

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE QUÍMICA
PROGRAMA DE PÓS-GRADUAÇÃO EM QUÍMICA

**Transições entre estados espaciais não homogêneos em sistemas químicos
e biológicos**

Leonardo Silva Dias*

Dissertação apresentada como parte dos
requisitos para obtenção do título de
MESTRE EM CIÊNCIAS, área de
concentração: FÍSICO-QUÍMICA

Orientador: Prof. Dr. Alejandro López Castillo

* Bolsista CAPES

SÃO CARLOS - SP
2019



Folha de Aprovação


Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Leonardo Silva Dias, realizada em 22/07/2019:



Prof. Dr. Alejandro Lopez Castillo
UFSCar



Prof. Dr. Marcus Aloizio Martinez de Aguiar
UNICAMP



Prof. Dr. Raphael Nagao de Sousa
UNICAMP

Dedico a minha Família
minha mãe, Simone, ao meu pai, Luiz, e ao meu irmão, Guilherme.

"We can only see a short distance ahead, but we can see plenty there that needs to be done. "

Alan Mathison Turing

Agradecimentos

Agradeço a minha mãe, meu pai e meu irmão pelo amor e suporte durante todas as etapas da minha vida.

Agradeço ao Professor Alejandro pela orientação, paciência, exemplo acadêmico e colaboração durante todos esses anos.

Agradeço aos meus colegas de laboratório, em especial ao Anderson Catão e Alexandre pelas relevantes discussões e apoio para o desenvolvimento deste trabalho.

Agradeço aos meus amigos-irmãos da praça Barbieri - Jaboticabal por tantos anos de amizade e, em particular, pelo incentivo no desenvolvimento deste trabalho.

Agradeço aos meus amigos da república Etanóis, membros e agregados, pela amizade e pelos momentos de descontração.

Agradeço a Izadora Lirango Furlani pela paciência, incentivo e companheirismo durante os anos de desenvolvimento deste trabalho.

Agradeço aos amigos da turma de Licenciatura em Química/011, assim como, aos demais amigos que fiz na UFSCar.

Agradeço a Cloud UFSCar por providenciar recursos computacionais que contribuíram para o desenvolvimento deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Assim sendo, agradeço a CAPES pelo financiamento.

Lista de Abreviaturas e Símbolos

EDP - Equação Diferencial Parcial

EDO - Equação Diferencial Ordinária

CIMA - *Chlorite - Iodide - Malonic Acid*

CDIMA - *Chlorine Dioxide - Iodine - Malonic Acid*

MI - Modelo isotérmico

MNI - Modelo não isotérmico

MPI - *Message Passing Interface*

CFN - Condição de Fronteira de Neumann

CFD - Condição de Fronteira de Dirichlet

GQIF - Gradiente Químico Imposto através das Fronteiras

ADI - *Alternating Direction Implicit Method*

t - Tempo

\mathbf{D} - Matriz diagonal dos coeficientes de difusão

\vec{f} - Função de termos de velocidade de reação

∇^2 - Operador Laplaciano

W - Taxa de absorção de fótons actínicos

Φ - Rendimento quântico

k_B - Constante de Boltzmann

η - Viscosidade do fluido

U - Concentração do reagente ativador U

V - Concentração do reagente inibidor V

A - Fonte de reagente U

E - Sumidouro de reagente U

K_f - Constante de velocidade da reação direta

K_r - Constante de velocidade da reação reversa

R - Constante universal dos gases ideias

E_i - Energia de ativação da i -ésima reação química

D_U - Constante de difusão de U

D_V - Constante de difusão de V

T - Temperatura

T_E - Temperatura externa

λ_v - Espaço de velocidade do reator

C_T - Concentração molar total do meio reacional

C_{ps} - Capacidade calorífica do meio reacional

$C_{p,U}$ - Capacidade calorífica de U

β - Coeficiente de transferência de calor do reator

A_r - Área de transferência de calor

v_0 - Volume do reator

\dot{v}_0 - Fluxo de entrada e saída do reagente U

α - Coeficiente de difusão térmica

T_Ω - Temperatura da fronteira

U_Ω - Concentração de U na fronteira

V_Ω - Concentração de V na fronteira

ΔH_i - Variação de entalpia da i -ésima reação química

$\Delta C_{p,i}$ - Variação da capacidade calorífica da i -ésima espécie química

x - Comprimento adimensional do reator na direção x

y - Comprimento adimensional do reator na direção y

τ - Tempo adimensional

u - Concentração adimensional do ativador U

v - Concentração adimensional do inibidor V

θ - Temperatura adimensional

d_u - Constante de difusão adimensional do reagente u

d_v - Constante de difusão adimensional do reagente v

d_θ - Coeficiente de difusão térmica adimensional da temperatura

θ_Ω - Temperatura de fronteira adimensional

k_f - Constante de velocidade adimensional da reação direta

k_r - Constante de velocidade adimensional da reação inversa

e_i - Energia de ativação adimensional da i -ésima reação química

ξ - Espaço de velocidade do reator adimensional

ϕ - Coeficiente de transferência de calor do reator adimensional

γ_u - Capacidade calorífica de U adimensional

$\Delta H'_{r_i}$ - Variação de entalpia da i -ésima reação química adimensional

λ - Expoente de Lyapunov

$\langle e \rangle$ - Excentricidade média

A_w - Área de pixels brancos

A_b - Área de pixels pretos

dS - Variação de entropia

dq - Variação de calor

$d_e S$ - Variação de entropia relativa a processos reversíveis

$d_i S$ - Variação de entropia relativa a processos irreversíveis

F_k - Forças termodinâmicas

J_k - Fluxos termodinâmicos

e_{max} - Erro máximo

e_{L^2} - Erro quadrático médio

σ - Taxa de produção de entropia

σ_T - Taxa de produção de entropia total

σ_{RQ} - Taxa de produção de entropia relativo a reação química

σ_M - Taxa de produção de entropia relativo ao transporte de massa

σ_H - Taxa de produção de entropia relativo a condução/difusão de calor

Γ - Funcional de Minkowski

p - Perímetro do meio reacional

$\vec{\nabla}$ - Operador gradiente

Lista de tabelas

TABELA 5.1 – EDPs em 1D e suas respectivas soluções analíticas utilizadas para o teste de acurácia das aproximações numéricas obtidas a partir do programa criado.	25
TABELA 5.2 – EDPs em 2D e suas respectivas soluções analíticas utilizadas para o teste de acurácia das aproximações numéricas obtidas a partir do programa criado.	25
TABELA 5.3 – Erro máximo e erro quadrático médio da aproximação numérica calculados para os Casos 1-3, presentes na TABELA 5.1, em 1D. . . .	26
TABELA 5.4 – Erro máximo e erro quadrático médio da aproximação numérica calculados para os Casos 1-3, presentes na TABELA 5.2, em 2D. . . .	26
TABELA 5.5 – Valores numéricos dos parâmetros físico-químicos. * Em Resultados 5.6 também utilizou-se $\xi = 1.0$	27
TABELA 5.6 – Taxa de produção de entropia e funcionais de Minkowski ao resultados obtidos a partir dos modelos MI e MNI (CFN com fluxo zero para a temperatura), e sistemas com diferentes perímetros.	33
TABELA 5.7 – Taxa de produção de entropia e funcionais de Minkowski ao resultados obtidos a partir do modelo MNI (CFD para a temperatura), e sistemas com diferentes perímetros.	34
TABELA 5.8 – Taxa de produção de entropia total para MI e MNI em diferentes regiões do sistema.	45
TABELA 5.9 – Taxa de produção de entropia relativa à cada processo irreversível e funcional de Minkowski para o MI e MNI dos dois novos reatores.	48
TABELA 7.1 – Forças e fluxos termodinâmicos associados aos processos irreversíveis. Os termos J_H e J_M são provenientes das leis de Fourier e Fick, respectivamente.* Os termos associados às reações químicas estão bem definidos na subseção 7.4.1.	70

Lista de figuras

- FIGURA 3.1 – Representação esquemática da dinâmica do sistema construído. 12
- FIGURA 4.1 – Retratos de fase de um sistema de EDO's de termos não lineares (à esquerda) e lineares (à direita), com variáveis dependentes u e v , e ponto de equilíbrio hiperbólico $(u^*, v^*) = (0, 0)$ (ponto em vermelho). Como o ponto de equilíbrio é hiperbólico, os retratos de fase dos dois sistemas são qualitativamente equivalentes, diferenciando entre si por distorções. Nas vizinhanças do ponto de equilíbrio, as órbitas são muito similares, o que permite classificar através da versão linear a estrutura e a estabilidade da órbita nesta região. Neste exemplo genérico, as órbitas apresentam uma estrutura de ponto de sela, implicando comportamento instável. 17
- FIGURA 4.2 – Determinação dos funcionais de Minkowski (Γ). **(A)** A partir de um sistema do tipo reação-difusão e em condições de instabilidade de Turing, observa-se a emergência de padrões de Turing. **(B)** Através do *software Mathematica* é possível obter uma imagem binária dos padrões de Turing. **(C)** Calcula-se as áreas formadas por pixels brancos e pretos a partir da imagem binária, assim como a média das excentricidades $\langle e \rangle$. Como esta representado na figura, para o cálculo de $\langle e \rangle$, elipses (em vermelho) são utilizadas para aproximar da forma das estruturas que compõe o padrão. Cada uma delas apresentará um valor de excentricidade, e com isso é possível determinar a média entre elas. Para formas circulares $e \rightarrow 0$ e para estruturas com forma de listra $e \rightarrow 1$. **(D)** Associação entre estruturas e valores esperados para Γ 19
- FIGURA 4.3 – Representação das mudanças de entropia devido à processos irreversíveis, $d_i S$, e reversíveis, $d_e S$. Pela segunda lei da termodinâmica, $d_i S \geq 0$ sempre, ao passo que $d_e S$ pode ser maior, menor ou igual a zero dependendo da classe de sistema considerado, *i.e.* isolado, fechado ou aberto, e transformação estudada. 20
- FIGURA 5.1 – Esquema de modularização do programa para o método de Peaceman-Rachford. Os passos correspondentes aos Módulos 2, 3 e 4 estão apresentados explicitamente em Apêndice 7.2. A construção é similar para o método de Crank-Nicolson. 23
- FIGURA 5.2 – Análise da estabilidade linear de MI, onde T: instabilidade de Turing e PS: ponto de sela. 27
- FIGURA 5.3 – Análise da estabilidade linear de MNI, onde T: instabilidade de Turing, PS: ponto de sela e O: oscilação. 28

FIGURA 5.4 – Diferença absoluta da taxa de produção de entropia (δ) entre simulações executadas com condição de fronteira de Dirichlet e Neumann para sistemas com diferentes tamanhos. Os dados em vermelho estão associados à MI e os em preto com MNI.	29
FIGURA 5.5 – Padrões de Turing referência. a) MI; b) MNI	30
FIGURA 5.6 – Simulações do MI para sistemas com área 900 u.a. e diferentes perímetros (p). a) $p = 60$ e área 30×30 u.a. b) $p = 61$ e área 36×25 u.a. c) $p = 65$ e área 45×20 u.a. d) $p = 75$ e área 60×15 u.a. e) $p = 100$ e área 90×10 u.a..	31
FIGURA 5.7 – Simulações do MNI para sistemas com área 900 u.a., CFN com fluxo zero para a temperatura e diferentes perímetros (p). a) $p = 60$ e área 30×30 u.a. b) $p = 61$ e área 36×25 u.a. c) $p = 65$ e área 45×20 u.a. d) $p = 75$ e área 60×15 u.a. e) $p = 100$ e área 90×10 u.a..	32
FIGURA 5.8 – Perfil espacial da temperatura para o MNI em um sistema 30×30 u.a.. Referente a temperatura da simulação apresentada na FIGURA 5.7 a).	32
FIGURA 5.9 – Simulações do MNI para sistemas com área 900 u.a., CFD para a temperatura e diferentes perímetros (p). a) $p = 60$ e área 30×30 u.a. b) $p = 61$ e área 36×25 u.a. c) $p = 65$ e área 45×20 u.a. d) $p = 75$ e área 60×15 u.a. e) $p = 100$ e área 90×10 u.a..	33
FIGURA 5.10 – Representação das trocas de energia térmica dos pontos da malha de integração para os casos de fontes térmicas localizadas nas fronteiras do reator, à esquerda, e banho térmico, à direita. Os termos em vermelho representam os pontos da malha que trocam energia térmica através do processo de condução.	34
FIGURA 5.11 – Simulação do MNI com diferentes fontes térmicas localizadas nas fronteiras do reator. a) $\theta_{\Omega} = 1.0$ b) $\theta_{\Omega} = 1.1$ c) $\theta_{\Omega} = 1.2$ d) $\theta_{\Omega} = 1.3$ e) $\theta_{\Omega} = 1.4$ f) $\theta_{\Omega} = 1.5$ g) $\theta_{\Omega} = 1.6$	35
FIGURA 5.12 – A taxa de produção de entropia total para as diferentes fontes térmicas localizadas nas bordas do meio reacional. As letras (a, b, c, d, e, f, g) indicam os diferentes padrões apresentados na FIGURA 5.11.	36
FIGURA 5.13 – Simulação do MNI com reator de tamanho 60×60 u.a. e diferentes temperaturas de banho térmico. a) $\theta_{EX} = 1.0$ b) $\theta_{EX} = 1.1$ c) $\theta_{EX} = 1.2$ d) $\theta_{EX} = 1.3$ e) $\theta_{EX} = 1.4$ f) $\theta_{EX} = 1.5$ g) $\theta_{EX} = 1.6$	36
FIGURA 5.14 – Funcionais de Minkowski para banhos térmicos com diferentes temperaturas. As letras (a, b, c, d, e, f, g) indicam os diferentes padrões apresentados na FIGURA 5.13.	37

- FIGURA 5.15 – A taxa de produção de entropia associada ao processo de difusão para banhos térmicos com diferentes temperaturas. As letras (*a, b, c, d, e, f, g*) indicam os diferentes padrões apresentados na FIGURA 5.13. . . . 37
- FIGURA 5.16 – Simulação do MNI com gradiente térmico na direção x criado através de um banho térmico (placa de aquecimento), de modo que, $x_i = 1.0$ e $x_f = 1.2$ 38
- FIGURA 5.17 – Versão do MI com diferentes fontes de reagentes químicos nas fronteiras, onde (U_Ω, V_Ω) é para a) (1.80, 2.09), b) (1.90, 2.19), c) (2.00, 2.29), d) (2.10, 2.39), e) (2.20, 2.49), f) (2.30, 2.59). 39
- FIGURA 5.18 – Versão do MNI com diferentes fontes de reagentes químicos nas fronteiras, onde (U_Ω, V_Ω) é para a) (2.10, 1.80), b) (2.20, 1.90), c) (2.30, 2.00), d) (2.40, 2.10), e) (2.50, 2.20), f) (2.60, 2.30). 39
- FIGURA 5.19 – A taxa de produção de entropia para diferentes fontes de reagentes químicos nas fronteiras. Os modelos MI e MNI são representados pelas cores vermelha e preta, respectivamente. As letras (*a, b, c, d, e, f*) indicam as mesmas condições consideradas em simulações prévias, presentes nas FIGURAS 5.17 e 5.18. 40
- FIGURA 5.20 – Funcional de Minkowski para diferentes fontes de reagentes químicos nas fronteiras. Os modelos MI e MNI são representados pelas cores vermelha e preta, respectivamente. As letras (*a, b, c, d, e, f*) indicam as mesmas condições consideradas em simulações prévias, presentes nas FIGURAS 5.17 e 5.18. 40
- FIGURA 5.21 – Versão do MI com $(U_\Omega, V_\Omega) = (2.0666, 2.3567)$ 41
- FIGURA 5.22 – Solução espacial de $u(x, y)$ (à esquerda) e $v(x, y)$ (à direita) no regime estacionário. a) $(U_\Omega, V_\Omega) = (2.00, 2.29)$ b) $(U_\Omega, V_\Omega) = (2.10, 2.39)$ - Ponto de equilíbrio c) $(U_\Omega, V_\Omega) = (2.20, 2.49)$ 42
- FIGURA 5.23 – Taxa de produção de entropia total para o MI, com fontes de ativador (vermelho) e inibidor (azul) nas fronteiras. As letras (*a, b, c, d, e, f*) indicam as mesmas condições consideradas nas simulações prévias, apresentada na FIGURA 5.17, para apenas um dos reagentes. 43
- FIGURA 5.24 – Taxa de produção de entropia total para o MNI, com fontes de ativador (vermelho) e inibidor (azul) nas fronteiras. As letras (*a, b, c, d, e, f*) indicam as mesmas condições consideradas nas simulações prévias, apresentada na FIGURA 5.18, para apenas um dos reagentes. 43
- FIGURA 5.25 – GQIF linear na direção x . As concentrações dos pontos de equilíbrio estão indicados pelas linhas verdes tracejadas. A concentração de U_Ω e V_Ω são representados pelas cores azul e vermelho, respectivamente. O MI e o MNI são dados pelas linhas pontilhadas e sólidas, respectivamente. Note que, o GQIF para 30×30 u.a. está cercado pelas linhas pretas pontilhadas. 44

FIGURA 5.26 – Padrões de Turing modulados por GQIF na direção x . a) MI; b) MNI.	45
FIGURA 5.27 – Simulações de MI e MNI, representados por a e b , respectivamente, em reatores com dimensões iguais a 60×30 u.a..	46
FIGURA 5.28 – Simulações de dois reatores sobre GQIF, considerando o MI, representado por a , e o MNI, representado por b . O gradiente toma a direção do eixo x e em $x = 30$ a concentração é igual ao valor do ponto de equilíbrio.	47
FIGURA 5.29 – Etapas para diferenciação celular, linhas vermelhas, e embriogênese, linha azul, e biogênese, linhas verdes. M - Modulação, QS - Quebra de Simetria, CR - Crescimento, D - Divisão, DA - Divisão Assimétrica, CO - Congelamento.	50
FIGURA 7.1 – Material condutor em contato com reservatórios térmicos nas extremidades, de modo que $T_0 > T_l$. Esta diferença de temperatura induz à um fluxo irreversível de energia térmica (J_H), provocando mudanças de entropia.	72
FIGURA 7.2 – Meio reacional com regiões de diferente concentração de reagentes. A diferença espacial de potencial químico induz um fluxo irreversível de massa (J_M), provocando alterações na entropia.	73

Lista de Algoritmos

ALGORITMO 5.1	Representação simplificada do código relativo ao método de Crank-Nicolson.	23
ALGORITMO 5.2	Representação simplificada do código relativo ao método de Peaceman-Rachford.	24
ALGORITMO 5.3	Representação simplificada do código paralelizável através do uso da biblioteca MPI referente ao método de Peaceman-Rachford.	24

Resumo

TRANSIÇÕES ENTRE ESTADOS ESPACIAIS NÃO HOMOGÊNEOS EM SISTEMAS QUÍMICOS E BIOLÓGICOS: No artigo intitulado “*The Chemical Basis of Morphogenesis*” - 1952, Alan M. Turing propõe que a morfogênese pode ser descrita através de um mecanismo reacional específico que possibilita a emergência de estruturas espaciais não homogêneas e estacionárias de reagentes químicos, conhecidas como "padrões de Turing". Diversos estudos teóricos-experimentais revelam que a influência de parâmetros físico-químicos como incidência de luz, variação de temperatura e condições de fronteira interferem na dinâmica de formação destes padrões. Entre os citados, as condições de fronteira são as fontes de perturbação menos exploradas, porém apresentam grande importância no estudo da dinâmica de sistemas de pequeno tamanho, *e.g.* células. Nesta perspectiva, o presente trabalho apresenta a construção de um programa em linguagem FORTRAN 90 capaz de resolver sistemas de equações diferenciais parciais parabólicas, do tipo reação-difusão, e resultados da investigação teórica dos efeitos das condições de fronteira, realizado através do perímetro do domínio reacional, gradiente de temperatura e gradiente de concentração na emergência dos padrões de Turing. Estas investigações foram realizadas considerando as versões isotérmica e não isotérmica do modelo *Brusselator* em um pequeno reator. Os resultados mostram que: 1) Em situações isotérmicas a variação do perímetro reacional possibilita a emergência de diferentes estruturas espaciais através de um possível fenômeno de sobreposição de padrões de Turing, sem que haja transição de estados. No caso não isotérmico, a temperatura regula fortemente a forma dos padrões emergentes, para qualquer perímetro, impedindo a emergência de estruturas espaciais de diferentes configurações. 2) A temperatura pode ser utilizada como parâmetro de controle na sintonização de diferentes padrões de Turing, de modo que, um sistema sob influência de um gradiente térmico possibilita a emergência de diferentes padrões espaciais simultaneamente. 3) Fontes de reagentes químicos definidos nas fronteiras podem induzir a quebra de simetria espacial de padrões de Turing se a seguinte condição for satisfeita: A concentração dos reagentes químicos, localizados na fronteira do meio reacional, apresentar um valor equivalente ao ponto de equilíbrio do sistema dinâmico homogêneo. Baseados nas ideias de Alan M. Turing e considerando os resultados obtidos neste trabalho, propõe-se modelos químicos mínimos para descrever as etapas iniciais dos processos de diferenciação celular, embriogênese e biogênese.

Palavras-chave: Dinâmica não linear, EDP, condições de fronteira, quebra de simetria.

Abstract

TRANSITIONS AMONG INHOMOGENEOUS SPATIAL STATES IN CHEMICAL AND BIOLOGICAL SYSTEMS: In the article "*The Chemical Basis of Morphogenesis*- 1952, Alan M. Turing proposed that the morphogenesis could be described through a specific reaction mechanism which enables the emergence of inhomogeneous and stationary spatial structures, known as "Turing patterns". Many theoretical and experimental studies revealed that physical and chemical parameters, e.g. radiation, temperature and boundary condition, can interfere on the emergence of those patterns. Among them, the boundary condition is the source of perturbation least explored, however it has a great importance on the study of the dynamics of small systems, e.g. cells. From these perspectives, this work presents the construction of a code, written in FORTRAN 90, capable of solving systems of partial differential equations, reaction-diffusion type, and the results of the theoretical investigation of the effects of the boundary conditions, performed through the perimeter of the system's domain, gradient of temperature and gradient of concentration. This investigation was carried out considering the isothermal and nonisothermal versions of the Brusselator model in a small domain reactor. The outcomes showed that: 1) In isothermal situations the variation of the perimeter enables the emergence of different spatial structures by a possible phenomenon of superposition of Turing patterns. In nonisothermal conditions, the temperature strongly regulates the geometrical formats of the patterns, for any perimeter, preventing the emergence of spatial structures with different configurations. 2) The temperature can be used as parameter of control in the syntonization of different Turing patterns, in a way that a thermal gradient can induce the emergence of different Turing patterns simultaneously. 3) Sources of chemicals defined at the bounders can induce the spatial symmetry breaking of Turing patterns if the following condition is satisfied: The concentration of chemicals, located at the bounders, presents the equivalent value of the equilibrium point of the homogeneous dynamical system. Based on Alan M. Turing ideas and considering the results obtained in this work, a minimal chemical model is proposed to describe the initial steps of the cellular differentiation, embriogenesis, and biogenesis.

Keywords: Nonlinear dynamics, PDE, boundary conditions, symmetry breaking.

Sumário

1 – Introdução	1
1.1 – Alan M. Turing e a morfôgenese	1
1.2 – Observações experimentais	2
1.3 – Fontes de perturbação	5
1.3.1 – Radiação	5
1.3.2 – Temperatura	6
1.3.3 – Geometria e escala do domínio reacional	7
1.4 – Instabilidades através das fronteiras	8
2 – Objetivos	10
3 – Modelo	11
4 – Teoria	15
4.1 – Método numérico de integração	15
4.2 – Análise da estabilidade linear - Instabilidade de Turing	16
4.3 – Funcionais de Minkowski	17
4.4 – Taxa de produção de entropia	18
5 – Resultados e Discussão	22
5.1 – O Programa	22
5.1.1 – Estrutura	22
5.1.2 – Validação	24
5.2 – Regiões de instabilidade de Turing	26
5.3 – Definição do tamanho do meio reacional	28
5.4 – Padrões de Turing referência	30
5.5 – Perímetro das fronteiras	30
5.6 – Gradiente de temperatura	34

5.6.1 – Fonte térmica nas fronteiras do sistema	35
5.6.2 – Banho térmico	36
5.7 – Gradiente de concentração	38
5.8 – Modelos mínimos: Diferenciação Celular, Embriogênese e Biogênese	48
6 – Conclusões	51
Referências bibliográficas	53
7 – Apêndices	59
7.1 – Transformações adimensionais	59
7.2 – Método implícito da direção alternada (ADI)	60
7.3 – Instabilidade de Turing	65
7.4 – Determinação de σ	70
7.4.1 – Reações químicas (σ_{RQ})	71
7.4.2 – Condução de energia térmica (σ_H)	72
7.4.3 – Difusão de massa (σ_M)	73
7.5 – Código computacional em FORTRAN 90	75
7.6 – Pontos de equilíbrio	131

1 – Introdução

1.1 – Alan M. Turing e a morfogênese

Alan Mathison Turing (1912-1954), matemático, pioneiro da ciência da computação e da dinâmica não linear, nasceu em Londres em 23 de junho de 1912.¹ Foi uma personalidade ímpar, com ideias à frente de seu tempo não se restringiu apenas a questionamentos da sua área de formação, colaborando no desenvolvimento da computação, lógica, criptografia e biologia teórica.^{1,2} Seu grande interesse por lógica matemática, influenciado por ideias de cientistas como Arthur Eddington, Bertrand Russel e John von Neumann, levou o matemático britânico à desenvolver trabalhos fundamentais relacionados a teoria da computação a partir do dispositivo metal nomeado "Máquina de Turing".²⁻⁴ Seus principais trabalhos estão entre os mais citados na literatura acadêmica e devido sua relevância conferem a A. Turing o título de "pai da computação".⁶ As ideias mais relevantes do matemático britânico estão apresentadas em três artigos:⁷ "On computable numbers with an application to the *Entscheidungsproblem*" (1936), "Computing machinery and intelligence" (1950) e "The chemical basis of morphogenesis" (1952).^{5,8,9}

No trabalho publicado em 14 de outubro de 1952, A. Turing propõe um modelo teórico para a descrição da morfogênese, construído a partir de considerações físicas e químicas do cenário biológico em questão.^{4,9,10} Apesar de se tratar de sua única contribuição na área da química, o artigo é um marco na história da teoria de sistemas dinâmicos químicos devido suas ideias impactantes.^{1,4} A morfogênese é um dos processos biológicos que participa nos estágios iniciais de desenvolvimento de embriões em organismos eucarióticos.^{11,12} Tal processo é responsável pela aglomeração de populações celulares para possibilitar novas interações bioquímicas e a construção de estruturas complexas em 3D, *e.g.* órgãos e tecidos.¹² Apesar da definição formal dada, no seu artigo A. Turing correlaciona especificamente o fenômeno da morfogênese com o processo de formação de padrões espaciais em tecidos biológicos, *e.g.* manchas das peles em leopardos, em asas de borboletas, periodicidades em listras de peixes e moluscos.¹³⁻¹⁷ Para a descrição deste fenômeno o matemático britânico argumenta que a morfogênese pode ser regulada por sinais químicos de ativação e inibição, ao qual ele nomeou "morfogens", através de um mecanismo reacional específico com cinética química não linear e transporte de massa.⁹ Mais detalhadamente, o modelo mental de A. Turing é composto por um anel de N células, de modo que os componentes químicos reagem no interior de uma dada célula localizada na posição r e difundem para as células vizinhas $r + 1$ e $r - 1$ através de fronteiras semipermeáveis.^{9,18} Este modelo pode ser representado matematicamente e de forma simplificada pela Eq.(1.1):

$$\frac{\partial \vec{u}(r, t)}{\partial t} = \vec{f}(\vec{u}) + \mathbf{D}\nabla^2 \vec{u}(r, t) \quad (1.1)$$

O vetor $\vec{u}(r, t)$ representa as variáveis dependentes macroscópicas, *e.g.* concen-

^a"Máquina" que executa tarefas pré-definidas presentes em um código por meios finitos, mas que não é capaz de decidir, a partir de suas intruções, se determinado processo executará infinitamente.⁵

tração molar, em função das variáveis independentes espaciais r e temporal t . A função \vec{f} representa os termos cinéticos de velocidade de reação, \mathbf{D} é a matriz diagonal dos coeficientes de difusão e ∇^2 o operador Laplaciano associado ao transporte de massa.^{18,19} Na ausência de fenômenos de convecção e em situações de estabilidade temporal, as instabilidades geradas pelo processo de transporte de massa, possibilitam a emergência de estruturas espaciais não homogêneas e estacionárias, chamadas de “Padrões de Turing”, a partir de estados homogêneos.^{4,9,10} Desta forma, as atividades biológicas, reguladas por esses sinais, variam espacialmente em um meio reacional conforme a configuração do padrão, formando diversos tipos de manchas como aquelas observadas em organismos biológicos.⁹ Tais mecanismos ficaram conhecidos como reação-difusão. A partir deste modelo A. Turing consegue então prever a formação de seis padrões espaciais possíveis, caracterizados por pares de vetor de onda e frequência.^{9,20}

Destaca-se o genial vislumbre do matemático britânico na formulação de mecanismos reacionais com cinética química não linear, única forma capaz de gerar estados espaciais não homogêneos.²¹ A. Turing rompe, com a publicação do artigo "The Chemical Basis of Morphogenesis", com as vertentes acadêmicas da época, mais especificamente na matemática aplicada, que se restringia à estudos com sistemas lineares, apenas capazes de amplificar padrões e não formá-los, devido a falta de recursos para o trabalho com problemas não lineares.²¹ A perspicácia de suas ideias reside no fato que os termos não lineares presentes no modelo podem ser tratados de forma aproximada através de suas versões lineares, como foi feito no artigo. E mais, estes termos definem estados temporais estáveis, de modo que o processo de difusão de massa é o responsável pela formação dos padrões e não pela sua destruição, como normalmente é observado.^{9,21} As ideias contidas neste documento marcam uma transição entre a matemática analítica e a matemática computacional, possibilitando consequentemente o desenvolvimento da teoria de sistemas dinâmicos químicos não lineares com problemas relacionados à estados oscilatórios, emergência de padrões espaciais não-estacionários, fenômenos de quebra de simetria e caos.⁴

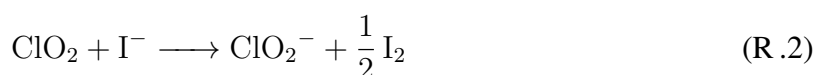
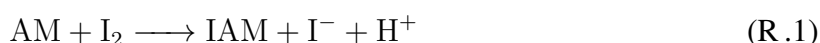
1.2 – Observações experimentais

A primeira observação experimental de padrões espaciais estacionários foi constatada aproximadamente quatro décadas após a publicação do seminal artigo de A. Turing sobre a morfogênese, devido as dificuldades experimentais associadas aos critérios necessários para a emergência destas estruturas.^{4,22} A elegância do mecanismo proposto por A. Turing teve um efeito profundo na comunidade científica, verificado no grande número de pesquisas desenvolvidas sobre este tema.⁴ Tais contribuições foram prioritariamente de cunho teórico uma vez que a observação experimental do fenômeno se mostrou uma tarefa complicada e não trivial.²² Em sistemas químicos, as dificuldades relacionadas a formação de padrões de Turing residem basicamente em dois fatos: (1) O sistema deve ser aberto e muito bem misturado, garantindo que a concentração de certos reagentes seja constante no tempo e no espaço; (2) Os coeficientes

de difusão do ativador e do inibidor devem apresentar diferentes escalas de magnitude;^{4,19} ^b A condição (2) é mais crítica, principalmente em sistemas de solução aquosa.¹⁹ Estas dificuldades foram superadas acidentalmente com a adição de amido na reação clorito - iodeto - ácido malônico (*chlorite - iodide - malonic acid* reaction, CIMA) em um meio reacional bem misturado e com fluxo contínuo de reagentes.^{19,22} A priori o amido foi utilizado como indicador para realçar as ondas químicas presentes nesta reação, contudo, fortuitamente a macromolécula e os íons iodeto ligam-se reversivamente, possibilitando a redução das taxas de difusão dos íons e consequentemente a formação de padrões de Turing.^{19,22}

A reação dióxido de cloro - iodo - ácido malônico (*chlorine dioxide - iodine - malonic acid*, CDIMA), sucessora da reação CIMA, possui mecanismo reacional complexo e não é completamente conhecido.⁴ Com a inédita constatação experimental dos padrões de Turing, muitos trabalhos foram desenvolvidos afim de compreender com mais detalhes o mecanismo reacional da reação CIMA. Tais investigações verificaram que o clorito e o iodeto são consumidos rapidamente, formando as espécies ClO_2 e I_2 , que por sua vez reagem com o ácido malônico (AM), resultando em ondas químicas.⁴ Considerando estas informações, outros estudos demonstraram que se a reação é iniciada com as espécies ClO_2 , I_2 e AM , *i.e.* reação CDIMA, oscilações com mesma frequência e forma de onda da reação CIMA original são obtidas, evitando qualquer evento transiente inicial.^{23,24} Assim sendo, a reação CDIMA difere da reação CIMA apenas em um aspecto, o ativador e o inibidor são produzidos no meio reacional e não bombeados de fontes externas.¹⁹

Apesar dos avanços citados, o mecanismo reacional completo é desconhecido e isto está relacionado com a parte orgânica da reação. A presença do ácido malônico dificulta a formulação de passos elementares devido ao grande número de intermediários orgânicos que podem ser formados.⁴ A parte inorgânica é bem entendida e possibilita a descrição do processo por meio do conjunto de reações (R.1) a (R.3):⁴



Os componentes I^- e ClO_2^- atuam como ativador e inibidor nesta reação, respectivamente.²⁵ Este fato, que não é óbvio, é justificado através das características de auto-inibição do iodeto e de consumo de íons iodeto pelo clorito.²⁵ ^c O AM , como em outras reações os-

^bDe outra forma, $\frac{D_v}{D_u} > 1$, onde D_u e D_v são os coeficientes de difusão do ativador e do inibidor, respectivamente.^{4,9}

^c Pode-se afirmar que o ClO_2^- não atua diretamente nem como um ativador nem como um inibidor, contudo como ele remove espécies ativadoras do meio reacional, o íon pode ser considerado um inibidor para a instabilidade de Turing.²⁵

cilantes, e.g. Belousov-Zhabotinsky^{26,27} e Briggs-Rauscher²⁸, é responsável pela formação do ativador no meio reacional.¹⁹ Na reação CDIMA este fato é representado pela reação (R.1), onde o *AM* reage com I_2 , formando I^- .²⁵

As características mecánísticas apresentadas previamente sugerem que a reação CDIMA seja capaz de produzir comportamentos espaço-temporais muito interessantes, contudo não satisfazem diretamente a principal condição para a emergência de padrões de Turing, *i.e.* o coeficiente de difusão do inibidor deve ser maior que o coeficiente de difusão do ativador.⁴ Experimentalmente esta condição foi satisfeita com a adição de amido (*starch*, *S*) no meio reacional.^{19,29} Os aspectos estruturais do amido, macromolécula constituída por uma combinação de amilóse e amilopectina que confere ao polissacarídeo resultante uma estrutura helicoidal e ramificada, possibilitam o aprisionamento reversível de íons iodeto, na forma de íon triiodeto, reduzindo significativamente sua capacidade difusiva.^{19,29,30} A redução da mobilidade dos íons I^- é acentuado com a inclusão do amido em uma fase gelatinosa composta por gel agarose.^{19,29} A interação entre o amido e o ativador é representada pela reação (R.4):



A partir destas considerações os critérios de A. Turing são satisfeitos, possibilitando assim a emergência de padrões espaciais estacionários de reagentes químicos.²⁹ Os resultados obtidos a partir da reação CDIMA impulsionaram investigações experimentais com outros sistemas químicos similares o que proporcionou a observação de padrões de Turing, *e.g.* nas reações de iodato-sulfeto-tiosulfato (*iodate-sulfite-thiosulfate*) e poliacrilamida-azul de metileno-sulfeto-oxigênio (*polyacrylamide-methylene blue-sulfide-oxygen*).^{31,32}

A dinâmica da reação CDIMA é descrita pelo modelo de Lengyel-Epstein, nomeado em reconhecimento aos cientistas que o propuseram.²⁵ O modelo, composto por duas equações diferenciais parciais que descrevem o comportamento dos agentes ativador e inibidor, é obtido a partir de reduções embasadas em aspectos físico-químicos relacionados à cinética da reação de um sistema inicialmente formado por cinco variáveis dependentes, ou seja, um sistema de cinco equações diferenciais parciais. O modelo de Lengyel-Epstein é reconhecido e muito utilizado em investigações sobre esse tema devido a boa correspondência entre os resultados teóricos e experimentais.²⁵

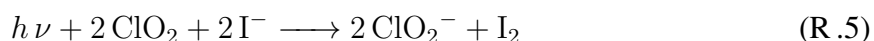
Vale observar que em sistemas biológicos as dificuldades para a observação de padrões de Turing estão relacionadas a identificação dos agentes ativador e inibidor e na demonstração de que o sistema homogêneo é estável.⁴ Apesar destes obstáculos, os avanços científicos na área de desenvolvimento biológico possibilitaram a identificação do primeiro morfogen no zigoto da mosca *Drosophila melanogaster*, o fator de transcrição Bicoid (BCD).³³ Outros trabalhos também apresentam pequenas moléculas com características similares, *e.g.* AMP cíclico na *Dictyostelium discoideum* e do ácido retinoico no desenvolvimento de vertebrados.^{13,34}

1.3 – Fontes de perturbação

Estados químicos espacialmente organizados são altamente afetados por fontes de perturbação externa, *e.g.* incidência de radiação, variação de temperatura, geometria e escala do domínio reacional. Estas perturbações levam o sistema a estados finais imprevisíveis possibilitando grande variabilidade de estruturas formadas.

