

Pedro Augusto Ferreira Rocha

**Emprego de GPGPUs para Acelerar Simulações do Sistema Humano Inato**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Orientador: Prof. D.Sc. Marcelo Lobosco

Coorientador: Prof. D.Sc. Rodrigo Weber dos Santos

Juiz de Fora

2012

Rocha, Pedro Augusto Ferreira

Emprego de GPGPUs para Acelerar Simulações do Sistema Humano Inato/Pedro Augusto Ferreira Rocha. – Juiz de Fora: UFJF/MMC, 2012.

XIV, 94 p.: il.; 29, 7cm.

Orientador: Marcelo Lobosco

Coorientador: Rodrigo Weber dos Santos

Dissertação (mestrado) – UFJF/MMC/Programa de Modelagem Computacional, 2012.

Referências Bibliográficas: p. 87 – 90.

1. Sistema imune inato. 2. Análise de sensibilidade.  
3. Equações diferenciais parciais. 4. Método das diferenças finitas. 5. Computação paralela. I. Lobosco, Marcelo *et al.* II. Universidade Federal de Juiz de Fora, MMC, Programa de Modelagem Computacional.

Pedro Augusto Ferreira Rocha

## **Emprego de GPGPUs para Acelerar Simulações do Sistema Humano Inato**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Aprovada em 27 de Agosto de 2012.

### BANCA EXAMINADORA

---

Prof. D.Sc. Marcelo Lobosco - Orientador  
Universidade Federal de Juiz de Fora

---

Prof. D.Sc. Rodrigo Weber dos Santos - Coorientador  
Universidade Federal de Juiz de Fora

---

Profa. D.Sc. Gizelle Kupac Vianna  
Universidade Federal Rural do Rio de Janeiro

---

Profa. D.Sc. Flávia de Souza Bastos  
Universidade Federal de Juiz de Fora

*Este trabalho dedico a toda a  
minha família, aos meus amigos  
e colegas e professores do  
mestrado...*

## AGRADECIMENTOS

Agradeço a toda minha família, principalmente meus pais e meus irmãos, pelo suporte nos momentos difíceis que encontrei durante todo este tempo. Agradeço aos meus amigos que convivi durante estes anos. Minha namorada que sempre que possível tentou me acompanhar de perto. Agradeço também aos colegas de trabalho e orientadores que me ajudaram bastante no desenrolar deste trabalho.

*A dream you dream alone is only  
a dream. A dream you dream  
together is reality.*

*John Lennon*

## RESUMO

Dois mecanismos são utilizados pelo Sistema Imunológico Humano (SIH) para defender o organismo contra doenças causadas pelos mais distintos agentes patogênicos: o sistema inato e o sistema adaptativo. O primeiro é composto por células e substâncias químicas que utilizam um mecanismo genérico de defesa para prevenir ou limitar infecções ocasionadas pela maioria dos patógenos. Já o segundo mecanismo é ativado pelo primeiro, baseando-se na habilidade de reconhecer e de recordar agentes patogênicos específicos, colaborando para a montagem de um ataque mais potente a cada vez que o mesmo patógeno é encontrado. Apesar de ser muito estudado, muitas questões sobre o funcionamento do SIH ainda estão em aberto em virtude de sua complexidade e do grande número de interações, nos mais diversos níveis, entre seus distintos componentes. Neste sentido, ferramentas computacionais podem se constituir em um poderoso ferramental para auxiliar nas pesquisas sobre o tema. O presente trabalho está inserido neste escopo, dividindo-se em duas partes. Na primeira parte, o trabalho apresenta os resultados de uma análise de sensibilidade em um modelo matemático-computacional que simula a resposta imunológica inata ao lipopolissacarídeo (LPS), com o objetivo de encontrar os parâmetros mais sensíveis deste modelo. Além disto, a segunda parte do trabalho propõe uma adaptação do modelo original para um modelo tridimensional. As simulações realizadas nas duas partes do trabalho mostraram-se computacionalmente caras, demandando longos períodos de tempo para serem concluídas. Assim, GPGPUs (*General Purpose Graphics Processing Units*) foram utilizadas para reduzir os tempos de execução. O uso de GPGPUs permitiu que acelerações de 276 vezes para a análise de sensibilidade massiva e de 87 vezes para a computação do modelo em três dimensões fossem obtidas.

**Palavras-chave:** Sistema imune inato. Análise de sensibilidade. Equações diferenciais parciais. Método das diferenças finitas. Computação paralela.

## ABSTRACT

Two mechanisms are used by the Human Immune System (HIS) to protect the body against diseases caused by distinct pathogens: the innate and the adaptive immune system. The first one is composed of cells and chemicals that use a generic mechanism of defense to prevent or limit infections caused by most pathogens. The second mechanism is activated by the first one. It has the ability to recognize and remember specific pathogens, contributing to the assembly of a more powerful attack each time the same pathogen is encountered again. Despite being widely studied, many questions about the functioning of the HIS are still open because of its complexity and the large number of interactions of its components on distinct levels. In this sense, computational tools are a powerful instrument to assist researchers on this field of study. This work is inserted in this scope and it is split into two parts. In the first part, this work presents the results of a sensitivity analysis on a mathematical-computational model that simulates the innate immune response to lipopolysaccharide (LPS). The main objective of the sensitivity analysis was to find the most sensitive parameters of the mathematical model. The second part of this work proposes the extension of the original model to a three-dimensional one. The simulations in the two parts of the work proved to be computationally expensive, requiring long periods of time to complete. Thus, GPGPUs (General Purpose Graphics Processing Units) were used to reduce execution times. The use of GPGPUs allowed speedups of 276 times for sensitivity analysis, when compared to the sequential one, and of 87 times for computations using the three dimensions model.

