=1}^N V_{2_m} \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \varphi_m \right) \right]
\end{aligned}$$

com $i, j = 1, \dots, N$ para cada n . Este sistema será resolvido iterativamente no tempo, a partir da condição inicial $\mathbf{c}^{(0)}$ pontualmente fornecida ou dada implicitamente por

$$\sum_{j=1}^N c_j^{(0)} (\varphi_j | \varphi_i) = (C_0 | \varphi_i)$$

1.5 Implementação computacional

Na implementação computacional do sistema (1.7), o domínio espacial foi discretizado através de elementos finitos triangulares de segunda ordem. A malha da discretização é mesma usada em SOSSAE (2003), obtida através do software *Triangle*.

As funções

$$\begin{aligned}\psi_1 &= \omega(2\omega - 1), & \psi_2 &= \xi(2\xi - 1), & \psi_3 &= \eta(2\eta - 1), \\ \psi_4 &= 4\xi\eta, & \psi_5 &= 4\eta\omega, & \psi_6 &= 4\xi\omega,\end{aligned}$$

onde $\omega = 1 - \xi - \eta$, foram usadas como “*shape functions*” no triângulo padrão para, a partir delas, serem construídas as sub-matrizes de rigidez. Uma proposta algorítmica em ambiente *Matlab* para a construção analítica dessas matrizes se encontra no apêndice.

A escolha do método de Crank-Nicolson nas aproximações temporais se deve à sua estabilidade numérica e à sua margem de erro, que é da ordem de $(\Delta t)^2$. Maiores detalhes sobre os métodos e critérios numéricos usados nesta seção e na anterior podem ser encontrados em CARNAHAN *et al.* (1969) e KARDESTUNCER e NORRIE (1987).

Finalmente, e como no início efetivo do próximo capítulo, deve-se pontuar que o domínio em estudo é de baixa circulação, produzindo efeitos residuais a médio e longo prazos.

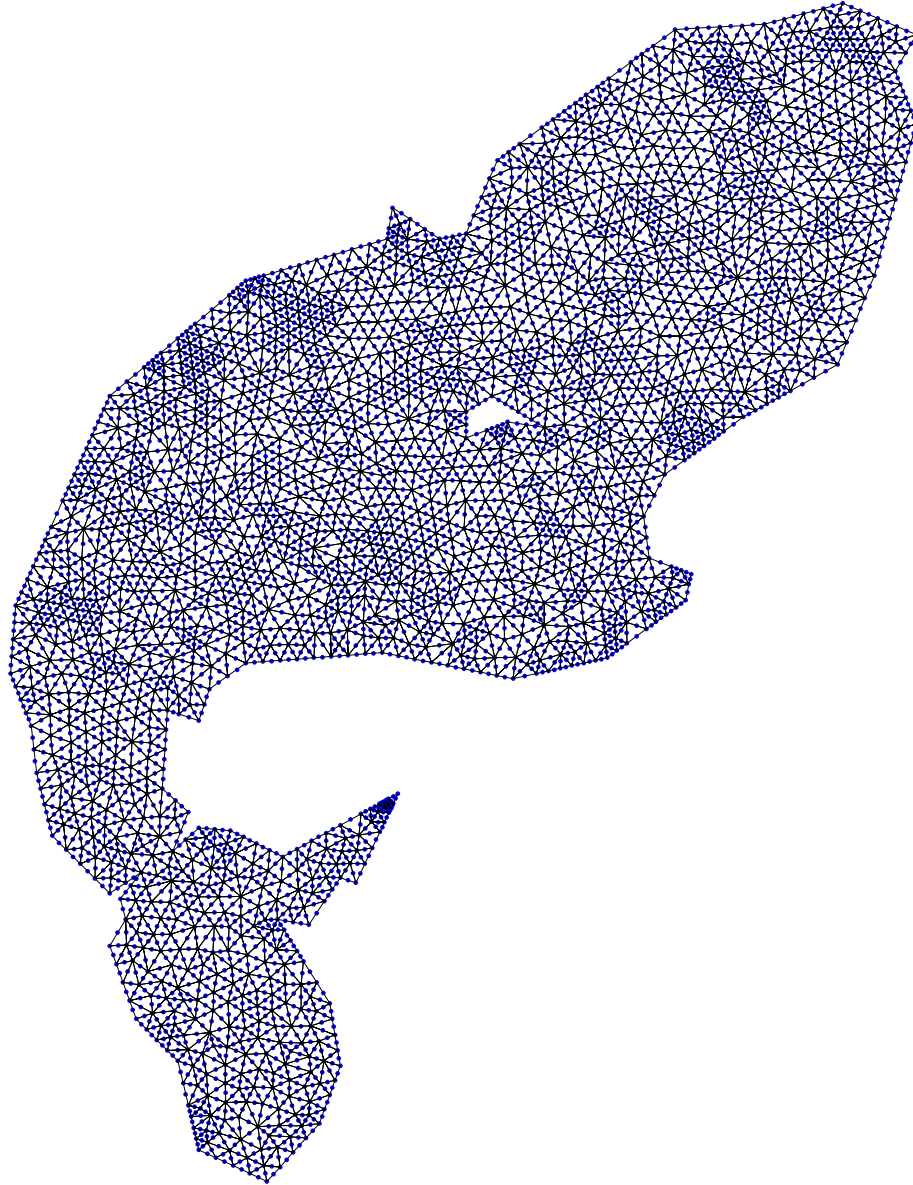


Figura 1.1: Discretização da lagoa Iberá através de elementos finitos triangulares de segunda ordem usada nas simulações: 6853 nós e 3297 elementos

Capítulo 2

Resultados das simulações – Parte I

Os parâmetros utilizados nas simulações deste trabalho foram baseados em SOSSAE (2003), porém fizeram-se algumas modificações com o intuito de serem sugeridas outras prováveis situações.

Parâmetros	Valores
α_c	0.75
σ_c	0.001
Δt	0.25

Tabela 2.1: Parâmetros utilizados na simulação da dispersão de um poluente na lagoa

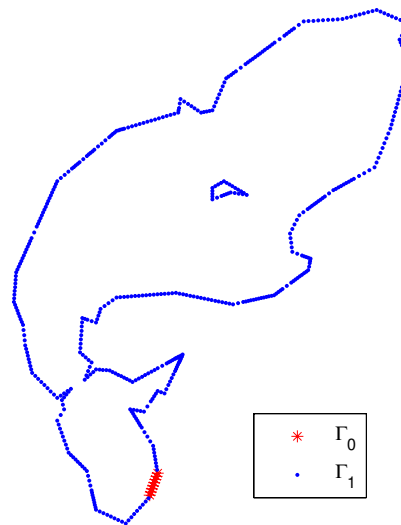


Figura 2.1: Determinação das fronteiras

A figura 2.2 fornece uma visualização do campo vetorial que simula o mapa de circulação da lagoa obtido por CANTÃO e D'AFONSECA (1998).

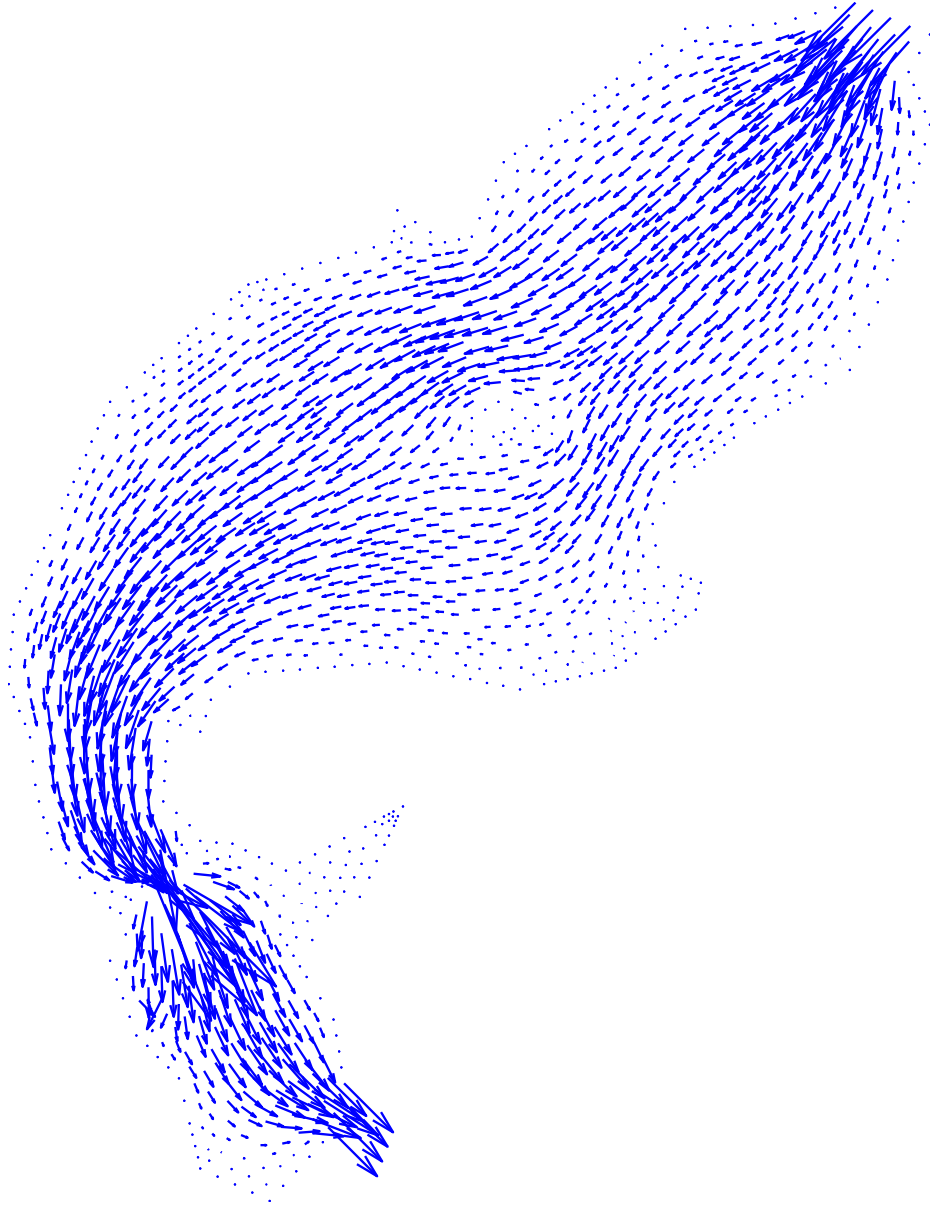


Figura 2.2: O campo de velocidades da lagoa

Para simular uma distribuição inicial da concentração do polente na lagoa, foram escolhidos alguns nós próximos à uma porção da margem, sugerindo uma descarga do material impactante tóxico advinda das plantações de arroz nos seus arredores, como mostra a figura 2.3.