1.3.1 – Radiação

Defini-se processos fotoquímicos como aqueles que são iniciados pela absorção de radiação por pelo menos um componente da mistura reacional. Estes processos podem ser classificados de acordo com o mecanismo de formação de produtos: *processos primários*, onde os produtos são formados a partir do estado excitado do reagente, e *processos secundários*, onde os produtos são produzidos por intermediários formados a partir do estado excitado do reagente.³⁵ Estas fontes de radiação agem como forças externas de interação, induzindo ao consumo/produção de reagentes de acordo com sua intensidade e frequência.³⁵ Devido às estas características uma fonte de radiação pode ser utilizada como parâmetro de controle de uma reação química.³⁶ Investigações considerando a fotossensibilidade do iodo na reação CDIMA mostram que a incidência de luz visível sobre as estruturas auto-organizadas é uma ferramenta muito conveniente de controle externo de padrões.³⁶ Neste caso, a radiação visível suprime oscilações através do consumo de íons iodeto e produção de íons clorito.^{19,36} A reação global deste fenômeno pode ser apresentada como:



de modo que a cinética deste processo é descrita pela Eq. (1.2):

$$r = \Phi W \quad (1.2)$$

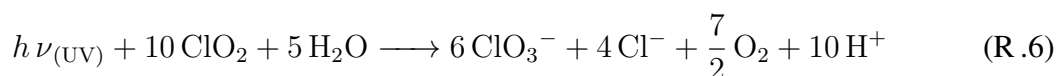
Onde $W(Ms^{-1})$ é a taxa de absorção de fótons actínicos^d dividido pelo volume do reator e Φ é o rendimento quântico.³⁶ ^e Desta forma, diferentes padrões de Turing podem ser modulados através da variação da intensidade de radiação luminosa no meio reacional para um mesmo conjunto de parâmetros.

A radiação ultravioleta (UV) também foi utilizada para perturbar padrões de Turing através da reação CDIMA. Os estudos desenvolvidos considerando luz UV mostraram que na ausência de outras fontes de radiação há a emergência de estruturas compostas por pontos encolhidos e organizados em uma rede hexagonal irregular.³⁸ Diferentemente da luz visível que atua principalmente no consumo de íons iodeto, as investigações experimentais sugerem que a radiação UV contribui de forma efetiva para a decomposição do dióxido de cloro, afetando in-

^d Associado a radiação eletromagnética que é capaz de iniciar uma reação química.³⁷

^e $\Phi = \frac{v}{I_{abs}}$, onde v é a velocidade do processo e I_{abs} intensidade de luz absorvida.³⁵

diretamente os agentes inibidor e ativador da reação. A reação global proposta para representar este processo é dada abaixo:³⁹



Uma relação de competitividade é estabelecida quando as fontes de radiação UV e luz visível são combinadas através da reação fotoquímica de decomposição do dióxido de cloro, impedindo a supressão total de padrões em intensidades intermediárias de luz visível e possibilitando então a formação de novas estruturas auto-organizadas.³⁸

1.3.2 – Temperatura

A dinâmica de processos químicos depende da temperatura através de parâmetros de equilíbrio e não-equilíbrio como: coeficientes térmicos e mecânicos, viscosidade, coeficientes de transporte e constante de velocidade.⁴⁰⁻⁴² Apesar desta relação de dependência ser de forma indireta, através de outros parâmetros físicos, ela também é sensível, ou seja, pequenas variações nesta variável de estado podem dirigir o sistema para regiões de comportamento imprevisível. Em mecanismos de reação-difusão (Eq. (1.1)), sem influências de campos externos, mudanças de temperatura alteram essencialmente dois parâmetros fundamentais^{40,42} : as constantes de velocidade (k), que podem ser descritas pela equação de Arrhenius,

$$k = Ae^{-\frac{E_a}{RT}} \quad (1.3)$$

e a constante de difusão (D), dada pela equação de Eintein-Stokes,

$$D = \frac{k_B T}{6\pi\eta r} \quad (1.4)$$

Na Eq. (1.3), A é o fator pré-exponencial, E_a é a energia de ativação, R é a constante universal dos gases ideais e T a temperatura.³⁵ Na Eq. (1.4), k_B é a constante de Boltzmann, T é a temperatura, η é a viscosidade do fluido que compõe o meio e r é o raio de uma partícula esférica.⁴³ Estas equações demonstram que a temperatura afeta exponencialmente a cinética das reações e linearmente o transporte de massa, em outras palavras, todos os fenômenos físico-químicos deste processo são perturbados – fato que explica a sensibilidade da dinâmica às alterações de temperatura. Desta forma, a temperatura, como a radiação apresentada na seção anterior, pode ser tomada como parâmetro de controle de reações químicas, induzindo a emergência de diferentes estados dinâmicos.⁴⁰⁻⁴²

Considerando as informações expostas previamente, investigações teóricas e experimentais foram conduzidas a fim de se analisar os efeitos das perturbações associadas à variação da temperatura no comportamento dinâmico de sistemas químicos fora do equilíbrio, confirmando as previsões esperadas. Observações experimentais de transições entre estados

espaciais, mais especificamente de um estado uniforme para padrões de Turing hexagonais e listrados, foram feitas a partir da reação CIMA, conduzidas pela variação da temperatura.⁴⁴ Transições de mesma natureza também foram reportadas a partir de uma abordagem teórica através da versão não isotérmica do modelo de Grey-Scott.⁴⁰ Mais ainda, outras versões não isotérmicas de modelos conhecidos, *e.g.* Oregonator, foram estudadas a fim de se analisar os efeitos da temperatura em regimes oscilatórios, demonstrando a existência de limites de temperatura para a observação destes estados.⁴⁵ Vale ressaltar que nas abordagens teóricas de modelos não isotérmicos de reação-difusão, a equação que descreve a dinâmica da temperatura é acoplada às equações de velocidade dos reagentes químicos apenas através das constantes de velocidade, *i.e.* equação de Arrhenius, de modo que a difusão permanece constante no tempo e no espaço.⁴⁰ Isto é justificado a partir da premissa que o meio reacional é idealmente diluído, de modo que as variações na temperatura não alteram significativamente o valor da constante de difusão.⁴⁰

1.3.3 – Geometria e escala do domínio reacional

A análise detalhada dos padrões formados em tecidos de animais permite a constatação de que tais estruturas não são uniformes em todo o corpo, principalmente padrões formados em caudas e em junções das patas dianteiras com o dorso, e se alteram em forma durante o processo de desenvolvimento dos indivíduos.^{14,15} A não uniformidade de padrões em peles de animais está diretamente relacionada com as alterações da geometria do domínio em que tais estruturas emergem. A presença de listras nas pontas das caudas do leopardo, jaguar e chetaah são as evidências mais notáveis da relação entre padrão e geometria do domínio.^{14,15} Trabalhos teóricos considerando um modelo de reação-difusão e um domínio reacional com formato cônico, a fim de mimetizar uma cauda, obtiveram resultados qualitativamente equivalentes àqueles observados em animais, com uma "transição"^f de padrões em forma de bolas para listras conforme se aproxima da ponta do cone.^{14,15} Resultados similares foram obtidos através de simulações com domínios representando junções entre dorso e pata de zebras e a pele do peixe-anjo (*Pomacanthus*).¹⁵⁻¹⁷ De modo mais fundamental, outro trabalho foi desenvolvido com o intuito de compreender como os padrões se auto-organizam em domínios esféricos. Neste caso, os pesquisadores relatam que as estruturas com forma de listras são orientadas através do formato do domínio do meio reacional, gerando anéis.⁴⁶ E mais, estruturas compostas por bolas se organizam em redes bem definidas, de modo que as bolas ocupam posições relacionadas a vértices de polígonos, *e.g.* uma rede de seis bolas se organizam no sentido de ocupar as posições dos vértices de um octaédrio.⁴⁶

O desenvolvimento dos indivíduos é acompanhada por mudanças nas estruturas de Turing devido o aumento da escala dos domínios. O tamanho do domínio reacional é decisivo na forma dos padrões emergentes, provocando alterações significativas, *e.g.* sistemas de

^fA partir das investigações deste trabalho, acredita-se que não há transições entre estados espaciais a partir da variação da área, mas sim um fenômeno sobreposição de padrões, formando estruturas com formato diferente. Veja em Resultados e Discussão 5.5.

pequeno tamanho podem impossibilitar a emergência de padrões espaciais, ao passo que o aumento contínuo da escala do domínio pode provocar bifurcações sucessivas e conseqüentemente o aparecimento e desaparecimento de diferentes estruturas de Turing.^{14,15,17} De maneira geral, o aumento do tamanho do domínio propicia a formação de mais estruturas espaciais, entretanto em sistema de tamanho muito grande a estrutura dos padrões é tão fina que acaba levando a formação de estados uniformes, ou seja, desaparecimento dos padrões.^{14,15,17}

Observa-se que, no processo de desenvolvimento biológico dos animais alterações tanto na escala quanto na forma do domínio reacional ocorrem simultaneamente. Assim sendo, os efeitos da variação destes dois componentes possibilitam a emergência de novos estados espaciais com diferentes configurações.^{15,17}

1.4 – Instabilidades através das fronteiras

A hipótese de A. Turing para a descrição da morfogênese pressupõe que as atividades biológicas variam espacialmente conforme a configuração da estruturas espaciais emergentes dos agentes ativador e inibidor. Assim sendo, a formação de diferentes padrões implica na formação de células com diferentes especializações. Essa variabilidade nas especificações das células podem ser atingidas a partir de mudanças dos parâmetros apresentados na seção anterior, o que justifica os estudos sobre seus efeitos na emergência de padrões de Turing. Entretanto, estas investigações negligenciam os efeitos das condições de fronteira, focando apenas nas alterações produzidas no bulk da reação.⁴⁷ Esta abordagem pode surtir conseqüências onerosas, uma vez que as instabilidades geradas pelas condições de fronteira podem alterar significativamente a dinâmica do meio reacional, principalmente no tratamento de sistemas de pequeno tamanho e sob influência de gradientes de concentração química.¹¹

As soluções de modelos de reação-difusão podem ser estudadas teóricamente a partir de três condições de fronteira possíveis: Dirichlet, Neumann e Periódica. Matematicamente tais fronteiras podem ser definidas da seguinte forma⁴⁸ :

Dirichlet	Neumann	Periódica
$u(r_0) = f_1(r)$	$\left. \frac{du(r)}{dr} \right _{r=r_0} = f_1(r)$	$u(r_0) = u(r_F)$
$u(r_F) = f_2(r)$	$\left. \frac{du(r)}{dr} \right _{r=r_F} = f_2(r)$	

com $u = u(r)$, $r_0 \leq r \leq r_F$ e $f_1(r)$, $f_2(r)$ são funções.

Físicamente as condições de fronteira de Dirichlet representam fontes/sumidouros da variável dependente nas bordas, as fronteiras de Neumann estão associadas à fluxos de entrada ou saída de u do sistema e as condições de fronteira Periódica representam a formação de estruturas toroidais, ou seja, um sistema com configuração espacial que não possui fronteiras físicas.⁴⁸

Trabalhos apontam que os estudos de padrões de Turing são em geral desenvolvidos com fronteiras de Neumann, mais especificamente com $f_1(r) = f_2(r) = 0$, ou seja, fronteiras com fluxo zero.^{11,47} Esta abordagem implica que as bordas do meio reacional possuem um caráter puramente reflexivo, perturbando minimamente a dinâmica do processo em questão.⁴⁹ Entretanto, a desconsideração dos efeitos da fronteira na formação de padrões de Turing, principalmente em sistemas biológicos, pode empobrecer a descrição da dinâmica do processo.¹⁶

Os poucos trabalhos desenvolvidos a partir de condições de Dirichlet mostraram que as instabilidades geradas pelas fronteiras são de curto alcance, de modo que as perturbações em um sistema de grande dimensão espacial são locais (próximas a fronteira) e não atingem o bulk da reação.¹⁶ Outro resultado reportado é sobre a existência de uma tendência de orientação dos padrões a partir da localização da fonte/sumidouro de reagente químico no meio reacional.¹⁶ Mais investigações também indicam que condições de fronteira não homogêneas[§] podem gerar soluções menos sensíveis a alterações do tamanho do domínio, que como foi já foi apresentado, provoca transições entre estados espaciais em sistemas com fronteiras de fluxo zero.⁴⁹ Estas constatações comprovam o papel central que as condições de fronteira possuem no fenômeno de emergência de padrões de Turing, sendo parâmetro fundamental para a completa descrição da formação de manchas em asas das borboletas *Lepidoptera*, na pele do peixe-anjo e em padrões de esqueleto em membros em desenvolvimento.^{16,50,51}

[§]Situação em que as condições de fronteira são definidas por uma função e não por um escalar, *i.e.* condições homogêneas.

2 – Objetivos

O presente trabalho tem como objetivo principal desenvolver um código computacional, utilizando a linguagem FORTRAN 90, capaz de resolver sistemas de equações diferenciais parciais (EDP's), do tipo reação-difusão, considerando os métodos numéricos de integração de Crank-Nicolson (para 1-D), Peaceman-Rachford (para 2-D) e Runge-Kutta. Posteriormente à construção do *software*, realizar investigações sistemáticas dos efeitos das perturbações associadas as condições de fronteiras através do perímetro do domínio reacional, gradiente de temperatura e gradiente de concentração na emergência de padrões de Turing. Tais investigações são feitas considerando as versões isotérmica e não isotérmica do modelo *Brusselator*. Os resultados são analisados com o auxílio de funcionais de Minkowski, para a caracterização morfológica dos padrões formados, e da taxa de produção de entropia, para a identificação de fenômenos de quebra de simetria. A discussão dos dados obtidos está fundamentada nas considerações de Alan Turing sobre a morfogênese, com a intenção de identificar eventos de relevância significativa principalmente para o processo de desenvolvimento biológico.

3 – Modelo

A construção do modelo foi inspirada pelas referências^{40,42}.

O modelo *Brusselator* foi proposto por Ilya Prigogine e colaboradores como um exemplo de mecanismo capaz de se auto-organizar na presença de instabilidades termodinâmicas.^{52,53} Tais características e a presença de uma etapa de autocatálise trimolecular resultaram em um grande número de críticas da comunidade acadêmica com respeito a validade do modelo para a representação de processos reais e ao cumprimento das leis de conservação. Entretanto, foi mostrado que tais alegações são infundadas, que o passo trimolecular pode ser representado por etapas bimoleculares sucessivas sem alteração da dinâmica e que as leis de conservação são obedecidas.⁵³ Como este modelo é muito conhecido e através dele diversos processos dinâmicos foram amplamente explorados, o *Brusselator* foi considerado para o desenvolvimento deste trabalho. O modelo é escrito da seguinte forma:



Nas reações acima, U e V são intermediários reacionais tomados como variáveis dependentes do tempo e do espaço, classificados como ativador e inibidor, respectivamente. B e D são espécies químicas com concentração constante no tempo, e A e E são a fonte e o sumidouro do intermediário U , respectivamente. As etapas irreversíveis, (R.7) e (R.10), representam a entrada ($A \xrightarrow{\lambda_v} U$) e a saída ($U \xrightarrow{\lambda_v} E$) de matéria, neste caso U , entre o sistema e as vizinhanças, respectivamente. As etapas reversíveis, (R.8) e (R.9), caracterizam as transformações químicas dentro do sistema.

A FIGURA 3.1 representa as principais características do sistema. A fonte, A , fornece constantemente o reagente químico U para o reator, com fluxo volumétrico, \dot{v}_0 , e temperatura externa, T_E . De forma similar, U sai do reator através do sumidouro E com temperatura interna T e mesmo fluxo \dot{v}_0 . O reator, que apresenta geometria retangular e volume constante (v_0), possui fronteira bem definida com temperatura constante, T_Ω , e um gradiente de concentração químico dos dois intermediários reacionais, U_Ω e V_Ω , imposto através de todo o seu perímetro. Dentro do sistema, os reagentes químicos interagem com as bordas, trocando energia térmica e matéria, e reagem entre si exotermicamente.

A partir da construção apresentada e assumindo a lei de ação das massas e equação Arrhenius, para o acoplamento entre energia térmica e matéria através das constantes de

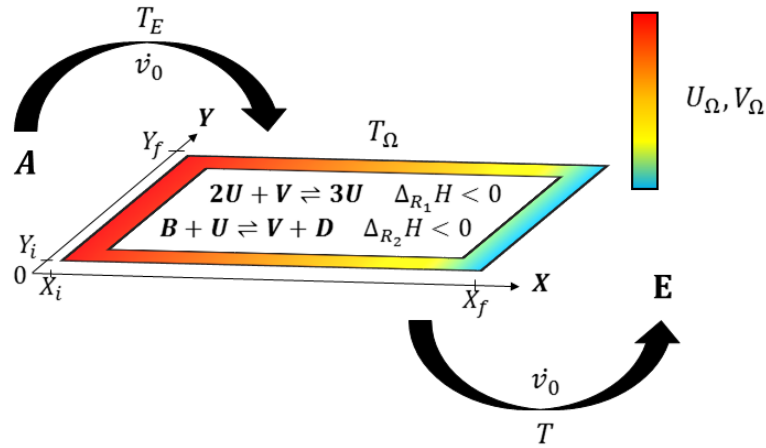


FIGURA 3.1 – Representação esquemática da dinâmica do sistema construído.

velocidade, as componentes de velocidade das reações (R.8) e (R.9) podem ser escritas da seguinte forma:

$$R_1 = -K_{f_1} e^{\frac{E_1}{R}(\frac{1}{T_E} - \frac{1}{T})} U^2 V + K_{r_1} e^{\frac{E_2}{R}(\frac{1}{T_E} - \frac{1}{T})} U^3 \quad (3.1)$$

$$R_2 = -K_{f_2} e^{\frac{E_3}{R}(\frac{1}{T_E} - \frac{1}{T})} U B + K_{r_2} e^{\frac{E_4}{R}(\frac{1}{T_E} - \frac{1}{T})} V D \quad (3.2)$$

Nas Eqs. (3.1) e (3.2), K_{f_j} e K_{r_j} são as constantes de velocidade das reações direta e reversa, respectivamente. ^h E_i é a energia de ativação de cada reação, R é a constante universal dos gases ideais e a temperatura externa T_E é tomada como referência para a definição das constantes de velocidade. O modelo isotérmico (MI) é composto pelas Eqs. (3.3) e (3.4).

$$\frac{\partial U}{\partial t} = \lambda(A - U) - R_1 + R_2 + D_U \nabla^2 U \quad (3.3)$$

$$\frac{\partial V}{\partial t} = R_1 - R_2 + D_V \nabla^2 V \quad (3.4)$$

Nas equações acima D_U e D_V são os coeficientes de difusão das variáveis U e V , respectivamente, e λ_v é o espaço de velocidade do reator. ⁱ Nesta versão, a temperatura interna é mantida constante durante todo o tempo. Diferentemente, no modelo não isotérmico (MNI) a energia térmica é tratada como uma variável dependente e sua dinâmica é descrita pela Eq. (3.5).

Na Eq. (3.5), o primeiro termo à direita do sinal de igualdade ($\beta A_r(T_\Omega -$

^hNote que, a equação de Arrhenius é dada por $k = A e^{-E/RT_E}$, de modo que o fator pré-exponencial pode ser escrito como: $A = k e^{E/RT_E}$. Assim, nas Eqs. (3.1) e (3.2) K_{f_j} e K_{r_j} são constantes de velocidade definidas em T_E .

ⁱDe outra forma, λ_v representa o inverso do tempo de residência.⁵⁴

T) representa a lei de resfriamento de Newton, o segundo termo ($\dot{v}_0 \lambda C_{p,U}(UT - AT_E)$) refere-se a energia térmica associada aos fluxos de entrada e saída da matéria, o terceiro termo ($v_0(\Delta H_{R_1} R_1 + \Delta H_{R_2} R_2)$) é relativo a energia térmica produzida/consumida na reação química e o último termo ($v_0 \alpha \nabla^2 T$) representa a difusão de energia térmica. Nesta mesma equação, C_T é a concentração molar total do meio, C_{ps} é a capacidade calorífica do meio, $C_{p,U}$ capacidade calorífica do reagente U, β é o coeficiente de transferência de calor do reator, A_r é a área de transferência de calor, ΔH_{R_1} e ΔH_{R_2} são as variações de entalpia das reações (R.8) e (R.9), respectivamente, e α é o coeficiente de difusão térmica do meio, que é expresso como $\alpha = \frac{\kappa}{C_T C_{ps}}$ com κ o coeficiente de condutividade térmica.

$$v_0 C_T C_{ps} \frac{\partial T}{\partial t} = \beta A_r (T_\Omega - T) - \dot{v}_0 \lambda C_{p,U} (UT - AT_E) + v_0 (\Delta H_{R_1} R_1 + \Delta H_{R_2} R_2) + v_0 \alpha \nabla^2 T \quad (3.5)$$

A entalpia é uma função da temperatura, de modo que estes termos estão relacionados através da diferença da capacidade calorífica entre os produtos e reagentes. Matematicamente esta relação é expressa da seguinte forma:

$$\Delta H_i = \Delta H_i^0 + \int_{T_E}^T \Delta C_{p,i} dT \quad i = 1, 2 \quad (3.6)$$

Onde ΔH_i^0 representa a variação da entalpia padrão da reação i e $\Delta C_{p,i}$ corresponde à diferença da capacidade calorífica entre os produtos e reagentes da reação i . Como foi proposto pela referência⁴⁰, este trabalho admite que a reação ocorre em uma fase líquida homogênea, o que implica $\Delta C_{p,i} \approx 0$ e $\Delta H_i = \Delta H_i^0$, ou seja, a entalpia passar a ser uma constante.

A partir das considerações feitas previamente, observa-se que as equações que descrevem a dinâmica dos modelos são compostas por um grande número de parâmetros. Apesar destes termos serem constantes, em grande quantidade eles tornam o tratamento analítico das equações, no estudo qualitativo das trajetórias das soluções, uma tarefa árdua. Para superar tal impasse, os modelos são reescritos na forma adimensional. Esta estratégia possibilita a redução do número de parâmetros físico-químicos, através de considerações físicas e matemáticas, sem alterar a estrutura matemática das equações que descrevem a dinâmica do processo. Os modelos adimensionais estão apresentados a baixo:

$$\frac{\partial u}{\partial \tau} = \xi(a - u) - r_1 + r_2 + d_u \nabla^2 u \quad (3.7)$$

$$\frac{\partial v}{\partial \tau} = r_1 - r_2 + d_v \nabla^2 v \quad (3.8)$$

$$\frac{\partial \theta}{\partial \tau} = \phi(\theta_\Omega - \theta) - \xi \gamma_u (u\theta - a) + \Delta H'_{r_1} r_1 + \Delta H'_{r_2} r_2 + d_\theta \nabla^2 \theta \quad (3.9)$$

com,

$$r_1 = -k_{f_1} e^{e_1(1-\frac{1}{\theta})} u^2 v + k_{r_1} e^{e_2(1-\frac{1}{\theta})} u^3 \quad (3.10)$$

$$r_2 = -k_{f_2} e^{e_3(1-\frac{1}{\theta})} u b + k_{r_2} e^{e_4(1-\frac{1}{\theta})} v d \quad (3.11)$$

Onde τ , u , v e θ são o tempo, a concentração de U , a concentração de V e a temperatura adimensionais, respectivamente. As transformações propostas podem ser encontradas no Apêndice 7.1 .

As Eqs. (3.7) e (3.8) representam o MI adimensional e acoplando a Eq. (3.9) à estas duas, o MNI adimensional. Estes dois modelos foram utilizados em todas as investigações realizadas neste trabalho.

4 – Teoria

Nesta seção estão apresentados os conceitos fundamentais utilizados para o desenvolvimento desta pesquisa.

Como foi previamente exposto, o principal objetivo deste estudo é compreender efeitos de determinadas perturbações na emergência de padrões de Turing. Para completar tal tarefa ferramentas matemáticas são necessárias para a identificação de situações em que as estruturas de Turing são formadas, assim como, para descrever a dinâmica de tais processos. Uma vez que os padrões espaciais são formados, ferramentas físico/matemáticas são utilizadas para a caracterização quantitativa dos padrões, e de maneira similar, para a identificação de fenômenos de transição entre estados.

4.1 – Método numérico de integração

A evolução espaço-temporal de um modelo de reação-difusão é obtida através da integração das equações que descrevem a dinâmica do sistema. Entretanto, devido as características destas equações, *i.e.* sistemas acoplados com termos não lineares, não é possível obter soluções analíticas, exigindo então a utilização de aproximações numéricas. Neste trabalho os métodos de Crank-Nicolson e Peaceman-Rachford foram considerados para a descrição dos termos lineares das equações em uma e duas dimensões espaciais, respectivamente, e o método de Runge-Kutta de 4ª ordem para a aproximação dos termos não lineares.⁵⁵ Este tipo de tratamento pode ser exemplificado da seguinte forma:

Considere a seguinte equação diferencial parcial:

$$\frac{\partial u(x, y, t)}{\partial t} = f(u) + D_u \left(\frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} \right) \quad (4.1)$$

Na Eq. (4.1), $f(u)$ é uma função de termos não lineares de u e D_u uma constante. Esta equação pode ser discretizada, sem perda de informação física, através de expansões e combinações de séries de Taylor:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = f(u) + D_u \left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta y^2} \right) \quad (4.2)$$

Na Eq. (4.2), os termos em azul são lineares e são tratados através dos métodos de Crank-Nicolson ou Peaceman-Rachford, dependendo da dimensão espacial considerada, e o termo em vermelho é a função não linear, tratada através do método de Runge-Kutta. Note que tal abordagem só pode ser feita se $f(u)$ não depender explicitamente das variáveis independentes. Caso contrário, este termo deve ser aproximado pelos métodos de Crank-Nicolson ou Peaceman-Rachford.

Veja no Apêndice 7.2 uma descrição explícita do método numérico de Peaceman-Rachford, também conhecido como Método Implícito da Direção Alternada (*Alternating Direction Implicit Method*, ADI).

Todas as simulações executadas neste trabalho foram realizadas com passo numérico no tempo $\Delta t = 0.01$ e passos numéricos no espaço $\Delta x = \Delta y = 0.5$. As equações foram resolvidas em sistemas com geometria retangular, condições de fronteira do tipo Dirichlet (CFD), Neumann (CFN) e periódicas e com tempo suficiente para as soluções atingirem os estados estacionários.

4.2 – Análise da estabilidade linear - Instabilidade de Turing

A análise da estabilidade linear é uma estratégia matemática utilizada para caracterizar localmente a geometria da estrutura da órbita de um sistema dinâmico específico. Esta caracterização qualitativa é feita ao redor de pontos de equilíbrio, *i.e.* estados estacionários, considerando as mudanças na estrutura da órbita em função de parâmetros do sistema na sua forma linear.⁵⁶ Esta investigação possibilita a classificação das soluções do sistema como estáveis ou instáveis através da determinação dos expoentes de Lyapunov (λ_i).⁵⁶

O estudo do comportamento das órbitas de um sistema dinâmico não linear através de sua versão linearizada é garantido pelo *teorema de Hartman-Grobman* sob certas condições.⁵⁷ Este teorema afirma que um sistema de equações não lineares de dimensão- n apresenta comportamento qualitativamente equivalente ao do sistema linear correspondente, em torno de um ponto de equilíbrio hiperbólico. Neste caso é dito que o retrato de fases de tal sistema é topologicamente orbitalmente equivalente ao do sistema linear. Veja a FIGURA 4.1.⁵⁷ Lembrando que:

- Um ponto é de equilíbrio hiperbólico quando todos os autovalores, calculados a partir da versão linearizada das equações originais, apresentam parte real não nula ($Re(\lambda_i) \neq 0$);^{56,57}
- Dois espaços vetoriais são topologicamente orbitalmente equivalentes quando um é a versão distorcida do outro;^{56,57}

Assim sendo, só é possível classificar as soluções de um sistema com relação à sua estabilidade se os expoentes de Lyapunov apresentarem parte real não nula. Caso contrário, outras abordagens devem ser consideradas, como por exemplo o método direto de Lyapunov.^{56,57}

Através desta estratégia matemática é possível determinar condições de emergência de padrões de Turing. A instabilidade de Turing é gerada por perturbações espaciais associadas ao processo de transporte de massa em um sistema que possua soluções temporais estáveis.⁹ As condições para a formação de padrões de Turing são bem conhecidas para sistemas com duas variáveis dependentes e elas são expressas em uma forma analítica elegante e

^jOs expoentes de Lyapunov são os autovalores calculados a partir da versão linear do sistema.

genérica.⁹ Entretanto, a determinação de expressões analíticas para a instabilidade de Turing para sistemas com mais de duas variáveis dependentes pode ser muito complicado. Devido a isto, nestes casos as características dos expoentes de Lyapunov são analisadas considerando os critérios de estabilidade de Routh-Hurwitz a partir do polinômio característico.^{40,57} Uma descrição completa da determinação de regiões de instabilidade de Turing em sistemas com duas e três variáveis dependentes estão apresentadas no Apêndice 7.3.

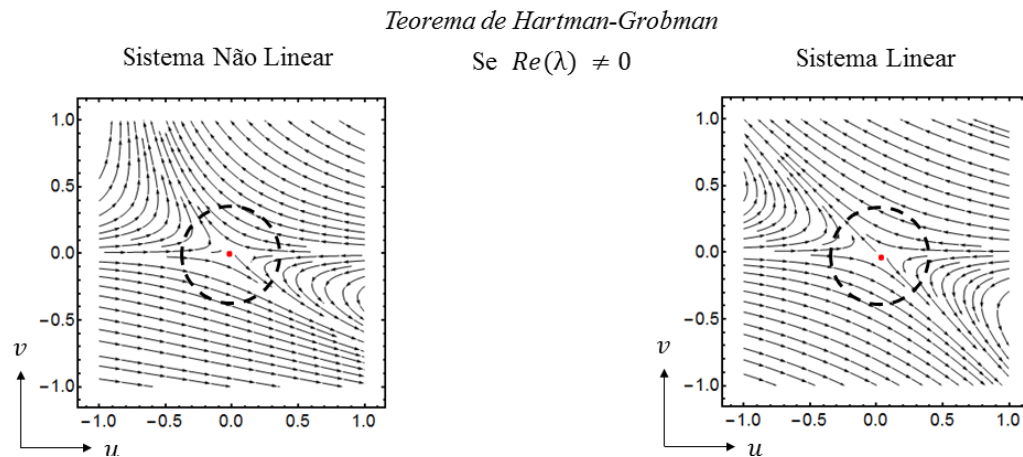


FIGURA 4.1 – Retratos de fase de um sistema de EDO's de termos não lineares (à esquerda) e lineares (à direita), com variáveis dependentes u e v , e ponto de equilíbrio hiperbólico $(u^*, v^*) = (0, 0)$ (ponto em vermelho). Como o ponto de equilíbrio é hiperbólico, os retratos de fase dos dois sistemas são qualitativamente equivalentes, diferenciando entre si por distorções. Nas vizinhanças do ponto de equilíbrio, as órbitas são muito similares, o que permite classificar através da versão linear a estrutura e a estabilidade da órbita nesta região. Neste exemplo genérico, as órbitas apresentam uma estrutura de ponto de sela, implicando comportamento instável.

4.3 – Funcionais de Minkowski

De forma geral, os padrões de Turing bidimensionais podem emergir com três configurações espaciais completamente diferentes: arranjos hexagonais de pontos/bolas, listras e hexágonos invertidos.^k Entretanto, estruturas com configurações resultantes da combinação destes três padrões podem ser formadas em situações em que o sistema é perturbado. Casos como estes dificultam a identificação e classificação das estruturas espaciais sem a utilização de um método quantitativo. Tais impasses são superados através de funcionais de Minkowski (Γ).^{40,58}

Os funcionais de Minkowski são utilizados para fazer medidas morfológicas dos padrões de Turing e com isso descrever propriedades geométricas e topológicas, *e.g.* área, conectividade entre as estruturas, comprimento das fronteiras e outros.⁵⁸ Os funcionais mais comuns são de cálculo de áreas pretas e brancas, comprimento das fronteiras de grupos formados e excentricidade média dos objetos que compõe a imagem.⁵⁸ Apesar deste grande número

^kObserve que é possível classificar listras como invertidas ou não dependendo da cor do fundo do gráfico.

de funcionais apresentados, neste trabalho a excentricidade média e a área de grupos em preto e branco são considerados de forma combinada⁴⁰ :

$$\Gamma = \begin{cases} \langle e \rangle, & \text{se } A_w < A_b \\ -\langle e \rangle, & \text{se } A_w > A_b \end{cases}$$

Onde $\langle e \rangle$ representa a excentricidade média, A_w e A_b são as áreas de pixel brancos e pretos, respectivamente.

Para a determinação de Γ é necessário transformar as imagens dos padrões de Turing em figuras binárias, preto e branco. Em seguida, com a utilização de um algoritmo computacional de análise de imagens (através do *software Mathematica* - "*ComponentMeasurements*"), determina-se as áreas formadas por pixels brancos e pretos e a média da excentricidade das estruturas que compõe a imagem. A essência das estruturas formadas depende diretamente das áreas, de modo que se a área em branco é maior que a área em preto implica que os padrões são construídos em preto e considera-se que o padrão é invertido. A situação oposta classifica as estruturas como não invertidas.⁵⁸ A média da excentricidade possibilita identificação das formas das estruturas. A descrição das etapas para o cálculo do funcional de Minkowski estão apresentados de forma esquemática na FIGURA 4.2.

O termo Γ é capaz de classificar quantitativamente a morfologia das estruturas espaciais bidimensionais de Turing.