**Keywords:** Innate immune system. Sensitivity analysis. Partial differential equations. Finite difference method. Parallel computing.



## SUMÁRIO

1	INTRODUÇÃO .....	15
1.1	Motivação e Objetivos .....	15
1.2	Método .....	17
1.3	Trabalhos Correlatos .....	17
1.4	Organização .....	18
2	SISTEMA IMUNOLÓGICO HUMANO INATO .....	19
2.1	Visão geral do sistema imunológico humano .....	19
2.2	Células do sistema imunológico humano inato .....	20
2.3	Reconhecimento entre próprio e não próprio .....	22
2.4	Processo de resposta inflamatória .....	22
2.4.1	<i>Lipopolissacarídeo .....</i>	23
2.4.2	<i>Ativação dos macrófagos e recrutamento de neutrófilos.....</i>	23
3	MODELO MATEMÁTICO .....	25
3.1	Modelo EDO .....	25
3.2	Modelo EDP .....	28
4	GPGPUs: ARQUITETURA E PROGRAMAÇÃO .....	31
4.1	NVIDIA's CUDA .....	31
4.2	Processo de Execução em CUDA .....	34
5	MÉTODOS .....	35
5.1	Análise de Sensibilidade .....	35
5.1.1	<i>Principais Conceitos sobre Análise de Sensibilidade .....</i>	35
5.1.2	<i>Método .....</i>	36
5.1.3	<i>Implementação em GPUs .....</i>	37
5.2	Modelo 3D .....	41
5.2.1	<i>Modelo Computacional.....</i>	41
5.2.2	<i>Implementação em GPUs .....</i>	44
5.3	Otimizações .....	44

5.3.1	<i>Otimizações Adotadas na Análise de Sensibilidade</i> .....	44
5.3.2	<i>Otimizações Adotadas no Sistema 3D</i> .....	46
5.3.2.1	<i>Diminuição de usos de Ramificações em um Kernel</i> .....	46
5.3.2.2	<i>Redução do Número de Acessos à Memória Global</i> .....	48
6	<b>RESULTADOS</b> .....	50
6.1	<b>Análise de Sensibilidade</b> .....	50
6.1.1	<i>Parâmetro <math>N^{max}</math></i> .....	51
6.1.2	<i>Parâmetro <math>\lambda_{N A}</math></i> .....	58
6.1.3	<i>Parâmetro <math>\lambda_{MA A}</math></i> .....	64
6.1.4	<i>Observações Finais sobre a Análise de Sensibilidade</i> .....	64
6.2	<b>Simulação do Modelo 3D</b> .....	67
6.2.1	<i>Caso 1</i> .....	67
6.2.2	<i>Caso 2</i> .....	67
6.2.3	<i>Caso 3</i> .....	74
6.2.4	<i>Comparação entre casos</i> .....	75
6.3	<b>Aceleração Obtida</b> .....	82
6.3.1	<i>Aceleração da Análise de Sensibilidade</i> .....	83
6.3.2	<i>Aceleração do Modelo Tridimensional</i> .....	83
7	<b>CONCLUSÕES E PERSPECTIVAS FUTURAS</b> .....	85
	<b>REFERÊNCIAS</b> .....	87
	<b>APÊNDICES</b> .....	90

## LISTA DE ILUSTRAÇÕES

2.1	Etapas do processo de migração dos neutrófilos da corrente sanguínea para os tecidos. Adaptado de [7]. . . . .	24
3.1	Relacionamentos entre os componentes. . . . .	26
4.1	Representação de blocos em uma grade de duas dimensões e de <i>threads</i> em blocos organizados em três dimensões. Adaptado de [2]. . . . .	32
4.2	Distribuição de blocos de <i>threads</i> em núcleos da GPU. Adaptado de [2]. . . . .	33
4.3	Hierarquia de Memória. Adaptado de [2]. . . . .	33
5.1	Organização adotada para estruturar a grade e os blocos. . . . .	37
5.2	Identificadores das <i>threads</i> . . . . .	38
5.3	Distribuição de cenários por <i>threads</i> . . . . .	38
5.4	Distribuição de cenários e acessos à memória pelas <i>threads</i> . . . . .	40
5.5	Representação do Fluxo exercido pelo processo de quimiotaxia. . . . .	43
5.6	Distribuição de pontos por <i>thread</i> e acesso a vizinhos. . . . .	45
5.7	(a) divergências por ramificações e (b) queda de desempenho devido à serialização do código por causa da ramificação. Adaptado de [34]. . . . .	47
6.1	Evolução temporal de LPS quando $A(0) = 20 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	52
6.2	Evolução temporal de neutrófilos quando $A(0) = 20 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	52
6.3	Evolução temporal de citocinas pró-inflamatórias quando $A(0) = 20 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	53
6.4	Evolução temporal de macrófagos em repouso quando $A(0) = 20 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	53
6.5	Evolução temporal de macrófagos ativos quando $A(0) = 20 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	54
6.6	Evolução temporal de LPS quando $A(0) = 40 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	55