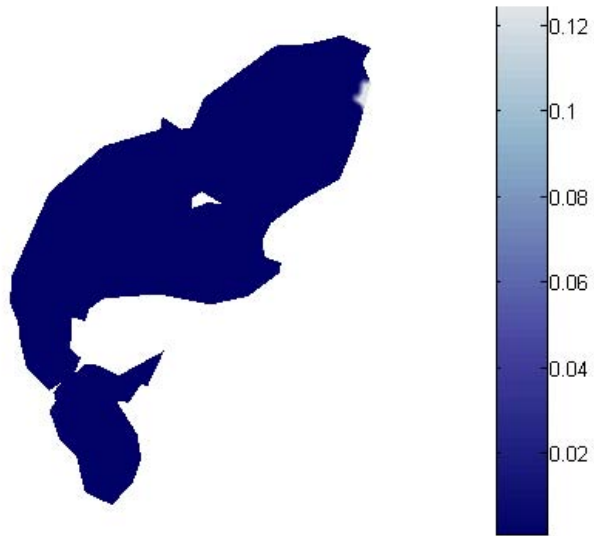


Figura 2.3: Distribuição inicial da concentração do polente na lagoa

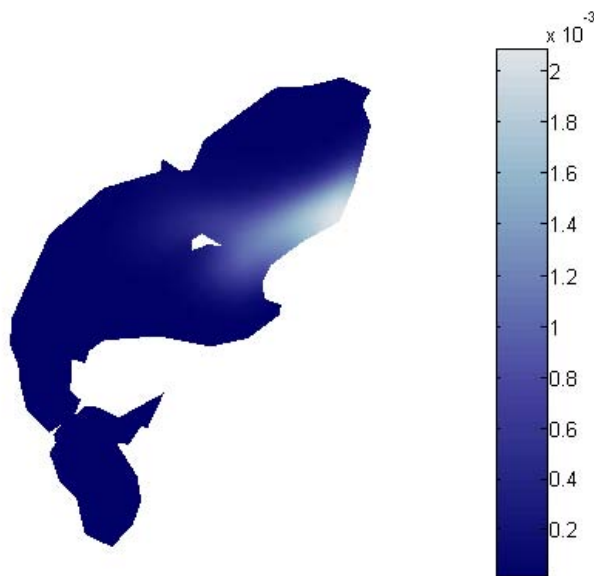


Figura 2.4: Distribuição da concentração do polente na lagoa após 1000 iterações

Nas figuras 2.4 e 2.5 pode-se perceber claramente o efeito difusivo-advectivo na evolução do material impactante. Efetuadas todas as iterações, obtém-se um cenário para a distribuição final da concentração do poluente na lagoa, como mostra a figura 2.6.

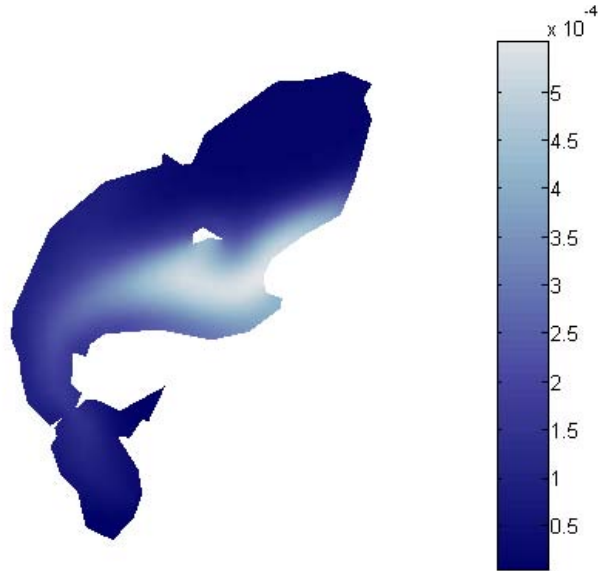


Figura 2.5: Distribuição da concentração do poluente na lagoa após 2000 iterações

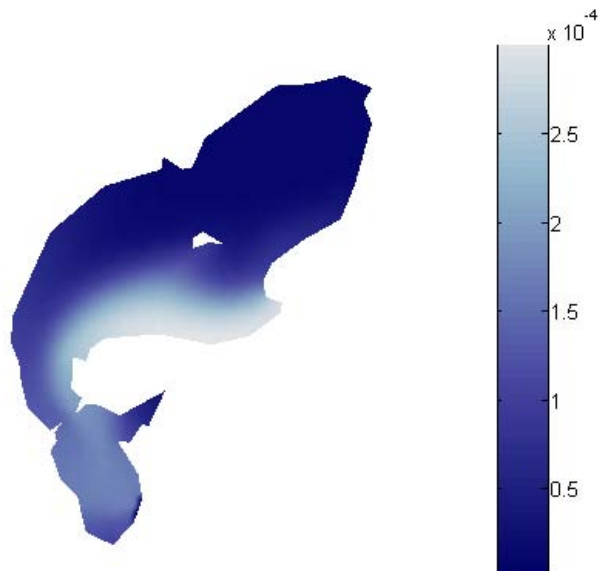


Figura 2.6: Distribuição final da concentração do poluente na lagoa (após 3000 iterações)

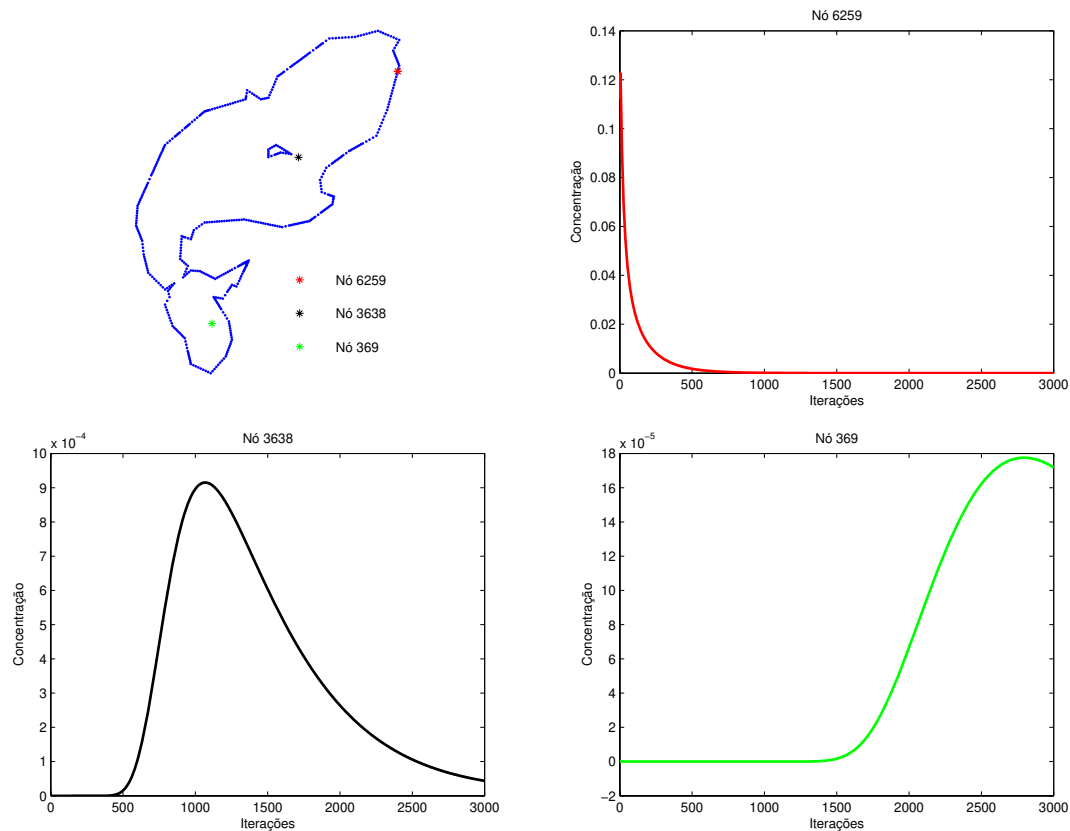


Figura 2.7: Localização dos nós 6259, 3638 e 369 e o acompanhamento do comportamento evolutivo do poluente nestes pontos

Três nós da malha foram escolhidos a fim de ser observado o comportamento evolutivo do material impactante nas regiões da lagoa que compreendem esses nós (*cf* figura 2.7). No gráfico do nó 6259, situado na região onde o poluente se concentra inicialmente, há um rápido decaimento na concentração, o que já era esperado pela modelagem que foi feita neste capítulo, pois a ausência de uma fonte poluente, a difusibilidade e a degradação moderadas e a influência da circulação da água fazem com que a dispersão do material impactante nesse ponto aconteça de forma gradativa. No gráfico do nó 3638, onde não há concentração inicial do poluente, percebe-se a sua chegada nesta região em determinado instante e após um certo período de tempo há um decaimento na concentração. Já no gráfico do nó 369, mais distante da região onde se concentra inicialmente o poluente, observa-se a sua chegada neste ponto já próximo ao fim das iterações, e ainda assim havendo um decaimento da concentração nessa região, correspondendo totalmente às expectativas da modelagem proposta. Estes resultados serão usados nos capítulos 3 e 4 para se estudar a influência da presença desse material impactante nas populações das duas espécies-chave do ecossistema local.

Capítulo 3

O convívio entre duas espécies competidoras sob efeito do poluente

Modelos de espécie única vêm sendo combatidos há tempo com o argumento de que nenhuma espécie existe isoladamente de outras. Desde os trabalhos de LOTKA (1925) e VOLTERRA (1926), sistemas de Equações Diferenciais Ordinárias que incorporam relações interespecíficas começaram a ser usados de diversos modos no estudo e na compreensão de muitos fenômenos de convívio entre espécies. No entanto, estes modelos consideravam apenas a variação temporal, pressupondo uma distribuição espacial homogênea das populações envolvidas. Os fenômenos de dispersão e migração, que são observados em quase todo tipo de população existente na natureza, tornaram necessária a inclusão das variáveis espaciais nas modelagens, cujo pioneiro foi SKELLAM (1951).

Sobre os Esteros de Iberá, região que abrange o domínio das aplicações do presente estudo, PREGNOLATTO (2002) estudou a modelagem e possibilidades de simulação computacional de uma determinada espécie carismática – a capivara (*Hydrocaeris hydrocaeris*) – afligida por uma epizootia local (o mal-das-cadeiras).

Aqui chega-se, então, ao ponto culminante do presente trabalho. O que se fará é uma combinação dos resultados obtidos nos capítulos 1 e 2 com um sistema de Equações Diferenciais Parciais de Dispersão-Migração envolvendo termos do tipo Lotka-volterra e dinâmicas vitais do tipo Verhulst. Esta idéia foi partejada primeiramente em SOSSAE (2003), num sistema que envolvia interações interespecíficas do tipo presa-predador e competição.

3.1 O modelo matemático

Dentro do contexto da presente proposta de modelagem, os fenômenos principalmente considerados serão:

- A dispersão populacional de cada espécie;
- Processos migratórios de cada espécie;
- O decaimento das espécies devido à presença de um material impactante tóxico;
- Dinâmicas vitais;
- Relações inter e intra-específicas.