4.4 – Taxa de produção de entropia

As palavras de Rudolf Clausius "A energia do universo é constante. A entropia do universo tende a um máximo." resumem os resultados fundamentais da termodinâmica clássica, incorporados na primeira e segunda leis.¹⁰ Essencialmente, esta teoria busca descrever as mudanças de energia e entropia, *i.e.* funções de estado, usando o conceito de processos reversíveis infinitamente lentos, através de parâmetros como temperatura, pressão, volume e quantidade de matéria, *i.e.* variáveis de estado.^{10,59} Nesta perspectiva, as mudanças de entropia podem ser expressas pela equação $dS = dq/T$. Entretanto para processos irreversíveis, que acontecem em tempos finitos, as mudanças nesta função de estado são apresentadas por uma desigualdade (de Clausius):

$$dS \geq \frac{dq}{T} \quad (4.3)$$

A desigualdade de Clausius não relaciona diretamente as mudanças de entropia com os processos irreversíveis, *e.g.* reações químicas, apenas com processos de troca de calor.¹⁰ Estes aspectos estão relacionados com uma premissa básica da termodinâmica clássica: transformações irreversíveis que ocorrem na Natureza podem ser reproduzidas através de processos reversíveis. No século XX, Onsager, De Donder e Van Rysselberghe, Prigogine e outros estenderam os conceitos da teoria clássica da termodinâmica para descrever explicitamente mu-

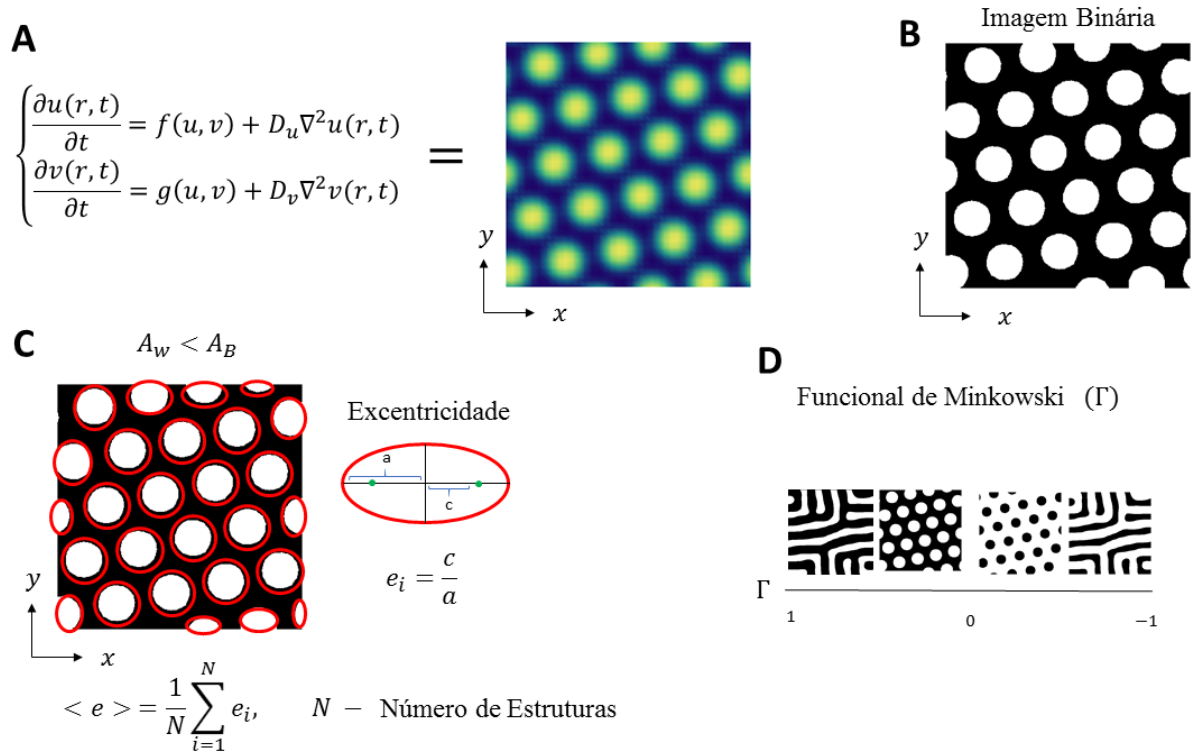


FIGURA 4.2 – Determinação dos funcionais de Minkowski (Γ). (A) A partir de um sistema do tipo reação-difusão e em condições de instabilidade de Turing, observa-se a emergência de padrões de Turing. (B) Através do *software Mathematica* é possível obter uma imagem binária dos padrões de Turing. (C) Calcula-se as áreas formadas por pixels brancos e pretos a partir da imagem binária, assim como a média das excentricidades $\langle e \rangle$. Como esta representado na figura, para o cálculo de $\langle e \rangle$, elipses (em vermelho) são utilizadas para aproximar da forma das estruturas que compõe o padrão. Cada uma delas apresentará um valor de excentricidade, e com isso é possível determinar a média entre elas. Para formas circulares $e \rightarrow 0$ e para estruturas com forma de listra $e \rightarrow 1$. (D) Associação entre estruturas e valores esperados para Γ .

danças da entropia de processos irreversíveis.⁵⁹ Neste formalismo, as mudanças de entropia são dadas por:¹⁰

$$dS = d_eS + d_iS \quad (4.4)$$

Onde d_eS representa as mudanças de entropia devido processos reversíveis, *e.g.* troca de matéria e energia térmica entre sistema e vizinhança, e d_iS são as mudanças de entropia causadas por processos irreversíveis que ocorrem no interior do sistema, *e.g.* reações químicas, condução/difusão de calor e difusão de massa. Veja a FIGURA 5.19.

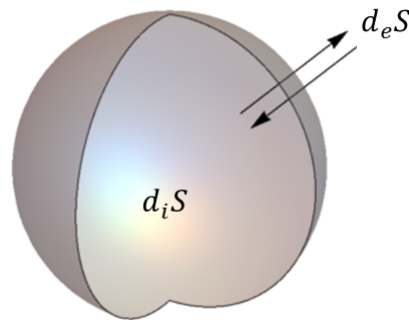


FIGURA 4.3 – Representação das mudanças de entropia devido à processos irreversíveis, d_iS , e reversíveis, d_eS . Pela segunda lei da termodinâmica, $d_iS \geq 0$ sempre, ao passo que d_eS pode ser maior, menor ou igual a zero dependendo da classe de sistema considerado, *i.e.* isolado, fechado ou aberto, e transformação estudada.

Na formulação moderna, os processos irreversíveis são descritos por forças termodinâmicas (F_k) e fluxos termodinâmicos (J_k).¹⁰ Assim sendo, a variação temporal da entropia por unidade de volume é dada pela expressão:

$$\sigma = \frac{1}{V} \frac{d_iS}{dt} = \sum_k F_k J_k \quad (4.5)$$

O termo $\sigma = \frac{1}{V} \frac{d_iS}{dt}$ é conhecido como taxa de produção de entropia.¹⁰

A taxa de produção de entropia é a base de uma robusta teoria de estabilidade e oferece informações associadas à flutuações internas de sistemas fora do equilíbrio.⁶⁰ Diversos trabalhos científicos investigaram o comportamento deste parâmetro em processos afastados do equilíbrio termodinâmico,^{61–63} mais especificamente em processos de quebra de simetria quiral.¹⁵⁹ Os resultados mostram que transições entre estados de assimetria quiral são marcadas por mudanças bruscas em σ , possibilitando assim a identificação destas transições.⁵⁹ Considerando tais informações e que a emergência de padrões de Turing também se configura em um processo de quebra de simetria, a taxa de produção de entropia pode ser usada para definir quantitativamente transições entre padrões espaciais deste tipo. Neste trabalho, três transformações

¹Síntese de duas moléculas com configurações diferentes em proporções desiguais.⁶⁴

irreversíveis são consideradas: reações química, difusão de massa e difusão/condução de energia térmica. As expressões que possibilitam o cálculo das quantidades de entropia produzidas devido a cada processo estão apresentadas no apêndice 7.4.

5 – Resultados e Discussão

Esta seção refere-se a descrição do código computacional desenvolvido para a simulação de modelos do tipo reação-difusão e apresentação dos resultados relativos às investigações sobre os efeitos das fronteiras através do perímetro do meio reacional, gradiente de temperatura e gradiente de concentração na emergência de padrões de Turing.

5.1 – O Programa

5.1.1 – Estrutura

Como foi apresentado em Teoria 4.1, os métodos numéricos utilizados para descrição da evolução espaço-temporal dos modelos isotérmico (MI) e não isotérmico (MNI) são Crank-Nicolson e Peaceman-Rachford. Estas abordagens numéricas foram implementadas computacionalmente na forma como está apresentado no Apêndice 7.2 (*i.e.* para o caso do método de Peaceman-Rachford. Similar para o método de Crank-Nicolson), em linguagem FORTRAN 90 e com a utilização de Módulos. De maneira geral, o módulo é uma unidade de programa que tem uma definição sintática explícita, método de ativação explícito, é semanticamente independente, realiza sub tarefas específicas e pode conter os seguintes elementos: variáveis, funções, sub rotinas e outros módulos. Ele possibilita o agrupamento de operações básicas, que podem ser acessados por diferentes partes de um programa ou por diferentes programas.^{65–67}

A fim de manter o código computacional organizado, optou-se pelo uso de módulos. A FIGURA 5.1 apresenta o esquema de modularização proposto para a construção do programa.

Os algoritmos 5.1 e 5.2 representam de forma simplificada a estrutura do código criado para os problemas em 1D e 2D, respectivamente, considerando o esquema de modularização apresentado previamente. Note que a implementação computacional do método de Crank-Nicolson difere do método de Peaceman-Rachford unicamente pelo Módulo 4, que não é necessário naquela abordagem.

Após a construção do código completo, com estrutura básica apresentada nos algoritmos anteriores, propôs-se modificações a fim de tornar o programa paralelizável. Sucintamente, a computação paralela tem como objetivo principal atingir o aumento do desempenho computacional, reduzindo conseqüentemente o tempo de execução de tarefas.^{68,69} Isto é feito a partir da divisão das tarefas em diversos processadores. Vale ressaltar que a construção de um código paralelizável não é uma atividade trivial. Existem diferentes arquiteturas computacionais paralelas, classificadas segundo as relações entre fluxos de instruções e fluxos de dados num processo computacional que podem ser consideradas e nem sempre a estrutura computacional proposta atinge o objetivo principal esperado, *i.e.* máxima eficiência de processamento. Isto normalmente é causado por latência de memória e de rede e algoritmos pouco paralelizados.⁶⁸

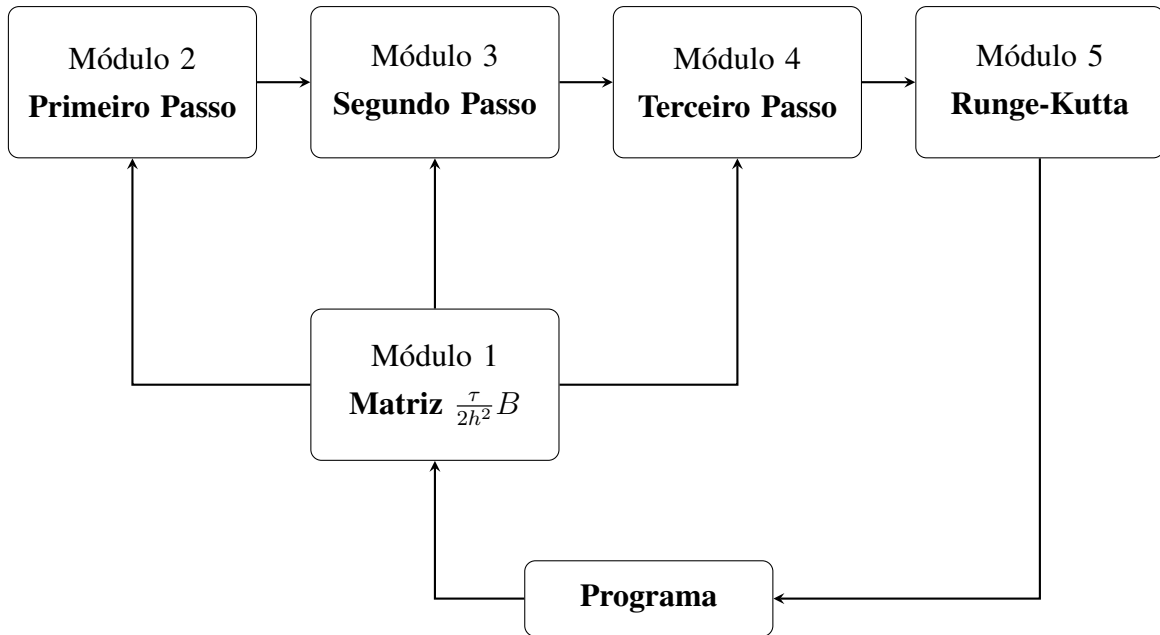


FIGURA 5.1 – Esquema de modularização do programa para o método de Peaceman-Rachford. Os passos correspondentes aos Módulos 2, 3 e 4 estão apresentados explicitamente em Apêndice 7.2. A construção é similar para o método de Crank-Nicolson.

Entrada: $t, x_L, u_0, v_0, u_x, v_x$
Saída: $u(t + 1, x), v(t + 1, x)$

```

1 início
2   ! Definição das condições iniciais;
3    $u(0, x) = u_0; v(0, x) = v_0$ 
4   ! Definição das condições de contorno;
5    $u(t, 0) = u(t, x_L) = u_x$ 
6    $v(t, 0) = v(t, x_L) = v_x$ 
7   para  $u$  e  $v$  faça
8      $\frac{\tau}{2h^2} B \leftarrow$  Módulo 1
9      $w_i^n \leftarrow$  Módulo 2
10     $u_i^{n+1} \leftarrow$  Módulo 3
11     $u_i^{n+1} \leftarrow$  Módulo 5
12  fim
13 fim
  
```

Algoritmo 5.1– Representação simplificada do código relativo ao método de Crank-Nicolson.

Neste trabalho, o código computacional foi paralelizado através do padrão MPI (*Message Passing Interface*). Este recurso é uma biblioteca de sub rotinas de comunicação de dados entre processos distribuídos de forma paralela.^{68,70} Entre as diversas formas de comunicação possíveis, foi considerada a comunicação de **ponto-a-ponto** devido a estrutura do problema em questão. Basicamente, esta forma de comunicação é feita através do envio e recebimento de mensagens entre dois pontos específicos do programa.^{68,70} Para exemplificar, o Algoritmo 5.3

Entrada: $t, x_L, y_L, u_0, v_0, u_x, u_y, v_x, v_y$

Saída: $u(t + 1, x, y), v(t + 1, x, y)$

```

1 início
2   ! Definição das condições iniciais;
3    $u(0, x, y) = u_0; v(0, x, y) = v_0$ 
4   ! Definição das condições de contorno;
5    $u(t, 0, y) = u(t, x_L, y) = u_x$  e  $u(t, x, 0) = u(t, x, y_L) = u_y$ 
6    $v(t, 0, y) = v(t, x_L, y) = v_x$  e  $v(t, x, 0) = v(t, x, y_L) = v_y$ 
7   para  $u$  e  $v$  faça
8      $\frac{\tau}{2h^2} B \leftarrow$  Módulo 1
9      $w_{i,j}^n \leftarrow$  Módulo 2
10     $u_{i,j}^{n+1/2}; v_{i,j}^{n+1/2} \leftarrow$  Módulo 3
11     $u_{i,j}^{n+1}; v_{i,j}^{n+1} \leftarrow$  Módulo 4
12     $u_{i,j}^{n+1}; v_{i,j}^{n+1} \leftarrow$  Módulo 5
13  fim
14 fim
```

Algoritmo 5.2– Representação simplificada do código relativo ao método de Peaceman-Rachford.

apresenta a ideia da paralelização do código em questão.

<pre> 1 para u faça 2 $\frac{\tau}{2h^2} B \leftarrow$ Módulo 1 3 $w_{i,j}^n \leftarrow$ Módulo 2 4 $u_{i,j}^{n+1/2} \leftarrow$ Módulo 3 5 $u_{i,j}^{n+1} \leftarrow$ Módulo 4 6 $u_{i,j}^{n+1} \leftarrow$ Módulo 5 7 fim</pre>	<pre> 8 para v faça 9 $\frac{\tau}{2h^2} B \leftarrow$ Módulo 1 10 $w_{i,j}^n \leftarrow$ Módulo 2 11 $v_{i,j}^{n+1/2} \leftarrow$ Módulo 3 12 $v_{i,j}^{n+1} \leftarrow$ Módulo 4 13 $v_{i,j}^{n+1} \leftarrow$ Módulo 5 14 fim</pre>
---	--

Algoritmo 5.3– Representação simplificada do código paralelizável através do uso da biblioteca MPI referente ao método de Peaceman-Rachford.

O *loop* central, relativo à cada variável dependente, é realizado por diferentes processadores e as demais tarefas como, definição das condições iniciais, definição das condições de contorno e escrita dos resultados é feita por um processador adicional. Desta forma, para um sistema de EDPs composto por n –variáveis dependentes, o programa pode ser paralelizado em um número m de processadores segundo a relação $m = n + 1$. O algoritmo paralelizado proposto conseguiu atingir uma redução média de aproximadamente $2/5$ e $1/2$ do tempo necessário para a simulação de sistemas de EDPs de duas e três variáveis dependentes, respectivamente.

O código escrito em FORTRAN 90 está apresentado no Apêndice 7.5.

5.1.2 – Validação

Equações diferenciais parciais parabólicas que apresentam solução analítica foram consideradas para comparar a acurácia da aproximação numérica do programa desenvol-

vido. Assim sendo, tais equações foram utilizadas para a validação do resultado final através do cálculo de erros entre o resultado analítico exato e o numérico aproximado. Basicamente, três equações foram consideradas para situações em 1D e 2D, e elas estão apresentadas nas TABELAS 5.1 e 5.2.

Os cálculos numéricos foram feitos tomando $\Delta x = \Delta y = \Delta t$, ou seja, os passos numéricos no tempo e no espaço são iguais, e seus tamanhos foram variados da seguinte forma:

$$\Delta t = \Delta x = \Delta y = \frac{1}{5 \times 2^{n-1}} \quad (5.1)$$

onde $n = 1, 4$.

Os erros foram determinados de duas maneiras distintas, o primeiro deles é o maior erro absoluto, o ponto que apresenta maior diferença com o resultado analítico, dado por:

$$e_{max} = \max_{i,j=1,\dots,N-1} |U_{i,j}^m - u(x_i, y_j, t_m)| \quad (5.2)$$

onde, $U_{i,j}^M$ é a aproximação numérica e $u(x_i, y_j, t_m)$ o resultado proveniente da equação analítica.

O segundo erro calculado foi o erro quadrático médio, dado pela seguinte equação (para o caso em 2D):

$$e_{L^2} = \left(\sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \Delta x \Delta y (U_{i,j}^m - u(x_i, y_j, t_m))^2 \right)^{1/2} \quad (5.3)$$

Caso	EDP	Solução Analítica
1	$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 4xe^{t+x}$	$x(1-x)e^{x+t}$
2	$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = e^{xt}(x-t^2)$	e^{xt}
3	$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0$	e^{x+t}

TABELA 5.1 – EDPs em 1D e suas respectivas soluções analíticas utilizadas para o teste de acurácia das aproximações numéricas obtidas a partir do programa criado.

Caso	EDP	Solução Analítica
1	$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = -xy(xy + 3y + 3x - 7)e^{t+x+y}$	$x(1-x)y(1-y)e^{x+y+t}$
2	$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = e^{xyt}(xy - t^2(x^2 + y^2))$	e^{xyt}
3	$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = -e^{t+x+y}$	e^{x+y+t}

TABELA 5.2 – EDPs em 2D e suas respectivas soluções analíticas utilizadas para o teste de acurácia das aproximações numéricas obtidas a partir do programa criado.

Nas TABELAS 5.3 e 5.4 os erros calculados a partir das Eqs. (5.2) e (5.3) para

os Casos 1-3, presentes na TABELA 5.1, estão apresentados para situações em 1D e 2D, respectivamente.

Δx	Caso 1		Caso 2		Caso 3	
	e_{max}	e_{L^2}	e_{max}	e_{L^2}	e_{max}	e_{L^2}
$\frac{1}{5}$	2.60×10^{-2}	1.40×10^{-2}	9.26×10^{-2}	3.88×10^{-2}	3.40×10^{-3}	1.82×10^{-3}
$\frac{1}{10}$	5.77×10^{-3}	5.22×10^{-3}	4.20×10^{-2}	1.56×10^{-2}	8.58×10^{-4}	4.30×10^{-4}
$\frac{1}{20}$	1.37×10^{-3}	2.86×10^{-3}	2.01×10^{-2}	6.98×10^{-3}	2.10×10^{-4}	1.02×10^{-4}
$\frac{1}{40}$	3.47×10^{-4}	1.56×10^{-3}	9.86×10^{-3}	3.30×10^{-3}	1.85×10^{-5}	6.77×10^{-6}

TABELA 5.3 – Erro máximo e erro quadrático médio da aproximação numérica calculados para os Casos 1-3, presentes na TABELA 5.1, em 1D.

Δx	Caso 1		Caso 2		Caso 3	
	e_{max}	e_{L^2}	e_{max}	e_{L^2}	e_{max}	e_{L^2}
$\frac{1}{5}$	3.99×10^{-2}	2.03×10^{-2}	1.07×10^{-2}	5.60×10^{-3}	1.13×10^{-3}	3.43×10^{-4}
$\frac{1}{10}$	9.98×10^{-3}	4.92×10^{-3}	2.44×10^{-3}	1.29×10^{-3}	2.46×10^{-4}	3.55×10^{-5}
$\frac{1}{20}$	2.42×10^{-3}	1.19×10^{-3}	5.80×10^{-4}	3.03×10^{-4}	5.14×10^{-5}	1.57×10^{-5}
$\frac{1}{40}$	5.99×10^{-4}	2.93×10^{-4}	1.54×10^{-4}	8.30×10^{-5}	1.18×10^{-4}	6.29×10^{-5}

TABELA 5.4 – Erro máximo e erro quadrático médio da aproximação numérica calculados para os Casos 1-3, presentes na TABELA 5.2, em 2D.

Como é possível notar através dos dados expostos nas tabelas, as aproximações numéricas das soluções das EDPs consideradas apresentam boa convergência e acurácia com as soluções analíticas correspondentes. A partir disto, conclui-se que o programa desenvolvido pode ser utilizado para o estudo de EDPs do tipo reação-difusão em uma e duas dimensões espaciais.

5.2 – Regiões de instabilidade de Turing

Para prosseguir com as investigações é preciso definir as condições necessárias para a formação de padrões espaciais estacionários e não homogêneos.

As condições para a emergência de padrões de Turing são determinados em função dos parâmetros $a = \frac{A}{C_T}$ e $b = \frac{B}{C_T}$ através da análise de estabilidade linear. Para isso, os parâmetros físico-químicos que compõem os modelos devem assumir valores numéricos, que estão apresentados na TABELA 5.5.

Parâmetro	Valor Numérico
γ_u	1.0
k_{f_1}	1.0
k_{f_2}	1.0
k_{r_1}	0.01
k_{r_2}	0.01
d	1.0
ξ	0.25*
ϕ	1.0
$e_i, i = 1, \dots, 4$	4.0
$\Delta H'_{r_1}$	0.01
$\Delta H'_{r_2}$	0.01
d_u	1.0
d_v	5.0
d_θ	10.0

TABELA 5.5 – Valores numéricos dos parâmetros físico-químicos. * Em Resultados 5.6 também utilizou-se $\xi = 1.0$.

A partir das informações na TABELA 5.5, o espaço de parâmetros ($a \times b$) relacionados com MI e MNI são apresentados nas FIGURAS 5.2 e 5.3, respectivamente. As figuras mostram regiões de instabilidade de Turing, pontos de sela e oscilação, que é particular para MNI.

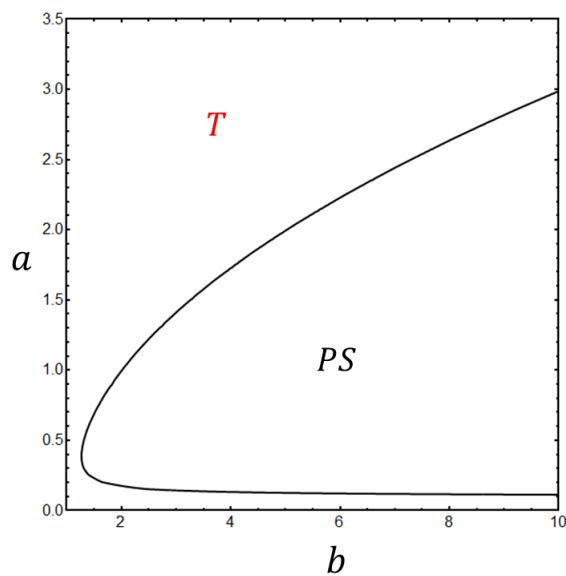


FIGURA 5.2 – Análise da estabilidade linear de MI, onde T: instabilidade de Turing e PS: ponto de sela.

Note que na FIGURA 5.3 há duas regiões de instabilidade de Turing demarcadas com a letra "T" em vermelho e preto. Aquelas regiões são diferentes entre si com respeito a magnitude das instabilidades geradas por perturbações espaciais não homogêneas, verificado através do número de autovalores com parte real positiva. A região marcada com "T" em vermelho é caracterizada com três autovalores com parte real positiva, ao passo que a região com "T" em preto apresenta apenas um autovalor com esta característica. Embora ambas regiões sa-

tisfaçam as condições necessárias para a formação das estruturas de Turing, a emergência de tais estruturas tomam mais tempo em situações de baixa perturbação espacial, como pode ser verificado por simulação. Devido a isto, os parâmetros foram escolhidos nas regiões marcadas com "T" em vermelho.

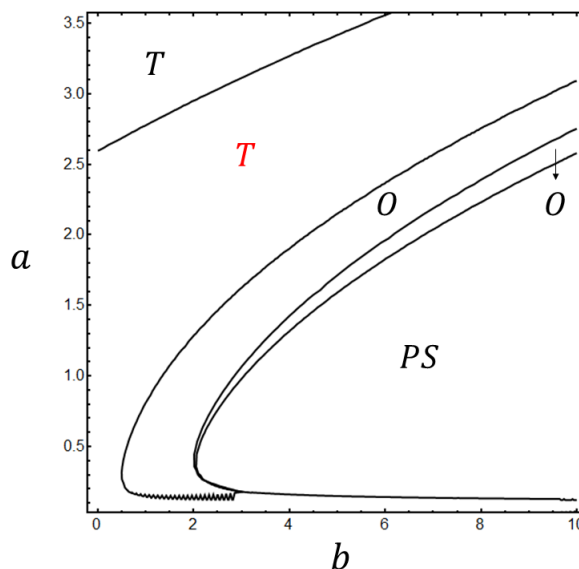


FIGURA 5.3 – Análise da estabilidade linear de MNI, onde T: instabilidade de Turing , PS: ponto de sela e O: oscilação.

Considerando os resultados obtidos a partir da análise da estabilidade linear as simulações do MI foram executadas tomando $a = 2.1$, $b = 5.0$ e a temperatura constante $\theta = 0.75$. No caso do MNI, $a = 2.4$, $b = 5.0$, e a temperatura nas fronteiras $\theta_{\Omega} = 0.96$. Em ambos sistemas as condições iniciais são definidas randomicamente ao redor dos pontos de equilíbrio. Os pontos de equilíbrio estão apresentados no Apêndice 7.6 .

5.3 – Definição do tamanho do meio reacional

A fim de entender os efeitos da perturbação das fronteiras na emergência de padrões de Turing, necessita-se definir o tamanho do meio reacional.

Considerando as informações expostas na Introdução 1.4, o sistema precisa ter uma dimensão ideal, com relação a dimensão das estruturas espaciais, para o desenvolvimento do estudo das influências das fronteiras sobre a formação de padrões de Turing. Com estas considerações, simulações foram realizadas em reatores quadrados de tamanho $(x \times y)$ unidades de área (u.a.), onde $x, y = 20, 30, 40, 50, 60$, e 70 , e diferentes condições de fronteira, Dirichlet (CFD) e Neumann (CFN) de fluxo zero, para cada um deles. A influência de CFN sobre o bulk da reação é pequena quando comparada com a CFD. A CFN de fluxo zero não interage quimicamente com o meio reacional, ao passo que a CFD modifica fortemente as suas vizinhanças devido a imposição de fontes/sumidouros de reagentes químicos. Com isso, torna-se possível medir quantitativamente os efeitos das fronteiras na dinâmica do processo em questão

comparando a taxa de produção de entropia em cada uma das situações apresentadas, através da seguinte equação:

$$\delta = \frac{|\sigma_T^D - \sigma_T^N|}{A_r} \quad (5.4)$$

Na Eq. (5.4), δ é a diferença absoluta das taxas de produção de entropia total entre simulações realizadas com CFD (σ_T^D) e CFN (σ_T^N) para sistema com diferentes tamanhos, normalizado pela área do sistema (A_r). Os termos σ_T^D e σ_T^N são calculados no estado estacionário, uma vez que nessa situação este parâmetro tende à valores constantes.⁷¹ No caso de CFD as concentrações dos reagentes químicos nas fronteiras são $U_\Omega = 2.0$ e $V_\Omega = 2.2$ para MI, e $U_\Omega = 2.3$ e $V_\Omega = 2.0$ para MNI. A FIGURA 5.4 mostra δ em função da área.

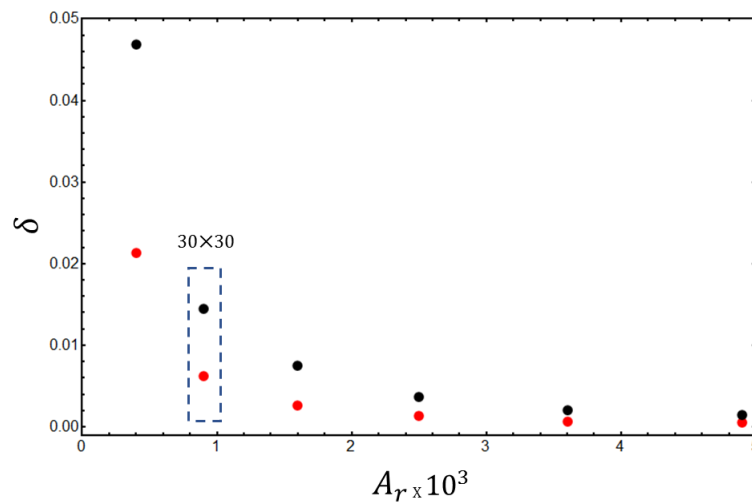


FIGURA 5.4 – Diferença absoluta da taxa de produção de entropia (δ) entre simulações executadas com condição de fronteira de Dirichlet e Neumann para sistemas com diferentes tamanhos. Os dados em vermelho estão associados à MI e os em preto com MNI.

Através da figura é possível verificar que a diferença da taxa de entropia total para um sistema de pequeno tamanho é maior do que em um sistema de grande tamanho. Este fato concorda com as informações relatadas na literatura de que os efeitos da perturbação das fronteiras são mais pronunciadas em sistemas de pequeno tamanho.¹⁶ Considerando os efeitos da fronteira, uma escolha inteligente para o tamanho do sistema seria o 20×20 u.a., entretanto a dimensão das estruturas de Turing nesta situação são comparáveis com as dimensão do reator, e isto dificulta a caracterização dos padrões. Então, considerando o caso 30×30 u.a. é possível afirmar, analisando a FIGURA 5.4, que a perturbação da fronteira é relativamente alta, quando comparada com as demais situações, podendo afetar o bulk da reação. Mais ainda, diferentemente do caso 20×20 u.a., as estruturas emergentes são menores em tamanho do que as dimensões do sistema, permitindo a identificação dos padrões de Turing. Conforme as dimensões aumentam, a partir de 40×40 u.a., a perturbação é minimizada, tendendo a zero. Portanto, o tamanho do sistema escolhido para a realização das simulações deste trabalho é 30×30 u.a..

Uma vez que as condições necessárias para a emergência de padrões de Turing e

o tamanho do meio reacional foram definidos, os efeitos relacionados ao perímetro das fronteiras, gradiente de temperatura e concentração podem ser investigados.

5.4 – Padrões de Turing referência

Considerando os propósitos deste trabalho, os padrões de Turing obtidos a partir de sistemas 30×30 u.a. e condições de fronteira de Neumann (CFN) de fluxo zero serão considerados os "padrões de Turing referência", ou seja, aqueles que são formados sob influência mínima de reagentes químicos e energia térmica, provenientes de fontes externas. Estas estruturas podem ser usadas como referência para a comparação dos padrões de Turing modulados por condições de fronteira de Dirichlet (CFD). Tais padrões estão apresentados na FIGURA 5.5. Note que estes padrões, assim como em todas as figuras deste trabalho, referem-se a concentração do ativador.

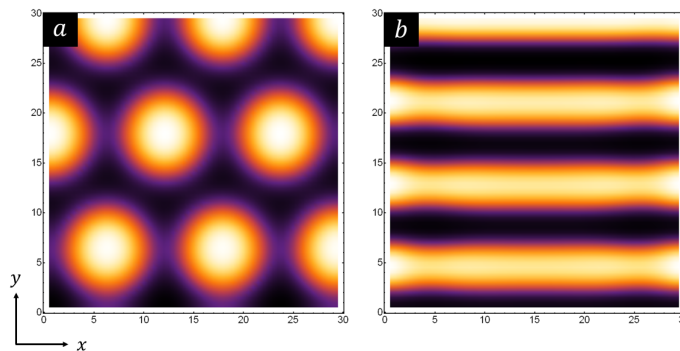


FIGURA 5.5 – Padrões de Turing referência. a) MI; b) MNI

5.5 – Perímetro das fronteiras

Nesta subseção buscou-se compreender as influências do perímetro e geometria das fronteiras na emergência dos padrões de Turing. Para isso, simulações foram executadas com sistemas de mesma área, 900 u.a., e diferentes perímetros (p), com $p = 60, 61, 65, 75$ e 100. Para o MI utilizou-se CFN com fluxo zero. Para o caso do MNI duas situações foram consideradas: 1) CFN de fluxo zero para os reagentes químicos e temperatura; 2) CFN de fluxo zero para os reagentes químicos e CFD para a temperatura. Apenas enfatizando, na situação (2) as bordas do meio reacional apresentam temperatura bem definida, $\theta_{\Omega} = 0.96$.

Os resultados obtidos a partir das simulações descritas estão apresentadas nas FIGURAS 5.6, 5.7 e 5.9 para MI e MNI, respectivamente.

Para o caso do MI, FIGURA 5.6, nota-se que entre as simulações realizadas os padrões mais frequentes possuem a forma de bolas. Entretanto, para o caso de $p = 65$ e $p = 100$, há a formação de listras, com diferentes orientações, *i.e.* em $p = 65$ listras na horizontal e em $p = 100$ listras na vertical. O último caso, relaciona-se diretamente com os resultados apresentados nas referências^{14,15}, onde devido a diminuição da largura do meio reacional as bolas se alongam parecendo listras. Contudo, este efeito não representa uma transição de

estados espaciais, apesar da morfologia do padrão ter alterado. Isto é confirmado através da análise das taxas de produção de entropia apresentadas na TABELA 5.6. Os valores desse parâmetro para todas as simulações são muito similares, evidenciando que não há transição de estados espaciais. Conclusões similares podem ser tiradas do caso em que $p = 65$, mas as razões para o surgimento das faixas são desconhecidas. Acredita-se que as dimensões do meio reacional possibilitam a sobreposição de bolas em linha reta formando as listras observadas.

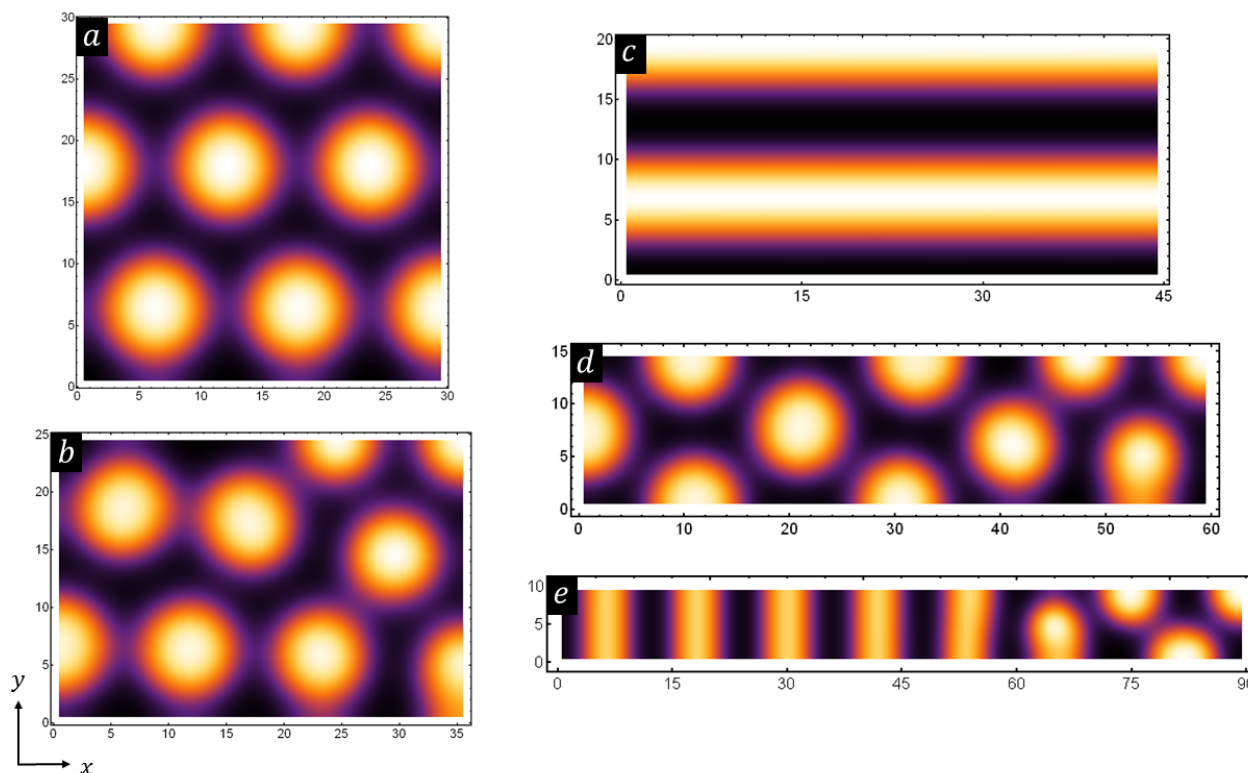


FIGURA 5.6 – Simulações do MI para sistemas com área 900 u.a. e diferentes perímetros (p). a) $p = 60$ e área 30×30 u.a. b) $p = 61$ e área 36×25 u.a. c) $p = 65$ e área 45×20 u.a. d) $p = 75$ e área 60×15 u.a. e) $p = 100$ e área 90×10 u.a..