6.7	Evolução temporal de neutrófilos quando $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	55
6.8	Evolução temporal de citocinas pró-inflamatórias quando $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	56
6.9	Evolução temporal de macrófagos em repouso quando $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	56
6.10	Evolução temporal de macrófagos ativos quando $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $N^{max}$ . . . . .	57
6.11	Evolução temporal de LPS quando $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	58
6.12	Evolução temporal de neutrófilos quando $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	59
6.13	Evolução temporal de citocinas pró-inflamatórias quando $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	59
6.14	Evolução temporal de macrófagos em repouso quando $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	60
6.15	Evolução temporal de macrófagos ativos quando $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	60
6.16	Evolução temporal de LPS quando $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	61
6.17	Evolução temporal de neutrófilos quando $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	61
6.18	Evolução temporal de citocinas pró-inflamatórias quando $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	62
6.19	Evolução temporal de macrófagos em repouso quando $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	62
6.20	Evolução temporal de macrófagos ativos quando $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{N A}$ . . . . .	63
6.21	Evolução temporal de LPS quando $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{MA A}$ . . . . .	64
6.22	Evolução temporal de neutrófilos quando $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{MA A}$ . . . . .	65

6.23	Evolução temporal de citocinas pró-inflamatórias quando $A(0) = 20 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{MA A}$ . . . . .	65
6.24	Evolução temporal de macrófagos em repouso quando $A(0) = 20 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{MA A}$ . . . . .	66
6.25	Evolução temporal de macrófagos ativos quando $A(0) = 20 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ para distintos valores de $\lambda_{MA A}$ . . . . .	66
6.26	Distribuição espacial e temporal de antígenos. . . . .	68
6.27	Distribuição espacial e temporal do LPS. . . . .	70
6.28	Distribuição espacial e temporal de macrófagos em repouso. . . . .	71
6.29	Distribuição espacial e temporal de citocinas pró-inflamatórias. . . . .	72
6.30	Distribuição espacial e temporal de macrófagos ativos. . . . .	73
6.31	Distribuição espacial e temporal do LPS. . . . .	76
6.32	Distribuição espacial e temporal de macrófagos em repouso. . . . .	77
6.33	Distribuição espacial e temporal de citocinas pró-inflamatórias. . . . .	78
6.34	Distribuição espacial e temporal de macrófagos ativos. . . . .	79
6.35	Distribuição espacial e temporal de neutrófilos. . . . .	80
6.36	Distribuição espacial e temporal de LPS. . . . .	81
6.37	Distribuição espacial e temporal de neutrófilos. . . . .	82
6.38	Distribuição espacial e temporal de macrófagos em repouso. . . . .	82

## LISTA DE TABELAS

5.1	Discretização temporal e espacial . . . . .	41
6.1	Tempo de execução sequencial e paralelo (utilizando CUDA em execução na placa GTX 285). Tempo representado em segundos. . . . .	83
6.2	Aceleração obtida pelo uso de GPUs. O tempo médio de execução é expresso em segundos. . . . .	84
A.1	Condições iniciais, parâmetros e unidades da Análise de Sensibilidade. . . . .	91
A.2	Condições iniciais do modelo tridimensional. . . . .	92
A.3	Parâmetros do modelo tridimensional. . . . .	92
A.4	Lista de Símbolos 1 . . . . .	93
A.5	Lista de Símbolos 2 . . . . .	94

# 1 INTRODUÇÃO

## 1.1 Motivação e Objetivos

O sistema imunológico (ou sistema imune) está presente em diversas formas de vida, pois exerce um papel importantíssimo para a manutenção da vida. Seu papel primário é o de eliminar agentes patogênicos externos, possíveis causadores de doenças que podem levar o organismo à morte. Além da eliminação dos agentes patogênicos externos, o sistema imune também é responsável por eliminar células mortas do próprio organismo, bem como as que apresentem comportamento anormal, sendo possíveis causadores de tumores e de outros danos, caso não sejam eliminadas.

Devido à grande interação entre seus muitos componentes, a compreensão do sistema imunológico torna-se uma tarefa complexa, porém essencial para o desenvolvimento de novos medicamentos e vacinas, bem como para o tratamentos de doenças. Uma das ferramentas que pode ser usada para auxílio e estudo deste sistema é a modelagem matemática-computacional, que permite que estudos *in silico* sejam feitos, de tal forma que pesquisadores da área possam testar hipóteses e realizar um grande volume de experimentos em um período mais curto de tempo do que seria necessário se os experimentos fossem realizados *in vivo* ou *in vitro*. O presente trabalho está inserido neste escopo, da imunologia computacional, ao propor ferramentas para o estudo do sistema imunológico *in silico*. Neste sentido, o trabalho está dividido em duas partes.