Então, levando em conta duas populações P_1 e P_2 que interagem entre si, o sistema não-linear que descreve tais fenômenos para as densidades populacionais $P_1(x, y; t)$ e $P_2(x, y; t)$ com $(x, y) \in \Omega \subset \mathbb{R}^2$ e $t \in (0, T]$ é dado por

$$\begin{cases} \frac{\partial P_1}{\partial t} - \operatorname{div}(\alpha_1 \nabla P_1) + \operatorname{div}(\mathbb{U}P_1) + \rho_1 \sigma P_1 = a_1 P_1 \left(1 - \frac{P_1 + P_2}{K}\right) - \delta_1 P_1 P_2 \\ \frac{\partial P_2}{\partial t} - \operatorname{div}(\alpha_2 \nabla P_2) + \operatorname{div}(\mathbb{W}P_2) + \rho_2 \sigma P_2 = a_2 P_2 \left(1 - \frac{P_1 + P_2}{K}\right) - \delta_2 P_1 P_2 \end{cases} \quad (3.1)$$

onde

- $\alpha_1 = \alpha_1(x, y; t)$ e $\alpha_2 = \alpha_2(x, y; t)$ representam as dispersões populacionais de cada espécie;
- $\mathbb{U} = (U_1(x, y; t), U_2(x, y; t))$ e $\mathbb{W} = (W_1(x, y; t), W_2(x, y; t))$ representam os campos de velocidades de migração de cada espécie, com $\operatorname{div}(\mathbb{U}) = \operatorname{div}(\mathbb{W}) = 0$;
- $\sigma = \sigma(x, y; t)$ representa a concentração de um poluente no meio, obtida pela resolução da equação 1.1;
- ρ_1 e ρ_2 representam os decaimentos populacionais de cada espécie devido à mortalidade causada pela presença deste poluente;
- a_1 e a_2 representam as taxas de crescimento intrínsecas de cada espécie;
- K representa a capacidade de suporte do meio para as duas espécies;
- δ_1 e δ_2 representam as taxas de relação interespecífica (o sinal negativo caracteriza a competição).

Cabe ressaltar aqui um ponto que diferencia este trabalho dos demais que utilizaram uma modelagem matemática para os fenômenos que estão sendo estudados. Nas equações do sistema (3.1), utiliza-se uma mesma capacidade de suporte do meio para as populações das duas espécies. Esta proposta se deve ao fato de que, competindo as duas espécies basicamente pelos mesmos recursos e sendo estes recursos limitados, não se consideram capacidades de suportes do meio distintas para cada população, incluindo assim, na modelagem proposta, mais uma hipótese que exclui o tratamento isolado de cada espécie.

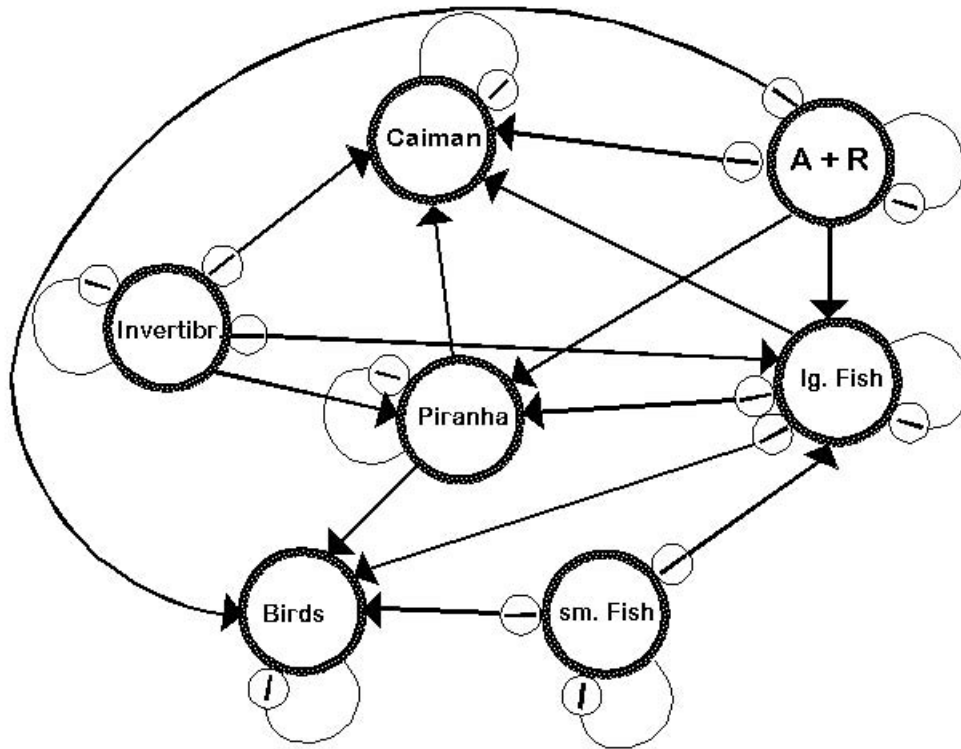


Figura 3.1: Diagrama ilustrativo da cadeia trófica do ecossistema da lagoa Iberá (site do projeto *The sustainable management of wetland resources*)

3.2 Simplificações do modelo

Usando argumentos análogos aos da seção 1.2, aqui neste caso serão considerados \mathbb{U} , \mathbb{W} , α_1 e α_2 constantes. Dessa forma, completando o sistema (3.1) com as condições iniciais e com as condições de contorno Dirichlet homogênea em Γ_0 e von Neumann homogênea em Γ_1 , tem-se

$$\left\{ \begin{array}{l}
\frac{\partial P_1}{\partial t} - \alpha_1 \Delta P_1 + U_1 \frac{\partial P_1}{\partial x} + U_2 \frac{\partial P_1}{\partial y} + \rho_1 \sigma P_1 = a_1 P_1 - b_1 P_1^2 - d_1 P_1 P_2 \\
\frac{\partial P_2}{\partial t} - \alpha_2 \Delta P_2 + W_1 \frac{\partial P_2}{\partial x} + W_2 \frac{\partial P_2}{\partial y} + \rho_2 \sigma P_2 = a_2 P_2 - b_2 P_2^2 - d_2 P_1 P_2 \\
(x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in (0, T] \\
P_1(x, y; 0) = P_{1_0}(x, y), \quad P_2(x, y; 0) = P_{2_0}(x, y) \quad \forall (x, y) \in \Omega \\
P_1|_{\Gamma_0} = P_2|_{\Gamma_0} = 0 \quad \forall t \in (0, T] \\
\left. \frac{\partial P_1}{\partial \eta} \right|_{\Gamma_1} = \left. \frac{\partial P_2}{\partial \eta} \right|_{\Gamma_1} = 0 \quad \forall t \in (0, T]
\end{array} \right. \quad (3.2)$$

onde $b_1 = \frac{a_1}{K}$, $b_2 = \frac{a_2}{K}$, $d_1 = b_1 + \delta_1$ e $d_2 = b_2 + \delta_2$.

3.3 Formulação variacional

Usando o mesmo argumento que no capítulo 1, aqui também será tomada como opção a formulação fraca do sistema (3.2) em vez da clássica. Adotando as mesmas notações da seção 1.3, o problema aqui torna-se encontrar $P_1, P_2 \in \mathcal{S}$ tais que

$$\left\{ \begin{array}{l}
\left(\frac{\partial P_1}{\partial t} | \nu \right) + \alpha_1 (\nabla P_1 | | \nabla \nu) + U_1 \left(\frac{\partial P_1}{\partial x} | \nu \right) + U_2 \left(\frac{\partial P_1}{\partial y} | \nu \right) + \rho_1 (\sigma P_1 | \nu) = \\
= a_1 (P_1 | \nu) - b_1 (P_1^2 | \nu) - d_1 (P_1 P_2 | \nu) \\
\left(\frac{\partial P_2}{\partial t} | \nu \right) + \alpha_2 (\nabla P_2 | | \nabla \nu) + W_1 \left(\frac{\partial P_2}{\partial x} | \nu \right) + W_2 \left(\frac{\partial P_2}{\partial y} | \nu \right) + \rho_2 (\sigma P_2 | \nu) = \\
= a_2 (P_2 | \nu) - b_2 (P_2^2 | \nu) - d_2 (P_1 P_2 | \nu) \\
\forall \nu \in \mathcal{V}
\end{array} \right. \quad (3.3)$$

3.4 Aproximação da solução

Aqui, como no capítulo anterior, as soluções do sistema (3.3) serão aproximadas nas suas variáveis espaciais através do método de Galerkin e na sua variável temporal através do método de Crank-Nicolson. Considerando o subespaço $\mathcal{V}_h \subset \mathcal{V}$ e a base \mathcal{B} como antes e separando-se as variáveis espaciais da temporal de P_1 e P_2 tem-se

$$\begin{aligned}
 P_1(x, y; t) &\cong P_{1h}(x, y; t) = \sum_{j=1}^N p_{1j}(t) \varphi_j(x, y), \\
 \frac{\partial P_{1h}}{\partial t} &= \sum_{j=1}^N \frac{dp_{1j}}{dt} \varphi_j, \quad \frac{\partial P_{1h}}{\partial x} = \sum_{j=1}^N p_{1j} \frac{\partial \varphi_j}{\partial x} \text{ e } \frac{\partial P_{1h}}{\partial y} = \sum_{j=1}^N p_{1j} \frac{\partial \varphi_j}{\partial y}; \\
 P_2(x, y; t) &\cong P_{2h}(x, y; t) = \sum_{j=1}^N p_{2j}(t) \varphi_j(x, y), \\
 \frac{\partial P_{2h}}{\partial t} &= \sum_{j=1}^N \frac{dp_{2j}}{dt} \varphi_j, \quad \frac{\partial P_{2h}}{\partial x} = \sum_{j=1}^N p_{2j} \frac{\partial \varphi_j}{\partial x} \text{ e } \frac{\partial P_{2h}}{\partial y} = \sum_{j=1}^N p_{2j} \frac{\partial \varphi_j}{\partial y}.
 \end{aligned}$$

Para a concentração σ do poluente, tem-se

$$\sigma(x, y; t) \cong \sigma_h(x, y; t) = \sum_{j=1}^N u_j(t) \varphi_k(x, y)$$

Usando-se estas aproximações na formulação variacional (3.3) e escrevendo o sistema

para os elementos da base \mathcal{B} obtém-se

$$\left(\begin{aligned}
 & \sum_{j=1}^N \frac{dp_{1j}}{dt} (\varphi_j | \varphi_i) + \alpha_1 \sum_{j=1}^N p_{1j} (\nabla \varphi_j | | \nabla \varphi_i) + U_1 \sum_{j=1}^N p_{1j} \left(\frac{\partial \varphi_j}{\partial x} | \varphi_i \right) + \\
 & + U_2 \sum_{j=1}^N p_{1j} \left(\frac{\partial \varphi_j}{\partial y} | \varphi_i \right) + \rho_1 \sum_{j=1}^N \sum_{k=1}^N p_{1j} u_k (\varphi_j \varphi_k | \varphi_i) = a_1 \sum_{j=1}^N p_{1j} (\varphi_j | \varphi_i) - \\
 & - b_1 \sum_{j=1}^N \sum_{k=1}^N p_{1j} p_{1k} (\varphi_j \varphi_k | \varphi_i) - d_1 \sum_{j=1}^N \sum_{k=1}^N p_{1j} p_{2k} (\varphi_j \varphi_k | \varphi_i) \\
 & \sum_{j=1}^N \frac{dp_{2j}}{dt} (\varphi_j | \varphi_i) + \alpha_2 \sum_{j=1}^N p_{2j} (\nabla \varphi_j | | \nabla \varphi_i) + W_1 \sum_{j=1}^N p_{2j} \left(\frac{\partial \varphi_j}{\partial x} | \varphi_i \right) + \\
 & + W_2 \sum_{j=1}^N p_{2j} \left(\frac{\partial \varphi_j}{\partial y} | \varphi_i \right) + \rho_2 \sum_{j=1}^N \sum_{k=1}^N p_{2j} u_k (\varphi_j \varphi_k | \varphi_i) = a_2 \sum_{j=1}^N p_{2j} (\varphi_j | \varphi_i) - \\
 & - b_2 \sum_{j=1}^N \sum_{k=1}^N p_{2j} p_{2k} (\varphi_j \varphi_k | \varphi_i) - d_2 \sum_{j=1}^N \sum_{k=1}^N p_{1j} p_{2k} (\varphi_j \varphi_k | \varphi_i) \\
 & \forall i = 1 \dots, N
 \end{aligned} \right. \tag{3.4}$$