Os resultados obtidos para o MNI com CFN com fluxo zero, apresentados na FIGURA 5.7, revelam pequenas diferenças na organização espacial dos padrões, como pode ser notado através do funcional de Minkowski na TABELA 5.6. Entretanto, para todos os perímetros considerados os padrões possuem a forma de listras, ou seja, as estruturas formadas representam o mesmo estado dinâmico, veja as taxas de produção de entropia total na TABELA 5.6.

Neste caso, pode-se afirmar que a orientação e organização espacial das estruturas formadas é regulada pela temperatura, que é tratada como variável dependente. É possível notar que o perfil espacial da temperatura apresenta uma forma complementar aos padrões de Turing, veja FIGURA 5.8. Supõe-se que estes fatos constatados estejam relacionados com a alta difusibilidade térmica, que faz com que a temperatura controle a dinâmica dos reagentes químicos.

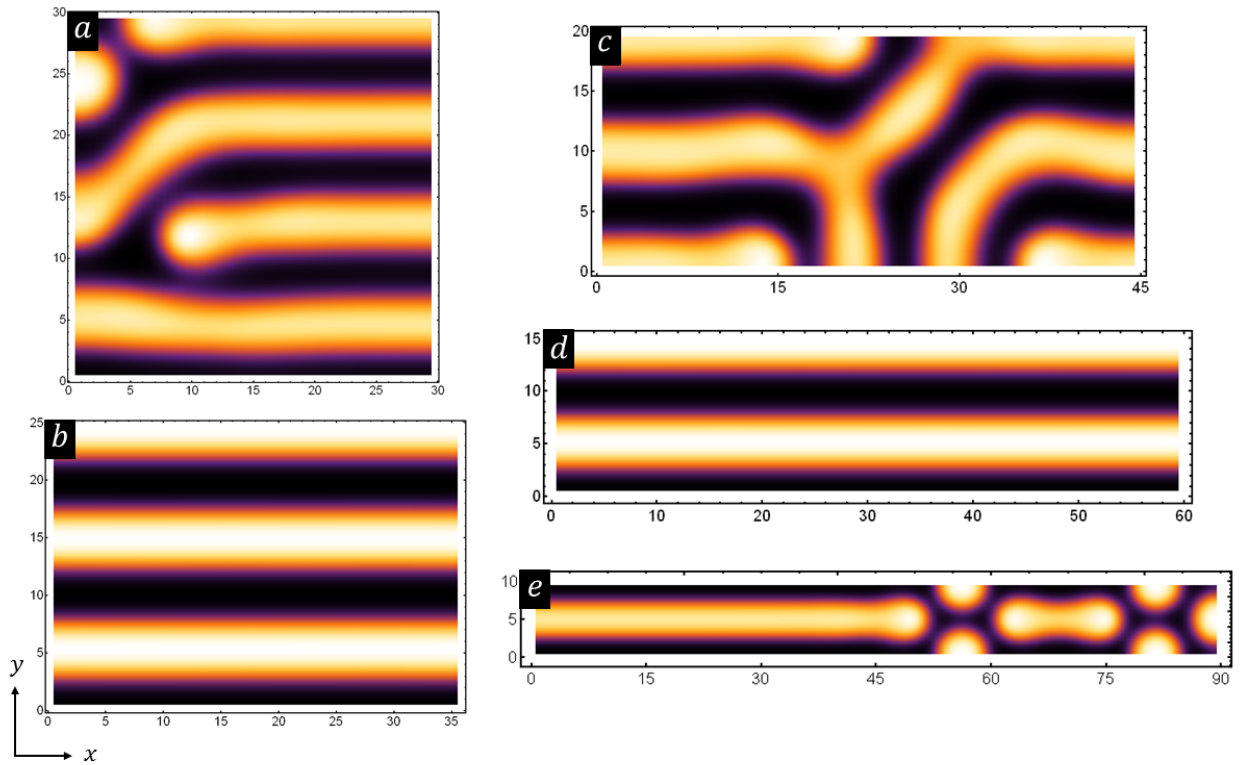


FIGURA 5.7 – Simulações do MNI para sistemas com área 900 u.a., CFN com fluxo zero para a temperatura e diferentes perímetros (p). a) $p = 60$ e área 30×30 u.a. b) $p = 61$ e área 36×25 u.a. c) $p = 65$ e área 45×20 u.a. d) $p = 75$ e área 60×15 u.a. e) $p = 100$ e área 90×10 u.a..

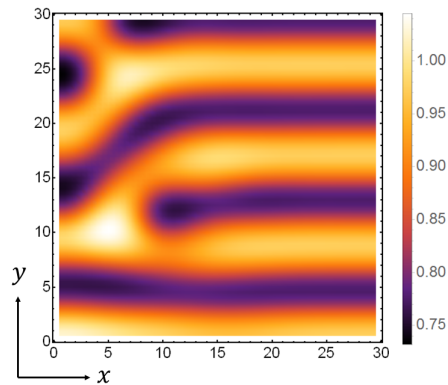


FIGURA 5.8 – Perfil espacial da temperatura para o MNI em um sistema 30×30 u.a.. Referente a temperatura da simulação apresentada na FIGURA 5.7 a).

Com relação aos resultados relativos ao MNI com CFD para a temperatura, verifica-se que há majoritariamente a emergência de padrões com formatos de listras, como no último caso. Entretanto, devido à presença de fontes térmicas nas fronteiras, o efeito observado anteriormente é intensificado conforme o perímetro aumenta, por uma razão geométrica, *i.e.* o aumento perimétrico implica na aproximação das fronteiras. Apesar de algumas diferenças entre as taxa de produção de entropia total calculadas para cada perímetro, TABELA 5.7, acredita-se que não há transição de estados, mesmo que os padrões se diferenciam morfolo-

camente em alguns aspectos.

Modelo	Parâmetros	Perímetro				
		60	61	65	75	100
MI	σ_T	27.2	27.2	27.2	27.2	26.4
	Γ	0.23	0.35	-0.90	0.49	-0.77
MNI	σ_T	51.9	51.6	51.9	50.3	50.0
	Γ	-0.71	-0.72	-0.89	-0.97	-0.9

TABELA 5.6 – Taxa de produção de entropia e funcionais de Minkowski ao resultados obtidos a partir dos modelos MI e MNI (CFN com fluxo zero para a temperatura), e sistemas com diferentes perímetros.

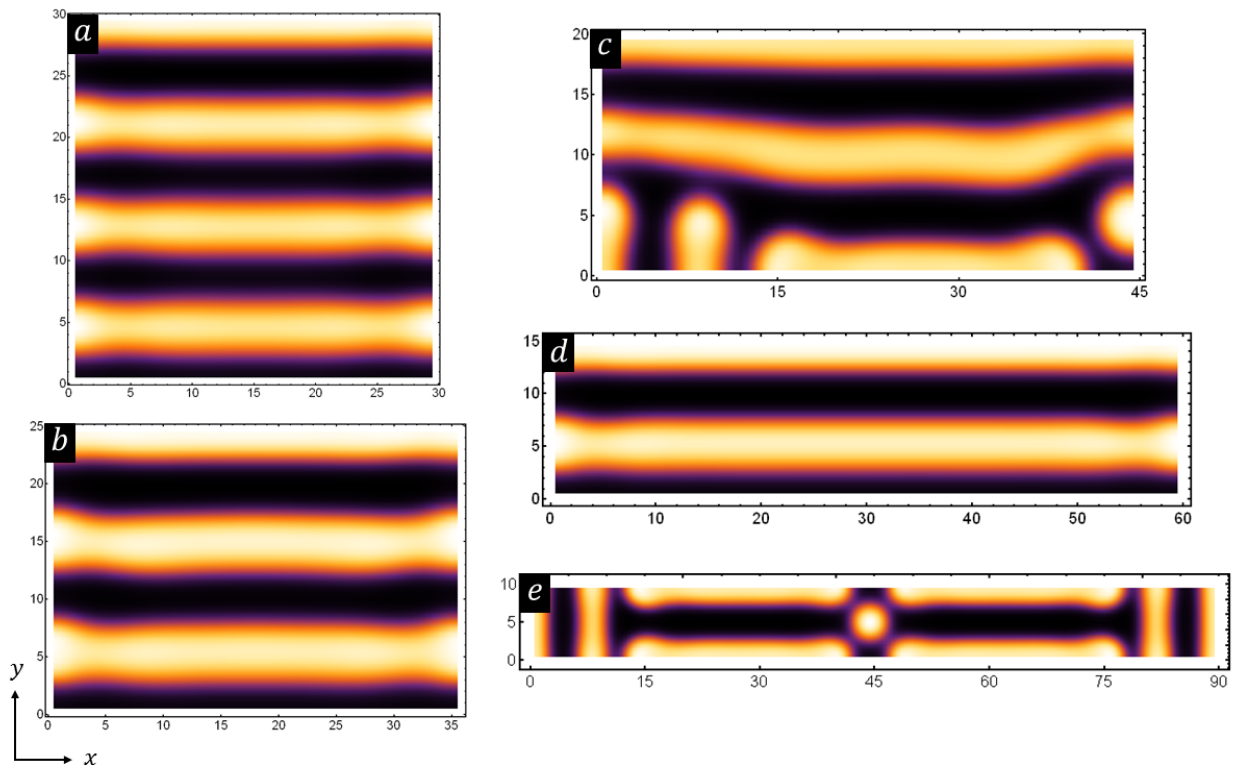


FIGURA 5.9 – Simulações do MNI para sistemas com área 900 u.a., CFD para a temperatura e diferentes perímetros (p). a) $p = 60$ e área 30×30 u.a. b) $p = 61$ e área 36×25 u.a. c) $p = 65$ e área 45×20 u.a. d) $p = 75$ e área 60×15 u.a. e) $p = 100$ e área 90×10 u.a..

Modelo	Parâmetros	Perímetro				
		60	61	65	75	100
MNI	σ_T	61.1	60.0	71.6	60.5	79.7
	Γ	-0.70	-0.72	-0.89	-0.97	-0.78

TABELA 5.7 – Taxa de produção de entropia e funcionais de Minkowski ao resultados obtidos a partir do modelo MNI (CFD para a temperatura), e sistemas com diferentes perímetros.

5.6 – Gradiente de temperatura

Esta subseção apresenta os resultados dos efeitos de perturbações associadas à presença de fontes térmicas localizadas nas fronteiras do reator na emergência de padrões de Turing. Mais ainda, considerando as investigações realizadas pela da referência ⁴⁰, simulações também foram executadas considerando o sistema em um banho térmico. Desta forma é possível fazer comparações entre as formas de perturbação.

Em termos de simulação, os casos apresentados são diferenciados da seguinte forma: no primeiro caso - fontes térmicas localizadas nas fronteiras - o termo relacionado a lei de resfriamento de Newton na Eq.(3.9) assume valores diferentes de zero para os primeiros termos da malha de integração, próximos às fronteiras, e zero para os demais. Ou seja, há troca de energia térmica apenas entre partículas em contato com as fronteiras. No segundo caso - banho térmico - o termo relacionado à lei de resfriamento de Newton é diferente de zero e igual para todos os pontos da malha de integração. De forma mais clara, todos as partículas do meio reacional trocam energia térmica com o banho. Veja a FIGURA 5.10 para melhor esclarecimento.

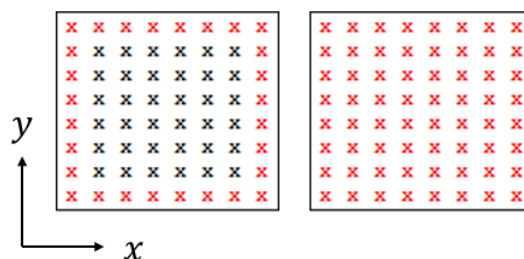


FIGURA 5.10 – Representação das trocas de energia térmica dos pontos da malha de integração para os casos de fontes térmicas localizadas nas fronteiras do reator, à esquerda, e banho térmico, à direita. Os termos em vermelho representam os pontos da malha que trocam energia térmica através do processo de condução.

A motivação principal da análise destas fontes distintas de perturbação é encontrar uma forma eficiente de sintonização de estados espaciais não homogêneos através do controle externo da temperatura e assim propor um modelo químico mínimo para descrever a biogênese.

Vale ressaltar que nesta seção o utilizou-se $\xi = 1.0$. Isto foi feito pois, nesta condição é possível observar um número maior de estados espaciais para uma pequena variação de temperatura com relação à situação em que $\xi = 0.25$.

5.6.1 – Fonte térmica nas fronteiras do sistema

Inicialmente buscou-se compreender através do MNI o papel da presença de fontes térmicas localizadas nas fronteiras do reator. Para isso, simulações foram realizadas com CFN de fluxo zero para os reagentes químicos e CFD para a temperatura, de forma que as fontes térmicas são iguais nas quatro paredes do reator. Os valores da temperatura nas bordas (θ_Ω) considerados são 1.0, 1.1, 1.2, 1.3, 1.4, 1.5 e 1.6, veja a FIGURA 5.11.

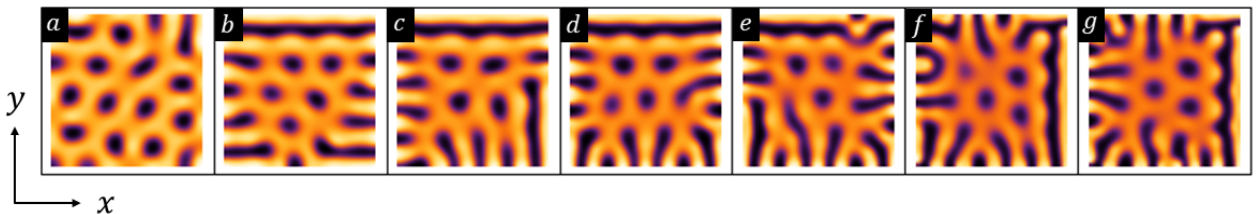


FIGURA 5.11 – Simulação do MNI com diferentes fontes térmicas localizadas nas fronteiras do reator. a) $\theta_\Omega = 1.0$ b) $\theta_\Omega = 1.1$ c) $\theta_\Omega = 1.2$ d) $\theta_\Omega = 1.3$ e) $\theta_\Omega = 1.4$ f) $\theta_\Omega = 1.5$ g) $\theta_\Omega = 1.6$.

Visualmente observa-se que há modificações locais do padrão próximo às paredes do reator conforme a temperatura da borda aumenta, entretanto no centro do reator estruturas na forma de bolas são preservadas, para todo θ_Ω considerado. Através da análise da taxa de produção de entropia total, apresentada na FIGURA 5.12, é possível verificar que não há transição de estados uma vez que não se observa descontinuidades bruscas na tendência deste parâmetro. Na verdade, essa taxa é diretamente proporcional às temperaturas consideradas, praticamente servindo como um "termômetro" linear.

Estes resultados revelam que a indução de quebra de simetria espacial através de variações da temperatura de fontes térmicas localizadas nas bordas do reator é inviável, mesmo para sistemas de pequeno tamanho. É fácil prever que para meios reacionais de grandes dimensões espaciais, o efeito da temperatura a partir desta construção é mínimo, alterando muito pouco a dinâmica do sistema.

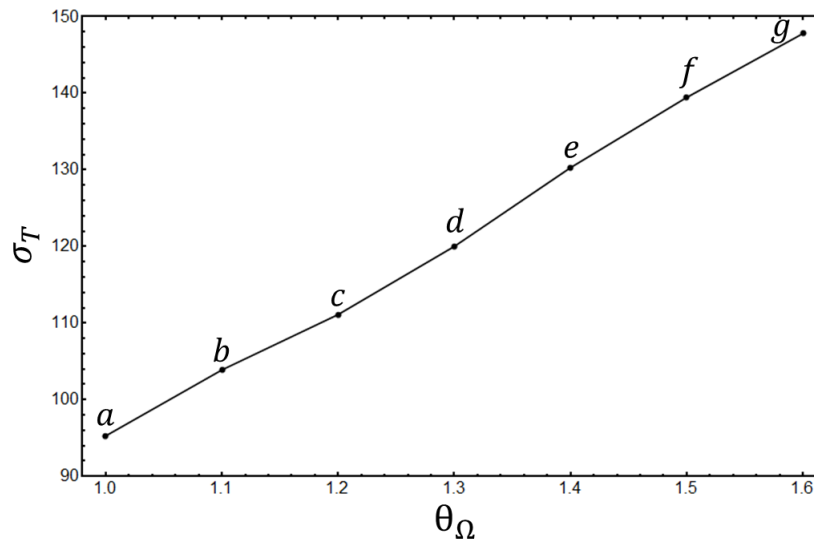


FIGURA 5.12 – A taxa de produção de entropia total para as diferentes fontes térmicas localizadas nas bordas do meio reacional. As letras (*a,b,c,d,e,f,g*) indicam os diferentes padrões apresentados na FIGURA 5.11.

5.6.2 – Banho térmico

Como foi apresentado no início desta subseção, buscou-se entender também o efeito do banho térmico na emergência de padrões de Turing. Para isso, simulações foram realizadas com condições de fronteira periódicas, reatores com dimensão igual à 60×60 u.a. e diferentes temperaturas de banho (θ_{EX}), sendo elas $\theta_{EX} = 1.0, 1.1, 1.2, 1.3, 1.4, 1.5$ e 1.6 . Os resultados destas simulações estão apresentados na FIGURA 5.13.

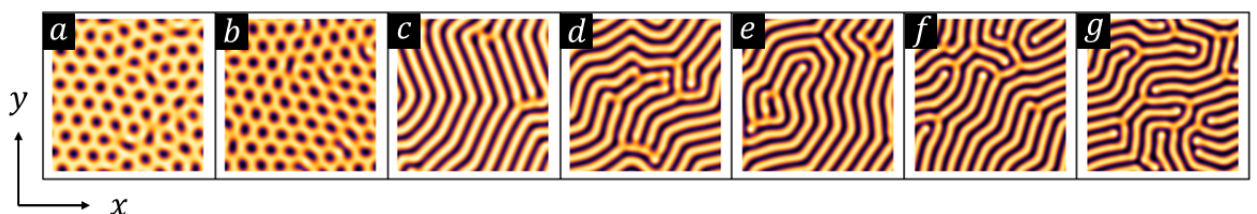


FIGURA 5.13 – Simulação do MNI com reator de tamanho 60×60 u.a. e diferentes temperaturas de banho térmico. a) $\theta_{EX} = 1.0$ b) $\theta_{EX} = 1.1$ c) $\theta_{EX} = 1.2$ d) $\theta_{EX} = 1.3$ e) $\theta_{EX} = 1.4$ f) $\theta_{EX} = 1.5$ g) $\theta_{EX} = 1.6$.

Visualmente é possível notar que diferentes padrões de Turing emergem conforme a temperatura do banho térmico varia, indo de bolas para listras. A análise dos funcionais de Minkowski confirma estes fatos, demonstrando que os padrões *a* e *b*, na FIGURA 5.13, apresentam similaridades morfológicas entre si, assim como os demais padrões também compartilham certas similaridades estruturais. Veja a FIGURA 5.14.

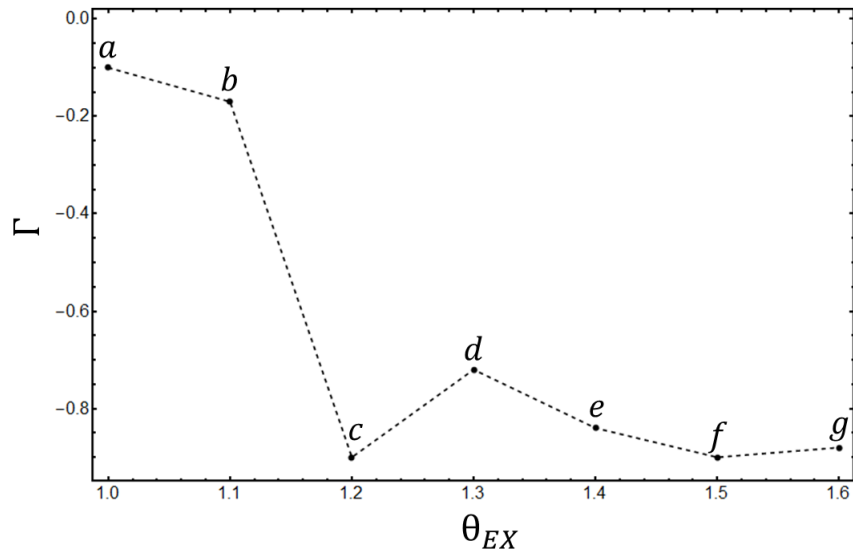


FIGURA 5.14 – Funcionais de Minkowski para banhos térmicos com diferentes temperaturas. As letras (*a, b, c, d, e, f, g*) indicam os diferentes padrões apresentados na FIGURA 5.13.

As diferenças morfológicas encontradas na série de padrões obtidos indica quebra de simetria espacial e uma possível transição de estado. Através da taxa de produção de entropia associada ao processo de difusão (σ_M) é possível notar um desnível na tendência do gráfico entre $\theta_{EX} = 1.1$ e $\theta_{EX} = 1.2$, evidenciando uma transição de estado, como esperado. Neste caso, utilizou-se σ_M pois o termo entrópico associado às reações químicas é muito grande e não varia muito, impossibilitando a identificação do processo de quebra de simetria espacial através da taxa de produção de entropia total. Veja a FIGURA 5.15.

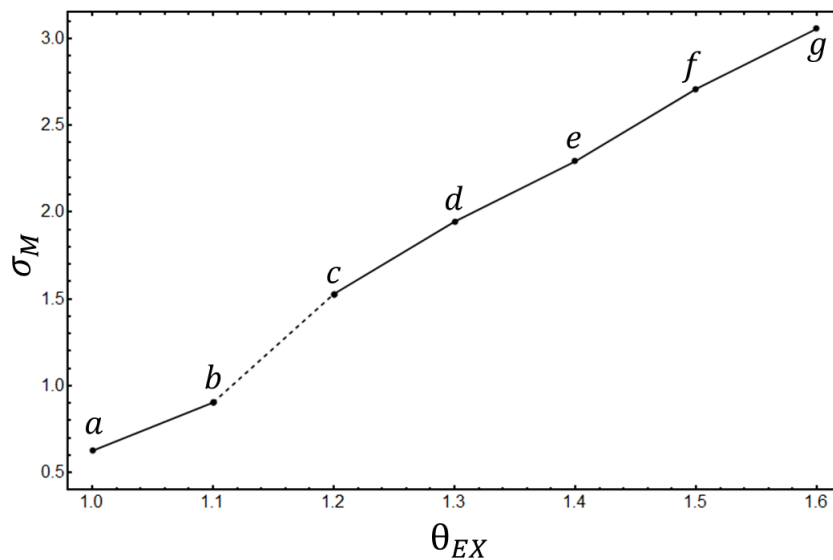


FIGURA 5.15 – A taxa de produção de entropia associada ao processo de difusão para banhos térmicos com diferentes temperaturas. As letras (*a, b, c, d, e, f, g*) indicam os diferentes padrões apresentados na FIGURA 5.13.

Assim sendo, é possível utilizar a temperatura de um banho térmico como parâmetro de controle de padrões de Turing. Este resultado já foi reportado na literatura ⁴⁰. A

partir disso, realizou-se uma simulação considerando um sistema de dimensões 90×30 u.a. com gradiente térmico sobre toda área do reator na direção- x , de modo que $\theta_{EX}(x_0) = 1.0$ e $\theta_{EX}(x_F) = 1.2$, com $x_0 \leq x \leq x_F$. Tal construção pode ser reproduzida experimentalmente utilizando-se uma chapa de aquecimento. A estrutura formada esta apresentada na FIGURA 5.16.

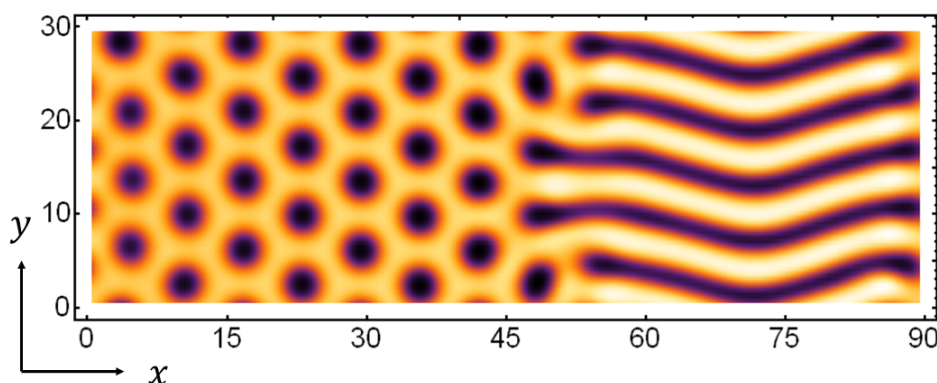


FIGURA 5.16 – Simulação do MNI com gradiente térmico na direção x criado através de um banho térmico (placa de aquecimento), de modo que, $x_i = 1.0$ e $x_f = 1.2$.

Como pode ser observado, um sistema sob a influência de um gradiente térmico criado sobre toda sua área induz quebra de simetria espacial dentro de um mesmo reator. Desta forma é possível obter diferentes padrões de Turing a partir da variação da temperatura do banho ou através de um gradiente de temperatura.

5.7 – Gradiente de concentração

Continuando com as investigações, esta subseção apresenta o estudo dos efeitos da existência de fontes de reagentes químicos nas fronteiras do meio reacional, assim como situações de gradiente de concentração imposto através das fronteiras, na emergência de padrões de Turing. A motivação principal do estudo desenvolvido aqui esta relacionado à construção de um modelo químico mínimo para descrever as etapas iniciais do processo de diferenciação celular e embriogenesis.

Entretanto, a fim de evitar confusões, destaca-se inicialmente a diferença de significado e notação entre os termos: gradiente químico imposto através das fronteiras e vetor gradiente. O primeiro termo, gradiente químico imposto através das fronteiras (GQIF), é relativo a definição de fontes de reagente químico nas fronteiras do sistema, criando situações de diferença de concentração em diferentes regiões do sistema. A última, o vetor gradiente, refere-se ao operador matemático $\vec{\nabla}$ e a sua notação usual será mantida. Esta ferramenta será utilizada em análises do campo vetorial do processo de difusão dos modelos em uma condição de dinâmica estacionária. Uma vez que estes pontos foram esclarecidos, continua-se com a apresentação dos resultados.

Baseado nas considerações apresentadas em Introdução 1.4 sobre perturbações

causadas pela presença de fontes/sumidouros de reagentes químicos na emergência de padrões de Turing, simulações foram executadas com fontes de reagentes químicos, ativador e inibidor, nas fronteiras para MI e MNI, de forma que as fontes são iguais nas quatro paredes do reator. Cada simulação é feita com fontes específicas (U_{Ω}, V_{Ω}) . Para MI são $(1.80, 2.09)$, $(1.90, 2.19)$, $(2.00, 2.29)$, **$(2.10, 2.39)$** , $(2.20, 2.49)$, e $(2.30, 2.59)$, e para MNI são $(2.10, 1.80)$, $(2.20, 1.90)$, $(2.30, 2.00)$, **$(2.40, 2.10)$** , $(2.50, 2.20)$, $(2.60, 2.30)$. Note que os valores das fontes em negrito são exatamente os pontos de equilíbrio dos modelos em questão. Os resultados são apresentados nas FIGURAS 5.17 e 5.18.

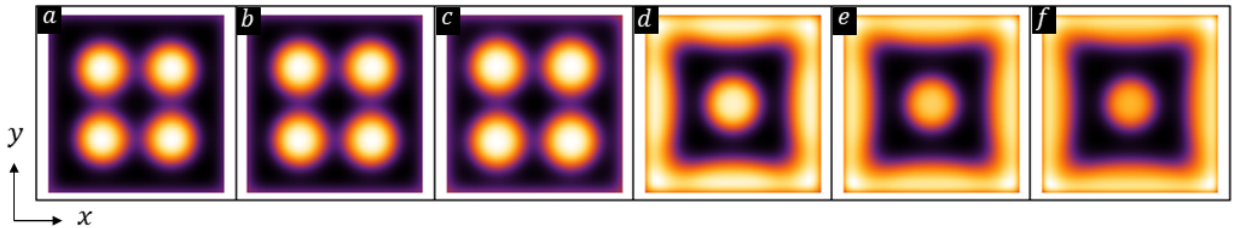


FIGURA 5.17 – Versão do MI com diferentes fontes de reagentes químicos nas fronteiras, onde (U_{Ω}, V_{Ω}) é para a) $(1.80, 2.09)$, b) $(1.90, 2.19)$, c) $(2.00, 2.29)$, d) **$(2.10, 2.39)$** , e) $(2.20, 2.49)$, f) $(2.30, 2.59)$.

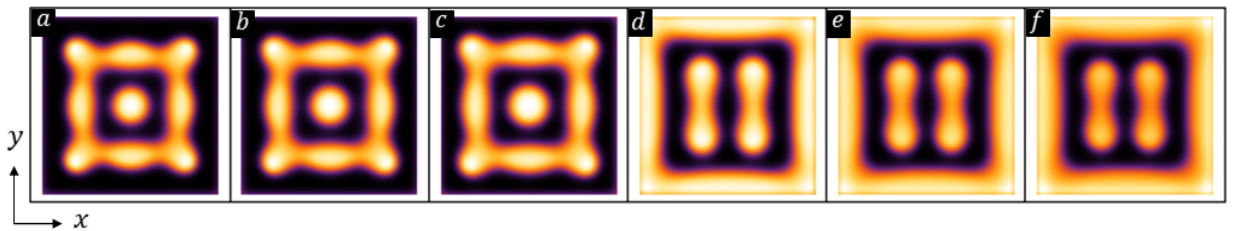


FIGURA 5.18 – Versão do MNI com diferentes fontes de reagentes químicos nas fronteiras, onde (U_{Ω}, V_{Ω}) é para a) $(2.10, 1.80)$, b) $(2.20, 1.90)$, c) $(2.30, 2.00)$, d) **$(2.40, 2.10)$** , e) $(2.50, 2.20)$, f) $(2.60, 2.30)$.

Visualmente, é possível notar que há uma quebra de simetria espacial em ambos modelos com a mudança da concentração de reagentes químicos nas fronteiras. Esse fenômeno é confirmado através da análise da taxa de produção de entropia total e do funcional de Minkowski. Nos dois modelos a transição é evidenciada pela mudança abrupta na inclinação de σ_T , como pode ser visto na FIGURA 5.19. A caracterização morfológica complementa a análise da taxa de produção de entropia, indicando a formação de diferentes estruturas espaciais, veja FIGURA 5.20. No caso do MI, as estruturas apresentam componentes topológicos com formatos geométricos similares, porém com diferentes organizações/distribuições espaciais. No caso de MNI, há expressivas diferenças topológicas entre os padrões finais.

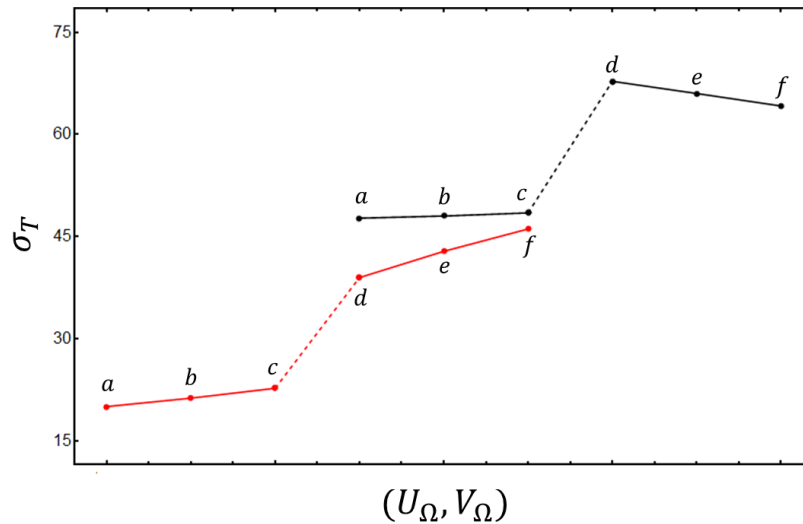


FIGURA 5.19 – A taxa de produção de entropia para diferentes fontes de reagentes químicos nas fronteiras. Os modelos MI e MNI são representados pelas cores vermelha e preta, respectivamente. As letras (a, b, c, d, e, f) indicam as mesmas condições consideradas em simulações prévias, presentes nas FIGURAS 5.17 e 5.18.

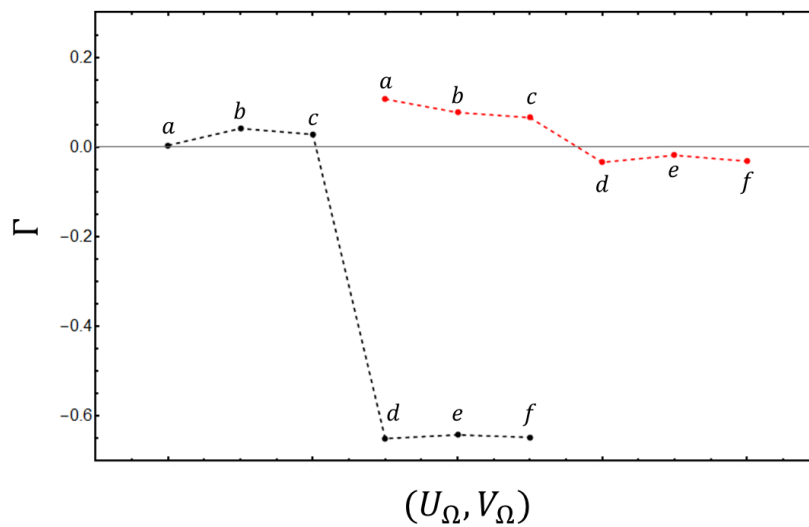


FIGURA 5.20 – Funcional de Minkowski para diferentes fontes de reagentes químicos nas fronteiras. Os modelos MI e MNI são representados pelas cores vermelha e preta, respectivamente. As letras (a, b, c, d, e, f) indicam as mesmas condições consideradas em simulações prévias, presentes nas FIGURAS 5.17 e 5.18.

Verificou-se também, através do MI, que em situações próximas ao processo de quebra de simetria, região definida entre as FIGURAS 5.17 (c) e 5.17 (d), há a formação de uma estrutura espacial intermediária, veja a FIGURA 5.21. A taxa de produção de entropia total associada a este padrão é igual a 33.0, de modo que se ajusta linearmente com os σ_T das estruturas 5.17 (d), (e) e (f). Apesar desta relação entre as taxas de produção de entropia, considera-se que o processo de quebra de simetria ocorre após a estrutura intermediária se modificar completamente, dando origem aos demais padrões.

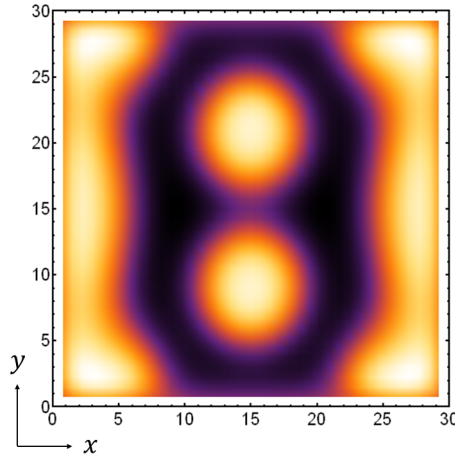


FIGURA 5.21 – Versão do MI com $(U_\Omega, V_\Omega) = (2.0666, 2.3567)$.