A primeira parte deste trabalho consiste da análise de sensibilidade de um modelo matemático-computacional do sistema imune inato, adaptado de Pigozzo [1]. Ou seja, o trabalho consiste do estudo dos impactos causados por uma gama de variações dos valores de entrada do modelo matemático sobre a saída gerada. Encontrar os valores exatos na literatura para os parâmetros utilizados no modelo matemático do sistema imunológico humano (SIH) inato é uma tarefa extremamente árdua, visto que a literatura reporta diversos tipos de experimentos, cuja grande maioria não está necessariamente relacionada ao SIH. Assim, inúmeros parâmetros necessários no modelo e utilizados neste trabalho sequer foram encontrados, e tiveram seus valores ajustados de modo qualitativo. A motivação para realizar a análise de sensibilidade é a de ajudar a identificação de que

parâmetros devem ser mais cuidadosamente ajustados ou mesmo aqueles que deveriam de algum modo ser obtidos experimentalmente a fim de se obter um resultado mais próximo ao do comportamento real do SIH. Entretanto, esta análise de sensibilidade pode acarretar um grande custo computacional, devido a gigantesca gama de cenários possíveis em um modelo.

A segunda parte deste trabalho consiste na extensão da modelagem da dinâmica dos principais elementos envolvidas na resposta imune inata [1] para ser aplicada a uma seção tridimensional de um tecido genérico. Deste modo, o modelo simula o comportamento das células e moléculas ao decorrer do tempo em uma seção cúbica de tecido, considerando para tal aspectos como a difusão das células e substâncias no tecido bem como o fenômeno da quimiotaxia, exercida por certas substâncias (citocinas pró-inflamatórias, detalhadas no próximo capítulo) produzidas por células e tecidos do organismo. A quimiotaxia é um fenômeno extremamente importante para uma resposta imune eficiente. Este estímulo químico orienta as células do SIH para o local de maior concentração de antígenos. A motivação desta parte do trabalho é criar um modelo mais próximo ao do tecido real. Todavia, o custo computacional deste sistema também é elevado e, se nenhuma alternativa for empregada em contrapartida à programação sequencial, as simulações tridimensionais podem levar muito tempo de execução.

Visto que tanto o processo de análise de sensibilidade quanto o de computação do modelo matemático-computacional do SIH inato em uma seção tridimensional do tecido são caros do ponto de vista computacional, é essencial que sejam empregadas formas de reduzir o tempo gasto na computação e geração de resultados. Neste sentido, este trabalho utiliza-se de GPGPUs (*General-Purpose Computing on Graphics Processing Units*) como plataforma para computação paralela.

Assim, podemos listar os principais objetivos deste trabalho como sendo:

- elaboração de um modelo temporal utilizando equações diferenciais ordinárias (EDOs) com base no modelo originalmente proposto por Pigozzo[1];
- realizar uma análise de sensibilidade no modelo temporal elaborado a fim de se encontrar as variáveis sensíveis do sistema;
- transformar o modelo temporal em um modelo de tecido tridimensional por meio de equações diferenciais parciais (EDP); e



- empregar GPGPUs para reduzir o tempo de execução necessário para realizar a análise de sensibilidade e da simulação do modelo tridimensional.

## 1.2 Método

Em um primeiro momento, desenvolveu-se um modelo matemático para a) três células do sistema imunológico humano inato (neutrófilos, macrófagos em repouso e macrófagos ativados); b) citocinas pró-inflamatórias (TNF- $\alpha$  e IL-8 - modeladas por uma única equação), que são responsáveis pela quimiotaxia (processo de locomoção de células); e c) LPS (uma endotoxina, substância altamente imunogênica). Este modelo foi utilizado em dois cenários distintos: temporal e tridimensional. No cenário temporal, o modelo foi utilizado para realizar uma análise de sensibilidade que envolve a resolução mínima de 469.145 sistemas de EDOs. Já o cenário tridimensional demonstra as interações que ocorrem durante a resposta imune entre o LPS e as células do SIH inato em uma malha de tecido tridimensional. Foram empregados o método das diferenças finitas para realizar a discretização espacial e um método explícito para a evolução temporal na implementação computacional. Para acelerar o desempenho das implementações computacionais, utiliza-se CUDA[2](*Compute Unified Device Architecture*), desenvolvida pela NVIDIA, para o desenvolvimento da versão paralela dos códigos das aplicações em GPGPUs.

## 1.3 Trabalhos Correlatos

O trabalho de Pigozzo [3] apresenta um modelo baseado em EDPs que simula a resposta do SIH inato à presença de lipopolissacarídeos (LPS) em uma seção microscópica de tecido. Tem-se como finalidade registrar o início, a manutenção e a resolução da resposta imune. Seu modelo simplificado [3] considera os seguintes elementos em uma seção unidimensional de tecido: LPS, neutrófilos e citocinas pró-inflamatórias. O modelo foi então paralelizado [4] usando MPI [5], OpenMP[6] e uma versão híbrida usando ambas as ferramentas de paralelização. Um outro modelo proposto por Pigozzo [1] apresenta um extensão de seu modelo inicial [3], com adição de outras células e moléculas do SIH inato, tais como: a) macrófagos, b) neutrófilos apoptóticos, c) citocinas anti-inflamatórias e d) grânulos proteicos. As principais diferenças entre este trabalho de dissertação e o de Pigozzo [3] são:

- é realizada uma análise de sensibilidade na versão temporal do modelo;
- novas células são incluídas no modelo simplificado: macrófagos em repouso e macrófagos ativos;
- o modelo é estendido para um espaço tridimensional; e
- a paralelização do código é realizada para reduzir os custos computacionais envolvidos na análise de sensibilidade e na simulação da versão tridimensional do modelo utilizando GPGPUs.