Aplicando o método de Crank-Nicolson nas variáveis temporais de P_1 , P_2 e σ , o sistema (3.4) torna-se

$$\left\{ \begin{aligned}
& \sum_{j=1}^N p_{1_j}^{(n+1)} \left\{ \left(1 - a_1 \frac{\Delta t}{2} \right) (\varphi_i | \varphi_j) + \frac{\Delta t}{2} \left[\alpha_1 (\nabla \varphi_i | | \nabla \varphi_j) + U_1 \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \right) + \right. \right. \\
& \left. \left. + U_2 \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \right) + \sum_{k=1}^N \left(b_1 p_{1_k}^{(n+1/2)} + d_1 p_{2_k}^{(n+1/2)} + \rho_1 u_k^{(n+1/2)} \right) (\varphi_i | \varphi_j \varphi_k) \right] \right\} = \\
& = \sum_{j=1}^N p_{1_j}^{(n)} \left\{ \left(1 + a_1 \frac{\Delta t}{2} \right) (\varphi_i | \varphi_j) - \frac{\Delta t}{2} \left[\alpha_1 (\nabla \varphi_i | | \nabla \varphi_j) + U_1 \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \right) + \right. \right. \\
& \left. \left. + U_2 \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \right) + \sum_{k=1}^N \left(b_1 p_{1_k}^{(n+1/2)} + d_1 p_{2_k}^{(n+1/2)} + \rho_1 u_k^{(n+1/2)} \right) (\varphi_i | \varphi_j \varphi_k) \right] \right\} \\
& \sum_{j=1}^N p_{2_j}^{(n+1)} \left\{ \left(1 - a_2 \frac{\Delta t}{2} \right) (\varphi_i | \varphi_j) + \frac{\Delta t}{2} \left[\alpha_2 (\nabla \varphi_i | | \nabla \varphi_j) + W_1 \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \right) + \right. \right. \\
& \left. \left. + W_2 \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \right) + \sum_{k=1}^N \left(b_2 p_{2_k}^{(n+1/2)} + d_2 p_{1_k}^{(n+1/2)} + \rho_2 u_k^{(n+1/2)} \right) (\varphi_i | \varphi_j \varphi_k) \right] \right\} = \\
& = \sum_{j=1}^N p_{2_j}^{(n)} \left\{ \left(1 + a_2 \frac{\Delta t}{2} \right) (\varphi_i | \varphi_j) - \frac{\Delta t}{2} \left[\alpha_2 (\nabla \varphi_i | | \nabla \varphi_j) + W_1 \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \right) + \right. \right. \\
& \left. \left. + W_2 \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \right) + \sum_{k=1}^N \left(b_2 p_{2_k}^{(n+1/2)} + d_2 p_{1_k}^{(n+1/2)} + \rho_2 u_k^{(n+1/2)} \right) (\varphi_i | \varphi_j \varphi_k) \right] \right\} \\
& \forall i = 1, \dots, N
\end{aligned} \right. \tag{3.5}$$

que, na forma matricial, escreve-se

$$\left\{ \begin{aligned}
& \mathbf{C}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(n+1)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(n+1)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)}) \mathbf{p}_1^{(n+1)} = \mathbf{D}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(n+1)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(n+1)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)}) \mathbf{p}_1^{(n)} \\
& \mathbf{E}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(n+1)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(n+1)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)}) \mathbf{p}_2^{(n+1)} = \mathbf{F}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(n+1)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(n+1)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)}) \mathbf{p}_2^{(n)}
\end{aligned} \right.$$

onde

$$c_{ij} = \left(1 - a_1 \frac{\Delta t}{2}\right) (\varphi_i | \varphi_j) + \frac{\Delta t}{2} \left\{ \alpha_1 (\nabla \varphi_i | | \nabla \varphi_j) + U_1 \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \right) + U_2 \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \right) + \right. \\ \left. + \sum_{k=1}^N \left[b_1 \frac{p_{1_k}^{(n)} + p_{1_k}^{(n+1)}}{2} + d_1 \frac{p_{2_k}^{(n)} + p_{2_k}^{(n+1)}}{2} + \rho_1 \frac{u_k^{(n)} + u_k^{(n+1)}}{2} \right] (\varphi_i | \varphi_j \varphi_k) \right\}$$

$$d_{ij} = \left(1 + a_1 \frac{\Delta t}{2}\right) (\varphi_i | \varphi_j) - \frac{\Delta t}{2} \left\{ \alpha_1 (\nabla \varphi_i | | \nabla \varphi_j) + U_1 \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \right) + U_2 \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \right) + \right. \\ \left. + \sum_{k=1}^N \left[b_1 \frac{p_{1_k}^{(n)} + p_{1_k}^{(n+1)}}{2} + d_1 \frac{p_{2_k}^{(n)} + p_{2_k}^{(n+1)}}{2} + \rho_1 \frac{u_k^{(n)} + u_k^{(n+1)}}{2} \right] (\varphi_i | \varphi_j \varphi_k) \right\}$$

$$e_{ij} = \left(1 - a_2 \frac{\Delta t}{2}\right) (\varphi_i | \varphi_j) + \frac{\Delta t}{2} \left\{ \alpha_2 (\nabla \varphi_i | | \nabla \varphi_j) + W_1 \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \right) + W_2 \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \right) + \right. \\ \left. + \sum_{k=1}^N \left[b_2 \frac{p_{2_k}^{(n)} + p_{2_k}^{(n+1)}}{2} + d_2 \frac{p_{1_k}^{(n)} + p_{1_k}^{(n+1)}}{2} + \rho_2 \frac{u_k^{(n)} + u_k^{(n+1)}}{2} \right] (\varphi_i | \varphi_j \varphi_k) \right\}$$

$$f_{ij} = \left(1 + a_2 \frac{\Delta t}{2}\right) (\varphi_i | \varphi_j) - \frac{\Delta t}{2} \left\{ \alpha_2 (\nabla \varphi_i | | \nabla \varphi_j) + W_1 \left(\varphi_i | \frac{\partial \varphi_j}{\partial x} \right) + W_2 \left(\varphi_i | \frac{\partial \varphi_j}{\partial y} \right) + \right. \\ \left. + \sum_{k=1}^N \left[b_2 \frac{p_{2_k}^{(n)} + p_{2_k}^{(n+1)}}{2} + d_2 \frac{p_{1_k}^{(n)} + p_{1_k}^{(n+1)}}{2} + \rho_2 \frac{u_k^{(n)} + u_k^{(n+1)}}{2} \right] (\varphi_i | \varphi_j \varphi_k) \right\}$$

com $i, j = 1, \dots, N$ para cada n . Este sistema será resolvido iterativamente no tempo, a partir das condições iniciais $\mathbf{p}_1^{(0)}$ e $\mathbf{p}_2^{(0)}$ pontualmente fornecidas ou dadas implicitamente por

$$\sum_{j=1}^N p_{1_j}^{(0)} (\varphi_j | \varphi_i) = (P_{1_0} | \varphi_i) \text{ e } \sum_{j=1}^N p_{2_j}^{(0)} (\varphi_j | \varphi_i) = (P_{2_0} | \varphi_i)$$

3.5 Implementação computacional

A implementação computacional do sistema deste capítulo é análoga à que foi feita no sistema (1.7). As diferenças mais notáveis são que agora aparecem termos variáveis nas matrizes do sistema, que são sobrescritos a cada iteração temporal. Outro detalhe

que pode ser observado é o cálculo do parâmetro σ . No programa usado nas simulações, é considerado $\mathbf{u}^{(n)} = \frac{\mathbf{c}^{(n)}}{\|\mathbf{c}^{(n)}\|}$, numa tentativa de se ter um bom tratamento numérico em relação à concentração do poluente em cada instante de tempo.

Serão feitas duas iterações internas a cada passo no tempo, caracterizando-se assim um método preditor-corretor que tende à uma aproximação da ordem de $(\Delta t)^2$ conforme se aumentam as iterações (ver MEYER, 1988). Este processo é feito da seguinte maneira:

- obtém-se $\mathbf{p}_1^{(*)}$ e $\mathbf{p}_2^{(*)}$ a partir de

$$\begin{cases} \mathbf{C}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(n)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(n)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)})\mathbf{p}_1^{(*)} = \mathbf{D}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(n)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(n)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)})\mathbf{p}_1^{(n)} \\ \mathbf{E}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(*)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(n)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)})\mathbf{p}_2^{(*)} = \mathbf{F}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(*)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(n)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)})\mathbf{p}_2^{(n)} \end{cases}$$

- depois obtém-se $\mathbf{p}_1^{(n+1)}$ e $\mathbf{p}_2^{(n+1)}$ a partir de

$$\begin{cases} \mathbf{C}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(*)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(*)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)})\mathbf{p}_1^{(n+1)} = \mathbf{D}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(*)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(*)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)})\mathbf{p}_1^{(*)} \\ \mathbf{E}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(n+1)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(*)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)})\mathbf{p}_2^{(n+1)} = \mathbf{F}(\mathbf{p}_1^{(n)}, \mathbf{p}_1^{(n+1)}, \mathbf{p}_2^{(n)}, \mathbf{p}_2^{(*)}, \mathbf{u}^{(n)}, \mathbf{u}^{(n+1)})\mathbf{p}_2^{(*)} \end{cases}$$

O algoritmo usado, programado em ambiente *Matlab*, é apresentado no apêndice mostrando como são feitas essas rotinas computacionalmente.

Capítulo 4

Resultados das simulações – Parte II

Nestas simulações, as densidades populacionais iniciais P_{1_0} e P_{2_0} das duas espécies (*chajá* e jacaré, respectivamente) serão consideradas distribuídas de forma homogênea em todo o domínio, respeitando as condições de contorno e assumindo valores pontuais iguais a 1. A tabela a seguir mostra os parâmetros utilizados.

Parâmetros	Valores	Parâmetros	Valores
α_1	0.05	α_2	0.025
U_1	0.0	U_2	0.01
W_1	0.0	W_2	0.0
ρ_1	0.05	ρ_2	0.02
δ_1	1.5×10^{-6}	δ_2	10^{-6}
a_1	0.004	a_2	0.005
K	20	Δt	0.25

Tabela 4.1: Parâmetros utilizados na simulação do convívio entre duas espécies competidoras sob efeito do poluente

Há um motivo para se considerar os parâmetros U_1 , W_1 e W_2 nulos. Em certas épocas, os *chajás* possuem uma tendência de migração para o norte, e em outras épocas essas tendências são para o sul, enquanto que os jacarés não possuem uma característica migratória expressiva quando os *chajás* migram para o norte. Portanto é suposto aqui que as simulações tentam retratar esta época específica, ou seja, em que $U_1 = W_1 = W_2 = 0$.