Espera-se que a quebra a simetria espacial dos padrões de Turing seja causada pela ação de perturbações espaciais. Portanto, a fim de entender as razões que levam à observação do fenômeno constatado, as soluções espaciais do MI são estudadas, de forma aproximada, no regime estacionário, *i.e.* $\frac{\partial u}{\partial t} = \frac{\partial v}{\partial t} = 0$, restando:

$$\xi(a - u) - r_1 + r_2 + d_u \nabla^2 u = 0 \quad (5.5)$$

$$r_1 - r_2 + d_v \nabla^2 v = 0 \quad (5.6)$$

Resolvendo as Eqs. (5.5) e (5.6), com condições de fronteira de Dirichlet, e tomando o vetor gradiente das soluções, *i.e.* $\vec{\nabla}u(x, y)$ e $\vec{\nabla}v(x, y)$, é possível verificar uma mudança da orientação do processo de difusão conforme a concentração dos reagente químicos aumenta nas fronteiras, veja a FIGURA 5.22. Mais especificamente, quando $(U_\Omega, V_\Omega) = (2.10, 2.39)$, que é o ponto de equilíbrio de MI, então $\vec{\nabla}u(x, y) = \vec{\nabla}v(x, y) = (0, 0)$. Quando a concentração é menor que a do ponto de equilíbrio $(U_\Omega, V_\Omega) = (2.00, 2.29)$, tem-se $\vec{\nabla}u(x, y) < (0, 0)$ e $\vec{\nabla}v(x, y) > (0, 0)$, e quando a concentração é maior do que o valor do ponto de equilíbrio $(U_\Omega, V_\Omega) = (2.20, 2.49)$, obtém-se $\vec{\nabla}u(x, y) > (0, 0)$ e $\vec{\nabla}v(x, y) < (0, 0)$. Logo, a alteração da orientação do processo de difusão induz o sistema a novos estados espaciais auto-organizados. Um explicação similar é dada para o MNI.

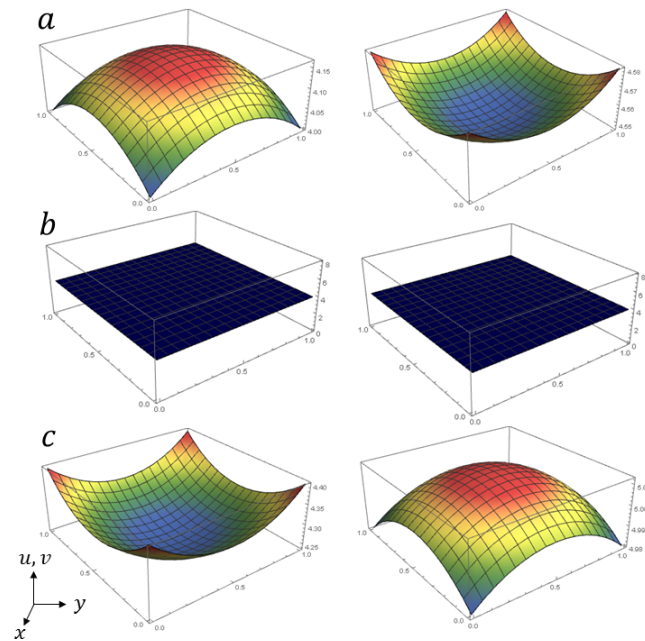


FIGURA 5.22 – Solução espacial de $u(x, y)$ (à esquerda) e $v(x, y)$ (à direita) no regime estacionário. a) $(U_{\Omega}, V_{\Omega}) = (2.00, 2.29)$ b) $(U_{\Omega}, V_{\Omega}) = (2.10, 2.39)$ - Ponto de equilíbrio c) $(U_{\Omega}, V_{\Omega}) = (2.20, 2.49)$.

Observa-se que a simetria dos padrões de Turing quebra (ou tem uma grande modificação indicando uma mudança de fase) quando a concavidade da concentração (ou o sinal do vetor gradiente) aproxima do valor do ponto de equilíbrio, veja a FIGURA 5.22. Baseado nisto, supõe-se que este comportamento é geral para sistemas químicos não lineares, com pequena dimensão espacial (área), sob instabilidade de Turing e com CFD. A razão para tal suposição esta relacionada ao comportamento aproximadamente linear quando o sistema se encontra próximo ao regime estacionário e com as fontes das fronteiras definidas ao redor dos pontos de equilíbrio – característica geral de qualquer sistema dinâmico não linear.

Mais simulações foram consideradas para entender o papel dos reagentes químicos, ativador e inibidor, nos processos de quebra de simetria. Basicamente, apenas um dos reagentes químicos foi definido com CFD, ao passo que o outro foi definido com CFN fluxo zero. Os resultados estão apresentados nas FIGURAS 5.23 e 5.24 e eles mostram que as perturbações geradas na dinâmica reacional com a presença de fontes de ativador não induzem aos fenômenos de quebra de simetria espacial. Diferentemente, fontes de inibidor nas fronteiras induzem a esses fenômenos à medida que a sua concentração se aproxima do valor do ponto de equilíbrio. Assim sendo, o agente responsável pela transição observada é o agente inibidor, e este fato deve estar relacionado com sua alta capacidade difusiva, quando comparada com a do agente ativador.

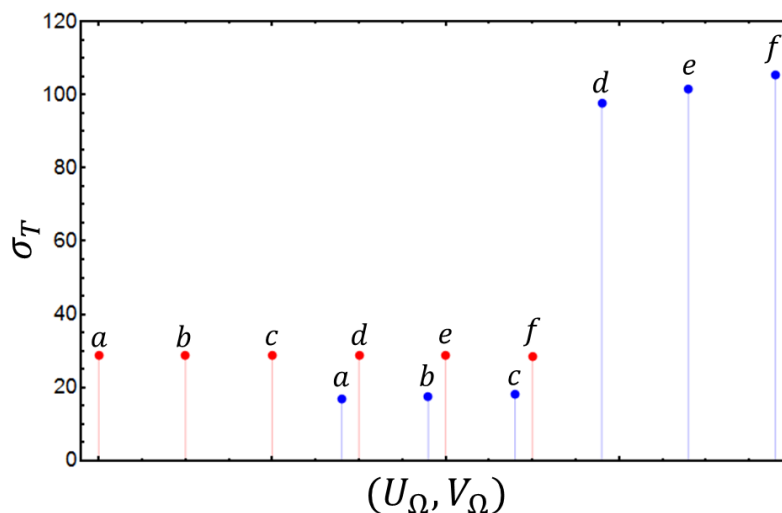


FIGURA 5.23 – Taxa de produção de entropia total para o MI, com fontes de ativador (vermelho) e inibidor (azul) nas fronteiras. As letras (*a, b, c, d, e, f*) indicam as mesmas condições consideradas nas simulações prévias, apresentada na FIGURA 5.17, para apenas um dos reagentes.

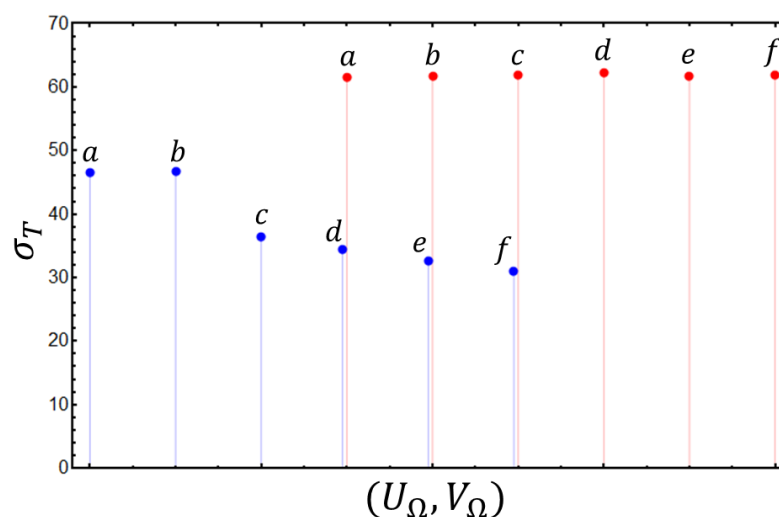


FIGURA 5.24 – Taxa de produção de entropia total para o MNI, com fontes de ativador (vermelho) e inibidor (azul) nas fronteiras. As letras (*a, b, c, d, e, f*) indicam as mesmas condições consideradas nas simulações prévias, apresentada na FIGURA 5.18, para apenas um dos reagentes.

A partir dos resultados obtidos e considerando o fato que a simetria espacial é quebrada conforme a concentração dos reagentes químicos nas fronteiras se aproximam do ponto de equilíbrio, um gradiente de reagentes químicos foi definido através das fronteiras (GQIF). O GQIF toma a direção do eixo x , de modo que em $x = 0$ e $x = x_f$ a concentração química do ativador e inibidor assumem valores mínimo e máximo, respectivamente, em uma faixa específica. Para $0 \leq x \leq x_f$, a concentração aumenta linearmente de forma que quando $x = \frac{x_f}{2}$ a concentração dos reagentes químicos é igual aos respectivos pontos de equilíbrio. A suposição do aumento linear da concentração nas fontes das fronteiras é baseada na solução

assintótica da lei de Fick. Para o MI considerou-se $u(0, y) = 1.85$, $u(x_f, y) = 2.35$, $v(0, y) = 2.15$, e $v(x_f, y) = 2.65$. E no caso do MNI $u(0, y) = 2.15$, $u(x_f, y) = 2.65$, $v(0, y) = 1.85$ e $v(x_f, y) = 2.35$. Veja a representação gráfica do GQIF na FIGURA 5.25.

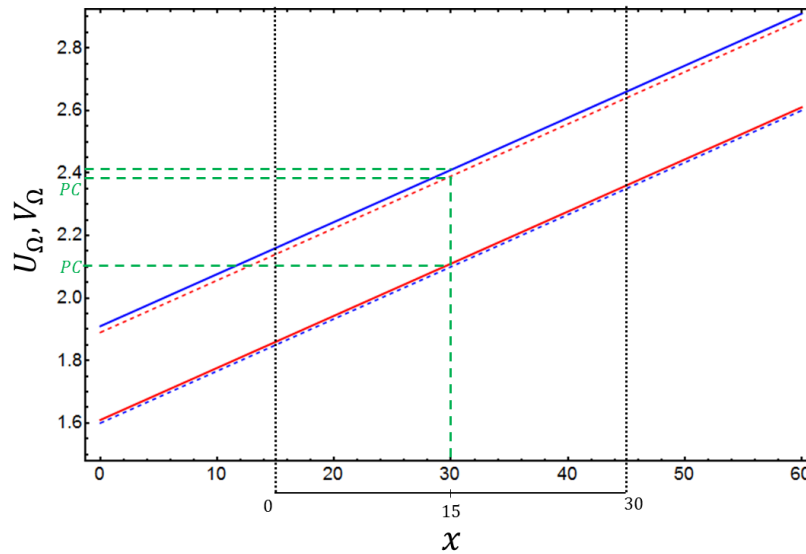


FIGURA 5.25 – GQIF linear na direção x . As concentrações dos pontos de equilíbrio estão indicados pelas linhas verdes tracejadas. A concentração de U_Ω e V_Ω são representados pelas cores azul e vermelho, respectivamente. O MI e o MNI são dados pelas linhas pontilhadas e sólidas, respectivamente. Note que, o GQIF para 30×30 u.a. está cercado pelas linhas pretas pontilhadas.

As simulações com GQIF são apresentadas na FIGURA 5.26. A partir disso, nota-se que diferentes estruturas espaciais emergem nas faixas $0 \leq x \lesssim 15$ e $15 \lesssim x \leq 30$, em ambos modelos. De outra forma, diferentes estruturas são formadas em condições que a concentração de reagentes químicos nas fronteiras são maior e menor que o ponto de equilíbrio, definido no meio do eixo x . Isto é confirmado pela taxa de produção de entropia calculada nos diferentes lados do reator, veja a TABELA 5.8, evidenciando quebra de simetria dentro do sistema em questão. Este é um importante resultado, uma vez que diferentes padrões de Turing podem ser modulados no mesmo reator por um gradiente químico imposto através das fronteiras.

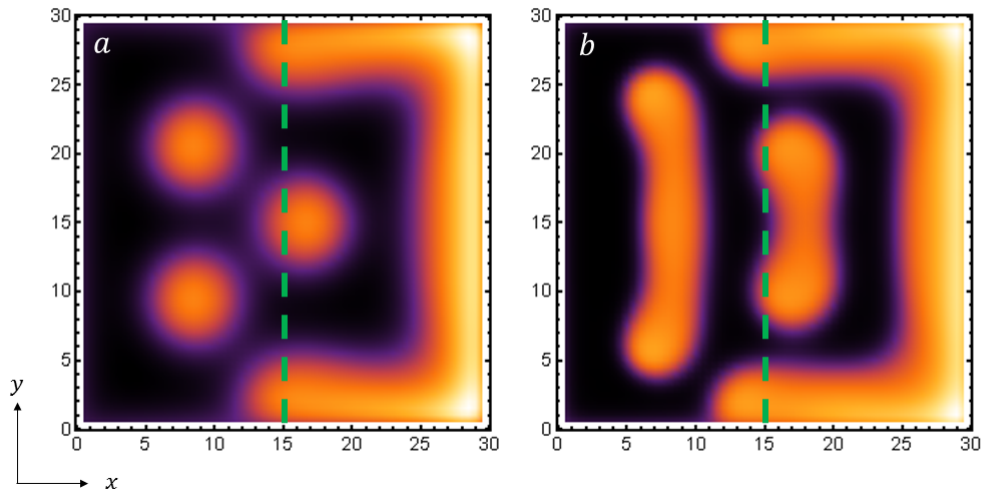


FIGURA 5.26 – Padrões de Turing modulados por GQIF na direção x . a) MI; b) MNI.

Modelo	σ_T	
	$0 \leq x \leq 15$	$15 < x \leq 30$
MI	9.7	25.5
MNI	22.2	40.0

TABELA 5.8 – Taxa de produção de entropia total para MI e MNI em diferentes regiões do sistema.

Considerando o interesse deste trabalho de obter um modelo mínimo de diferenciação celular, a quebra de simetria espacial dos padrões de Turing poderia ocorrer simultaneamente com o aumento de tamanho da célula (reator). O modelo estático proposto aqui não contempla a dinâmica de aumento do sistema, que pode ser modelado através de um esquema de acúmulo de componentes químicos e subsequente distensão do reator ao longo do GQIF. Este processo de aumento do tamanho do meio reacional pode ser representado por um reator de grande dimensão espacial.

Assim sendo, considere um sistema com dimensões 60×30 u.a., sob o mesmo GQIF considerado previamente, *i.e.* com a mesma inclinação do gradiente usado na simulação anterior. Veja a FIGURA 5.25. Nesta situação dois padrões de Turing diferentes emergem nas regiões $0 \leq x \lesssim 30$ e $30 \lesssim x \leq 60$, como era esperado, FIGURA 5.27.

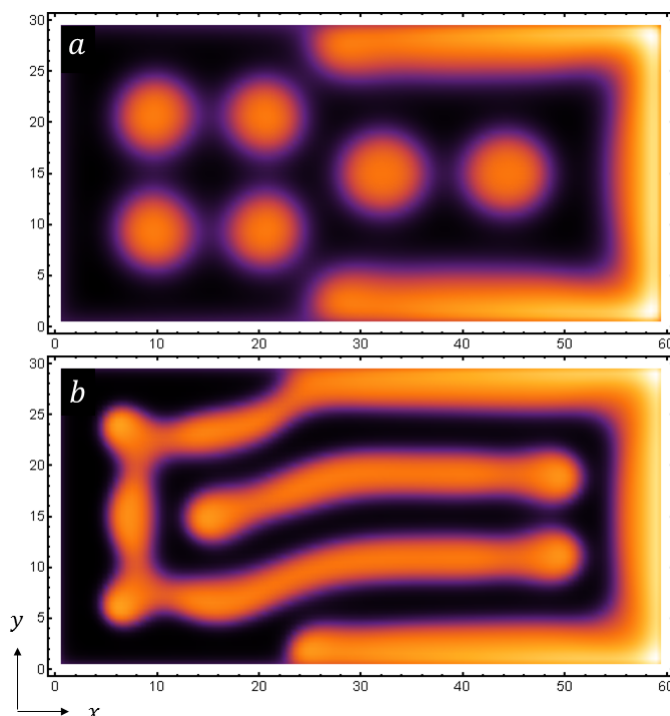


FIGURA 5.27 – Simulações de MI e MNI, representados por a e b , respectivamente, em reatores com dimensões iguais a 60×30 u.a..

Seguindo com os principais objetivos deste trabalho, considera-se agora que o reator de 60×30 u.a. pode ser dividido em dois novos reatores, devido a uma dobra de borda (membrana), convenientemente próximo ao ponto da fronteira que contém a concentração do ponto de equilíbrio. A dobra completa da membrana resultaria em dois reatores separados, com fontes de reagentes químicos nas fronteiras internas iguais ao ponto de equilíbrio, para ambos reatores. Este processo pode ser representado de maneira simplificada por dois reatores separados, que é exatamente a configuração final do sistema após a dobra da membrana.

Sendo assim, assumamos que o reator 60×30 u.a. é dividido em dois sub-reatores 30×30 u.a. exatamente na posição em que a concentração dos reagentes químicos nas fronteiras são iguais ao ponto de equilíbrio, *i.e.* $x = 30$. Dois novos reatores são formados e eles continuam sob as influências de gradiente químico. Após a ocorrência desta divisão, diferentes padrões de Turing emergem em cada novo reator. Veja as estruturas formadas a partir de MI e MNI na FIGURA 5.28. A taxa de produção de entropia e funcional de Minkowski são apresentadas na TABELA 5.9. A partir destes dados é fácil verificar que as estruturas formadas são diferentes nos dois novos reatores, para ambos modelos.

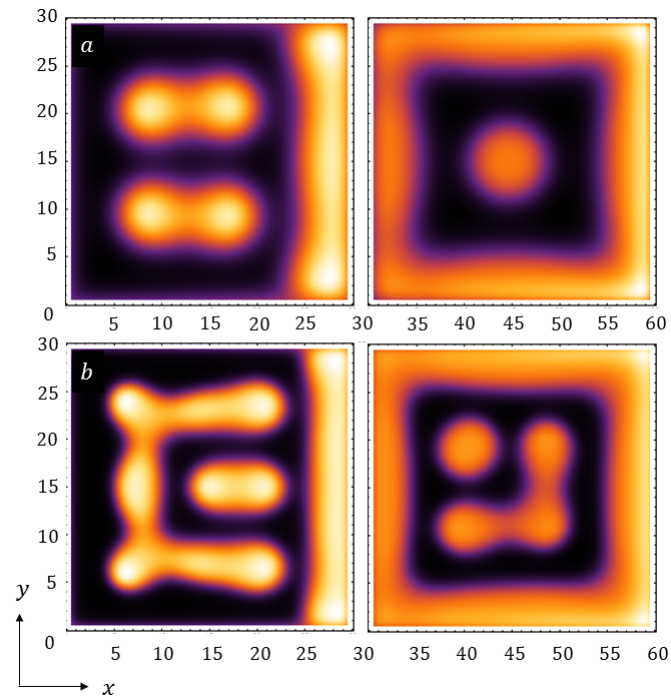


FIGURA 5.28 – Simulações de dois reatores sobre GQIF, considerando o MI, representado por a , e o MNI, representado por b . O gradiente toma a direção do eixo x e em $x = 30$ a concentração é igual ao valor do ponto de equilíbrio.

Modelo	Propiedades	Extensão	
		$0 \leq x < 30$	$30 < x \leq 60$
MI	σ_{CH}	25.3	45.8
	σ_M	0.4	0.7
	σ_T	25.5	46.4
	Γ	0.7	-0.2
MNI	σ_{CH}	51.9	61.4
	σ_M	1.4	1.1
	σ_H	0.04	0.05
	σ_T	53.3	62.5
	Γ	0.61	-0.42

TABELA 5.9 – Taxa de produção de entropia relativa à cada processo irreversível e funcional de Minkowski para o MI e MNI dos dois novos reatores.

5.8 – Modelos mínimos: Diferenciação Celular, Embriogênese e Biogênese

Considerando os resultados obtidos, argumenta-se que este trabalho pode indicar possíveis modelos químicos mínimos para a descrição dos passos iniciais da diferenciação celular, embriogênese e biogênese. Tais modelos são apresentados através das seguintes etapas:

• Diferenciação celular

- 1) Modulação de diferentes padrões de Turing (célula diferenciada) a partir de padrões de Turing referência (estrutura interna de célula tronco), FIGURA 5.5, por CFD;
- 2) Aumento do tamanho do sistema devido acumulação interna de reagentes químicos (crescimento celular), similar à FIGURA 5.27 a);
- 3) Divisão do sistema (divisão da célula) como resultado de uma dobra das bordas (membrana) após alcance de um tamanho crítico, resultando em sistemas similares (mesma célula), FIGURA 5.28 a);
- 4) Congelamento dos padrões de Turing (polimerização de proteínas) a fim de manter as estruturas internas do sistema estáveis após supressão de reagentes químicos na

fronteira.

- **Embriogênese**

- 1) Padrões de Turing referência (estrutura interna de células tronco) sofrem quebra de simetria espacial devido GQIF do reator, FIGURA 5.26 a);

- 2) Aumento do tamanho do sistema devido acumulação interna de reagentes químicos (crescimento celular), FIGURA 5.27 a));

- 3) Divisão do sistema em duas partes distintas (células) como resultado de dobra das bordas (membrana) próximo a posição da fronteira que possui fonte de reagente químico equivalente ao valor do ponto de equilíbrio do sistema dinâmico geral após o alcance do tamanho crítico, FIGURA 5.28 a);

- 4) Divisão do sistema (divisão celular) como resultado da dobra da fronteiras (membrana) após alcance do tamanho crítico, produzindo um sistema similar (mesma célula), FIGURA 5.28 a);

- 5) Continuação de quebra de simetria espacial para obter diferentes tipos de padrões de Turing (células diferentes). Sistema pode apresentar mais que um ponto de equilíbrio para outros gradiente e diferentes reagentes químicos a fim de obter estruturas de Turing distintas (células diferentes com diferentes estruturas celulares), similar à FIGURA 5.26;

- 6) Congelamento de padrões de Turing (polimerização de proteínas) a fim de manter as estruturas internas do sistema estáveis (célula) após supressão do gradiente químico.

- **Biogênese**

- 1) Modulação de diferentes padrões de Turing, a partir do padrão referência e do MNI por gradiente térmico (ou também gradiente químico) no reator (célula), veja FIGUR 5.13;

- 1') Padrão de Turing referência sofre quebra de simetria espacial devido gradiente térmico (ou também gradiente químico) formando diferentes padrões de Turing, FIGURA 5.16 ;

- 2) Aumento do tamanho do sistema devido acumulação interna de reagentes químicos (crescimento celular), FIGURA 5.27 b);

- 3) Divisão do sistema (possível divisão celular) como resultado de dobras das bordas (membrana) próximo a posição da fronteira que possui fonte de reagente químico equivalente ao valor do ponto de equilíbrio após atingir o tamanho crítico, produzindo sistemas similares (possíveis células), FIGURA 5.28 b);

- 3') Divisão do sistema (possível divisão celular) como resultado de dobra das bordas(membrana) após alcance um tamanho crítico reproduzindo sistema similares (possíveis células), FIGURA 5.28 b);

6 – Conclusões

Este trabalho buscou compreender os efeitos das condições de fronteira de Dirichlet na emergência dos padrões de Turing através de três abordagens: variações no perímetro do reator, presença de fontes térmicas e fontes de reagentes químicos nas bordas do meio reacional.

Com relação aos resultados relativos ao perímetro do meio reacional, observou-se através do MI que a variação do perímetro do reator, mantendo a área constante, possibilita a emergência de estruturas espaciais com diferentes configurações para determinadas situações. Este fenômeno possivelmente ocorre como resultado da sobreposição de padrões de Turing e não representa uma transição de estado dinâmico, verificado através das análises da taxa de produção de entropia. Tais resultados são similares aqueles reportados na literatura referentes a formação de diferentes padrões em diferentes regiões do corpo de animais, *e.g.* leopardo.

A mesma abordagem foi considerada para o MNI, de modo que os resultados mostram que a temperatura controla fortemente a forma dos padrões para casos em que há a presença ou não de fontes térmicas nas fronteiras, formando padrões complementares às estruturas espaciais de reagentes químicos. Devido a isto, não é possível obter diferentes padrões de Turing pela simples variação do perímetro reacional como foi obtido para o caso do MI.

Com relação as investigações associadas a fontes de energia térmica, estudos foram feitos a partir do MNI considerando duas formas de perturbação: fontes térmicas localizadas nas fronteiras do sistema e banho térmico. Constatou-se que não é possível obter diferentes estados espaciais a partir da variação de fontes térmicas localizadas nas fronteiras, mesmo em sistemas de pequeno tamanho. Diferentemente, em situações de banho térmico, estados espaciais podem ser sintonizados a partir do controle da temperatura do banho (temperatura externa) ou através de um gradiente de temperatura. As diferenças observadas na formação dos padrões de Turing está relacionada a magnitude das perturbações de cada situação considerada, de modo que as transições somente podem ser induzidas através do banho térmico pois, neste caso, todo o sistema químico é perturbado.

Com relação as fontes de reagentes químicos determinou-se condições suficientes para induzir quebra de simetria espacial de padrões de Turing em reatores de pequeno tamanho: A concentração de fontes de reagentes químicos, localizados nas fronteiras, devem ser equivalentes aos valores do ponto de equilíbrio do sistema dinâmico geral. Este processo inesperado é causado por uma inversão da orientação do processo de transporte de massa dos reagentes químicos. Este comportamento é facilmente verificado através das resoluções dos modelos em um regime estacionário com CFD definidas ao redor dos pontos de equilíbrio. Mais ainda, considerando os aspectos físicos e matemáticos deste problema espera-se que a quebra de simetria espacial possa ser induzida em qualquer sistema dinâmico não linear genérico, se a condição suficiente estabelecida aqui é satisfeita, em situações de instabilidade de Turing.

Também foi possível demonstrar que o inibidor tem um papel crucial na transição da quebra de simetria espacial. Investigações considerando fontes de apenas um reagente

químico, ativador ou inibidor, mostraram que apenas fontes de inibidor são capazes de induzir estas transições. Este fato é relacionado à maior capacidade difusiva do inibidor quando comparado com o ativador.

Considerando que os padrões de Turing podem ser modulados por condições de fronteira, estudou-se as perturbações causadas por um gradiente químico imposto através das fronteiras. Como esperado, observou-se que conforme a concentração dos reagentes químicos, em fontes localizadas nas fronteiras, se aproxima do ponto de equilíbrio a simetria espacial é quebrada, resultando na emergência de duas estruturas de Turing diferentes no mesmo reator.

Os resultados relacionados à capacidade de indução de quebra de simetria espacial do inibidor e da emergência de diferentes padrões de Turing no mesmo reator devido gradiente químico pode ter consequências interessantes para sistemas biológicos. A presença de fontes de inibidor, e a formação de situações de gradiente de concentração, pode possibilitar a emergência de diferentes estruturas espaciais de reagentes químicos. Partindo das ideias de Turing, processos biológicos vão ser ativados/inibidos em diferentes regiões do sistema, resultando em variabilidade biológica. Em outras palavras, apenas fontes de inibidor podem gerar variabilidade biológica, ao passo que fontes de ativador não podem.

A partir dos resultados expostos e motivados pelo legado de A. Turing, propõe-se modelos para descrever as etapas iniciais da diferenciação celular, embriogênese e biogênese, baseados na construção físico-química do sistema considerado aqui. Tais modelos são essencialmente compostos por etapas de quebra de simetria espacial dos padrões de Turing em um mesmo reator devido gradiente químico/térmico, aumento do tamanho do meio reacional e divisão do reator em dois novos sub-reatores. Este trabalho não apresenta a dinâmica de aumento e divisão, entretanto demonstra que diferentes padrões de Turing emergem na configuração final destes dois últimos processos.

Referências bibliográficas

1. NEWMAN, M. H. A. “Alan Mathison Turing, 1912-1954”, 1955.
2. HODGES, A. Turing um filósofo da natureza. Unesp, 1999.
3. DYSON, G. “The dawn of computing: Alan Turing’s bridging of logic and machines laid the foundation for digital computers”. *Nature*, London, **482** n. 7386: 459–461, 2012.
4. EPSTEIN, I. R. & POJMAN, J. A. An introduction to nonlinear chemical dynamics: oscillations, waves, patterns, and chaos. Oxford University Press, 1998.
5. TURING, A. M. “On computable numbers, with an application to the Entscheidungsproblem”. *Proceedings of the London mathematical society*, **2** n. 1: 230–265, 1937.
6. HODGES, A. “The man behind the machine”. *Nature News*, **482** n. 7386: 441, 2012.
7. BRENNER, S. “Turing centenary: life’s code script”. *Nature*, London, **482** n. 7386: 461, 2012.
8. TURING, A. M. “Computing machinery and intelligence”. *Mind*, **59** n. 236: 433, 1950.
9. TURING, A. M. “The chemical basis of morphogenesis”. *Phylosophical Transactions of the Royal Society of London*, **237** n. 641: 37–72, 1952.
10. KONDEPUDI, D. & PRIGOGINE, I. Modern thermodynamics: from heat engines to dissipative structures. John Wiley & Sons, 2014.
11. MAINI, P.; PAINTER, K. & CHAU, H. P. “Spatial pattern formation in chemical and biological systems”. *Journal of the Chemical Society, Faraday Transactions*, **93** n. 20: 3601–3610, 1997.
12. HOGAN, B. L. “Morphogenesis”. *Cell*, **96** n. 2: 225–233, 1999.
13. HOWARD, J.; GRILL, S. W. & BOIS, J. S. “Turing’s next steps: the mechanochemical basis of morphogenesis”. *Nature Reviews Molecular Cell Biology*, **12** n. 6: 392, 2011.
14. MURRAY, J. D. “A pre-pattern formation mechanism for animal coat markings”. *Journal of Theoretical Biology*, **88** n. 1: 161–199, 1981.
15. MURRAY, J. D. “How the leopard gets its spots”. *Scientific American*, **258** n. 3: 80–87, 1988.
16. BARRIO, R.; VAREA, C.; ARAGÓN, J. & MAINI, P. “A two-dimensional numerical study of spatial pattern formation in interacting Turing systems”. *Bulletin of mathematical biology*, **61** n. 3: 483–505, 1999.

17. VAREA, C.; ARAGÓN, J. & BARRIO, R. “Confined Turing patterns in growing systems”. *Physical Review E*, **56** n. 1: 1250, 1997.
18. MCGHEE, E. & PEACOCK-LOPEZ, E. “An introduction to Turing patterns in nonlinear chemical kinetics”. *Chem. Educ*, **10**: 84–94, 2005.
19. NAGAO, R. & VARELA, H. “Turing patterns in chemical systems”. *Química Nova*, **39** n. 4: 474–485, 2016.
20. TOMPKINS, N.; LI, N.; GIRABAWA, C.; HEYMANN, M.; ERMENTROUT, G. B.; EPSTEIN, I. R. & FRADEN, S. “Testing Turing’s theory of morphogenesis in chemical cells”. *Proceedings of the National Academy of Sciences*, **111** n. 12: 4397–4402, 2014.
21. REINITZ, J. “Turing centenary: pattern formation”. *Nature*, London, **482** n. 7386: 464, 2012.
22. CASTETS, V.; DULOS, E.; BOISSONADE, J. & DE KEPPER, P. “Experimental evidence of a sustained standing Turing-type nonequilibrium chemical pattern”. *Physical Review Letters*, Woodbury, **64** n. 24: 2953, 1990.
23. LENGYEL, I.; RABAI, G. & EPSTEIN, I. R. “Experimental and modeling study of oscillations in the chlorine dioxide-iodine-malonic acid reaction”. *Journal of the American Chemical Society*, Washington, **112** n. 25: 9104–9110, 1990.
24. LENGYEL, I.; RABAI, G. & EPSTEIN, I. R. “Systematic design of chemical oscillators. Part 65. Batch oscillation in the reaction of chlorine dioxide with iodine and malonic acid”. *Journal of the American Chemical Society*, Washington, **112** n. 11: 4606–4607, 1990.
25. LENGYEL, I. & EPSTEIN, I. R. “Modeling of Turing Structures in the Chlorite—Iodide—Malonic Acid—Starch Reaction System”. *Science*, Washington, **251** n. 4994: 650–652, 1991.
26. ZHABOTINSKY, A. M. “Periodical oxidation of malonic acid in solution (a study of the Belousov reaction kinetics)”. *Biofizika*, **9**: 306–311, 1964.
27. ZHABOTINSKY, A. M. “Periodic oxidizing reactions in the liquid phase”. In: . c1964. 157. 392–395.
28. FURROW, S. D.; CERVELLATI, R. & AMADORI, G. “New substrates for the oscillating Briggs- Rauscher reaction”. *The Journal of Physical Chemistry A*, **106** n. 24: 5841–5850, 2002.
29. LENGYEL, I. & EPSTEIN, I. R. “A chemical approach to designing Turing patterns in reaction-diffusion systems.”. *Proceedings of the National Academy of Sciences*, **89** n. 9: 3977–3979, 1992.

30. GREEN, M. M.; BLANKENHORN, G. & HART, H. “Which starch fraction is water-soluble, amylose or amylopectin?”. *Journal of Chemical Education*, **52** n. 11: 729, 1975.
31. LIU, H.; POJMAN, J. A.; ZHAO, Y.; PAN, C.; ZHENG, J.; YUAN, L.; HORVÁTH, A. K. & GAO, Q. “Pattern formation in the iodate–sulfite–thiosulfate reaction–diffusion system”. *Physical Chemistry Chemical Physics*, **14** n. 1: 131–137, 2012.
32. WATZL, M. & MÜNSTER, A. “Turing-like spatial patterns in a polyacrylamide-methylene blue-sulfide-oxygen system”. *Chemical physics letters*, **242** n. 3: 273–278, 1995.
33. FROHNHÖFER, H. G. & NÜSSLEIN-VOLHARD, C. “Organization of anterior pattern in the Drosophila embryo by the maternal gene bicoid”. *Nature*, London, **324** n. 6093: 120, 1986.
34. KONDO, S. & MIURA, T. “Reaction-diffusion model as a framework for understanding biological pattern formation”. *science*, **329** n. 5999: 1616–1620, 2010.
35. ATKINS, P. & DE PAULA, J. Atkins’ physical chemistry. Oxford University Press, 2002.
36. MUÑUZURI, A. P.; DOLNIK, M.; ZHABOTINSKY, A. M. & EPSTEIN, I. R. “Control of the chlorine dioxide- iodine- malonic acid oscillating reaction by illumination”. *Journal of the American Chemical Society*, Washington, **121** n. 35: 8065–8069, 1999.
37. RENNIE, R. & LAW, J. A dictionary of chemistry. Oxford University Press, 2016.
38. NAGAO, P. D. R.; DE MIRANDA, R. C. C.; EPSTEIN, I. R. & DOLNIK, M. “Modulation of Turing Patterns in the CDIMA Reaction by Ultraviolet and Visible Light”. *The Journal of Physical Chemistry A*, **123**: 992–998, 2019.
39. KARPEL VEL LEITNER, N.; DE LAAT, J. & DORE, M. “Photodecomposition of Chlorine Dioxide and Chlorite by U.V. Irradiation - Part I. Photo-Products”. *Water Research*, **26**: 1655–1664, 1992.
40. SERNA, H.; MUÑUZURI, A. P. & BARRAGÁN, D. “Thermodynamic and morphological characterization of Turing patterns in non-isothermal reaction–diffusion systems”. *Physical Chemistry Chemical Physics*, **19** n. 22: 14401–14411, 2017.
41. GRAY, P.; KAY, S. & SCOTT, S. K. “Oscillations of an exothermic reaction in a closed system-I. Approximate (exponential) representation of Arrhenius temperature-dependence”. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, **416** n. 1851: 321–341, 1988.
42. SIMAKOV, D. S. & PÉREZ-MERCADER, J. “Noise induced oscillations and coherence resonance in a generic model of the nonisothermal chemical oscillator”. *Scientific reports*, **3**: 2404, 2013.