## 1.4 Organização

Este trabalho foi organizado da seguinte forma. Os próximos três capítulos apresentam os principais conceitos necessários para a compreensão deste trabalho. Em particular, o capítulo 2 apresenta o funcionamento básico do sistema imunológico humano inato. O capítulo 3 apresenta o modelo matemático utilizado, descrevendo todas as equações utilizadas neste trabalho, tanto para o sistema temporal quanto para o tridimensional. O capítulo 4 apresenta os principais conceitos envolvidos na programação em GPGPUs utilizando CUDA. O capítulo 5 descreve como foi implementada a análise de sensibilidade, o modelo 3D e a implementação dos códigos paralelos. Já o capítulo 6 descreve os resultados obtidos nas simulações do sistema de EDOs, bem como o sistema de EDPs. Além disso, também será apresentada a aceleração obtida pelos dois sistemas em função do emprego de GPGPUs. Por fim, no capítulo 7 são realizadas as considerações finais deste trabalho e apresentadas as perspectivas para trabalhos futuros. O apêndice contém tabela dos valores das condições iniciais, parâmetros e lista de símbolos utilizados neste trabalho.

# 2 SISTEMA IMUNOLÓGICO

## HUMANO INATO

Este capítulo apresenta os principais conceitos referentes ao sistema imunológico humano (SIH) inato que serão utilizados neste trabalho. Primeiramente, será fornecida uma visão superficial de como é o funcionamento do sistema imunológico humano, que inicia-se pelas barreiras naturais, passa pela imunidade inata e, por fim, é auxiliado pelo SIH adaptativo. Também serão discutidas a função e a dinâmica das células/moléculas do sistema imunológico humano inato que foram utilizadas nos modelos deste trabalho. A maior parte das informações contidas neste capítulo foi retirada da literatura clássica da imunologia [7, 8, 9]. Trechos extraídos de outras fontes foram ressaltados e os autores explicitados ao longo do texto. O estudo deste capítulo é essencial para compreensão do modelo matemático reproduzido no capítulo 6 deste trabalho.

### 2.1 Visão geral do sistema imunológico humano

A resposta do sistema imunológico humano pode ser dividida em três etapas. Na primeira delas, o organismo é defendido por barreiras naturais inespecíficas, tais como: temperatura, saliva, mucos, pH e o próprio revestimento interno e externo do ser humano. As superfícies corporais humanas propiciam uma barreira física entre o meio interno e o externo, sendo o tecido epitelial responsável por essa proteção. Tanto a pele quanto os tratos gastrointestinal, geniturinário e respiratório compreendem toda malha formada pelo tecido epitelial. Os epitélios proveem um bloqueio efetivo contra o ambiente externo, mas, eventualmente, patógenos podem atravessar ou colonizar o tecido epitelial, causando assim infecções.

Uma vez vencidas as barreiras físicas e químicas do corpo, os patógenos encontram os componentes do SIH inato (células e moléculas), responsáveis por desenvolver uma resposta contra o invasor. A resposta imune inata é auxiliada pelo SIH adaptativo, capaz de eliminar a infecção com mais eficiência em relação ao SIH inato. Porém, a resposta adaptativa leva dias ao invés de horas para se desenvolver, por isso o SIH inato desempenha

um papel importantíssimo para a defesa do organismo.

## 2.2 Células do sistema imunológico humano inato

Esta seção apresenta e descreve o comportamento e interações entre as células e moléculas envolvidas no processo de resposta do sistema imunológico. Um maior foco foi dado aos componentes utilizados neste trabalho, devendo ser ressaltados que o sistema imunológico é composto por várias outras células, moléculas e interações.

O sistema imune é composto primariamente por células denominadas fagócitos, que são células do sangue que protegem o organismo através da fagocitose de indivíduos estranhos no organismo, como bactérias e vírus. São ainda responsáveis por ingerir células mortas. Este trabalho aborda dois tipos de fagócitos: os macrófagos e os neutrófilos (granulócitos comumente chamados de células brancas).

Os macrófagos são células de grandes dimensões e de vida relativamente longa. Enquanto estão em circulação pelo sistema sanguíneo, são chamados de monócitos. Somente ao fazer a migração para os tecidos é que se tornam maduros, e assim chamados de macrófagos. Os macrófagos realizam muitas e diferentes funções, tanto para resposta imune quanto na resposta adaptativa. A princípio, uma de suas funções é engolfar e matar os micro-organismos invasores. Além disso, eles auxiliam no aumento de permeabilidade do tecido epitelial e no processo de recrutamento de outras células do SIH através da liberação de proteínas sinalizadoras (citocinas e quimiocinas). Consequentemente, participam também na indução da inflamação [10, 11]. Ademais, os macrófagos são responsáveis por limpar o organismo de células mortas e restos celulares. Podem ser encontrados abaixo da superfície de todas as áreas do corpo que estão expostas ao mundo exterior, por isso são também chamados de células sentinelas.