Nas primeiras 1000 iterações pode-se ver (figura 4.1) que a espécie 1 começa a sofrer mais com os danos causados pela presença do poluente que a espécie 2, resultado numérico influenciado pela diferença entre os parâmetros ρ_1 e ρ_2 , que indicam os decaimentos proporcionais das densidades populacionais das duas espécies.

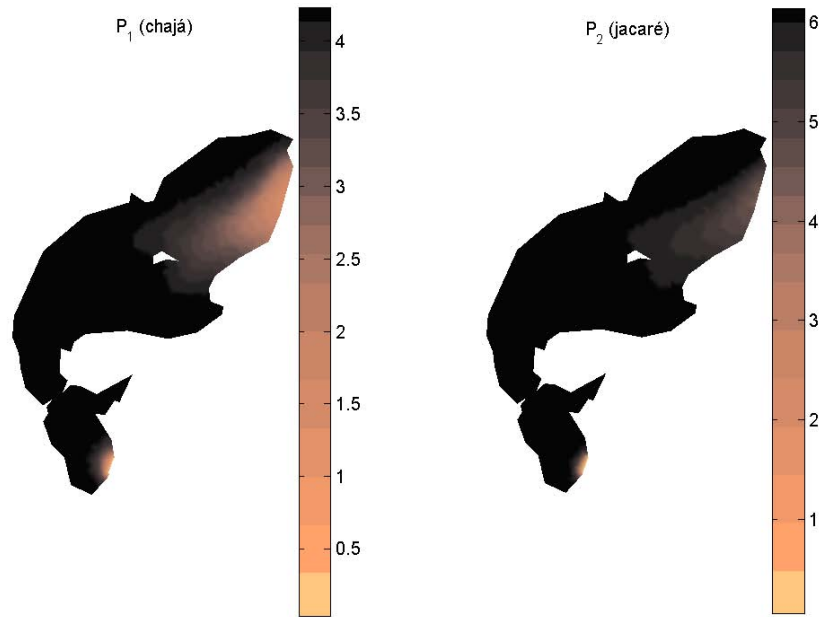


Figura 4.1: Distribuição das densidades populacionais das duas espécies na lagoa após 1000 iterações

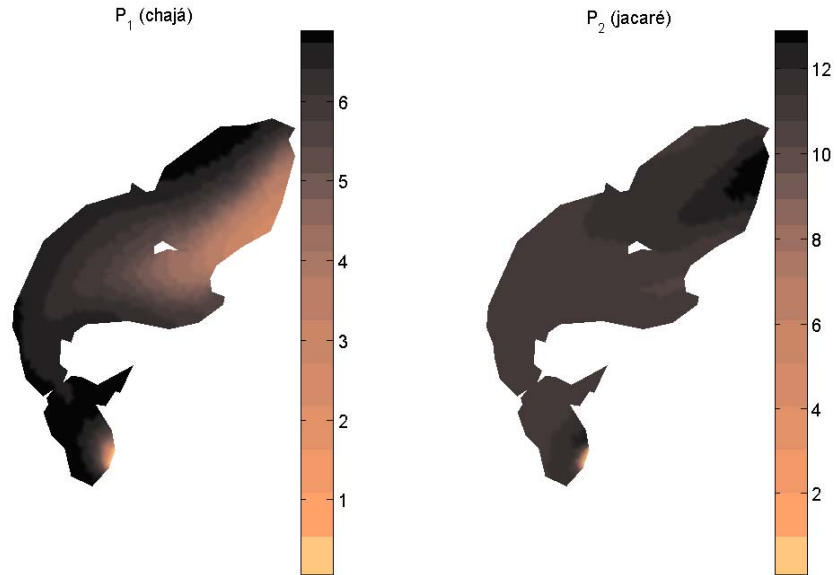


Figura 4.2: Distribuição das densidades populacionais das duas espécies na lagoa após 2000 iterações

Já após 2000 iterações, observa-se que a espécie 2 se recupera na região onde se tinha a presença inicial do material impactante tóxico enquanto que a espécie 1 ainda sofre por toda a região onde se dispersa o poluente, resultado numérico influenciado agora não só pelos decaimentos proporcionais mas também pela diferença entre os parâmetros a_1 e a_2 , que indicam o crescimento intrínseco das populações das duas espécies (ver figura 4.2).

Efetuada todas as 3000 iterações, o que corresponde a um tempo final de 750 unidades de tempo, pode-se perceber agora a influência de todos os parâmetros do modelo no resultado final (figuras 4.3, 4.4 e 4.5). As regiões onde se tem uma menor densidade populacional da espécie 1, são justamente aquelas onde se tem maior densidade populacional da espécie 2 – desse modo a competição interespecífica influenciando negativamente na dinâmica populacional – e por onde houve a passagem do poluente. Por outro lado, as regiões onde se tem maior densidade populacional da espécie 1 se situam ao norte da lagoa – efeito da característica de migração, entre outros – e na “ponta” a sudeste, onde não se teve a presença expressiva do material impactante tóxico em nenhum instante.

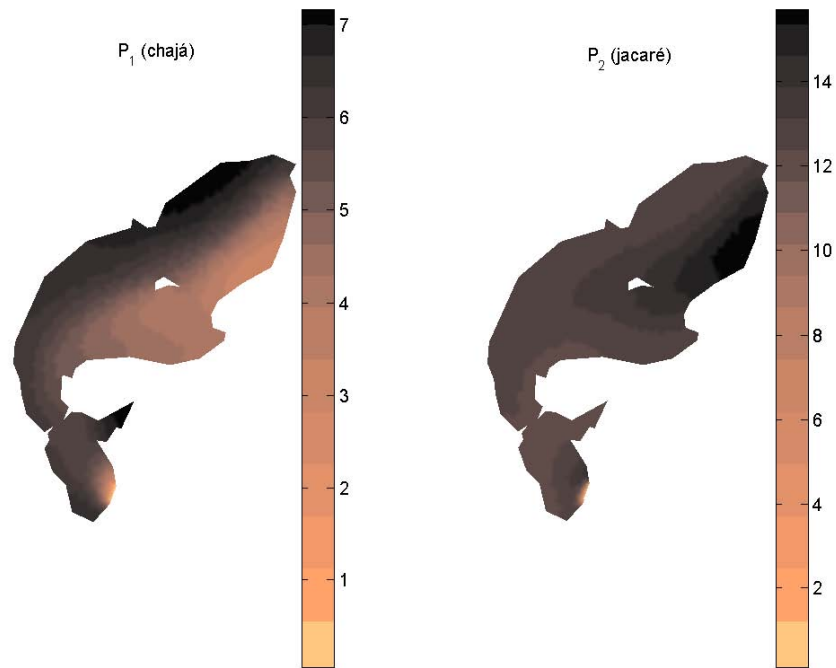


Figura 4.3: Distribuição final da densidade populacional da espécie 1 (*chajá*) na lagoa (após 3000 iterações)

Em contrapartida, a espécie 2 apresenta níveis populacionais mais altos nas regiões onde o poluente já se dispersou completamente ou quase que completamente, aproveitando-se da recuperação mais lenta da sua espécie competidora. A figura 4.6 mostra de forma detalhada estes fenômenos localmente em alguns pontos escolhidos.

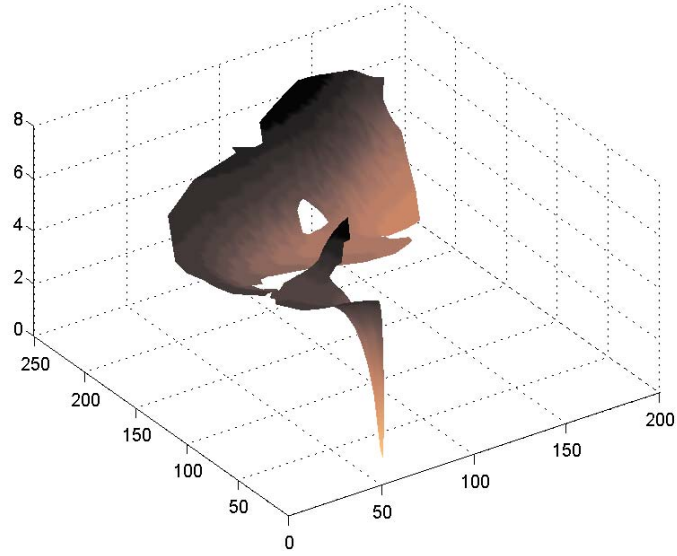


Figura 4.4: Visualização 3D da distribuição final da densidade populacional da espécie 1 (*chajá*) na lagoa (após 3000 iterações)

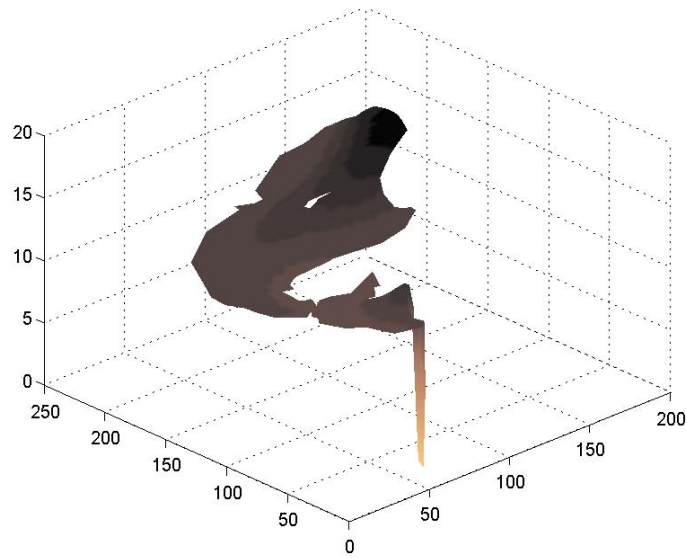


Figura 4.5: Visualização 3D da distribuição final da densidade populacional da espécie 2 (*jacaré*) na lagoa (após 3000 iterações)