43. EINSTEIN, A. Investigations on the theory of the brownian movement. Courier Corporation, 1956.
44. OUYANG, Q. & SWINNEY, H. L. “Transition from a uniform state to hexagonal and striped Turing patterns”. *Nature*, London, **352** n. 6336: 610, 1991.
45. PULLELA, S. R.; CRISTANCHO, D.; HE, P.; LUO, D.; HALL, K. R. & CHENG, Z. “Temperature dependence of the Oregonator model for the Belousov-Zhabotinsky reaction”. *Physical Chemistry Chemical Physics*, **11** n. 21: 4236–4243, 2009.
46. VAREA, C.; ARAGON, J. & BARRIO, R. “Turing patterns on a sphere”. *Physical Review E*, **60** n. 4: 4588, 1999.
47. HADELER, K.; AN DER HEIDEN, U. & ROTHE, F. “Nonhomogeneous spatial distributions of populations”. *Journal of Mathematical Biology*, **1** n. 2: 165–176, 1974.
48. FRANCO, N. B. Cálculo numérico. Pearson, 2006.
49. ARCURI, P. & MURRAY, J. “Pattern sensitivity to boundary and initial conditions in reaction-diffusion models”. *Journal of mathematical biology*, **24** n. 2: 141–165, 1986.
50. MURRAY, J. D. “On pattern formation mechanisms for lepidopteran wing patterns and mammalian coat markings”. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, **295** n. 1078: 473–496, 1981.
51. DILLON, R.; MAINI, P. & OTHMER, H. “Pattern formation in generalized Turing systems”. *Journal of Mathematical Biology*, **32** n. 4: 345–393, 1994.
52. PRIGOGINE, I. & LEFEVER, R. “Symmetry breaking instabilities in dissipative systems. II”. *The Journal of Chemical Physics*, **48** n. 4: 1695–1700, 1968.
53. LEFEVER, R.; NICOLIS, G. & BORCKMANS, P. “The Brusselator: it does oscillate all the same”. *Journal of the Chemical Society, Faraday Transactions 1: Physical Chemistry in Condensed Phases*, **84** n. 4: 1013–1023, 1988.
54. FOGLER, H. S. Elements of chemical reaction engineering. Pearson Education, 2006.
55. SETAYESHGAR, S. & CROSS, M. “Turing instability in a boundary-fed system”. *Physical Review E*, **58** n. 4: 4485, 1998.
56. WIGGINS, S. Introduction to applied nonlinear dynamical systems and chaos. Springer Science & Business Media, 2003. 2.
57. MONTEIRO, L. H. A. Sistemas dinâmicos. Editora Livraria da Física, 2006.

58. GUIU-SOUTO, J.; CARBALLIDO-LANDEIRA, J. & MUNUZURI, A. P. “Characterizing topological transitions in a Turing-pattern-forming reaction-diffusion system”. *Physical Review E*, **85** n. 5: 056205, 2012.
59. KONDEPUDI, D. & KAPCHA, L. “Entropy production in chiral symmetry breaking transitions”. *Chirality: The Pharmacological, Biological, and Chemical Consequences of Molecular Asymmetry*, **20** n. 3-4: 524–528, 2008.
60. GLANSDORFF, P. & PRIGOGINE, I. Thermodynamic theory of structure, stability and fluctuations. John Wiley & Sons, 1971.
61. YOSHIDA, N. “Entropy production in a chemical system involving an autocatalytic reaction in an isothermal, continuous stirred tank reactor”. *The Journal of chemical physics*, **92** n. 4: 2593–2597, 1990.
62. BOTRÈ, C.; LUCARINI, C.; MEMOLI, A. & D’ASCENZO, E. “On the entropy production in oscillating chemical systems”. *Journal of Electroanalytical Chemistry and Interfacial Electrochemistry*, **128**: 201–212, 1981.
63. YOSHIDA, N. “Thermodynamics of transitions between nonequilibrium states”. *The Journal of Physical Chemistry B*, **101** n. 40: 8006–8011, 1997.
64. SILVA-DIAS, L. & LÓPEZ-CASTILLO, A. “Stochastic chiral symmetry breaking process besides the deterministic one”. *Physical Chemistry Chemical Physics*, **19** n. 43: 29424–29428, 2017.
65. LAHEY, T. M. & ELLIS, T. Fortran 90 programming. Addison-Wesley Longman Publishing Co., Inc., 1994.
66. SAFONOV, V. O. Using aspect-oriented programming for trustworthy software development. John Wiley & Sons, 2008. 5.
67. BLANCO, A. L. et al. Melquiades: um programa de monte carlo para a simulação de sistemas multicomponentes utilizando modelos de potenciais arbitrários. Tese de Doutorado, Universidade Federal de São Carlos, São Carlos, SP, 2017.
68. GOMES, J. Paralelização de algoritmo de simulação de monte carlo para a adsorção em superfícies heterogêneas bidimensionais. Dissertação de Mestrado, Universidade Estadual de Maringá, Maringá, PR, 2009.
69. HWANG, K. & FAYE, A. Computer architecture and parallel processing. McGraw-Hill, New York, NY, USA, 1984.
70. PACHECO, P. Parallel programming with mpi. Morgan Kaufmann, 1997.
71. KONDEPUDI, D. Introduction to modern thermodynamics. Wiley, 2008.

72. WRAY, S. Alternating direction implicit finite difference methods for the heat equation on general domains in two and three dimensions. Masters thesis, Colorado School of Mines, Golden, CO, 2016.
73. PEACEMAN, D. W. & RACHFORD, JR, H. H. “The numerical solution of parabolic and elliptic differential equations”. *Journal of the Society for industrial and Applied Mathematics*, **3** n. 1: 28–41, 1955.
74. BIALECKI, B. & FERNANDES, R. I. “Orthogonal spline collocation Laplace-modified and alternating-direction methods for parabolic problems on rectangles”. *mathematics of computation*, **60** n. 202: 545–573, 1993.

7 – Apêndices

7.1 – Transformações adimensionais

Variáveis e parâmetros adimensionais:

$$\theta = \frac{T}{T_E}$$

$$\theta_\Omega = \frac{T_\Omega}{T_E}$$

$$u = \frac{U}{C_T}$$

$$v = \frac{V}{C_T}$$

$$a = \frac{A}{C_T}$$

$$b = \frac{B}{C_T}$$

$$d = \frac{D}{C_T}$$

$$\tau = C_T^2 K_{f_1} t$$

$$k_{f_1} = \frac{K_{f_1}}{K_{f_1}} = 1$$

$$k_{f_2} = \frac{K_{f_2}}{K_{f_1} C_T}$$

$$k_{r_1} = \frac{K_{r_1}}{K_{f_1}}$$

$$k_{r_2} = \frac{K_{r_2}}{K_{f_1} C_T}$$

$$\xi = \frac{\lambda}{K_{f_1} C_T^2}$$

$$x = \frac{X}{l}$$

$$y = \frac{Y}{l}$$

$$e_i = \frac{E_i}{RT_E}$$

$$\phi = \frac{\beta A_r}{v_0 C_{ps} K_{f_1} C_T^3}$$

$$d_u = \frac{D_U}{K_{f_1} l^2 C_T^2}$$

$$d_v = \frac{D_V}{K_{f_1} l^2 C_T^2}$$

$$d_\theta = \frac{\alpha}{K_{f_1} l^2 C_T^2}$$

$$\gamma_u = \frac{C_{p,u}}{C_{ps}}$$

$$\Delta H'_{r_1} = \frac{\Delta H_{R_1}}{C_{ps} T_E}$$

$$\Delta H'_{r_2} = \frac{\Delta H_{R_2}}{C_{ps} T_E}$$

7.2 – Método implícito da direção alternada (ADI)

Nesta seção o método numérico de integração de equações diferenciais parciais (EDP) parabólicas em duas dimensões espaciais de Peaceman-Rachford, também conhecido como método implícito da direção alternada, será descrito matematicamente de forma explícita. Para sistemas em três dimensões veja a referência⁷².

A aproximação numérica das soluções de EDP's em uma dimensão espacial é feita majoritariamente através do método implícito de Crank-Nicolson, devido a estabilidade numérica desta abordagem.⁴⁸ Este método também pode ser utilizado para descrever a evolução de EDP's em duas dimensões espaciais, entretanto, neste caso é necessário a resolução de um conjunto de sistemas de equações lineares cuja matriz tem dimensão igual ao número de pontos da malha das variáveis espaciais, tornando a abordagem desvantajosa em termos computacionais. Considerando estes fatos, modificações no método de Crank-Nicolson podem ser feitas a fim de manter a ordem de convergência e tornar a computação mais eficiente. Tais alterações resultam no método implícito das direções alternadas (ADI).

O método ADI ocorre em dois passos, de modo que em cada passo apenas uma variável espacial é tratada de forma implícita. Basicamente, no primeiro passo o termo u_{xx}^m é tratado implicitamente e u_{yy} explicitamente e no segundo passo as ordens são invertidas. Vale ressaltar que estes dois passos numéricos no espaço ocorrem dentro de um passo numérico do tempo, de modo que tais passos representam passos "intermediários". Assim, o primeiro passo intermediário não representa um passo espacial completo, mas sim um "meio" passo. Um passo espacial completo ocorre depois de dois intermediários.⁴⁸

A descrição apresentada a baixo é baseada na dissertação.⁷²

Considere a seguinte EDP em duas dimensões espaciais :

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x, y, t) \quad (7.1)$$

Onde o domínio espacial de simulação será $\Omega = (0, Z) \times (0, Z)$, um quadrado, e o tempo $t = [0, T]$. Considerando estas informações, defini-se as condições iniciais como:

$$u(x, y, 0) = g_1(x, y) \quad (x, y) \in \Omega \quad (7.2)$$

e as condições de fronteira:

$$u(x, y, t) = g_2(x, y, t) \quad (x, y) \in \partial\Omega \text{ e } t \in (0, T] \quad (7.3)$$

A abordagem numérica exige que as variáveis independentes sejam discretas. Assim sendo, o intervalo de tempo será dividido em M partes iguais, e o espaço, tanto na direção

^mSeja $u = u(x, y, t)$, então $u_{xx} = \frac{\partial^2 u(x, y, t)}{\partial x^2}$.

x quanto na y , em N partes iguais. Em consequência destas considerações o passo numérico no tempo tem tamanho $\tau = T/M$ e o passo numérico no espaço tem tamanho $h = Z/N$. Logo, $x_i = ih$ e $y_j = jh$ com $i, j = 0, 1, \dots, N$.

Considere também $U_{i,j}^m \approx u(x_i, y_j, t_m)$ a representação da aproximação discreta da solução.

O método de Peaceman-Rachford⁷³ foi apresentado em 1955 da seguinte forma:

$$(1 - \frac{\tau}{2}\delta_x^2 U_{i,j}^{m+1/2}) = (1 + \frac{\tau}{2}\delta_y^2)U_{i,j}^m + \frac{\tau}{2}f_{i,j}^{m+1/2} \quad (7.4)$$

$$(1 - \frac{\tau}{2}\delta_y^2 U_{i,j}^{m+1}) = (1 + \frac{\tau}{2}\delta_x^2)U_{i,j}^{m+1/2} + \frac{\tau}{2}f_{i,j}^{m+1/2} \quad (7.5)$$

De modo que δ_x^2 e δ_y^2 são os seguintes operadores:

$$\delta_x^2 U_{i,j}^m = \frac{U_{i-1,j}^m - 2U_{i,j}^m + U_{i+1,j}^m}{h^2}$$

$$\delta_y^2 U_{i,j}^m = \frac{U_{i,j-1}^m - 2U_{i,j}^m + U_{i,j+1}^m}{h^2}$$

Apesar das Eqs. (7.4) e (7.5) representarem a versão original do método implícito das direções alternadas a versão proposta por Bialecki e Fernandes⁷⁴ foi considerada neste trabalho. Tal versão é apresentada abaixo:

$$W_{i,j} = (1 + \frac{\tau}{2}\delta_y^2)U_{i,j}^m \quad (7.6)$$

$$(1 - \frac{\tau}{2}\delta_x^2 U_{i,j}^{m+1/2}) = V_{i,j} + \frac{\tau}{2}f_{i,j}^{m+1/2} \quad (7.7)$$

$$(1 - \frac{\tau}{2}\delta_y^2 U_{i,j}^{m+1}) = 2U_{i,j}^{m+1/2} - V_{i,j} \quad (7.8)$$

Para $i, j = 1, \dots, N-1$ e $m = 0, \dots, M-1$. Note que a variação dos termos i e j desconsidera as fronteiras. De modo geral, nesta versão o termo $W_{i,j}$ é uma nova função, proveniente de uma equação explícita, Eq. (7.6). Esta abordagem aumenta o número de equações, entretanto torna a sua programação mais simples.

Note que na Eq. (7.7) o termo $U_{i,j}^{m+1/2}$ representa a aproximação da solução no passo "meio", mencionado no início desta seção. Este termo pode ser escrito explicitamente a partir da combinação das Eqs (7.6) e (7.7):

$$U_{i,j}^{m+1/2} = \frac{1}{2}(1 - \frac{\tau}{2}\delta_y^2)U_{i,j}^{m+1} + \frac{1}{2}(1 + \frac{\tau}{2}\delta_y^2)U_{i,j}^m \quad (7.9)$$

Os valores da fronteira neste passo intermediário são obtidos através da Eq. (7.9), e podem ser escritos como:

$$U_{i,j}^{m+1/2} = \frac{1}{2} \left(1 - \frac{\tau}{2} \delta_y^2\right) g_2(x_i, y_j, t_{m+1}) + \frac{1}{2} \left(1 + \frac{\tau}{2} \delta_y^2\right) g_2(x_i, y_j, t_m) \quad (7.10)$$

Com a apresentação dos aspectos mais relevantes do método, as Eq's (7.6), (7.7) e (7.8) serão descritas através da notação matricial.

1. Primeiro Passo: Eq. (7.6)

A primeira aproximação ocorre na coordenada espacial y de modo que o termo i é fixo. Assim, com $j = 1, \dots, N - 1$ é possível escrever:

$$\begin{aligned} W_{i,1} &= U_{i,1}^m + \frac{\tau}{2h^2} (U_{i,0}^m - 2U_{i,1}^m + U_{i,2}^m) \\ W_{i,2} &= U_{i,2}^m + \frac{\tau}{2h^2} (U_{i,1}^m - 2U_{i,2}^m + U_{i,3}^m) \\ &\vdots \\ &\vdots \\ &\vdots \\ W_{i,N-1} &= U_{i,N-1}^m + \frac{\tau}{2h^2} (U_{i,N-2}^m - 2U_{i,N-1}^m + U_{i,N}^m) \end{aligned}$$

Defini-se também vetores em \mathbb{R}^{N-1} :

$$U_{i,\cdot}^m = \begin{pmatrix} U_{i,1}^m \\ U_{i,2}^m \\ \cdot \\ \cdot \\ \cdot \\ U_{i,N-1}^m \end{pmatrix} \quad W_{i,\cdot}^m = \begin{pmatrix} W_{i,1}^m \\ W_{i,2}^m \\ \cdot \\ \cdot \\ \cdot \\ W_{i,N-1}^m \end{pmatrix}$$

e também a matriz tridiagonal B em $\mathbb{R}^{N-1} \times \mathbb{R}^{N-1}$.

$$B = \begin{pmatrix} -2 & 1 & 0 & 0 & & \\ 1 & -2 & 1 & 0 & \cdot & \cdot & \cdot \\ 0 & 1 & -2 & 1 & & & \\ 0 & 0 & 1 & -2 & & & \\ & & & & \cdot & & \\ & & & & \cdot & & \\ & & & & \cdot & & \end{pmatrix}$$

Portanto, a Eq. (7.6) pode ser rescrita como:

$$W_{i,\cdot}^m = \left(I + \frac{\tau}{2h^2} B \right) U_{i,\cdot}^m + \begin{pmatrix} V_{i,0}^m \\ 0 \\ \cdot \\ \cdot \\ 0 \\ V_{i,N}^m \end{pmatrix} \quad (7.11)$$

Com I a matriz identidade $\mathbb{R}^{N-1} \times \mathbb{R}^{N-1}$.

2. Segundo Passo Eq. (7.7)

O mesmo pode ser feito com a Eq. (7.7). Expandindo os operadores e fixando j , uma vez que a aproximação é em x . As equações são escritas da seguinte forma:

$$\begin{aligned} U_{1,j}^{m+1/2} - \frac{\tau}{2}(U_{0,j}^{m+1/2} - 2U_{1,j}^{m+1/2} + U_{2,j}^{m+1/2}) &= W_{1,j}^m + \frac{\tau}{2}f_{1,j}^{m+1/2} \\ U_{2,j}^{m+1/2} - \frac{\tau}{2}(U_{1,j}^{m+1/2} - 2U_{2,j}^{m+1/2} + U_{3,j}^{m+1/2}) &= W_{2,j}^m + \frac{\tau}{2}f_{2,j}^{m+1/2} \\ &\cdot \\ &\cdot \\ &\cdot \\ U_{N-1,j}^{m+1/2} - \frac{\tau}{2}(U_{N-2,j}^{m+1/2} - 2U_{N-1,j}^{m+1/2} + U_{N,j}^{m+1/2}) &= W_{N-1,j}^m + \frac{\tau}{2}f_{N-1,j}^{m+1/2} \end{aligned}$$

De forma similar ao primeiro passo, vetores em \mathbb{R}^{N-1} são definidos:

$$U_{\cdot,j}^m = \begin{pmatrix} U_{1,j}^m \\ U_{2,j}^m \\ \cdot \\ \cdot \\ U_{N-1,j}^m \end{pmatrix} \quad W_{\cdot,j}^m = \begin{pmatrix} W_{1,j}^m \\ W_{2,j}^m \\ \cdot \\ \cdot \\ W_{N-1,j}^m \end{pmatrix} \quad f_{\cdot,j}^{m+1/2} = \begin{pmatrix} f_{1,j}^{m+1/2} \\ f_{2,j}^{m+1/2} \\ \cdot \\ \cdot \\ f_{N-1,j}^{m+1/2} \end{pmatrix}$$

e com isso a Eq. (7.7) é reescrita como:

$$(I - \frac{\tau}{2h^2}B)U_{.,j}^{m+1/2} = W_{.,j}^m + \frac{\tau}{2}f_{.,j}^{m+1/2} + \frac{\tau}{2h^2} \begin{pmatrix} U_{0,j}^{m+1/2} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ U_{N,j}^{m+1/2} \end{pmatrix} \quad (7.12)$$

3. Terceiro Passo Eq. (7.8)

O mesmo procedimento é feito para a última equação, Eq. (7.8),

$$\begin{aligned} U_{i,1}^{m+1} - \frac{\tau}{2}(U_{i,0}^{m+1} - 2U_{i,1}^{m+1} + U_{i,2}^{m+1}) &= 2U_{i,1}^{m+1/2} + W_{i,1}^m \\ U_{i,2}^{m+1} - \frac{\tau}{2}(U_{i,1}^{m+1} - 2U_{i,2}^{m+1} + U_{i,3}^{m+1}) &= 2U_{i,2}^{m+1/2} + W_{i,2}^m \\ &\cdot \\ &\cdot \\ &\cdot \\ U_{i,N-1}^{m+1} - \frac{\tau}{2}(U_{i,N-2}^{m+1} - 2U_{i,N-1}^{m+1} + U_{i,N}^{m+1}) &= 2U_{i,N-1}^{m+1/2} + W_{i,N-1}^m \end{aligned}$$

Obtendo,

$$(I - \frac{\tau}{2h^2}B)U_{i,.}^{m+1} = 2U_{i,.}^{m+1/2} - W_{i,.}^m + \frac{\tau}{2h^2} \begin{pmatrix} U_{i,0}^{m+1} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ U_{i,N}^{m+1} \end{pmatrix} \quad (7.13)$$

Assim sendo o programa desenvolvido neste trabalho foi baseado no método de integração de Peaceman-Rachford e as Eqs. (7.11), (7.12) e (7.13) são as formas utilizadas para implementação computacional.

7.3 – Instabilidade de Turing

Informações retiradas das referências.^{4,18}

Esta seção apresenta o tratamento de análise de estabilidade das soluções de um sistema genérico tipo reação-difusão, na forma linear, e determinação das regiões de instabilidade de Turing.

Considere o seguinte sistema genérico de equações diferenciais parciais, formado por duas variáveis dependentes (U, V):

$$\frac{\partial U(r, t)}{\partial t} = f(U, V) + D_U \nabla^2 U(r, t) \quad (7.14)$$

$$\frac{\partial V(r, t)}{\partial t} = g(U, V) + D_V \nabla^2 V(r, t) \quad (7.15)$$

Onde $f(U, V)$ e $g(U, V)$ são termos não lineares e D_U e D_V são as constantes de difusão dos termos U e V , respectivamente. A determinação das regiões de instabilidade de Turing inicia-se através da determinação dos pontos de equilíbrio (estados estacionários), ou seja, situação em que:

$$f(U^*, V^*) = g(U^*, V^*) = 0 \quad (7.16)$$

Para investigar a estabilidade do(s) par(es) (U^*, V^*), considera-se uma pequena perturbação ($u(r, t), v(r, t)$) em torno do ponto de equilíbrio e assim analisa-se o comportamento das soluções sob influência destas pequenas perturbações. Portanto,

$$U(r, t) = U^* + u(r, t) \quad (7.17)$$

$$V(r, t) = V^* + v(r, t) \quad (7.18)$$

Utilizando a série de Taylor é possível obter as versões lineares de $f(U, V)$ e $g(U, V)$ em torno do ponto (U^*, V^*):

$$f(U, V)|_{(U^*, V^*)} = f(U^*, V^*) + u(r, t) \left. \frac{\partial f}{\partial U} \right|_{(U, V)=(U^*, V^*)} + v(r, t) \left. \frac{\partial f}{\partial V} \right|_{(U, V)=(U^*, V^*)} + \dots \quad (7.19)$$

como, $f(U^*, V^*) = 0$, então

$$f(U, V)|_{(U^*, V^*)} = u(r, t) f_U + v(r, t) f_V \quad (7.20)$$

onde $f_U = \frac{\partial f}{\partial U} \Big|_{(U,V)=(U^*,V^*)}$ e $f_V = \frac{\partial f}{\partial V} \Big|_{(U,V)=(U^*,V^*)}$. O mesmo exercício pode ser feito para $g(U, V)$ gerando:

$$g(U, V)|_{(U^*,V^*)} = u(r, t)g_U + v(r, t)g_V \quad (7.21)$$

e de modo similar à f , $g_U = \frac{\partial g}{\partial U} \Big|_{(U,V)=(U^*,V^*)}$ e $g_V = \frac{\partial g}{\partial V} \Big|_{(U,V)=(U^*,V^*)}$.

Desta forma, obtemos a versão linearizada do sistema formado pelas Eqs. (7.14)

e (7.15):

$$\frac{\partial u(r, t)}{\partial t} = u(r, t)f_U + v(r, t)f_V + D_U \nabla^2 u(r, t) \quad (7.22)$$

$$\frac{\partial v(r, t)}{\partial t} = u(r, t)g_U + v(r, t)g_V + D_V \nabla^2 v(r, t) \quad (7.23)$$

É possível supor que as soluções do sistema de equações lineares (7.22) e (7.23) sejam da seguinte forma:

$$u(r, t) = e^{\lambda t} \sum_k u_k F_k(r) \quad (7.24)$$

$$v(r, t) = e^{\lambda t} \sum_k v_k F_k(r) \quad (7.25)$$

onde, F_k é uma série de Fourier e satisfaz as condições de fronteira do problema físico. Verifica-se também que:

$$\nabla^2 F_k = k^2 F_k \quad (7.26)$$

Considerando o que foi colocado e utilizando notação matricial é possível escrever:

$$\sum_k F_k(r) \begin{pmatrix} \lambda - f_U + D_U k^2 & f_V \\ g_U & \lambda - g_V + D_V k^2 \end{pmatrix} \begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (7.27)$$

Com,

$$J = \begin{pmatrix} \lambda - f_U + D_U k^2 & f_V \\ g_U & \lambda - g_V + D_V k^2 \end{pmatrix} \quad (7.28)$$

Como as funções $F_k(r)$ são linearmente independentes e formam um conjunto completo, as soluções não triviais são encontradas quando o determinante da matriz central (J)

da Eq. (7.28) é zero. Assim, obtém-se o seguinte polinômio característico:

$$\lambda^2 + [(D_U + D_V)k^2 - (f_U + g_V)]\lambda + [D_U D_V k^4 - (D_V f_U + D_U g_V)k^2 + (f_U g_V - f_V g_U)] = 0 \quad (7.29)$$

Para estudar a estabilidade temporal das soluções do sistema de Eqs. (7.22) e (7.23) basta tomar $k = 0$, reduzindo o polinômio característico Eq. (7.29) a:

$$\lambda^2 - Tr(J(k = 0)) \lambda + Det(J(k = 0)) = 0 \quad (7.30)$$

onde,

$$Tr(J(k = 0)) = f_U + g_V \quad (7.31)$$

$$Det(J(k = 0)) = (f_U g_V - f_V g_U) \quad (7.32)$$

Segundo os critérios de estabilidade de Routh-Hurwitz, as soluções da Eq. (7.30) apresentarão parte real negativa se:

$$-Tr(J(k = 0)) = -(f_U + g_V) > 0 \quad (7.33)$$

$$Det(J(k = 0)) = (f_U g_V - f_V g_U) > 0 \quad (7.34)$$

Tomando agora $k \neq 0$ e definindo:

$$M(k) \equiv [(D_U + D_V)k^2 - Tr(J(k = 0))] \quad (7.35)$$

$$h(k) \equiv [D_U D_V k^4 - (D_V f_U + D_U g_V)k^2 + Det(J(k = 0))] \quad (7.36)$$

O polinômio característico é reescrito da seguinte forma:

$$\lambda^2 + M(k) + h(k) = 0 \quad (7.37)$$

de modo, que as suas soluções são representadas por:

$$\lambda_{\pm} = \frac{1}{2}(-M(k) \pm \sqrt{M(k)^2 - 4h(k)}) \quad (7.38)$$

Neste momento busca-se por soluções em que a parte real dos autovalores seja positiva, gerando instabilidade espacial. Isto requer que $h(k) < 0$. Consequentemente, uma

condição necessária mas não suficiente é a condição em que:

$$D_V f_U + D_U g_V > 0 \quad (7.39)$$

Mas é possível encontrar uma solução que seja necessária e suficiente, garantindo que os valores de k sejam reais. Se fizermos $h(k) = 0$, tem-se as seguintes soluções:

$$k_{\pm}^2 = \frac{1}{2} \left(\frac{f_U}{D_U} + \frac{g_V}{D_V} \right)^2 \pm \frac{1}{2} \sqrt{\left(\frac{f_U}{D_U} + \frac{g_V}{D_V} \right)^2 - \frac{4 \text{Det}(J(k=0))}{D_U D_V}} \quad (7.40)$$

Para garantir que k seja um número real,

$$\left(\frac{f_U}{D_U} + \frac{g_V}{D_V} \right)^2 - \frac{4 \text{Det}(J(k=0))}{D_U D_V} > 0 \quad (7.41)$$

Assim sendo, para um sistema composto por duas variáveis dependentes os seguintes critérios devem ser satisfeitos para a observação da emergência de estruturas espaciais não homogêneas do tipo Turing:

$$-(f_U + g_V) > 0 \quad (7.42)$$

$$(f_U g_V - f_V g_U) > 0 \quad (7.43)$$

$$D_V f_U + D_U g_V > 0 \quad (7.44)$$

$$\left(\frac{f_U}{D_U} + \frac{g_V}{D_V} \right)^2 - \frac{4 \text{Det}(J(k=0))}{D_U D_V} > 0 \quad (7.45)$$

Para a situação em que o modelo é composto por três variáveis dependentes, a determinação de critérios analíticos elegantes como os apresentados previamente é um trabalho muito complicado. Assim, as condições de estabilidade/instabilidade são obtidas através do polinômio característico, agora de grau três (3), considerando os critérios de estabilidade de Routh-Hurwitz. O polinômio característico neste caso pode ser escrito como:

$$\lambda^3 - \text{Tr}(J(k)) \lambda^2 + \sum_{i=j=1}^3 |M(J(k))_{i,j}| \lambda - \text{Det}(J(k)) = 0 \quad (7.46)$$

com $M(J(k))$ os menores da matriz $J(k)$. A matriz de Hurwitz pode ser escrita

como:

$$H = \begin{pmatrix} -Tr(J(k)) & -Det(J(k)) & 0 \\ 1 & \sum_{i=j=1}^3 |M(J(k))_{i,j}| & 0 \\ 0 & -Tr(J(k)) & -Det(J(k)) \end{pmatrix} \quad (7.47)$$

Seguindo os critérios de Routh-Hurwitz a Eq. (7.46) apresentará $Re(\lambda_j) < 0$, ou seja, soluções estáveis, se os determinantes de Hurwitz, calculados a partir da matriz H (Eq. (7.47)), e definidos como:

$$\Delta_1 = -Tr(J(k)) \quad (7.48)$$

$$\Delta_2 = -Tr(J(k)) \sum_{i=j=1}^3 |M(J(k))_{i,j}| + Det(J(k)) \quad (7.49)$$

$$\Delta_3 = -\Delta_2 Det(J(k)) \quad (7.50)$$

e os termos do polinômio característico, satisfazerem as seguintes condições:

$$-Tr(J(k)) > 0 \quad \sum_{i=j=1}^3 |M(J(k))_{i,j}| > 0 \quad -Det(J(k)) > 0$$

$$\Delta_1 > 0$$

$$\Delta_2 > 0$$

$$\Delta_3 > 0$$

Caso estas condições não sejam satisfeitas, as soluções são instáveis. Para o estudo das soluções temporais, toma-se $k = 0$, e para o estudo das soluções espaciais, $k \neq 0$.

7.4 – Determinação de σ

Processo Irreversível	Força Termodinâmica (F_k)	Fluxo Termodinâmico (J_k)
Condução de Calor	$\nabla(1/T)$	$J_H = -\kappa\nabla T$
Difusão de Massa	$\nabla(\mu/T)$	$J_M = -D\nabla n$
Reações Químicas*	Afinidade, A	Velocidade da Reação

TABELA 7.1 – Forças e fluxos termodinâmicos associados aos processos irreversíveis. Os termos J_H e J_M são provenientes das leis de Fourier e Fick, respectivamente.* Os termos associados às reações químicas estão bem definidos na subseção 7.4.1.

Nesta seção as expressões que descrevem explicitamente a taxa de produção de entropia relativas às reações químicas, difusão de massa/energia térmica e condução de energia térmica são apresentadas.

Como foi exposto, a entropia produzida localmente em processos irreversíveis é dada de forma geral através de forças e fluxos termodinâmicos:

$$\sigma = \sum_k F_k J_k \quad (7.51)$$

onde k , representa todas as transformações irreversíveis que contribuem para a variação de $d_i S$. Assim sendo, em um sistema em que há reações químicas, difusão de massa e difusão/condução de energia térmica, a Eq. (7.51) pode ser escrita como:

$$\sigma = \sigma_{RQ} + \sigma_H + \sigma_M \quad (7.52)$$

Com, $\sigma = \frac{1}{V} \frac{d_i S}{dt}$ e :

- $\sigma_{RQ} = \sum_k F_{RQ_k} J_{RQ_k}$ - entropia produzida por reações químicas;
- $\sigma_H = F_H J_H$ - entropia produzida pelo processo de condução de energia térmica;
- $\sigma_M = \sum_k F_{M_k} J_{M_k}$ - entropia produzida pelo processo de difusão de massa;

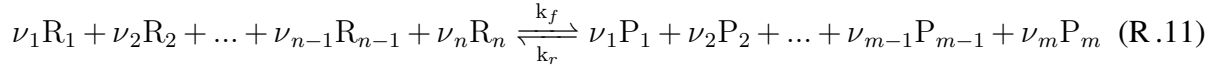
A taxa de produção de entropia total (σ_T) é então calculada da seguinte forma¹⁰ :

$$\sigma_T = \int_V \sigma dV \quad (7.53)$$

Nas subseções seguintes cada componente de taxa de produção de entropia é determinado.

7.4.1 – Reações químicas (σ_{RQ})

Considere a seguinte reação química:



Como está apresentado na Tabela 7.1, a força termodinâmica que induz reações químicas é chamada de Afinidade (A) e é definida da seguinte forma:

$$A = \sum_{k=1}^n \mu_{Rk} \nu_{Rk} - \sum_{q=1}^m \mu_{Pq} \nu_{Pq} \quad (7.54)$$

Onde μ_{Rk} e μ_{Pq} são os potenciais químicos dos reagentes e produtos, respectivamente. O fluxo termodinâmico é a velocidade das reações, como esta apresentado na TABELA 7.1. A partir disto, a taxa de produção de entropia relacionada à reação (R. 11) é

$$\sigma_{RQ} = \frac{A}{T} v = \frac{A}{T} \frac{d\xi}{dt} \quad (7.55)$$

Na equação acima ξ é o avanço da reação.ⁿ Tomando então, $\mu_k(p, T) = \mu_{k0}(T) + RT \ln(a_k)$, com R a constante universal dos gases, T a temperatura e a_k a atividade da espécie k , é possível reescrever a afinidade como:

$$A = \sum_{k=1}^n (\mu_{Rk0} + RT \ln(a_{Rk})) - \sum_{q=1}^m (\mu_{Pq0} + RT \ln(a_{Pq})) \quad (7.56)$$

e mais,

$$A = RT \ln K(T) + RT \ln \frac{\prod_{k=1}^n a_{Rk}}{\prod_{q=1}^m a_{Pq}} \quad (7.57)$$

De modo que, $K(T)$ representa a constante de equilíbrio da reação (R.11). No equilíbrio $K(T) = k_f/k_r$, k_f e k_r são as constantes de velocidade das reações direta e reversa. Logo a Eq. (7.57) torna-se:

$$A = RT \ln \frac{\prod_{k=1}^n k_f a_{Rk}}{\prod_{q=1}^m k_r a_{Pq}} \quad (7.58)$$

e portanto,

$$A = RT \ln \frac{R_f}{R_r} \quad (7.59)$$

com, $R_f = \prod_{k=1}^n k_f a_{Rk}$ e $R_r = \prod_{q=1}^m k_r a_{Pq}$, as componentes de velocidade da reação direta e reversa, respectivamente. Considerando que, $\frac{1}{V} \frac{d\xi}{dt} = R_f - R_r$, a Eq. (7.55) pode

ⁿOnde $\xi = \frac{n_x - n_x^0}{\nu_x}$, com n_x^0 quantidade de matéria inicial de x e ν_x coeficiente estequiométrico de x .

ser reescrita como:

$$\sigma_{RQ} = R \sum_j (R_{jf} - R_{jr}) \ln(R_{jf}/R_{jr}) \quad (7.60)$$

Em que o subíndice j representa o número de reações químicas.

Para mais informações veja a referência¹⁰.

7.4.2 – Condução de energia térmica (σ_H)

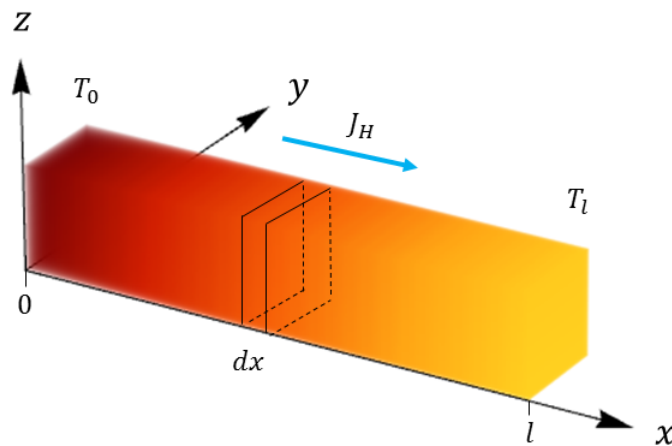


FIGURA 7.1 – Material condutor em contato com reservatórios térmicos nas extremidades, de modo que $T_0 > T_l$. Esta diferença de temperatura induz à um fluxo irreversível de energia térmica (J_H), provocando mudanças de entropia.