Os macrófagos podem assumir dois estados diferentes: macrófagos em repouso e macrófagos ativados. No estado de repouso, os macrófagos funcionam como sentinelas, coletando lixo dentro do organismo como restos celulares, proteínas desconhecidas, e células mortas. O segundo estado, macrófago ativado, é consequência de um sinal liberado pelo invasor, como o LPS. Quando os receptores dos macrófagos em repouso se conectam ao LPS, os macrófagos entram em estado ativado e passam a proliferar e focar a atenção na eliminação dos antígenos. Neste estado, os macrófagos se tornam

maiores e conseqüentemente há um aumento na taxa de fagocitose. Além disso, há a produção das citocinas IL-1 e TNF- $\alpha$  [12], responsáveis pelo aumento da permeabilidade e processo de quimiotaxia (processo pelo qual as células mudam de estado devido à presença de uma substância química para se aproximarem de um local favorável [13]). Por fim, os macrófagos atuam também como células apresentadoras de antígenos (APCs), que realizam a ligação entre a resposta imune inata e a resposta imune adaptativa com o intermédio dos linfócitos T.

As células mais numerosas e importantes na resposta do sistema inato são os neutrófilos polimorfonucleares (PMNs, *polimorphonucleated neutrophils*). Estas células fagocitam uma variedade de micro-organismos, destruindo-os nas vesículas intracelulares por meio de substâncias antibacterianas armazenadas em grânulos citoplasmáticos. A ativação de neutrófilos por citocinas liberadas devido a substâncias advindas de bactérias é ilustrada em modelos *in vivo* de depleção de neutrófilos[14, 15]. Citocinas como o IL-1, IL-2, TNF- $\alpha$ ; o LPS e várias outras substâncias são capazes de realizar a ativação dos neutrófilos, o que provoca uma grande produção de IL-8 [16, 17] - também produzida pelos macrófagos e outros tipos de células, como as epiteliais [18].

As citocinas são representadas por um conjunto de proteínas que, secretadas por algumas células, afetam o estado e comportamento de outras células portadoras de um receptor adequado para a citocina em questão. Já as quimiocinas são um subgrupo de citocinas responsáveis por atrair algumas células na direção percorrida por estas citocinas. O reconhecimento da quimiocina provoca uma mudança de estado e função das células afetadas, como monócitos e neutrófilos. O TNF- $\alpha$  e a IL-1 produzidos principalmente pelos macrófagos, mas também por outras células, são as citocinas inflamatórias primárias. Estas são responsáveis por aumentar a permeabilidade vascular, iniciando assim uma grande produção de quimiocinas (por exemplo a IL-8), importante para o recrutamento de novos macrófagos e neutrófilos para o local infeccionado [19]. A citocina IL-8 é responsável pelo recrutamento de mais neutrófilos ao local infeccionado [17], além de estar associada a uma reação pró-inflamatória [20, 21, 22].

## 2.3 Reconhecimento entre próprio e não próprio

Macrófagos, neutrófilos e outras células do SIH inato possuem receptores de reconhecimento de patógenos, conhecidos como receptores de reconhecimento de padrões (PRRs, *pattern recognition receptors*). PRR's reconhecem estruturas conservadas durante a evolução como: oligossacarídeos ricos em manose, peptidoglicanos, lipopolissacarídeos e DNA CpG não-metilado. Estas estruturas são conhecidas como padrões moleculares associados aos patógenos (PAMPs, *pathogen-associated molecular pattern*) e estão presentes em muitos micro-organismos, mas as próprias células do organismo hospedeiro não as apresentam. Assim, mesmo contando com um repertório fixo de receptores, os sistemas de defesa inata são bastantes efetivos contra uma gama de patógenos, sendo portanto capazes de distinguir entre próprio e não-próprio. Ao reconhecer um padrão como não-próprio, o SIH inato produz uma série de eventos chamada de resposta inflamatória.

## 2.4 Processo de resposta inflamatória

A caracterização da resposta inflamatória se dá devido a qualquer dano ao tecido, seja por ferimento ou por uma substância estranha, que induz a uma complexa sequência de eventos coletivos no local. Para que uma resposta inflamatória seja disparada, o LPS (um dos componentes moleculares presentes na parede celular de alguns invasor) deve interagir com os receptores presentes em algumas células do organismo defensor. Por exemplo, os macrófagos em estado de repouso estão a todo instante em circulação pelos tecidos. Estes macrófagos possuem um receptor, TLR4 (*Toll-like receptor 4*), capaz de detectar lipopolissacarídeos de bactérias gram-negativas. A interligação desses receptores ativa o macrófago, provocando a ingestão da bactéria, degradando-a internamente, o que acarreta a produção de proteínas (**citocinas** e **quimiocinas**), iniciando então a resposta inflamatória.

Calor, inchaço, rubor e dor são as quatro características que definem a inflamação. As citocinas provocam no organismo um aumento da dilatação, acarretando consequentemente uma maior permeabilidade dos vasos sanguíneos, causando assim o inchaço. Já o calor e o rubor são ocasionados devido a um maior fluxo sanguíneo e ao extravasamento de fluido para o local infeccionado. A transmigração de leucócitos

circulantes nas células endoteliais para o tecido é o grande responsável pela sensação de dor.

### **2.4.1 *Lipopolissacarídeo***

A endotoxina LPS (presente na membrana exterior de bactérias gram-negativas) é um potente imunestimulante capaz de induzir uma resposta inflamatória aguda, comparável a uma infecção bacteriana. No momento em que as células do sistema inato provocam a lise de algumas bactérias, o LPS é liberado no hospedeiro. Com esta liberação, células do sistema inato, como neutrófilos e macrófagos, intensificam sua resposta inflamatória. Uma maneira comumente utilizada para provocar a lise e liberação do LPS das bactérias é através da administração de antibióticos [23].