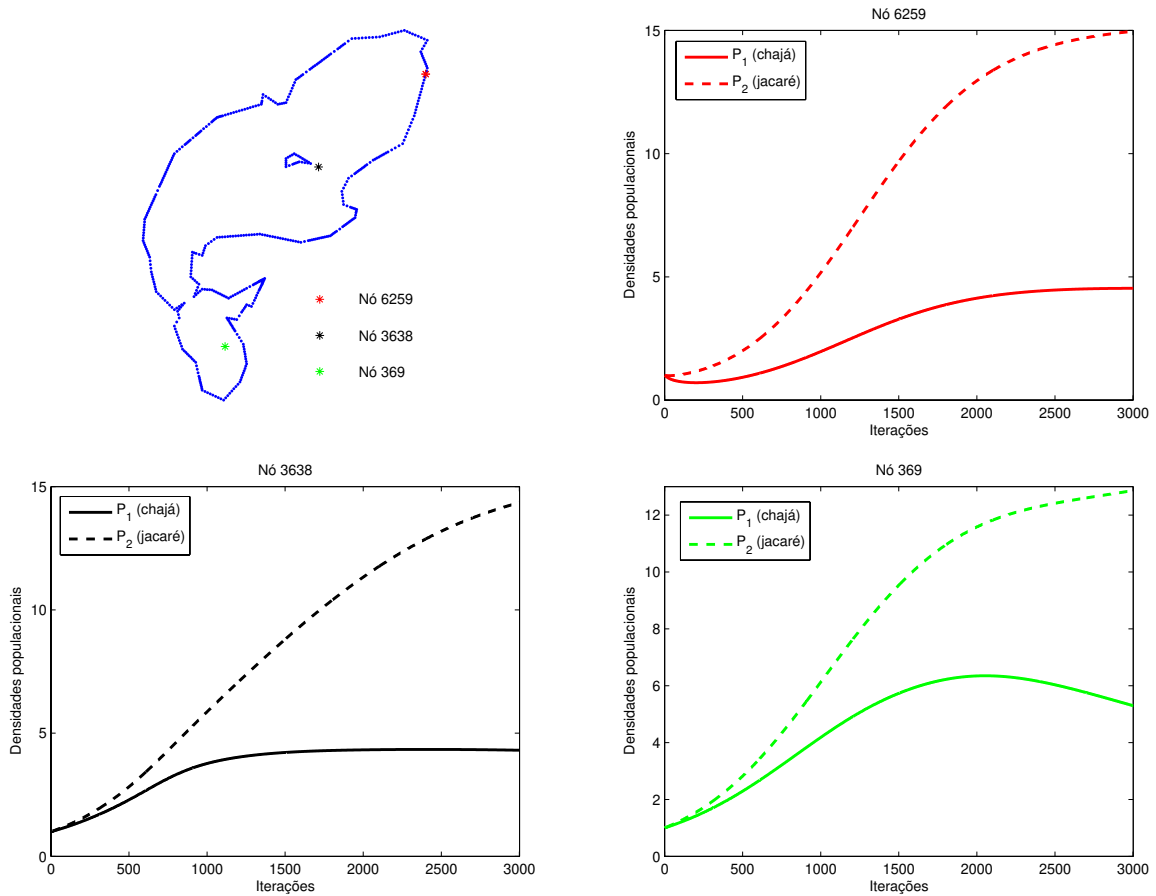


Figura 4.6: Localização do nós 6259, 3638 e 369 e o acompanhamento do comportamento evolutivo das densidades populacionais das duas espécies nestes pontos

Conclusões

É reconhecido o risco e o resultado muitas vezes fatal decorrente de um material impactante que afeta de modo irreversível o delicado e instável equilíbrio do convívio de duas espécies. No caso de espécies-chave, cuja presença e cuja permanência são essenciais na estabilidade do ecossistema, experimentos não podem ser feitos a não ser em condições muito restritas. No sentido de permitir reduzir o número de diferentes experimentos, e suas replicações, o que foi proposto neste trabalho é a elaboração de um programa que simule essa situação, permitindo reduzir a um mínimo as experiências de campo. O que se espera quanto à utilidade deste trabalho é, então:

1. A redução de experiências que impactam o meio;
2. A redução significativa do número daquelas experiências que não podem ser simuladas numericamente;
3. Criação de um instrumental básico para a formulação de planos de contingência;
4. Criação de um instrumental auxiliar para a formulação de estratégias de manejo integrado e sustentável em ambientes variados, e não só na região específica onde foram usadas as aplicações do presente estudo;
5. Criação de conhecimento computacional em simulação de fenômenos ecológicos e ambientais.

Portanto, espera-se que todas estas ferramentas ou seja, cada um destes resultados possa ou venha a contribuir de modo positivo para que desenvolvimentos regionais sejam efetuados de modo consciente e planejado.

Como já foi dito antes, este trabalho traz uma inovação, simples porém contundente, na proposta de modelagem matemática para os fenômenos que foram estudados. O uso de uma mesma capacidade de suporte para populações de espécies-chave de um ecossistema não visa simplesmente facilitar os cálculos, ou tornar o modelo útil apenas do ponto de

vista qualitativo. O objetivo deste estudo foi mesmo descrever de modo mais preciso* o convívio entre as espécies, considerando-se assim o máximo de hipóteses que excluem o tratamento isolado de cada uma.

As implementações computacionais feitas para a resolução dos sistemas propostos foram elaboradas de tal modo que, além de se tentar reduzir o tempo de processamento dos programas, se dispusesse de ferramentas mais acessíveis a profissionais e pesquisadores de áreas de ecologia e meio ambiente que não têm tanta familiaridade com linguagens de programação, tanto é que os algoritmos e as saídas gráficas foram implementados e obtidos em ambiente *Matlab* 7.0[†].

No tocante a esse mesmo intuito, o de se ter uma comunicação mais fácil e eficiente com especialistas de outras áreas, muitas oportunidades de trabalhos surgem para um futuro que já está se tornando realidade no próprio grupo de Biomatemática do IMECC – UNICAMP. Os avanços nas pesquisas do grupo vêm se tornando um alicerce sólido para se utilizar a lógica *fuzzy* na estimativa e calibração de parâmetros de modelos como os que foram apresentados neste trabalho, e também na resolução de problemas que podem ser modelados por sistemas *p-fuzzy*[‡], não abolindo deste modo os métodos numéricos usados nos trabalhos desenvolvidos até aqui mas apresentando-se como mais uma ferramenta auxiliar na busca de soluções para os problemas matemáticos, corroborando os resultados já obtidos e tornando o contato cada vez mais próximo com outros especialistas.

*Faz-se reconhecer aqui as limitações do uso deste termo e da ambição desse objetivo

[†]Os tempos de execução dos programas do poluente e das populações foram de cerca de 24min e 54h6min, respectivamente, num PC Pentium 4, CPU de 3.00 GHz e 1.00 GB de RAM.

[‡]Em outras palavras, puramente *fuzzy*

Apêndice

As sub-matrizes de rigidez

Aqui será apresentado de maneira breve uma proposta para a construção das sub-matrizes de rigidez usadas nas formulações algorítmicas para resolução dos sistemas expostos no trabalho.

O software *Triangle*, usado na discretização dos domínios espaciais, dispõe os nós dos triângulos como na figura 4.7

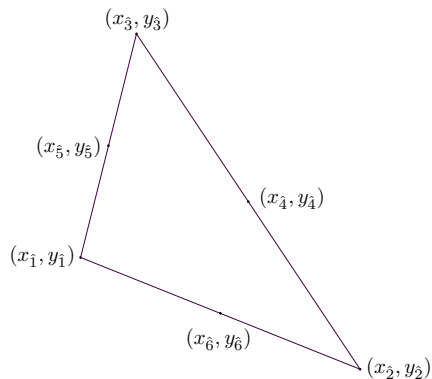


Figura 4.7: Triângulo genérico

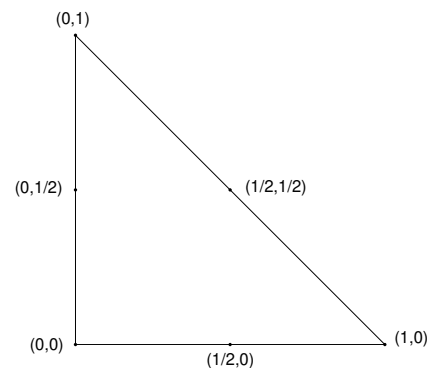


Figura 4.8: Triângulo padrão

Um arquivo denominado *malha.mat* foi criado para armazenar as informações sobre a malha de elementos e as coordenadas de cada nó. Este arquivo contém a matriz **malha** de dimensão $nel \times 6$, onde nel é o número de elementos da malha, em que cada linha representa um triângulo. As três primeiras colunas de cada linha são os números dos nós dos vértices, e as três restantes os pontos médios das faces. O arquivo também contém dois vetores de dimensão nn (número de nós da malha): um é o vetor \mathbf{x} , contendo as coordenadas no eixo- x de cada nó e o outro é o vetor \mathbf{y} contendo as coordenadas no eixo- y .

Para a aplicação do Método de Elementos Finitos, a transformação que leva o triângulo padrão \hat{K} num triângulo qualquer K da malha é

$$\begin{cases} x(\xi, \eta) = x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta \\ y(\xi, \eta) = y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta \end{cases}$$

A partir desta transformação, desenvolvem-se as integrais

$$\iint_K \varphi_i \varphi_j \, dydx = J \iint_{\hat{K}} \psi_i \psi_j \, d\eta d\xi$$

$$\begin{aligned} \iint_K \nabla \varphi_i \nabla \varphi_j \, dydx &= h_1 \iint_{\hat{K}} \frac{\partial \psi_i}{\partial \xi} \frac{\partial \psi_j}{\partial \xi} \, d\eta d\xi + h_2 \iint_{\hat{K}} \frac{\partial \psi_i}{\partial \eta} \frac{\partial \psi_j}{\partial \eta} \, d\eta d\xi - \\ &- h_3 \iint_{\hat{K}} \frac{\partial \psi_i}{\partial \xi} \frac{\partial \psi_j}{\partial \eta} + \frac{\partial \psi_j}{\partial \xi} \frac{\partial \psi_i}{\partial \eta} \, d\eta d\xi \end{aligned}$$

$$\iint_K \varphi_i \frac{\partial \varphi_j}{\partial x} \varphi_k \, dydx = (y_3 - y_1) \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \xi} \psi_k \, d\eta d\xi - (y_2 - y_1) \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \eta} \psi_k \, d\eta d\xi$$

$$\iint_K \varphi_i \frac{\partial \varphi_j}{\partial y} \varphi_k \, dydx = (x_2 - x_1) \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \eta} \psi_k \, d\eta d\xi - (x_3 - x_1) \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \xi} \psi_k \, d\eta d\xi$$

$$\iint_K \varphi_i \frac{\partial \varphi_j}{\partial x} \, dydx = (y_3 - y_1) \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \xi} \, d\eta d\xi - (y_2 - y_1) \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \eta} \, d\eta d\xi$$

$$\iint_K \varphi_i \frac{\partial \varphi_j}{\partial y} \, dydx = (x_2 - x_1) \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \eta} \, d\eta d\xi - (x_3 - x_1) \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \xi} \, d\eta d\xi$$

$$\iint_K \varphi_i \varphi_j \varphi_k = J \iint_{\hat{K}} \psi_i \psi_j \psi_k \, d\eta d\xi$$

onde

$$J = |(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)|$$

$$h_1 = \frac{1}{J} [(y_3 - y_1)^2 + (x_3 - x_1)^2]$$

$$h_2 = \frac{1}{J} [(y_2 - y_1)^2 + (x_2 - x_1)^2]$$

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

Então as sub-matrizes de rigidez são definidas por

$$(M1)_{i,j} = \iint_{\hat{K}} \psi_i \psi_j \, d\eta d\xi \quad (M2)_{i,j} = \iint_{\hat{K}} \frac{\partial \psi_i}{\partial \xi} \frac{\partial \psi_j}{\partial \xi} \, d\eta d\xi$$

$$(M3)_{i,j} = \iint_{\hat{K}} \frac{\partial \psi_i}{\partial \eta} \frac{\partial \psi_j}{\partial \eta} \, d\eta d\xi \quad (M4)_{i,j} = \iint_{\hat{K}} \frac{\partial \psi_i}{\partial \xi} \frac{\partial \psi_j}{\partial \eta} + \frac{\partial \psi_j}{\partial \xi} \frac{\partial \psi_i}{\partial \eta} \, d\eta d\xi$$

$$(M5)_{i,j,k} = \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \xi} \psi_k \, d\eta d\xi \quad (M6)_{i,j,k} = \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \eta} \psi_k \, d\eta d\xi$$

$$(M7)_{i,j} = \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \xi} \, d\eta d\xi \quad (M8)_{i,j} = \iint_{\hat{K}} \psi_i \frac{\partial \psi_j}{\partial \eta} \, d\eta d\xi$$

$$(M9)_{i,j,k} = \iint_{\hat{K}} \psi_i \psi_j \psi_k \, d\eta d\xi$$

Uma proposta algorítmica para a construção destas matrizes que são armazenadas no arquivo *smr2.mat* em ambiente *Matlab* é a seguinte:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  CALCULO DAS SUB-MATRIZES DE RIGIDEZ  %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funcoes de forma e suas derivadas

syms x y z=1-x-y;
f(1)=z*(2*z-1); f(2)=x*(2*x-1); f(3)=y*(2*y-1);
f(4)=4*x*y; f(5)=4*y*z; f(6)=4*x*z;

```