Considere a situação de não equilíbrio em que um material condutor está em contato com reservatórios com diferentes temperaturas em cada uma de suas extremidades, como apresentado pela FIGURA 7.1. Considere também que esta situação provoca um processo de condução em estado estacionário, o que implica fluxo J_H constante, ° e que não há trocas de energia térmica com as vizinhas, ou seja, o material se encontra isolado do meio externo. A partir das informações presentes na TABELA 7.1, a taxa de produção de entropia em um ponto x do material condutor é expressa da seguinte forma:

$$\sigma_H(x)dx = J_H \left(\frac{\partial}{\partial x} \frac{1}{T} \right) dx \quad (7.61)$$

Para obter a taxa de produção de entropia relativa ao fluxo de energia térmica entre as extremidades do condutor basta integrar a Eq. (7.61) sobre o comprimento (l) do

°A diferença de temperatura entre os reservatórios é sempre constante.

material:

$$\int_0^l \sigma_H(x) dx = \int_0^l J_H \left(\frac{\partial}{\partial x} \frac{1}{T} \right) dx = J_H \left(\frac{1}{T_l} - \frac{1}{T_0} \right) \quad (7.62)$$

A partir disto e de que $J_H = -\kappa \frac{dT}{dx} \approx -\kappa \frac{T_l - T_0}{\Delta x} \approx \alpha(T_0 - T_l)$, a Eq. (7.61) pode ser reescrita como:

$$\sigma_H = \frac{\alpha(T_0 - T_l)^2}{T_0 T_l} \quad (7.63)$$

Para os fins deste trabalho:

$$\sigma_H = \frac{\alpha(T - T_\Omega)^2}{T_\Omega T} \quad (7.64)$$

Para mais informações veja a referência.⁷¹

7.4.3 – Difusão de massa (σ_M)

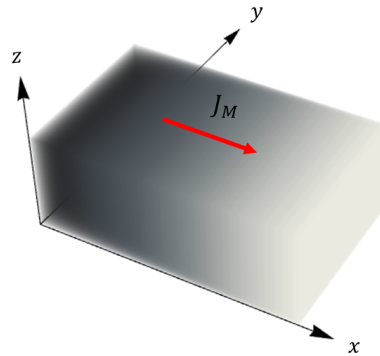


FIGURA 7.2 – Meio reacional com regiões de diferente concentração de reagentes. A diferença espacial de potencial químico induz um fluxo irreversível de massa (J_M), provocando alterações na entropia.

Como está apresentado na TABELA 7.1 a força termodinâmica que induz ao fluxo de massa é a variação de potencial químico ao longo de uma direção. Considere então por simplicidade um fluxo na direção x em uma situação isotérmica, veja a FIGURA 7.2. A taxa de produção de entropia relacionada ao processo de difusão de massa é dada por:

$$\sigma_M(x) = \sum_k \frac{1}{T} \left(\frac{\partial \mu_k}{\partial x} \right) J_M \quad (7.65)$$

Onde k representa o número de diferentes componentes que difundem no meio reacional, neste caso $k = 1, 2$. Supondo que a concentração $n_k(x)$ dependa da coordenada espacial, então:

$$\frac{\partial \mu_k}{\partial x} = \frac{-RT}{n_k} \frac{\partial n_k}{\partial x} \quad (7.66)$$

Logo, a Eq. (7.65) pode ser reescrita como:

$$\sigma_M = R \left(\frac{D_1}{n_1} \left(\frac{\partial n_1}{\partial x} \right)^2 + \frac{D_2}{n_2} \left(\frac{\partial n_2}{\partial x} \right)^2 \right) \quad (7.67)$$

Para fins deste trabalho podemos escrever:

$$\sigma_M = R \left(\frac{D_u}{n_u} \left(\frac{\partial n_u}{\partial x} \right)^2 + \frac{D_v}{n_v} \left(\frac{\partial n_v}{\partial x} \right)^2 \right) \quad (7.68)$$

Note que θ também difunde no sistema e apesar de não se tratar de "massa" as equações que descrevem esses processos são muito similares, de modo que é necessário contabilizar um termo de produção de entropia devido à difusão de energia térmica, $\frac{D_\theta}{\theta} \left(\frac{\partial \theta}{\partial x} \right)^2$.

Vale ressaltar que estes termos são definidos numericamente, de modo que o resultado final das taxas de produção de entropia são obtidos através de uma normalização pelo tamanho da malha de integração.

Para mais informações veja a referência.¹⁰

7.5 – Código computacional em FORTRAN 90

master.f90

```

program EDP
! *****
!           MÉTODO DAS DIFERENÇAS FINITAS NA SUA
!           FORMA IMPLÍCITA (PEACEMAN-RACHFORD)
!
! Programa capaz de resolver sistemas de equações diferenciais
! parciais não lineares sem dependência explícita das variáveis
! independentes, tempo e espaço.
! Estas simulações podem ser realizadas em uma ou duas dimensões
! espaciais.
! *****
! Declarando os módulos que serão utilizados. Estes módulos
! estão associados com a construção da matriz de coeficientes,
! os passos de integração para a EDP linear (PEACMAN-RACHFORD)
! e o método de Runge-Kutta de integração para a parte não
! linear (no tempo).

use matriz
use primeiro_passo
use segundo_passo
use terceiro_passo
use runge_kutta
use mpi
implicit none

!#####
!#####

integer :: nv_independente,nv_dependente,nv_espaco,chan
integer :: error,p,nk,i,j,m,k1,l,n1,n2,unidade,g,l1,NXYZ
integer, allocatable, dimension(:) :: N
real :: x_in,y_in,k,x_fin,y_fin,t_fin,xepsilon,a1,a2,a3,a4,a5,t, &
      & t1,t2,func1,func2,delta
real, allocatable, dimension(:) :: q,h,D,f,condin,er,z
real, allocatable, dimension(:,:) :: EPS,phi,xk,tetafr
real, allocatable, dimension(:, :, :) :: u,u0,uxi,uxf,uyl,uyf,w,A, &
      & v,u1,var1,var2

```

```

character(25):: name1,name2
character (len=200) :: line
character(:),allocatable :: dimensao,name0
character(len=1), allocatable, dimension(:) :: cond_frenteira_R

Integer :: STATUS(MPI_STATUS_SIZE),MsgType, Erro, Nproc,MyID

real :: start,finish
call cpu_time(start)

!#####
!           PRIMEIRO BLOCO - DEFINIDO PARÂMETROS
!#####
!*****
!***** PROGRAMA EM 1D OU 2D? *****
!*****

Call MPI_INIT(Erro)
Call MPI_COMM_SIZE(MPI_COMM_WORLD,Nproc, Erro)
Call MPI_COMM_RANK(MPI_COMM_WORLD, MyID,Erro)
!-----

chan=30
open(unit=chan,file='inputfile.dat')

NXYZ=0
! definindo o tamanho do character 'dimensao'
allocate(character(2) :: dimensao)

call pular_comentarios(chan,line)
read(line,'(A)',iostat=error) dimensao

! nv_dependente = número de variáveis independentes
if(dimensao.eq.'1D')nv_espaco=1
if(dimensao.eq.'2D')nv_espaco=2

! variáveis independentes espaciais são definidas como 'q'

```

```

allocate(q(nv_espaco),stat=error)
if(error.ne.0)then
print*, 'Espaco insuficiente 1'
endif

! desalocando o vetor dimensao
deallocate(dimensao)

!*****
!***** LIMITES E PASSOS DE INTEGRAÇÃO *****
!*****
! numero de variáveis independentes
nv_independente=3

! Determinando vetores 'N' que representam o numero de passos
! numéricos
allocate(N(nv_independente),h(nv_espaco),stat=error)
if(error.ne.0)then
print*, 'Espaco insuficiente 2'
endif

! Se o número de variáveis independentes e 3.
! N(1) e N(3) estão associadas ao espaço
! N(2) esta associada ao tempo
!-----
! Para a variável x

! Número de passos para x
call pular_comentarios(chan,line)
read(line,*,iostat=error) N(1)

! limite inicial
call pular_comentarios(chan,line)
read(line,*,iostat=error) x_in

! limite final
call pular_comentarios(chan,line)
read(line,*,iostat=error) x_fin

! tamanho do passo

```



```

      h(1)=(x_fin-x_in)/N(1)

!-----
! Para a variável y

! Número de passos para y
  call pular_comentarios(chan,line)
  read(line,*,iostat=error) N(3)

! limite inicial
  call pular_comentarios(chan,line)
  read(line,*,iostat=error) y_in

! limite final
  call pular_comentarios(chan,line)
  read(line,*,iostat=error) y_fin
! tamanho do passo
  if(nv_espaco.eq.1)N(3)=2
  h(2)=(y_fin-y_in)/N(3)
!-----
! Para a variável t

! Número de passos para t
  call pular_comentarios(chan,line)
  read(line,*,iostat=error) N(2)

! limite final
  call pular_comentarios(chan,line)
  read(line,*,iostat=error) t_fin

! tamanho do passo
  k=t_fin/N(2)
!-----

!*****
!***** VARIÁVEIS DEPENDENTES *****
!*****
! declarando o número de variáveis dependentes
  call pular_comentarios(chan,line)

```

```

read(line,*,iostat=error) nv_dependente

! definindo o número de constantes de difusão, e constante phi
allocate(D(nv_dependente),phi(nv_dependente,nv_espaco),stat=error)
if(error.ne.0)then
print*, 'Espaco insuficiente 3'
endif

do 1 p=1,nv_dependente

call pular_comentarios(chan,line)
read(line,*,iostat=error) D(p)

! definindo os termos phi
do 11 g=1,nv_espaco

phi(p,g)=D(p)*k/(2*h(g)*h(g))

11 continue
1 continue

!*****
!***** GAUSS-SEIDEL *****
!*****
! numero de interações do método de Gauss-Jacobi
nk=50000

! precisão das aproximações do método de Gauss-Seidel
xepsilon=1.0D-40

!*****
!***** TIPO DE FRONTEIRA *****
!*****
! Definindo o vetor u0 de condições de fronteira
allocate( cond_frenteira_R(nv_dependente))
! Condições de Fronteira
! Drechilet - D
! Periodic - P
! Neumann - N

```

```

do p=1,nv_dependente

    call pular_comentarios(chan,line)
    read(line,'(A)',iostat=error) cond_frenteira_R(p)
    !print*,cond_frenteira_R(p)
enddo

!*****
!  cria-se uma pasta chamada 'Resultados' onde serão alocados
!  as condições iniciais, erros e ordem de convergência
  if(MyID.eq.0)then

    allocate(character(10)::name0)
    write(name0,'(A)')'Resultados'
    call system ('mkdir '//name0)

    write(name1,'(A)')'Condicoes Iniciais.dat'
    open(unit=10,file=name0//' '//name1)

        write(10,99)'*****'
        write(10,99)'          CONDICOES INICIAIS          '
        write(10,99)'*****'
        write(10,105)'          DIMENSAO ESPACIAL   ',nv_espaco,'D'
        write(10,99)'*****'

do 2 p=1,nv_dependente
    write(10,99)'          '
    write(10,104)'          VARIAVEL DEPENDENTE          ',p
    write(10,99)'          '
    write(10,99)'-----'
do 222 g=1,nv_espaco

    if(g.eq.1)write(10,100)'Constante de difusao = ',D(p)
    if(g.eq.1)write(10,99)'-----'

    if(g.eq.1)write(10,100)'Passo tempo = ',k
    if(g.eq.1)write(10,99)'-----'

    if(g.eq.1)write(10,100)' Passo x = ',h(g)

```

```

        if((nv_espaco.eq.2).and.(g.eq.nv_espaco)) &
        & write(10,100)' Passo y = ',h(g)
        write(10,99) '-----'
        if(g.eq.1)write(10,100)'Constante phix = ',phi(p,g)
        if((nv_espaco.eq.2).and.(g.eq.nv_espaco)) &
        &write(10,100)'Constante phiy = ',phi(p,g)
        write(10,99) '-----'
222    continue
2     continue

        write(10,99)'
        write(10,99)'*****'
        write(10,102)'xf =',x_fin,' | nx = ',N(1)
        write(10,99) '-----'
        write(10,102)' yf =',y_fin,' | ny = ',N(3)
        write(10,99) '-----'
        if(nv_espaco.eq.2) write(10,102)' tf =',t_fin,' &
        & | t_fin = ',N(2)

        write(10,99)'
        write(10,99)'*****'
        write(10,101)'Interacoes Gauss-Seidel = ',nk,' |
        & Precisaao = ',xepsilon
        write(10,99) '-----'
do p=1,nv_dependente
    write(10,103)'Condicoes de Fronteira das Equacoes &
    & dos Reagentes= ',cond_frenteira_R(p)
    write(10,99) '-----'
enddo
        write(10,99) '-----'

! fechando unidades de armazenamento
close(unit=10)

99    FORMAT (A)
100   FORMAT (A,F24.10)
101   FORMAT (A,I6,A,E24.10)
102   FORMAT (A,F24.10,A,I6)
103   FORMAT (A,A)
104   FORMAT (A,I6)

```

```
105   FORMAT (A, I1, A)
```

```
endif
```

```
!*****
```

```
!#####
```

```
!           FIM DO PRIMEIRO BLOCO
```

```
!#####
```

```
!#####
```

```
! SEGUNDO BLOCO - DEFINIDO CONDIÇÕES INICIAS E FRONTEIRA
```

```
!#####
```

```
!*****
```

```
!***** CONDIÇÕES INICIAIS E FRONTEIRAS *****
```

```
!*****
```

```
! Alocando vetores para diferentes dimensões abordadas
```

```
if (nv_espaco.eq.1) then
```

```
allocate (u (nv_dependente, N(1)+1, N(2)+1), stat=error)
```

```
if (error.ne.0) then
```

```
print*, 'Espaco insuficiente 4'
```

```
endif
```

```
endif
```

```
!-----
```

```
! Definindo o vetor u0 de condições iniciais
```

```
if (nv_espaco.eq.2) then
```

```
allocate (u0 (nv_dependente, N(1)+1, N(3)+1), uxi (nv_dependente, &
```

```
& N(3)+1, N(2)+1), uxf (nv_dependente, N(3)+1, N(2)+1) &
```

```
&, uyi (nv_dependente, N(1)+1, N(2)+1), uyf (nv_dependente, N(1)+1, &
```

```
& N(2)+1), u (nv_dependente, N(1)+1, N(3)+1), stat=error)
```

```
if (error.ne.0) then
```

```

    print*, 'Espaco insuficiente 5'
    endif

endif

    allocate (condin(nv_dependente), tetafr(nv_dependente,2), stat=error)
    if(error.ne.0) then
    print*, 'Espaco insuficiente 7'
    endif

!-----
! definição das condições iniciais e fronteira feita
! pelo processador "0"
if(MyID.eq.0) then

do 9 p=1,nv_dependente

! condições iniciais (1) e condições de Fronteira (2)

! (1) condições iniciais:
    call pular_comentarios(chan,line)
    read(line,*,iostat=error) condin(p)

!-----
! (2) condições de fronteira:

    do l1=1,2

        call pular_comentarios(chan,line)
        read(line,*,iostat=error) tetafr(p,l1)

    enddo

!-----

```

```

do 4 j=1,N(2)+1

  do 5 i=1,N(1)+1

    q(1)=x_in+(i-1)*h(1)
    t=(j)*k

!*****
!  PARA O CASO EM 1D
!*****
if(nv_espaco.eq.1)then

  if(j.eq.1)u(p,i,1)=condin(p)+rand(0)*0.15

!-----
!

  if(cond_frenteira_R(p).eq.'D')then

    if((j.gt.1).and.(i.eq.1))u(p,1,j)=tetafr(p,1)
    if((j.gt.1).and.(i.eq.N(1)+1))u(p,N(1)+1,j)=tetafr(p,2)

  elseif(cond_frenteira_R(p).eq.'N')then

    if((j.gt.1).and.(i.eq.1))u(p,1,j)=tetafr(p,1)
    if((j.gt.1).and.(i.eq.N(1)+1))u(p,N(1)+1,j)=tetafr(p,2)

  endif

endif

endif
!*****
!  PARA O CASO EM 2D
!*****
if(nv_espaco.eq.2)then

  do 6 m=1,N(3)+1

```

```

q(2)=y_in+(m-1)*h(2)

if(j.eq.1)u0(p,i,m)=condin(p)+rand(0)*0.15

!-----
!

if(cond_frenteira_R(p).eq.'D')then

! (2) condições de fronteira:

delta=ABS(tetafr(p,1)-tetafr(p,2))/(N(1)+1)

if(i.eq.1)uxi(p,m,j)=tetafr(p,1)
if(i.eq.N(1)+1)uxf(p,m,j)=tetafr(p,2)
if(m.eq.1)uyi(p,i,j)=tetafr(p,1)+delta*i
if(m.eq.N(3)+1)uyf(p,i,j)=tetafr(p,1)+delta*i

elseif(cond_frenteira_R(p).eq.'N')then

if(i.eq.1)uxi(p,m,j)=tetafr(p,1)
if(i.eq.N(1)+1)uxf(p,m,j)=tetafr(p,1)
if(m.eq.1)uyi(p,i,j)=tetafr(p,1)
if(m.eq.N(3)+1)uyf(p,i,j)=tetafr(p,1)

endif

6   continue

endif

5   continue
4   continue
9   continue

!*****
! Informações compartilhadas com os demais processadores

```



```

do p=1,nv_dependente

!-----
  MsgType=100+p

! Para situações em 1D
  if(nv_espaco.eq.1)then

    Call MPI_SEND(u,nv_dependente*(N(1)+1)*(N(2)+1),MPI_REAL, &
      & p,MsgType,MPI_COMM_WORLD,Erro)
  else
! Para situações em 2D

    Call MPI_SEND(u0,nv_dependente*(N(1)+1)*(N(3)+1),MPI_REAL, &
      & p,MsgType,MPI_COMM_WORLD,Erro)

    Call MPI_SEND(uxi,nv_dependente*(N(2)+1)*(N(3)+1),MPI_REAL, &
      & p,MsgType,MPI_COMM_WORLD,Erro)
    Call MPI_SEND(uxf,nv_dependente*(N(2)+1)*(N(3)+1),MPI_REAL, &
      & p,MsgType,MPI_COMM_WORLD,Erro)
    Call MPI_SEND(uyi,nv_dependente*(N(2)+1)*(N(1)+1),MPI_REAL, &
      & p,MsgType,MPI_COMM_WORLD,Erro)
    Call MPI_SEND(uyf,nv_dependente*(N(2)+1)*(N(1)+1),MPI_REAL, &
      & p,MsgType,MPI_COMM_WORLD,Erro)

  endif

  enddo

!-----
else
!-----
  MsgType=MyID+100

! Para situações em 1D
  if(nv_espaco.eq.1)then

    Call MPI_RECV(u,nv_dependente*(N(1)+1)*(N(2)+1),MPI_REAL &
      & ,0,MsgType,MPI_COMM_WORLD,STATUS,Erro)

```

```

else
! Para situações em 2D
  Call MPI_RECV(u0,nv_dependente*(N(1)+1)*(N(3)+1),MPI_REAL &
& ,0,MsgType,MPI_COMM_WORLD,STATUS,Erro)

  Call MPI_RECV(uxi,nv_dependente*(N(2)+1)*(N(3)+1),MPI_REAL &
& ,0,MsgType,MPI_COMM_WORLD,STATUS,Erro)
  Call MPI_RECV(uxf,nv_dependente*(N(2)+1)*(N(3)+1),MPI_REAL &
& ,0,MsgType,MPI_COMM_WORLD,STATUS,Erro)
  Call MPI_RECV(uyi,nv_dependente*(N(2)+1)*(N(1)+1),MPI_REAL &
& ,0,MsgType,MPI_COMM_WORLD,STATUS,Erro)
  Call MPI_RECV(uyf,nv_dependente*(N(2)+1)*(N(1)+1),MPI_REAL &
& ,0,MsgType,MPI_COMM_WORLD,STATUS,Erro)

endif
endif
!*****

!#####
!
!          FIM DO SEGUNDO BLOCO
!#####

!#####
!
!          TERCEIRO BLOCO - DIFERENÇAS FINITAS
!#####

!*****
!***** INICIO DO MÉTODO DAS DIFERENÇAS FINITAS *****
!*****
! Alocando espaço para tensores utilizados
! Para o caso em 1D
if(nv_espaco.eq.1)then
  allocate(w(nv_dependente,N(1)+1,N(2)+1),v(nv_dependente,N(1)+1 &
& ,nk),er(nv_dependente),stat=error)
  if(error.ne.0)then
    print*, 'Espaco insuficiente 8'
  endif
endif
endif

```

```

if(nv_espaco.eq.1)then
  allocate(var1(nv_dependente,N(1)+1,N(2)+1),var2(nv_dependente, &
& N(1)+1,N(2)+1),stat=error)
  if(error.ne.0)then
    print*, 'Espaco insuficiente paralelo'
  endif
endif

!-----
! Para o caso em 2D
  if(nv_espaco.eq.2)then
    allocate(w(nv_dependente,N(1)+1,N(3)+1),v(nv_dependente,N(1)+1 &
& ,nk),er(nv_dependente),stat=error)
    if(error.ne.0)then
      print*, 'Espaco insuficiente 9'
    endif
  endif
endif
!

  if(nv_espaco.eq.2)then
    allocate(var1(nv_dependente,N(1)+1,N(3)+1),var2(nv_dependente &
&,N(1)+1,N(3)+1),stat=error)
    if(error.ne.0)then
      print*, 'Espaco insuficiente paralelo'
    endif
  endif
endif

!-----
! Início dos passos temporais
do 10 j=1,N(2)

! PARALELIZAÇÃO DO LOOP CENTRAL
  if((MyID.eq.1).or.(MyID.eq.2))then

    p=MyID

! Para o caso em 1D ou 2D

```

```

allocate (A(nv_dependente,N(1)+1,N(1)+1),EPS(nv_dependente,N(1)+1) &
&,stat=error)
if(error.ne.0)then
print*, 'Espaco insuficiente 10'
endif

!-----
do g=1,nv_espaco

! Formação da matriz associada a parte implícita de x
if(((nv_espaco.eq.1).or.(nv_espaco.eq.2)).and.(g.eq.1)).and.&
& (cond_frenteira_R(p).eq.'D')) &
& call matriz_imp_x_dirichlet(N,p,g,phi,A)
if(((nv_espaco.eq.1).or.(nv_espaco.eq.2)).and.(g.eq.1)).and. &
& (cond_frenteira_R(p).eq.'N')) &
& call matriz_imp_x_neumann(N,p,g,phi,A)
if(((nv_espaco.eq.1).or.(nv_espaco.eq.2)).and.(g.eq.1)).and.&
& (cond_frenteira_R(p).eq.'P')) &
& call matriz_imp_x_periodico(N,p,g,phi,A)

! Formação do vetor w para o caso em 1D

if((nv_espaco.eq.1).and.(cond_frenteira_R(p).eq.'D')) call &
& primeiropasso1D_dirichlet(N,j,p,g,phi,A,u,w)
if((nv_espaco.eq.1).and.(cond_frenteira_R(p).eq.'N')) call &
& primeiropasso1D_neumann(N,j,p,g,phi,h,A,u,w)
if((nv_espaco.eq.1).and.(cond_frenteira_R(p).eq.'P')) call &
& primeiropasso1D_periodico(N,j,p,g,phi,A,u,w)

! Formação do vetor w para o caso em 2D

if(((nv_espaco.eq.2).and.(g.eq.nv_espaco)).and.&
& (cond_frenteira_R(p).eq.'D')) &
& call primeiropasso2D_dirichlet(N,j,p,g,phi,A,u,uyi,uyf,u0,w)
if(((nv_espaco.eq.2).and.(g.eq.nv_espaco)).and.&
& (cond_frenteira_R(p).eq.'N')) &
& call primeiropasso2D_neumann(N,j,p,g,phi,h,A,u,uyi,uyf,u0,w)
if(((nv_espaco.eq.2).and.(g.eq.nv_espaco)).and. &
& (cond_frenteira_R(p).eq.'P')) &
& call primeiropasso2D_periodico(N,j,p,g,phi,A,u,u0,w)

```

```

        enddo

!-----
! início dos passos espaciais e aqueles associados ao método de
! Gauss-Seidel
do 20 m=2,N(3)
    do 30 k1=1,nk
        do 40 i=2,N(1)

! cálculo dos valores explícitos das coordenadas independentes
        if(nv_espaco.eq.2) q(2)=y_in+(m-1)*h(2)
        q(1)=x_in+(i-1)*h(1)
        t1=(j)*k
        t2=(j+1)*k

!-----
! funções que tem dependência de coordenadas espaciais e temporais
! em casos de EDPs com dependência
! explícita das variáveis independentes, ou seja, tempo e espaço.
        func1=0
        func2=0

!-----
        do g=1,nv_espaco

! Segundo passo do método de Crank-Nicholson

        if((nv_espaco.eq.1).and.(cond_frenteira_R(p).eq.'D')) call &
        & segundopassolD_dirichlet(i,p,g,N,k1,k,j,phi, &
        & func1,func2,A,w,u,v)
        if((nv_espaco.eq.1).and.(cond_frenteira_R(p).eq.'N')) call &
        & segundopassolD_neumann(i,p,g,N,k1,k,j,phi,h,func1,func2 &
        & ,A,w,u,v)
        if((nv_espaco.eq.1).and.(cond_frenteira_R(p).eq.'P')) call &
        & segundopassolD_periodico(i,p,g,N,k1,k,j,phi,func1,func2, &
        & A,w,u,v)

```

! Segundo passo do método de Peaceman-Rachford

```

    if((nv_espaco.eq.2).and.(g.eq.nv_espaco)).and.&
    &(cond_frenteira_R(p).eq.'D')) &
    & call segundopasso2D_dirichlet(N,i,m,p,g,k1,j,k,phi,&
    &func1,func2,A,uxi,uxf,w,v)
    if((nv_espaco.eq.2).and.(g.eq.nv_espaco)).and.&
    &(cond_frenteira_R(p).eq.'N')) call &
    & segundopasso2D_neumann(N,i,m,p,g,k1,j,k,phi,h,func1,func2 &
    &,A,uxi,uxf,w,v)
    if((nv_espaco.eq.2).and.(g.eq.nv_espaco)).and.&
    &(cond_frenteira_R(p).eq.'P')) &
    & call segundopasso2D_periodico(N,i,m,p,g,k1,j,k,phi,func1, &
    &func2,A,w,u,v)

```

enddo

!-----

40 continue

!! teste de convergência e acurácia

```
do 123 l=2,N(1)
```

```
    EPS(p,l)=ABS(v(p,l,k1+1)-v(p,l,k1))
```

```
    if(l.eq.2) then
```

```
        er(p)=EPS(p,l)
```

```
    endif
```

```
    if((l.gt.2).and.(EPS(p,l).gt.er(p))) er(p)=EPS(p,l)
```

123 continue

!-----

! Teste feito para analisar se a aproximação convergiu ou não.

```
    if(k1.eq.nk) goto 201
```

!-----

```

! se o erro da ultima variável dependente for menor que
! xepsilon então o resultado será contado e escrito
      if(er(p).lt.xepsilon) goto 401

!-----
30  continue

401  do 22 n1=2,N(1)
      ! if(j.eq.1)print*,v(1,n1,k1+1)

      if(nv_espaco.eq.1)var1(p,n1,j+1)=v(p,n1,k1+1)

!     Alocação dos primeiros resultados nos vetores específicos
      if(nv_espaco.eq.2)u(p,n1,m)=v(p,n1,k1+1)

22  continue

20  if(nv_espaco.eq.2)continue

      deallocate(A,EPS)

      if(nv_espaco.eq.1) goto 310

!-----
! TERCEIRA PARTE DO MÉTODO DE PEACEMAN_RACHFORD - Apenas para casos
! em 2D

      allocate(A(nv_dependente,N(3)+1,N(3)+1),EPS(nv_dependente, &
& N(3)+1),stat=error)
      if(error.ne.0)then
        print*, 'Espaco insuficiente 11'
      endif

!-----

      do g=1,nv_espaco

```

! Formação das matrizes do terceiro passo implícito

```

    if((g.eq.nv_espaco).and.(cond_frenteira_R(p).eq.'D')) call &
    & matriz_imp_y_dirichlet(N,p,g,phi,A)
    if((g.eq.nv_espaco).and.(cond_frenteira_R(p).eq.'N')) call &
    & matriz_imp_y_neumann(N,p,g,phi,A)
    if((g.eq.nv_espaco).and.(cond_frenteira_R(p).eq.'P')) call &
    & matriz_imp_y_periodico(N,p,g,phi,A)

```

enddo

!-----

```

do 50 i=2,N(1)
  do 60 k1=1,nk
    do 70 m=2,N(3)

```

!-----

! Determinação dos valores de $u(i,m,j+1)$

```

do g=1,nv_espaco

```

! Rotinas para os calculos

```

    if((g.eq.nv_espaco).and.(cond_frenteira_R(p).eq.'D')) call &
    & terceiropasso2D_dirichlet(N,p,g,i,m,k1,j,phi,A,uyi,uyf,w,u,v)
    if((g.eq.nv_espaco).and.(cond_frenteira_R(p).eq.'N')) call &
    & terceiropasso2D_neumann(N,p,g,i,m,k1,j,phi,h,A,uyi,uyf,w,u,v)
    if((g.eq.nv_espaco).and.(cond_frenteira_R(p).eq.'P')) call &
    & terceiropasso2D_periodico(N,p,g,i,m,k1,j,phi,A,w,u,v)

```

enddo

!-----

70 continue

! teste de convergência e acurácia

```

do l=2,N(3)

```



```

EPS(p,l)=ABS(v(p,l,k1+1)-v(p,l,k1))

    if(l.eq.2) then
    er(p)=EPS(p,l)

    endif
    if((l.gt.2).and.(EPS(p,l).gt.er(p))) er(p)=EPS(p,l)

    enddo

!  Teste feito para analisar se a aproximação convergiu ou não.
    if(k1.eq.nk) goto 205

!-----
!  se o erro da ultima variável dependente for menor que xepsilon
!  então o resultado será contado e escrito
    if(er(p).lt.xepsilon) goto 501

!-----

60  continue

!-----

501  do 52 n2=2,N(3)

!  Alocação dos primeiros resultados nos vetores específicos
    var1(p,i,n2)=v(p,n2,k1+1)

52  continue

50  continue

!-----
    deallocate(A,EPS)

```

```
310 endif
```

```
! compartilhamento das informações entre os processadores
```

```
If(MyID.NE.0) then
```

```
MsgType=100+MyID
```

```
if(nv_espaco.eq.1) Call MPI_SEND(var1,nv_dependente*(N(1)+1) &
& *(N(2)+1),MPI_REAL,0,MsgType,MPI_COMM_WORLD,Erro)
```

```
if(nv_espaco.eq.2) Call MPI_SEND(var1,nv_dependente*(N(1)+1) &
& *(N(3)),MPI_REAL,0,MsgType,MPI_COMM_WORLD,Erro)
```

```
Else
```

```
Do p=1,nv_dependente
```

```
MsgType=100+p
```

```
if(nv_espaco.eq.1) Call MPI_RECV(var2,nv_dependente*(N(1)+1) &
& *(N(2)+1),MPI_REAL,MyID+p,MsgType,MPI_COMM_WORLD,STATUS,Erro)
```

```
if(nv_espaco.eq.2) Call MPI_RECV(var2,nv_dependente*(N(1)+1) &
& *(N(3)),MPI_REAL,MyID+p,MsgType,MPI_COMM_WORLD,STATUS,Erro)
```

```
if(nv_espaco.eq.1) then
```

```
do 113 i=2,N(1)
```

```
do 117 m=2,j
```

```
u(p,i,m)=var2(p,i,m)
```

```
117 continue
```

```
113 continue
```

```
else
```

```

do 13 i=2,N(1)
  do 17 m=2,N(3)

    u(p,i,m)=var2(p,i,m)

17  continue
13  continue

endif

enddo

Endif

! FIM DA TERCEIRA PARTE DO MÉTODO DE PEACEMAN-RACHFOD

!#####
!
!          FIM DO TERCEIRO BLOCO
!#####

!#####
!
!          QUARTO BLOCO - MÉTODO DE RUNGE-KUTTA
!#####

!*****
!***** MÉTODO DE RUNGE-KUTTA *****
!*****
!Passagem associada há parte não linear de edps que apenas
! apresentam dependência explícita da variável independente
! "tempo".

if(MyID.eq.0)Then

allocate(xk(4,nv_dependente),z(nv_dependente),f(nv_dependente) &
& ,stat=error)

```

```

    if(error.ne.0)then
      print*, 'Espaco insuficiente 12'
    endif

!-----
    if(nv_espaco.eq.1)then
      allocate(u1(nv_dependente,N(1)+1,N(2)+1),stat=error)
      if(error.ne.0)then
        print*, 'Espaco insuficiente 13'
      endif
    endif

!-----
    if(nv_espaco.eq.2)then
      allocate(u1(nv_dependente,N(1)+1,N(3)+1),stat=error)
      if(error.ne.0)then
        print*, 'Espaco insuficiente 14'
      endif
    endif

!-----
    if(j.eq.1)then
      call pular_comentarios(chan,line)
      read(line,*,iostat=error) a1

      call pular_comentarios(chan,line)
      read(line,*,iostat=error) a2
    endif

do 85 n1=2,N(1)
  do 95 n2=2,N(3)

      t1=(j+1)*k

! Sub rotina que descreve o método de RUNGE-KUTTA.
      call rk(nv_dependente,nv_espaco,n1,n2,j,a1,a2,k,t1,f,z,xk,u,u1)

```

```

do p=1,nv_dependente
  if(nv_espaco.eq.1) u(p,n1,j+1)=u1(p,n1,j+1)
  if(nv_espaco.eq.2) u(p,n1,n2)=u1(p,n1,n2)
!   if(j.eq.1) print*,u(p,n1,n2)
  enddo
95  continue
85  continue

  deallocate(u1,z,f,xk)

! Repasse de informações entre processadores
do p=1,nv_dependente

  MsgType=100+p

  if(nv_espaco.eq.1) Call MPI_SEND(u,nv_dependente*(N(1)+1)* &
& (N(2)+1),MPI_REAL,p,MsgType,MPI_COMM_WORLD,Erro)

  if(nv_espaco.eq.2) Call MPI_SEND(u,nv_dependente*(N(1)+1)* &
& (N(3)+1),MPI_REAL,p,MsgType,MPI_COMM_WORLD,Erro)

  enddo

  else

  MsgType=MyID+100

  if(nv_espaco.eq.1) Call MPI_RECV(u,nv_dependente*(N(1)+1)* &
& (N(2)+1),MPI_REAL,0,MsgType,MPI_COMM_WORLD,STATUS,Erro)

  if(nv_espaco.eq.2) Call MPI_RECV(u,nv_dependente*(N(1)+1)* &
& (N(3)+1),MPI_REAL,0,MsgType,MPI_COMM_WORLD,STATUS,Erro)

endif

```

```

#####
!
!                                     FIM DO QUARTO BLOCO
!
#####

!#####
!
!                                     QUINTO BLOCO - IMPRESSÃO DOS RESULTADOS
!
#####

!*****
!***** IMPRIMINDO RESULTADOS *****
!*****

    if (MyID.eq.0) then

        do 155 p=1,nv_dependente

            do 80 n1=2,N(1)
                do 90 n2=2,N(3)

                    if (nv_espaco.eq.2) q(2)=y_in+(n2-1)*h(2)
                    q(1)=x_in+(n1-1)*h(1)
                    t=(j+1)*k

!-----
! PARA O CASO DE SIMULAÇÕES EM 1D
    if (nv_espaco.eq.1) then

        unidade=10+p

        write(name1,'(A,I1,A)')'resultados',p,'.dat'
        open(unit=unidade,file=name0///'///name1)
        write(unidade,600)q(1),t,u(p,n1,j+1)

!-----
!-----
! PARA O CASO DE SIMULAÇÕES EM 2D
    else

```

```
if(j.eq.N(2))then

    unidade=10+p

    write(name1,'(A,I1,A)')'resultados',p,'.dat'
    open(unit=unidade,file=name0//'/'/name1)
    write(unidade,600)q(1),q(2),u(p,n1,n2)

endif

endif

!-----

600 FORMAT(3(F24.10))

90    continue
80    continue

155 continue

endif

!*****
10 continue

close(unit=unidade)

201 if(k1.eq.nk) then
    print*, 'O metodo de Gauss-Seidel nao atingiu a acuracia
    pretendida com o numero de passos numericos definidos s'
    goto 305
endif
```

```

205 if(k1.eq.nk) then
    print*, 'O metodo de Gauss-Seidel nao atingiu a acuracia
    pretendida com o numero de passos numericos definidos t'
    goto 305
endif

```

```

if(MyID.eq.0)then
    call cpu_time(finish)
    print*,finish-start
endif

```

```

Call MPI_FINALIZE(ERRO)

```

```

305 stop
    end program EDP

```

```

!*****
!#####
!
!           FIM DO QUINTO BLOCO E PROGRAMA
!#####

```

```

subroutine pular_comentarios(chan,line)
    integer, intent(in) :: chan
    integer :: ioa
    character (len=200), intent(inout) :: line

```

```

111 continue
    read (chan,*,iostat=ioa) line
    if (line(1:1).eq. '#') goto 111
    return

```

```

end subroutine pular_comentarios

```


matriz.f90

```

module matriz
implicit none
contains

```

```

!***** Dirichelt *****
!#####
!#####

```

```

subroutine matriz_imp_x_dirichlet(N,p,g,phi,A1)
integer :: i1,i2
integer, intent(in) :: p,g
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:, :), intent(in) :: phi
real, allocatable, dimension (:, :, :) :: A1