### **2.4.2 *Ativação dos macrófagos e recrutamento de neutrófilos***

Os macrófagos estão a todo instante em circulação nos tecidos, atuando como sentinelas. Por isso, na grande maioria de infecções bacterianas, os macrófagos são as primeiras células do organismo a perceberem a invasão. Cada macrófago, então, passa ao seu estado ativo, onde realiza a fagocitose de invasores e provoca o aumento de permeabilidade do tecido devido a produção de citocinas pró-inflamatórias (IL-1 e o TNF- $\alpha$ ). O aumento da permeabilidade se dá pela mudança de comportamento da proteína selectina das células endoteliais, induzida por estas citocinas pró-inflamatórias a interagir com o ligante da selectina (região sialyl-Lewis) presente nos neutrófilos.

Os neutrófilos estão continuamente circulando pela corrente sanguínea em grande velocidade, porém, quando ocorre a interação entre a selectina e o ligante, os neutrófilos diminuem a velocidade, mantendo-se presos brevemente à célula endotelial. Eles são então rapidamente liberados da interação devido à força de cisalhamento provocada pela circulação sanguínea. O processo é repetido até o momento em que os neutrófilos reconheçam um quimioatraente como o IL-8. A ligação entre os receptores na membrana de um neutrófilo e este quimioatraente provoca a ativação do neutrófilo e emite um sinal que induz uma mudança estrutural nas moléculas de integrina presente naquela membrana. Esta mudança gera um aumento de afinidade da integrina com os ICAMS (*immunoglobulin cell adhesion molecule superfamily*) pertencentes às células endoteliais devido à indução provocada pelas citocinas pró-inflamatórias [19]. Através desta

interação, os neutrófilos adquirem uma adesão firme, realizam a diapedese (passagem pelas paredes dos capilares sanguíneos, vénulas e arteríolas), e por fim têm a transmigração efetuada através das células endoteliais para os tecidos. Este processo é chamado de extravasamento de neutrófilos, conforme ilustrado na Figura 2.1.

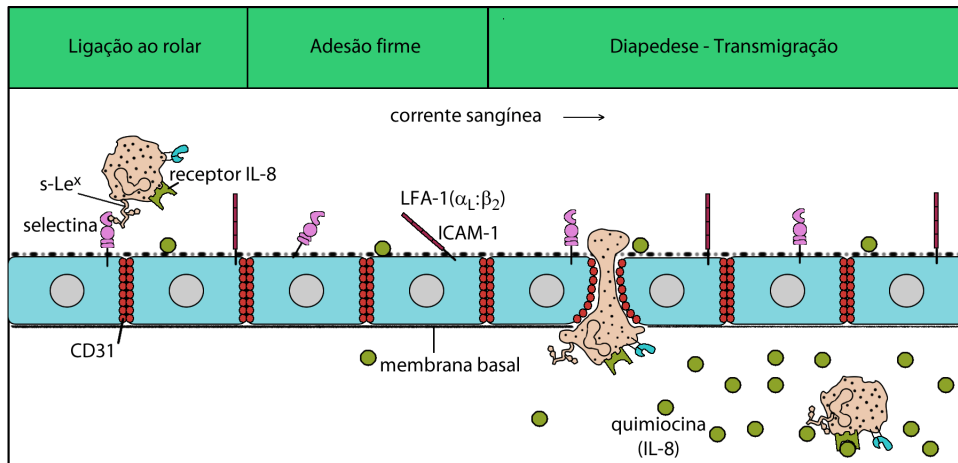


Figura 2.1: Etapas do processo de migração dos neutrófilos da corrente sanguínea para os tecidos. Adaptado de [7].



## 3 MODELO MATEMÁTICO

O modelo proposto neste capítulo é baseado em um trabalho anterior[3] que utiliza um sistema de EDPs para descrever a dinâmica da resposta do sistema imune ao LPS em uma seção microscópica unidimensional de tecido. Em particular, foram modeladas as interações entre LPS, neutrófilos e citocinas. Este capítulo apresenta dois modelos do SIH inato, um representado por um conjunto de EDOs e outro por um conjunto de EDPs.

### 3.1 Modelo EDO

Este é um modelo simplificado utilizando EDOs para simular o comportamento temporal do LPS, dos neutrófilos, dos macrófagos e das citocinas durante a primeira fase de resposta do sistema imune. As principais diferenças entre este modelo em relação ao modelo original [3] são:

- este modelo não considera a dinâmica espacial das células e moléculas; e
- foram acrescentados macrófagos em dois estados de prontidão: repouso e ativo.

A Figura 3.1 apresenta esquematicamente o relacionamento entre macrófagos, neutrófilos, citocinas pró-inflamatórias e LPS. O LPS estimula uma resposta tanto nos macrófagos quanto nos neutrófilos, que reconhecem o LPS e o degradam através de fagocitose. O processo de fagocitose induz, rapidamente, a apoptose dos neutrófilos. Esta indução está associada com a geração de espécies reativas ao oxigênio (*ROS*) [24]. Os macrófagos em repouso se tornam ativos ao encontrarem o LPS no tecido. As citocinas pró-inflamatórias são produzidas tanto pelos macrófagos ativos quanto pelos neutrófilos após o reconhecimento. As citocinas induzem o aumento da permeabilidade endotelial, permitindo que mais neutrófilos atravessem os vasos sanguíneos para entrar na área de tecido infectado.