```

for i=1:6
    dfdx(i)=diff(f(i),x);
    dfdy(i)=diff(f(i),y);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Construcao das matrizes

for i=1:6
    for j=1:6
        r1=int(f(i)*f(j),y,0,1-x);
        M1(i,j)=int(r1,0,1);
        r2=int(dfdx(i)*dfdx(j),y,0,1-x);
        M2(i,j)=int(r2,0,1);
        r3=int(dfdy(i)*dfdy(j),y,0,1-x);
        M3(i,j)=int(r3,0,1);
        r4=int(dfdx(i)*dfdy(j)+dfdx(j)*dfdy(i),y,0,1-x);
        M4(i,j)=int(r4,0,1);
        r7=int(f(i)*dfdx(j),y,0,1-x);
        M7(i,j)=int(r7,0,1);
        r8=int(f(i)*dfdy(j),y,0,1-x);
        M8(i,j)=int(r8,0,1);
        for k=1:6
            r5=int(f(i)*dfdx(j)*f(k),y,0,1-x);
            M5(i,j,k)=int(r5,0,1);
            r6=int(f(i)*dfdy(j)*f(k),y,0,1-x);
            M6(i,j,k)=int(r6,0,1);
            r9=int(f(i)*f(j)*f(k),y,0,1-x);
            M9(i,j,k)=int(r9,0,1);
        end
    end
end

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

M1=double(M1); M2=double(M2); M3=double(M3); M4=double(M4);
M5=double(M5); M6=double(M6); M7=double(M7); M8=double(M8);

```

```
M9=double(M9);
```

```
save smr2 M1 M2 M3 M4 M5 M6 M7 M8 M9
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Um arquivo denominado *transformacao.mat* foi construído separadamente armazenando as informações sobre a transformação que faz a correspondência entre o triângulo padrão e um triângulo qualquer da malha. Sua programação foi feita como segue:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CALCULOS PARA USO NOS SISTEMAS LINEARES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear all
```

```
load malha
```

```
nel=size(malha,1);
```

```
for e=1:nel
```

```
    x1(e)=x(malha(e,1));
```

```
    x2(e)=x(malha(e,2));
```

```
    x3(e)=x(malha(e,3));
```

```
    y1(e)=y(malha(e,1));
```

```
    y2(e)=y(malha(e,2));
```

```
    y3(e)=y(malha(e,3));
```

```
    J(e)=abs((x2(e)-x1(e))*(y3(e)-y1(e))-(x3(e)-x1(e))*(y2(e)-y1(e)));
```

```
    h1(e)=((y3(e)-y1(e))*(y3(e)-y1(e))+(x3(e)-x1(e))*(x3(e)-x1(e)))/J(e);
```

```
    h2(e)=((y2(e)-y1(e))*(y2(e)-y1(e))+(x2(e)-x1(e))*(x2(e)-x1(e)))/J(e);
```

```
    h3(e)=((y3(e)-y1(e))*(y2(e)-y1(e))+(x3(e)-x1(e))*(x2(e)-x1(e)))/J(e);
```

```
end
```

```
save transformacao x1 x2 x3 y1 y2 y3 J h1 h2 h3
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dados do dominio temporal

nit=3000;    % numero de iteracoes
dt=0.25;    % delta t

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Coeficientes do problema

alfac=0.75;    % difusividade
sigmac=0.001; % decaimento

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Preparacao dos coeficientes constantes

dt2=dt/2; sig1=1+sigmac*dt2; sig2=1-sigmac*dt2; alf=alfac*dt2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Montagem das matrizes do sistema

A=sparse(nn,nn); B=sparse(nn,nn);

for e=1:nel
    for il=1:nnl
        i=malha(e,il);
        if gama0(i)==0
            for jl=1:nnl
                j=malha(e,jl);
                if gama0(j)==0
                    A(i,j)=A(i,j)+sig1*J(e)*M1(il,jl)+...
                        +...+
                        +alf*(h1(e)*M2(il,jl)+h2(e)*M3(il,jl)-h3(e)*...
                            M4(il,jl));
                    B(i,j)=B(i,j)+sig2*J(e)*M1(il,jl)-...
                        -...-
                        -alf*(h1(e)*M2(il,jl)+h2(e)*M3(il,jl)-h3(e)*...

```

```

        M4(il,jl));
    for kl=1:nnl
        k=malha(e,kl);
        if gama0(k)==0
            A(i,j)=A(i,j)+dt2*V(k,1)*((y3(e)-y1(e))*...
                M5(il,jl,kl)-(y2(e)-y1(e))*M6(il,jl,kl))+...
                +...+
                +dt2*V(k,2)*((x2(e)-x1(e))*M6(il,jl,kl)-...
                (x3(e)-x1(e))*M5(il,jl,kl));
            B(i,j)=B(i,j)-dt2*V(k,1)*((y3(e)-y1(e))*...
                M5(il,jl,kl)-(y2(e)-y1(e))*M6(il,jl,kl))-...
                -...-
                -dt2*V(k,2)*((x2(e)-x1(e))*M6(il,jl,kl)-...
                (x3(e)-x1(e))*M5(il,jl,kl));
        end
    end
end
end
end
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inclusao da condicao de Dirichlet na matriz A

for i=1:nn
    if gama0(i)==1
        A(i,i)=1;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Condicao inicial

cn=zeros(nn,1);

for il=1:6

```

```

cn(malha(3071,il))=0.125;
cn(malha(3049,il))=0.125;
cn(malha(2998,il))=0.125;
cn(malha(2912,il))=0.125;
cn(malha(2976,il))=0.125;
end

%-----
% Armazenamento da condicao inicial para visualizacao e para efeito nas
% populacoes

cgraf(:,1)=cn;
c6259(1)=cn(6259); c3638(1)=cn(3638); c369(1)=cn(369);
uant=cn/norm(cn);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Resolucao do sistema

for n=1:nit
    cn=A\ (B*cn);

%-----
% Armazenamento das solucoes para efeito nas populacoes

un=cn/norm(cn);
unm=(un+uant)/2;
u(:,n)=unm;
uant=un;

%-----
% Armazenamento de algumas solucoes e acompanhamento em alguns pontos
% para visualizacao

c6259(n+1)=cn(6259);
c3638(n+1)=cn(3638);
c369(n+1)=cn(369);

```

```

if n==1000
    cgraf(:,2)=cn;
end

if n==2000
    cgraf(:,3)=cn;
end

if n==nit
    cgraf(:,4)=cn;
end

%-----
end

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

min=etime(clock,t0)/60; save poluente cgraf c6259 c3638 c369 u min

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Um arquivo chamado *poluente.mat* é salvo no fim do programa armazenando a matriz \mathbf{u} de dimensão $nn \times nit$, onde nit é o numero de iterações, que será usada no programa das populações. Cada coluna \mathbf{u}_j desta matriz tem a forma $\mathbf{u}_j = \frac{\mathbf{u}^{(j-1)} + \mathbf{u}^{(j)}}{2}$, em que $\mathbf{u}^{(j)}$ representa a normalização da concentração do poluente após a iteração j .

As espécies sob efeito do poluente

Agora é apresentado o algoritmo usado na resolução do sistema que envolve as duas espécies competidoras sob efeito do poluente.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% CONVIVIO ENTRE DUAS ESPECIES COMPETIDORAS SOB EFEITO DO POLUENTE %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all

```

```

t0=clock;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Carregando arquivos

load malha
load smr2
load transformacao
load fronteira
load poluente

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dados do dominio espacial

nn=length(x);      % numero de nos da malha
nel=size(malha,1); % numero de elementos
nnl=6;             % numero de nos locais

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dados do dominio temporal

nit=3000;    % numero de iteracoes
dt=0.25;     % delta t

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Coeficientes do problema

alfa1=0.05;    % dispersao de P1
alfa2=0.025;  % dispersao de P2
delta1=1.5e-6; % taxa de relacao interespecifica de P1
delta2=1e-6;  % taxa de relacao interespecifica de P2
U1=0.0;      % migracao de P1 na direcao do eixo x
U2=1e-2;     % migracao de P1 na direcao do eixo y
W1=0.0;      % migracao de P2 na direcao do eixo x
W2=0.0;     % migracao de P2 na direcao do eixo y
a1=4e-3;     % taxa de crescimento intrinseca de P1
a2=5e-3;     % taxa de crescimento intrinseca de P2

```



```

K=20;          % capacidade de suporte do meio para P1 e P2
ro1=5e-2;     % decaimento de P1 devido a presenca do poluente
ro2=2e-2;     % decaimento de P2 devido a presenca do poluente

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Preparacao dos coeficientes constantes

b1=a1/K; b2=a2/K;
d1=b1+delta1; d2=b2+delta2;
a11=1-a1*dt/2; a12=1+a1*dt/2;
alf1=alfa1*dt/2; alf2=alfa2*dt/2;
U1dt2=U1*dt/2; U2dt2=U2*dt/2;
W1dt2=W1*dt/2; W2dt2=W2*dt/2;
a21=1-a2*dt/2; a22=1+a2*dt/2;
bb1=b1*dt/4; bb2=b2*dt/4;
dd1=d1*dt/4; dd2=d2*dt/4;
rro1=ro1*dt/2; rro2=ro2*dt/2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Condicoes iniciais

p1n=ones(nn,1); p2n=ones(nn,1);

for i=1:nn
    if gama0(i)==1
        p1n(i)=0;
        p2n(i)=0;
    end
end

%-----
% Armazenamento das condicoes iniciais para visualizacao

p1graf(:,1)=p1n;    p2graf(:,1)=p2n;
p16259(1)=p1n(6259); p26259(1)=p2n(6259);
p13638(1)=p1n(3638); p23638(1)=p2n(3638);
p1369(1)=p1n(369);  p2369(1)=p2n(369);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Montagem das partes fixas das matrizes C, D, E e F

```