```

```

!-----

```

```

!   Determiação da matrix A para o segundo passo implícito em x

```

```

do 11 i1=2,N(1)
do 22 i2=2,N(1)

```

```

if(i2.eq.i1)then
A1(p,i1,i2)=1+2*phi(p,g)

```

```

elseif((i2.eq.i1+1).or.(i2.eq.i1-1))then
A1(p,i1,i2)=-phi(p,g)

```

```

else
A1(p,i1,i2)=0

```

```

endif

```

```

22 continue

```

```

11 continue

```

```

!-----

```

```

end subroutine matriz_imp_x_dirichlet

```

```

#####
#####
subroutine matriz_imp_y_dirichlet(N,p,g,phi,A1)
integer :: i1,i2
integer, intent(in) :: p,g
integer, allocatable, dimension(:),intent(in) :: N
real, allocatable, dimension(:,:),intent(in) :: phi
real,allocatable,dimension (:,:,) :: A1

!-----
!   Determiação da matrix A para o segundo passo implícito em x
      do 11 i1=2,N(3)
          do 22 i2=2,N(3)

              if(i2.eq.i1)then
                  A1(p,i1,i2)=1+2*phi(p,g)

              elseif((i2.eq.i1+1).or.(i2.eq.i1-1))then
                  A1(p,i1,i2)=-phi(p,g)

              else
                  A1(p,i1,i2)=0

              endif

          22 continue
      11 continue
!-----
end subroutine matriz_imp_y_dirichlet
#####
#####
!*****
!***** Periodica *****
!#####
!#####

```

```

subroutine matriz_imp_x_periodico(N,p,g,phi,A1)
integer :: i1,i2
integer, intent(in) :: p,g
integer, allocatable, dimension(:),intent(in) :: N
real, allocatable, dimension(:,,:),intent(in) :: phi
real,allocatable,dimension (:,:,) :: A1

```

```

do 11 i1=2,N(1)
  do 22 i2=2,N(1)

    if(i2.eq.i1)then
      A1(p,i1,i2)=1+2*phi(p,g)

    elseif((i1.eq.2).and.(i2.eq.N(1)))then
      A1(p,i1,i2)=-phi(p,g)

    elseif((i1.eq.N(1)).and.(i2.eq.2))then
      A1(p,i1,i2)=-phi(p,g)

    elseif((i2.eq.i1+1).or.(i2.eq.i1-1))then
      A1(p,i1,i2)=-phi(p,g)

    else
      A1(p,i1,i2)=0
    endif
  enddo
enddo

```

```

22      continue
11      continue

```

```

end subroutine matriz_imp_x_periodico

```

```

!#####

```

```

#####
subroutine matriz_imp_y_periodico(N,p,g,phi,A1)
integer :: i1,i2
integer, intent(in) :: p,g
integer, allocatable, dimension(:),intent(in) :: N
real, allocatable, dimension(:,,:),intent(in) :: phi
real,allocatable,dimension (:,:,) :: A1

do 11 i1=2,N(3)
  do 22 i2=2,N(3)

    if(i2.eq.i1)then
      A1(p,i1,i2)=1+2*phi(p,g)

    elseif((i1.eq.2).and.(i2.eq.N(3)))then
      A1(p,i1,i2)=-phi(p,g)

    elseif((i1.eq.N(3)).and.(i2.eq.2))then
      A1(p,i1,i2)=-phi(p,g)

    elseif((i2.eq.i1+1).or.(i2.eq.i1-1))then
      A1(p,i1,i2)=-phi(p,g)

    else
      A1(p,i1,i2)=0

    endif

  22 continue
11 continue

end subroutine matriz_imp_y_periodico

```

```
#####
#####
```

```
!***** neumann *****
```

```
#####
```

```
#####
```

```
subroutine matriz_imp_x_neumann(N,p,g,phi,A1)
```

```
integer :: i1,i2
```

```
integer, intent(in) :: p,g
```

```
integer, allocatable, dimension(:), intent(in) :: N
```

```
real, allocatable, dimension(:, :), intent(in) :: phi
```

```
real, allocatable, dimension (:, :, :) :: A1
```

```
!-----
```

```
! Determinação da matrix A para o segundo passo implícito em x
```

```
do 11 i1=2,N(1)
```

```
do 22 i2=2,N(1)
```

```
if(i2.eq.i1)then
```

```
A1(p,i1,i2)=1+2*phi(p,g)
```

```
elseif((i1.eq.2).and.(i2.eq.3))then
```

```
A1(p,i1,i2)=-2*phi(p,g)
```

```
elseif((i1.eq.N(1)).and.(i2.eq.N(1)-1))then
```

```
A1(p,i1,i2)=-2*phi(p,g)
```

```
elseif((i2.eq.i1+1).or.(i2.eq.i1-1))then
```

```
A1(p,i1,i2)=-phi(p,g)
```

```
else
```

```
A1(p,i1,i2)=0
```

```
endif
```

```
22 continue
```

```
11 continue
```

```

!-----
end subroutine matriz_imp_x_neumann

!#####
!#####
subroutine matriz_imp_y_neumann(N,p,g,phi,A1)
integer :: i1,i2
integer, intent(in) :: p,g
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:, :), intent(in) :: phi
real, allocatable, dimension (:, :, :) :: A1

!-----
!   Determiação da matrix A para o segundo passo implícito em x
      do 11 i1=2,N(3)
        do 22 i2=2,N(3)

          if(i2.eq.i1)then
            A1(p,i1,i2)=1+2*phi(p,g)

          elseif((i1.eq.2).and.(i2.eq.3))then
            A1(p,i1,i2)=-2*phi(p,g)

          elseif((i1.eq.N(3)).and.(i2.eq.N(3)-1))then
            A1(p,i1,i2)=-2*phi(p,g)

          elseif((i2.eq.i1+1).or.(i2.eq.i1-1))then
            A1(p,i1,i2)=-phi(p,g)

          else
            A1(p,i1,i2)=0

          endif

22          continue
11         continue

```

```
!-----  
end subroutine matriz_imp_y_neumann  
!#####  
!#####  
!*****  
  
end module matriz  
  
!#####  
!#####
```

primeiro-passo.f90

```

module primeiro_passo
implicit none
contains

!#####
subroutine primeiropasso1D_dirichlet(N,j,p,g,phi,A,u,w1)
integer :: i
integer, intent(in) :: j,p,g
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:, :), intent(in) :: phi
real, allocatable, dimension(:, :, :), intent(in) :: A,u
real, allocatable, dimension (:, :, :) :: w1

! Primeiro passo. Passo para determinar vetor w de forma explícita
do i=2,N(1)

    w1(p,i,j)=phi(p,g)*u(p,i-1,j)+(1-2*phi(p,g))*u(p,i,j) &
    & +phi(p,g)*u(p,i+1,j)

enddo

end subroutine primeiropasso1D_dirichlet

!#####
!#####
subroutine primeiropasso2D_dirichlet(N,j,p,g,phi,A,u,uyi,uyf,u0,w1)
integer :: i,m
integer, intent(in) :: j,p,g
real, allocatable, dimension(:, :), intent(in) :: phi
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:, :, :), intent(in) :: A,u,uyi,uyf,u0
real, allocatable, dimension (:, :, :) :: w1

! Primeiro passo. Passo para determinar vetor w de forma explícita
do i=2,N(1)
do m=2,N(3)

```



```

if(j.eq.1) w1(p,i,m)=phi(p,g)*u0(p,i,m-1)+ &
& (1-2*phi(p,g))*u0(p,i,m)+phi(p,g)*u0(p,i,m+1)

if((j.gt.1).and.(m.eq.2)) w1(p,i,m)=phi(p,g)*uyi(p,i,j)+&
& (1-2*phi(p,g))*u(p,i,m)+phi(p,g)*u(p,i,m+1)

if((j.gt.1).and.(m.eq.N(3))) w1(p,i,m)=phi(p,g)*uyf(p,i,j)+&
& (1-2*phi(p,g))*u(p,i,m)+phi(p,g)*u(p,i,m-1)

if(((j.gt.1).and.(m.gt.2)).and.(m.lt.N(3))) &
& w1(p,i,m)= phi(p,g)*u(p,i,m-1)+(1-2*phi(p,g))*u(p,i,m) &
& +phi(p,g)*u(p,i,m+1)

    enddo
enddo

end subroutine primeiropasso2D_dirichlet
#####
#####

!***** periodica *****
!#####
!#####

subroutine primeiropasso1D_periodico(N,j,p,g,phi,A,u,w1)
integer :: i
integer, intent(in) :: j,p,g
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:,,:), intent(in) :: phi
real, allocatable, dimension(:, :, :), intent(in) :: A,u
real, allocatable, dimension (:, :, :) :: w1

! Primeiro passo. Passo para determinar vetor w de forma explicita
do i=2,N(1)

    if(i.eq.2) w1(p,i,j)=phi(p,g)*u(p,N(1),j)+ &
& (1-2*phi(p,g))*u(p,i,j)+phi(p,g)*u(p,i+1,j)

```

```

        if(i.eq.N(1)) w1(p,i,j)=phi(p,g)*u(p,i-1,j)+ &
        & (1-2*phi(p,g))*u(p,i,j)+phi(p,g)*u(p,2,j)

        if((i.gt.2).and.(i.lt.N(1))) w1(p,i,j)= &
        & phi(p,g)*u(p,i-1,j)+(1-2*phi(p,g))*u(p,i,j) &
        & +phi(p,g)*u(p,i+1,j)

    enddo

end subroutine primeiropasso1D_periodico

!#####
!#####
subroutine primeiropasso2D_periodico(N,j,p,g,phi,A,u,u0,w1)
integer :: i,m
integer, intent(in) :: j,p,g
real, allocatable, dimension(:,,:), intent(in) :: phi
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:, :, :), intent(in) :: A,u,u0
real, allocatable, dimension (:, :, :) :: w1

! Primeiro passo. Passo para determinar vetor w de forma explícita
do i=2,N(1)
do m=2,N(3)

    if(j.eq.1)then

        if(m.eq.2) w1(p,i,m)=phi(p,g)*u0(p,i,N(3))+ &
        & (1-2*phi(p,g))*u0(p,i,m)+phi(p,g)*u0(p,i,m+1)

        if(m.eq.N(3)) w1(p,i,m)=phi(p,g)*u0(p,i,m-1)+ &
        & (1-2*phi(p,g))*u0(p,i,m)+phi(p,g)*u0(p,i,2)

        if((m.gt.2).and.(m.lt.N(3))) w1(p,i,m)= &
        & phi(p,g)*u0(p,i,m-1)+(1-2*phi(p,g))*u0(p,i,m) &
        & +phi(p,g)*u0(p,i,m+1)
    endif
enddo
enddo

```

```

else

  if(m.eq.2) w1(p,i,m)=phi(p,g)*u(p,i,N(3))+ &
& (1-2*phi(p,g))*u(p,i,m)+phi(p,g)*u(p,i,m+1)

  if(m.eq.N(3)) w1(p,i,m)=phi(p,g)*u(p,i,2)+ &
& (1-2*phi(p,g))*u(p,i,m)+phi(p,g)*u(p,i,m-1)

  if((m.gt.2).and.(m.lt.N(3))) w1(p,i,m)= &
& phi(p,g)*u(p,i,m-1)+(1-2*phi(p,g))*u(p,i,m) &
& +phi(p,g)*u(p,i,m+1)

endif

  enddo
enddo

end subroutine primeiropasso2D_periodico

!#####
!#####

!***** Neumann *****
!#####
subroutine primeiropasso1D_neumann(N,j,p,g,phi,h,A,u,w1)
integer :: i
integer, intent(in) :: j,p,g
real, allocatable, dimension(:), intent(in) :: h
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:, :), intent(in) :: phi
real, allocatable, dimension(:, :, :), intent(in) :: A,u
real, allocatable, dimension (:, :, :) :: w1

! Primeiro passo. Passo para determinar vetor w de forma explícita
do i=2,N(1)

  if(i.eq.2) w1(p,i,j)=-2*h(g)*phi(p,g)*u(p,i-1,j)+ &

```

```

& (1-2*phi(p,g))*u(p,i,j)+2*phi(p,g)*u(p,i+1,j)

if(i.eq.N(3)) w1(p,i,j)=2*h(g)*phi(p,g)*u(p,i-1,j)+ &
& (1-2*phi(p,g))*u(p,i,j)+2*phi(p,g)*u(p,i+1,j)

if((i.ne.2).and.(i.ne.N(3))) w1(p,i,j)= &
& phi(p,g)*u(p,i-1,j)+(1-2*phi(p,g))*u(p,i,j)+ &
& phi(p,g)*u(p,i+1,j)

enddo
end subroutine primeiropasso1D_neumann

!#####
!#####
subroutine primeiropasso2D_neumann(N,j,p,g,phi,h,A,u,uyi,uyf,u0,w1)
integer :: i,m
integer, intent(in) :: j,p,g
real, allocatable, dimension(:), intent(in) :: h
real, allocatable, dimension(:,,:), intent(in) :: phi
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:, :, :), intent(in) :: A,u,uyi,uyf,u0
real, allocatable, dimension (:, :, : ) :: w1

! Primeiro passo. Passo para determinar vetor w de forma explícita
do i=2,N(1)
do m=2,N(3)

if((j.eq.1).and.(m.eq.2))w1(p,i,m)= &
&-2*h(g)*phi(p,g)*u0(p,i,m-1)+(1-2*phi(p,g))*u0(p,i,m) &
& +2*phi(p,g)*u0(p,i,m+1)

if((j.eq.1).and.(m.eq.N(3)))w1(p,i,m)= &
& 2*h(g)*phi(p,g)*u0(p,i,m+1)+(1-2*phi(p,g))*u0(p,i,m) &
& +2*phi(p,g)*u0(p,i,m-1)

if(((j.eq.1).and.(m.ne.N(3))).and.(m.ne.2))w1(p,i,m)= &
& phi(p,g)*u0(p,i,m+1)+(1-2*phi(p,g))*u0(p,i,m)+ &
& phi(p,g)*u0(p,i,m-1)

if((j.gt.1).and.(m.eq.2))w1(p,i,m)= &

```

```

& -2*h(g)*phi(p,g)*uyi(p,i,j)+(1-2*phi(p,g))*u(p,i,m) &
& +2*phi(p,g)*u(p,i,m+1)

```

```

if((j.gt.1).and.(m.eq.N(3)))w1(p,i,m)= &
& 2*h(g)*phi(p,g)*uyf(p,i,j)+(1-2*phi(p,g))*u(p,i,m) &
& +2*phi(p,g)*u(p,i,m-1)

```

```

if(((j.gt.1).and.(m.gt.2)).and.(m.lt.N(3))) &
& w1(p,i,m)=phi(p,g)*u(p,i,m-1)+(1-2*phi(p,g))*u(p,i,m) &
& +phi(p,g)*u(p,i,m+1)

```

```

    enddo

```

```

enddo

```

```

end subroutine primeiropasso2D_neumann

```

```

!#####
!#####

```

```

end module primeiro_passo

```

```

!#####
!#####

```

segundo-passo.f90

```

module segundo_passo
  implicit none
  contains

!#####
subroutine segundopasso1D_dirichlet(i,p,g,N,k1,k,j,phi,func1, &
& func2,A,w,u,v11)

integer :: i2,q
integer, intent(in) :: i,k1,j,p,g
real, intent(in) :: k,func1,func2
integer, allocatable, dimension(:),intent(in) :: N
real, allocatable, dimension(:,:),intent(in) :: phi
real, allocatable, dimension(:, :, :),intent(in) :: A,w,u
real, allocatable, dimension(:, :, : ) :: v11
!-----
!   Definição dos pivos

      if((i.eq.2).or.(i.eq.N(1)))then

!   Determinação do primeiro pivo
      if(i.eq.2)v11(p,i,k1+1)=(w(p,i,j)+ &
& (k/2)*(func1+func2)+phi(p,g)*u(p,1,j+1))/A(p,i,i)!

      if(i.eq.N(1))v11(p,i,k1+1)=(w(p,i,j)+ &
& (k/2)*(func1+func2)+phi(p,g)*u(p,N(1)+1,j+1))/A(p,i,i)!

      else

      v11(p,i,k1+1)=(w(p,i,j)+(k/2)*(func1+func2))/A(p,i,i)

      endif

!-----

```

```

do 80 i2=2,N(1)
!      numeros de Seidel

      if(i2.eq.i)goto 80
      if(i2.gt.i)q=k1
      if(i2.lt.i)q=k1+1

!      implementacao do metodo de Gauss-Seidel
      v11(p,i,k1+1)=v11(p,i,k1+1)-A(p,i,i2)*v11(p,i2,q)/A(p,i,i)

80      continue

end subroutine segundopasso1D_dirichlet
#####
#####
subroutine segundopasso2D_dirichlet(N,i,m,p,g,k1,j,k,phi, &
& func1,func2,A,uxi,uxf,w,v11)
integer :: q,i2
integer, intent(in) :: i,m,k1,j,p,g
real, intent(in) :: k,func1,func2
integer, allocatable, dimension(:),intent(in) :: N
real, allocatable, dimension(:,,:),intent(in) :: phi
real, allocatable, dimension(:, :, :),intent(in) :: A,w,uxi,uxf
real, allocatable, dimension(:, :, :)) :: v11

! Segundo passo. Método iterativo de Gauss-Seidel para o
! passo implícito para determinar u(i,m,j+1/2)
      if((i.eq.2).or.(i.eq.N(1)))then

          if(i.eq.2) v11(p,i,k1+1)=(w(p,i,m)+(k/4)*(func1+func2) &
& -(phi(p,g-1)*phi(p,g)/2)*(uxi(p,m-1,j+1)+uxi(p,m+1,j+1))+ &
& (phi(p,g-1)*phi(p,g)/2)*(uxi(p,m-1,j)+uxi(p,m+1,j)) &
& +phi(p,g-1)*phi(p,g)*(uxi(p,m,j+1)-uxi(p,m,j))+ &
& (phi(p,g)/2)*(uxi(p,m,j+1)+uxi(p,m,j)))/A(p,i,i)

          if(i.eq.N(1)) v11(p,i,k1+1)=(w(p,i,m)+(k/4)*(func1+func2) &
& -(phi(p,g-1)*phi(p,g)/2)*(uxf(p,m-1,j+1)+uxf(p,m+1,j+1))+ &
& (phi(p,g-1)*phi(p,g)/2)*(uxf(p,m-1,j)+uxf(p,m+1,j)) &

```

```

& +phi (p,g-1) *phi (p,g) * (uxf (p,m,j+1) -uxf (p,m,j)) + &
& (phi (p,g) /2) * (uxf (p,m,j+1) +uxf (p,m,j)) /A(p,i,i)

else
    v11 (p,i,k1+1) = (w(p,i,m) + (k/4) * (func1+func2)) /A(p,i,i) !
endif

do 80 i2=2,N(1)
!     números de Seidel

        if(i2.eq.i) goto 80
        if(i2.gt.i) q=k1
        if(i2.lt.i) q=k1+1

!     implementação do método de Gauss-Seidel
        v11 (p,i,k1+1) = v11 (p,i,k1+1) - A(p,i,i2) * v11 (p,i2,q) /A(p,i,i)

80     continue

end subroutine segundopasso2D_dirichlet
!#####
!#####

!***** Periodico *****
!#####
!#####
subroutine segundopasso1D_periodico(i,p,g,N,k1,k,j,phi,func1 &
& ,func2,A,w,u,v11)
integer :: i2,q
integer, intent(in) :: i,k1,j,p,g
real, intent(in) :: k,func1,func2
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:,:), intent(in) :: phi
real, allocatable, dimension(:, :, :), intent(in) :: A,w,u
real, allocatable, dimension(:, :, : ) :: v11

```



```

v11(p,i,k1+1)=(w(p,i,j)+(k/2)*(func1+func2))/A(p,i,i)

!-----

do 80 i2=2,N(1)
!   numeros de Seidel

      if(i2.eq.i)goto 80
      if(i2.gt.i)q=k1
      if(i2.lt.i)q=k1+1

!   implementacao do metodo de Gauss-Seidel
      v11(p,i,k1+1)=v11(p,i,k1+1)-A(p,i,i2)*v11(p,i2,q)/A(p,i,i)

80      continue

end subroutine segundopasso1D_periodico
#####
#####
subroutine segundopasso2D_periodico(N,i,m,p,g,k1,j,k,phi,func1, &
& func2,A,w,u,v11)
integer :: q,i2
integer, intent(in) :: i,m,k1,j,p,g
real, intent(in) :: k,func1,func2
integer, allocatable, dimension(:),intent(in) :: N
real, allocatable, dimension(:,,:),intent(in) :: phi
real, allocatable, dimension(:, :, :),intent(in) :: A,w,u
real, allocatable, dimension(:, :, :)) :: v11

! Segundo passo. Método iterativo de Gauss-Seidel para o passo
! implícito para determinar u(i,m,j+1/2)

v11(p,i,k1+1)=(w(p,i,m)+(k/4)*(func1+func2))/A(p,i,i)

do 80 i2=2,N(1)
!   numeros de Seidel

      if(i2.eq.i)goto 80

```

```

        if(i2.gt.i)q=k1
        if(i2.lt.i)q=k1+1

!      implementacao do metodo de Gauss-Seidel
        v11(p,i,k1+1)=v11(p,i,k1+1)-A(p,i,i2)*v11(p,i2,q)/A(p,i,i)

80      continue

end subroutine segundopasso2D_periodico
!#####
!#####

!***** Neumann *****
!#####
!#####
subroutine segundopasso1D_neumann(i,p,g,N,k1,k,j,phi,h,func1, &
& func2,A,w,u,v11)

integer :: i2,q
integer, intent(in) :: i,k1,j,p,g
real, intent(in) :: k,func1,func2
real, allocatable,dimension(:), intent(in) :: h
integer, allocatable, dimension(:),intent(in) :: N
real, allocatable, dimension(:,:),intent(in) :: phi
real, allocatable, dimension(:, :, :),intent(in) :: A,w,u
real, allocatable, dimension(:, :, : ) :: v11
!-----
!      Definição dos pivos

        if((i.eq.2).or.(i.eq.N(1)))then

!      Determinação do primeiro pivo
        if(i.eq.2)v11(p,i,k1+1)=(w(p,i,j)+(k/2)*(func1+func2) &
& -(phi(p,g)*h(g))*u(p,1,j+1))/A(p,i,i)!

        if(i.eq.N(1))v11(p,i,k1+1)=(w(p,i,j)+(k/2)*(func1+func2) &
& +(phi(p,g)*h(g))*u(p,N(1)+1,j+1))/A(p,i,i)!

```

```

else

v11(p,i,k1+1)=(w(p,i,j)+(k/2)*(func1+func2))/A(p,i,i)

endif

!-----

do 80 i2=2,N(1)
!   numeros de Seidel

      if(i2.eq.i)goto 80
      if(i2.gt.i)q=k1
      if(i2.lt.i)q=k1+1

!   implementacao do metodo de Gauss-Seidel
      v11(p,i,k1+1)=v11(p,i,k1+1)-A(p,i,i2)*v11(p,i2,q)/A(p,i,i)

80      continue

end subroutine segundopasso1D_neumann

!#####
!#####
subroutine segundopasso2D_neumann(N,i,m,p,g,k1,j,k,phi,h,func1, &
& func2,A,uxi,uxf,w,v11)
integer :: q,i2
integer, intent(in) :: i,m,k1,j,p,g
real, intent(in) :: k,func1,func2
real, allocatable,dimension(:), intent(in) :: h
integer, allocatable, dimension(:),intent(in) :: N
real, allocatable, dimension(:,,:),intent(in) :: phi
real, allocatable, dimension(:,,:),intent(in) :: A,w,uxi,uxf
real, allocatable, dimension(:,,:),:) :: v11

! Segundo passo. Método iterativo de Gauss-Seidel para o passo

```

```

! implícito para determinar u(i,m,j+1/2)
  if((i.eq.2).or.(i.eq.N(1)))then

    if(i.eq.2) v11(p,i,k1+1)=(w(p,i,m)+(k/4)*(func1+func2)- &
& (phi(p,g)*h(g))*(uxi(p,m,j+1)+uxi(p,m,j)))/A(p,i,i)

    if(i.eq.N(1)) v11(p,i,k1+1)=(w(p,i,m)+(k/4)*(func1+func2)+ &
& (phi(p,g)*h(g))*(uxf(p,m,j+1)+uxf(p,m,j)))/A(p,i,i)

  else
    v11(p,i,k1+1)=(w(p,i,m)+(k/4)*(func1+func2))/A(p,i,i)!

  endif

  do 80 i2=2,N(1)
!      números de Seidel

    if(i2.eq.i)goto 80
    if(i2.gt.i)q=k1
    if(i2.lt.i)q=k1+1

!      implementação do método de Gauss-Seidel
    v11(p,i,k1+1)=v11(p,i,k1+1)-A(p,i,i2)*v11(p,i2,q)/A(p,i,i)

80      continue

  end subroutine segundopasso2D_neumann
!#####
!#####

  end module segundo_passo

!#####
!#####

```

terceiro-passo.f90

```

module terceiro_passo
  implicit none
  contains

!***** dirichlet *****
!#####
  subroutine terceiropasso2D_dirichlet(N,p,g,i,m,k1,j,phi,A,uyi, &
    & uyf,w,u,v11)
    integer :: q,m1
    integer, intent(in) :: i,m,k1,j,p,g
    integer, allocatable, dimension(:), intent(in) :: N
    real, allocatable, dimension(:,,:), intent(in) :: phi
    real, allocatable, dimension(:, :, :), intent(in) :: A,u,uyi,uyf,w
    real, allocatable, dimension(:, :, :)) :: v11

    if((m.eq.2).or.(m.eq.N(3)))then

      if(m.eq.2) v11(p,m,k1+1)=(2*u(p,i,m)-w(p,i,m)+ &
        & phi(p,g)*uyi(p,i,j+1))/A(p,m,m)

      if(m.eq.N(3))v11(p,m,k1+1)=(2*u(p,i,m)-w(p,i,m)+ &
        & phi(p,g)*uyf(p,i,j+1))/A(p,m,m)

    else
      v11(p,m,k1+1)=(2*u(p,i,m)-w(p,i,m))/A(p,m,m)
    endif

    do 81 m1=2,N(3)
!      números de Seidel

      if(m1.eq.m)goto 81
      if(m1.gt.m)q=k1
      if(m1.lt.m)q=k1+1

```

```

!      implementação do método de Gauss-Seidel
      v11(p,m,k1+1)=v11(p,m,k1+1)-A(p,m,m1)*v11(p,m1,q)/A(p,m,m)

81      continue

end subroutine terceiropasso2D_dirichlet

!#####
!#####

!***** Periodico *****
!#####
!#####

subroutine terceiropasso2D_periodico(N,p,g,i,m,k1,j,phi,A,w,u, &
& v11)
integer :: q,m1
integer, intent(in) :: i,m,k1,j,p,g
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:,,:), intent(in) :: phi
real, allocatable, dimension(:, :, :), intent(in) :: A,u,w
real, allocatable, dimension(:, :, : ) :: v11

      v11(p,m,k1+1)=(2*u(p,i,m)-w(p,i,m))/A(p,m,m)

do 81 m1=2,N(3)
!      numeros de Seidel

      if(m1.eq.m) goto 81
      if(m1.gt.m) q=k1
      if(m1.lt.m) q=k1+1

!      implementacao do metodo de Gauss-Seidel
      v11(p,m,k1+1)=v11(p,m,k1+1)-A(p,m,m1)*v11(p,m1,q)/A(p,m,m)

81      continue

end subroutine terceiropasso2D_periodico
!#####
!#####

```

```

!***** neumann *****
!#####
subroutine terceiropasso2D_neumann(N,p,g,i,m,k1,j,phi,h,A,uyi &
& ,uyf,w,u,v11)
integer :: q,m1
integer, intent(in) :: i,m,k1,j,p,g
real, allocatable, dimension(:), intent(in) :: h
integer, allocatable, dimension(:), intent(in) :: N
real, allocatable, dimension(:, :), intent(in) :: phi
real, allocatable, dimension(:, :, :), intent(in) :: A,u,uyi,uyf,w
real, allocatable, dimension(:, :, :)) :: v11

        if((m.eq.2).or.(m.eq.N(3)))then

                if(m.eq.2) v11(p,m,k1+1)=(2*u(p,i,m)-w(p,i,m)- &
& 2*h(g)*phi(p,g)*uyi(p,i,j+1))/A(p,m,m)

                if(m.eq.N(3))v11(p,m,k1+1)=(2*u(p,i,m)-w(p,i,m)+ &
& 2*h(g)*phi(p,g)*uyf(p,i,j+1))/A(p,m,m)

        else

                v11(p,m,k1+1)=(2*u(p,i,m)-w(p,i,m))/A(p,m,m)

        endif

do 81 m1=2,N(3)
!       números de Seidel

                if(m1.eq.m)goto 81
                if(m1.gt.m)q=k1
                if(m1.lt.m)q=k1+1

!       implementação do método de Gauss-Seidel
                v11(p,m,k1+1)=v11(p,m,k1+1)-A(p,m,m1)*v11(p,m1,q)/A(p,m,m)

81       continue

```

```
end subroutine terceiropasso2D_neumann
!#####
!#####

end module terceiro_passo

!#####
!#####
```


runge-kutta.f90

```

module runge_kutta
implicit none
contains

!#####
subroutine rk(nv_dependente,nv_espaco,n1,n2,j,a1,a2,k,t1,f,z,xk &
& ,u,u11)

integer :: p,error,nm
integer, intent(in) :: n1,n2,nv_dependente,j,nv_espaco
real, intent(in) :: k,t1,a1,a2
real :: t,a3,a4,a5,a6,k1,k2,k11,k22,kk1,kk2,kk11,kk22,e,Temp,xi
real, allocatable, dimension(:) :: f,z
real, allocatable, dimension(:, :) :: xk
real, allocatable, dimension(:, :, :), intent(in) :: u
real, allocatable, dimension(:, :, : ) :: u11
!-----

do nm=1,4
do 1 p=1,nv_dependente

if(nm.eq.1)xk(nm,p)=0

!-----
select case (nm)
case(2)
if(nv_espaco.eq.1) z(p)=u(p,n1,j)+xk(nm-1,p)/2
if(nv_espaco.eq.2) z(p)=u(p,n1,n2)+xk(nm-1,p)/2
t=t1+k/2

case(3)
if(nv_espaco.eq.1) z(p)=u(p,n1,j)+xk(nm-1,p)/2
if(nv_espaco.eq.2) z(p)=u(p,n1,n2)+xk(nm-1,p)/2
t=t1+k/2

case(4)
if(nv_espaco.eq.1) z(p)=u(p,n1,j)+xk(nm-1,p)
if(nv_espaco.eq.2) z(p)=u(p,n1,n2)+xk(nm-1,p)

```

```

        t=t1+k

        case default
        if (nv_espaco.eq.1) z(p)=u(p,n1,j)
        if (nv_espaco.eq.2) z(p)=u(p,n1,n2)
        t=t1

    end select
!-----

1  continue

    kk1=1.0
    kk11=0.01
    kk2=1.0
    kk22=0.01
    e=4
    xi=0.25
    Temp=0.75

    k1=kk1*Exp(e*(1-(1/Temp)))
    k11=kk11*Exp(e*(1-(1/Temp)))
    k2=kk2*Exp(e*(1-(1/Temp)))
    k22=kk22*Exp(e*(1-(1/Temp)))

!-----
!   BRUSSELATOR
    f(1)=xi*(a1-z(1))+k1*z(1)*z(1)*z(2)-k11*z(1)*z(1)*z(1) &
    & -k2*a2*z(1)+k22*z(2)
    f(2)=k2*a2*z(1)-k1*z(1)*z(1)*z(2)+k11*z(1)*z(1)*z(1)-k22*z(2)
!-----

    do 2 p=1,nv_dependente

        xk(nm,p)=k*f(p)

2  continue

enddo

```

```
do 3 p=1,nv_dependente
```

```
    nm=1
```

```
    if(nv_espaco.eq.1) u11(p,n1,j+1)=u(p,n1,j)+xk(nm,p)/6 &
    & +xk(nm+1,p)/3+xk(nm+2,p)/3+xk(nm+3,p)/6
```

```
    if(nv_espaco.eq.2) u11(p,n1,n2)=u(p,n1,n2)+xk(nm,p)/6 &
    & +xk(nm+1,p)/3+xk(nm+2,p)/3+xk(nm+3,p)/6
```

```
3    continue
```

```
end subroutine rk
```

```
!#####
!#####
```

```
end module runge_kutta
```

```
!#####
!#####
```

Exemplo de inputfile para a simulação do *Brusselator* isotérmico em duas dimensões espaciais.

inputfile.f90

```
# ***** INPUTFILE PARA O PROGRAMA EDP *****
# Defina a dimensão espacial em que a simulação será realizada:
# 1D ou 2D?
'2D'
# Defina o numero de passos numéricos para a variável independente x
39
# Defina a posição inicial em x
0
# Defina a posição final em x
30
# Defina o numero de passos numéricos para a variável independente
# (se a simulação for realizada em 1D, escreva zero da seguinte
# forma: 0)
39
# Defina a posição inicial em y
0
# Defina a posição final em y
30
# Defina o numero de passos numéricos para o tempo
1000
# Defina o tempo máximo de simulação
100
# Defina o numero de variáveis dependentes
2
# Defina o valor da constante de difusão da variável dependente 1
1.0
# Defina o valor da constante de difusão da variável dependente 2
5.0
# Defina o tipo de fronteira para o reagente 1: Periódica (P),
# Dirichlet (D), Neumann (N)?
'P'
# Defina o tipo de fronteira para o reagentes 2 Periódica (P),
# Dirichlet (D), Neumann (N)?
'P'
# Defina as condições iniciais para a variável dependente 1
2.1
```

```
# Defina as condições de fronteira para a variável dependente 1,  
# para (x,y) -> (0,0:N) (Menor concentração)  
0.0  
# Defina as condições de fronteira para a variável dependente 1,  
# para (x,y) -> (N,0:N) (Maior concentração)  
0.0  
# Defina as condições iniciais para a variável dependente 2  
2.39  
# Defina as condições de fronteira para a variável dependente 2,  
# para (x,y) -> (0,0:N) (Menor concentração)  
0.0  
# Defina as condições de fronteira para a variável dependente 2,  
# para (x,y) -> (N,0:N) (Maior concentração)  
0.0  
# Defina o valor do parâmetro a  
2.1  
# Defina o valor do parâmetro b  
5.0
```

7.6 – Pontos de equilíbrio

Como foi exposto no texto principal, o MI é descrito pelas seguintes equações:

$$\frac{\partial u}{\partial \tau} = f(u, v) + d_u \nabla^2 u \quad (7.69)$$

$$\frac{\partial v}{\partial \tau} = g(u, v) + d_v \nabla^2 v \quad (7.70)$$

com $f(u, v) = \xi(a - u) - r_1 + r_2$, $g(u, v) = r_1 - r_2$, e

$$r_1 = -k_{f_1} e^{e_1(1-\frac{1}{\theta})} u^2 v + k_{r_1} e^{e_2(1-\frac{1}{\theta})} u^3$$

$$r_2 = -k_{f_2} e^{e_3(1-\frac{1}{\theta})} u b + k_{r_2} e^{e_4(1-\frac{1}{\theta})} v d$$

O ponto de equilíbrio (u^*, v^*) é determinado com a condição $f(u^*, v^*) = g(u^*, v^*) = 0$ é satisfeita. Assim sendo, no caso do MI:

$$u^* = a \quad (7.71)$$

$$v^* = \frac{abk_{f_2} + a^3 k_{r_1}}{a^2 k_{r_1} + dk_{r_2}} \quad (7.72)$$

No caso do MNI o ponto de equilíbrio é (u^*, v^*, θ^*) , com:

$$\theta^* = \frac{1}{(a\gamma\xi + \phi)(a^2 k_{f_1} + dk_{r_2})} (-a^3 b \Delta H_1 k_{f_1} k_{f_2} - a^3 b \Delta H_2 k_{f_1} k_{f_2} + a^3 d \Delta H_1 k_{r_1} k_{r_2} + a^3 d \Delta H_2 k_{r_1} k_{r_2} + a^3 \gamma k_{f_1} \xi + a^2 k_{f_1} \theta_{\Omega} \phi + a\gamma dk_{r_2} \xi + dk_{r_2} \theta_{\Omega} \phi) \quad (7.73)$$