```

C1=sparse(nn,nn); D1=sparse(nn,nn); E1=sparse(nn,nn); F1=sparse(nn,nn);

```

```

for e=1:nel

```

```

    for il=1:nnl

```

```

        i=malha(e,il);

```

```

        if gama0(i)==0

```

```

            for jl=1:nnl

```

```

                j=malha(e,jl);

```

```

                if gama0(j)==0

```

```

                    C1(i,j)=C1(i,j)+a11*J(e)*M1(il,jl)+...

```

```

                    +...+

```

```

                    +alf1*(h1(e)*M2(il,jl)+h2(e)*M3(il,jl)-h3(e)*...
                    M4(il,jl))+...

```

```

                    +...+

```

```

                    +U1dt2*((y3(e)-y1(e))*M7(il,jl)-(y2(e)-y1(e))*...
                    M8(il,jl))+...

```

```

                    +...+

```

```

                    +U2dt2*((x2(e)-x1(e))*M8(il,jl)-(x3(e)-x1(e))*...
                    M7(il,jl));

```

```

                    D1(i,j)=D1(i,j)+a12*J(e)*M1(il,jl)-...

```

```

                    -...-

```

```

                    -alf1*(h1(e)*M2(il,jl)+h2(e)*M3(il,jl)-h3(e)*...
                    M4(il,jl))-...

```

```

                    -...-

```

```

                    -U1dt2*((y3(e)-y1(e))*M7(il,jl)-(y2(e)-y1(e))*...
                    M8(il,jl))-...

```

```

                    -...-

```

```

                    -U2dt2*((x2(e)-x1(e))*M8(il,jl)-(x3(e)-x1(e))*...
                    M7(il,jl));

```

```

                    E1(i,j)=E1(i,j)+a21*J(e)*M1(il,jl)+...

```

```

                    +...+

```

```

                    +alf2*(h1(e)*M2(il,jl)+h2(e)*M3(il,jl)-h3(e)*...
                    M4(il,jl))+...

```

```

+...+
+W1dt2*((y3(e)-y1(e))*M7(il,j1)-(y2(e)-y1(e))*...
M8(il,j1))+...
+...+
+W2dt2*((x2(e)-x1(e))*M8(il,j1)-(x3(e)-x1(e))*...
M7(il,j1));
F1(i,j)=F1(i,j)+a22*J(e)*M1(il,j1)-...
-...-
-alf2*(h1(e)*M2(il,j1)+h2(e)*M3(il,j1)-h3(e)*...
M4(il,j1))-...
-...-
-W1dt2*((y3(e)-y1(e))*M7(il,j1)-(y2(e)-y1(e))*...
M8(il,j1))-...
-...-
-W2dt2*((x2(e)-x1(e))*M8(il,j1)-(x3(e)-x1(e))*...
M7(il,j1));
end
end
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inicio das iteracoes

p1est=p1n; p2est=p2n;

for n=1:nit
%-----
% Iteracoes internas para aproximar p1^(n+1) e p2^(n+1): montagem das
% partes variaveis das matrizes C e D

for it=1:2
C=C1; D=D1;
for e=1:nel
for il=1:nnl
i=malha(e,il);

```

```

    if gama0(i)==0
        for jl=1:nnl
            j=malha(e,jl);
            if gama0(j)==0
                for kl=1:nnl
                    k=malha(e,kl);
                    if gama0(k)==0
                        aux1=(bb1*(p1n(k)+p1est(k))+dd1*...
                            (p2n(k)+p2est(k))+rro1*u(k,n))*...
                            J(e)*M9(il,jl,kl);
                        C(i,j)=C(i,j)+aux1;
                        D(i,j)=D(i,j)-aux1;
                    end
                end
            end
        end
    end
end

%-----
% Inclusao da condicao de Dirichlet na matriz C

for i=1:nn
    if gama0(i)==1
        C(i,i)=1;
    end
end

%-----
% Resolucao do sistema para p1

p1est=C\(D*p1est);

%-----
% Montagem das partes variaveis das matrizes E e F

```

```

E=E1; F=F1;
for e=1:nel
    for il=1:nnl
        i=malha(e,il);
        if gama0(i)==0
            for jl=1:nnl
                j=malha(e,jl);
                if gama0(j)==0
                    for kl=1:nnl
                        k=malha(e,kl);
                        if gama0(k)==0
                            aux2=(bb2*(p2n(k)+p2est(k))+dd2*...
                                (p1n(k)+p1est(k))+rro2*u(k,n))*...
                                J(e)*M9(il,jl,kl);
                            E(i,j)=E(i,j)+aux2;
                            F(i,j)=F(i,j)-aux2;
                        end
                    end
                end
            end
        end
    end
end

%-----
% Inclusao da condicao de Dirichlet na matriz E

for i=1:nn
    if gama0(i)==1
        E(i,i)=1;
    end
end

%-----
% Resolucao do sistema para p2

p2est=E\F*p2est);

```

```

        %-----
end

%-----
% Fim das iteracoes internas

p1n=p1est; p2n=p2est;

%-----
% Armazenamento de algumas solucoes e acompanhamento em alguns pontos
% para visualizacao

p16259(n+1)=p1n(6259); p26259(n+1)=p2n(6259);
p13638(n+1)=p1n(3638); p23638(n+1)=p2n(3638);
p1369(n+1)=p1n(369); p2369(n+1)=p2n(369);

if n==1000
    p1graf(:,2)=p1n; p2graf(:,2)=p2n;
end

if n==2000
    p1graf(:,3)=p1n; p2graf(:,3)=p2n;
end

if n==nit
    p1graf(:,4)=p1n; p2graf(:,4)=p2n;
end
end

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

horas=etime(clock,t0)/3600;
save populacoes p1graf p2graf p16259 p26259 p13638 p23638 p1369 p2369 horas

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Bibliografia

- BERNARDES, M., 1998. Poluição em corpos aquáticos de baixa circulação: Modelagem e simulação numérica. Dissertação de Mestrado, IMECC – UNICAMP, Campinas/SP.
- CANTÃO, R. F., 1998. Modelagem e simulação numérica de derrames de óleo no canal de São Sebastião, SP. Dissertação de Mestrado, IMECC – UNICAMP, Campinas/SP.
- CANTÃO, R. F., D'AFONSECA, L. A., 1998. Produção interna do grupo de Biomatemática em Matlab, software em Matlab, IMECC – UNICAMP.
- CANTRELL, R. S., COSNER, C., 1989. Diffusive logistic equations with indefinite weights: population models in disrupted environments. *Proceedings of Royal Society of Edinburgh* 112A, 293-318.
- CANTRELL, R. S., COSNER, C., 1996. Practical persistence in ecological models via comparison methods. *Proceedings of Royal Society in Edinburgh* 126A, 247-272.
- CANTRELL, R. S., COSNER, C., 1996. Spatially explicit models for the population dynamics of a species colonizing an island. *Mathematical Biosciences* 136, 65-107.
- COMMISSION of European Communities, 1991. Water pollution research report 27: Pesticides in ground and drinking water, Lecture Notes of Directorate General for Science, Research and Development.
- COSNER, C., 1996. Variability, vagueness and comparison methods for ecological models. *Bulletin of Math. Biology* 58 (2), 207-246.
- DINIZ, G. L., 2003. Dispersão de poluentes num sistema ar-água: modelagem, aproximação e aplicações. Tese de Doutorado, FEEC – UNICAMP, Campinas/SP.
- DUBEY, B., 1997. Modelling the depletion and conservation of resources: effects of two interacting populations. *Ecological Modelling* 101, 123-136.

- EDELSTEIN-KESHET, L. 1988. *Mathematical Models in Biology*. SIAM/Random-House, New York.
- GOURLEY, S. A., BRITTON, N. F., 1996. A predator-prey reaction-diffusion system with non-local effects. *J. Math. Biology* 34, 297-333.
- GOURLEY, S. A., BRITTON, N. F., CHAPLAIN, M. A. J., BYRNE, H. M., 1996. Mechanisms for stabilization of systems of reaction-diffusion equations. *J. Math. Biology* 34, 857-877.
- HADLER, K. P., HEIDEN, U. A. D., ROTH, F., 1974. Nonhomogeneous distribution of populations. *J. Math. Biology* 1, 165-175.
- HEIDERICH, R. H., 2004. *Dispersão estacionária de populações: modelagem e instrumentação numérica*. Dissertação de Mestrado, IMECC – UNICAMP, Campinas/SP.
- KARDESTUNCER, H., NORRIE, D. H., 1987. *Finite Element Handbook*. MacGraw-Hill, New York.
- LACAZ, T. M. V. S., 1999. *Análises de problemas populacionais intraespecíficos e interespecíficos com difusão densidade-dependente*. Tese de Doutorado, IMECC – UNICAMP, Campinas/SP.
- MARCHUK, G. I., 1986. *Mathematical models in environmental problems*. *Studies in Mathematics and its Applications*, Vol. 16. North-Holland, Amsterdam.
- MEYER, J. F. C. A., 1988. *Modelagem e simulação numérica do transiente térmico em meios compostos*. Tese de Doutorado, IMECC – UNICAMP, Campinas/SP.
- MEYER, J. F. C. A., SOSSAE, R. C., LOISELLE, S., ROSSI, C., 1999. Mathematical modeling and numerical simulation of space-dependent multispecies interactions. *Annals of the New York Academy of Sciences* 879, 440-443.
- MISTRO, D. C., 1992. *O problema de poluição em rios por mercúrio metálico: modelagem e simulação*. Dissertação de Mestrado, IMECC – UNICAMP, Campinas/SP.
- MURRAY, J. D., 1989. *Mathematical Biology*. Springer, Heidelberg.
- NISBERT, R. M., GURNEY, W. S. C., 1982. *Modelling Fluctuating Populations*. John Wiley and Sons Ltd.
- OKUBO, A., 1980. *Diffusion and Ecological Problems: Mathematical Models*. Springer, Berlin.

- OLIVEIRA, F. R., O comportamento evolutivo de uma mancha de óleo na Baía de Ilha Grande, RJ: modelagem, análise numérica e simulações. Tese de Doutorado, IMECC – UNICAMP, Campinas/SP.
- PEYRET, R., TAYLOR, T. D., 1985. Computational Methods for Fluid Flow. Springer - Verlag New York Heidelberg Berlin.
- PREGNOLATTO, S. A., 2002. Mal-das-cadeiras em capivaras: Estudo, modelagem e simulação de um caso. Tese de Doutorado, FEEC – UNICAMP, Campinas/SP.
- SKELLAM, J. C., 1951. Random dispersal in theoretical populations. *Biometrika* 38, 196-218.
- Site do projeto *The sustainable management of wetland resources*: www.unisi.it/wetland.
- SOSSAE, R. C., 1995. Dinâmica populacional densidade-dependente em processos de dispersão e migração. Dissertação de Mestrado, IMECC – UNICAMP, Campinas/SP.
- SOSSAE, R. C., 2003. A presença evolutiva de um material impactante e seu efeito no transiente populacional de espécies interativas: modelagem e simulação. Tese de Doutorado, IMECC – UNICAMP, Campinas/SP.
- VÁSQUEZ, J. C. S., 2005. Comportamento evolutivo de descarga de água de produção decorrente de atividade offshore: tratamento numérico e simulação computacional. Tese de Doutorado, IMECC – UNICAMP, Campinas/SP.