O conjunto de equações elaboradas para este modelo é apresentado abaixo, onde  $MR$ ,  $MA$ ,  $A$ ,  $N$  e  $CH$  representam a concentração de macrófagos em repouso, de macrófagos ativos, de LPS, de neutrófilos e de citocinas pró-inflamatórias, respectivamente. As

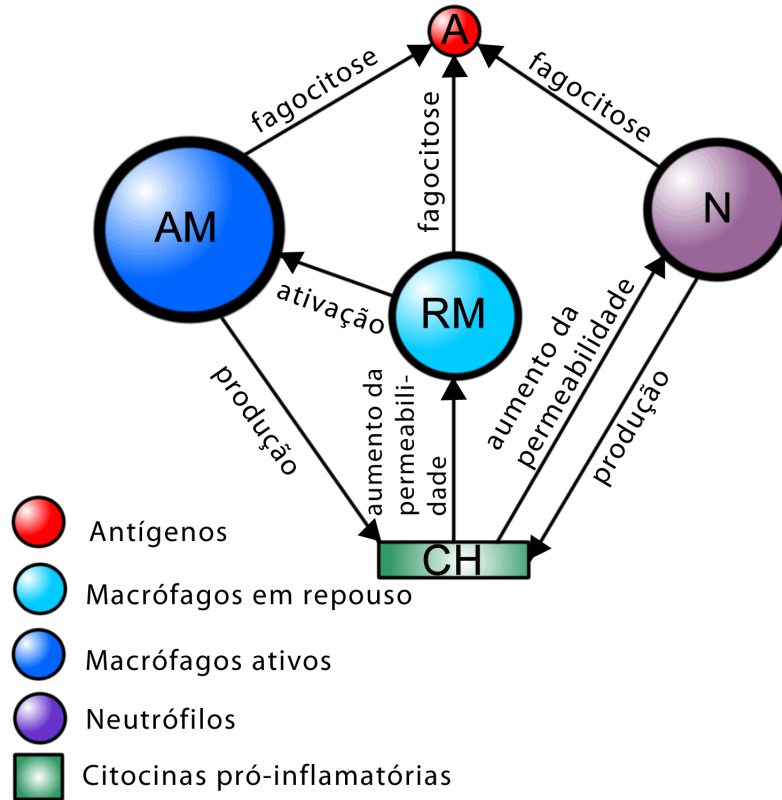


Figura 3.1: Relacionamentos entre os componentes.

citocinas pró-inflamatórias,  $\text{TNF-}\alpha$  e  $\text{IL-8}$ , são modeladas por um única equação no modelo. A dinâmica do LPS é modelada pela Equação 3.1.

$$\begin{cases} \frac{dA}{dt} = -\mu_A A - (\lambda_{N|A} \cdot N + \lambda_{MA|A} \cdot MA + \lambda_{MR|A} \cdot MR) \cdot A \\ A(0) = A_0 \end{cases} \quad (3.1)$$

O termo  $\mu_A A$  modela a taxa de decaimento do LPS no tecido, onde  $\mu_A$  é a taxa de decaimento. O termo  $-(\lambda_{N|A} \cdot N + \lambda_{MA|A} \cdot MA + \lambda_{MR|A} \cdot MR) \cdot A$  modela a fagocitose de LPS por macrófagos e neutrófilos, onde  $\lambda_{N|A}$  é a taxa de fagocitose por neutrófilos,  $\lambda_{MA|A}$  é a taxa de fagocitose por macrófagos ativos, e  $\lambda_{MR|A}$  é a taxa de fagocitose por macrófagos em repouso.

A Equação 3.2 modela a dinâmica dos neutrófilos.

$$\begin{cases} P_N = (P_N^{max} - P_N^{min}) \cdot \frac{CH}{CH + K_{CH}} + P_N^{min} \\ S_N = P_N \cdot (N^{max} - N) \\ \frac{dN}{dt} = -\mu_N N - \lambda_{A|N} A \cdot N + S_N \\ N(0) = N_0 \end{cases} \quad (3.2)$$

O termo  $P_N$  utiliza a equação de Hill [25] para modelar como a permeabilidade do endotélio dos vasos sanguíneos depende da concentração local de citocinas. Equações de Hill também são usadas, por exemplo, para modelar relações dose-resposta para o uso de medicamentos [26]. A ideia é modelar o aumento da permeabilidade do endotélio de acordo com a concentração de citocinas pró-inflamatórias. Na equação de Hill,  $P_N^{max}$  representa a taxa máxima do aumento de permeabilidade do endotélio para neutrófilos induzidos pela citocina pró-inflamatória;  $P_N^{min}$  representa a taxa mínima de aumento de permeabilidade induzida pelas citocinas; e  $K_{CH}$  é a concentração de citocinas pró-inflamatórias que exercem 50% do efeito máximo no aumento da permeabilidade. O termo  $\mu_N N$  modela a apoptose (morte programada [27]) de neutrófilos, onde  $\mu_N$  é a taxa de apoptose. O termo  $\lambda_{A|N} A \cdot N$  modela a apoptose de neutrófilos induzida pela fagocitose, onde  $\lambda_{A|N}$  representa a taxa de indução de apoptose. Já  $S_N$  representa a fonte de neutrófilos, isto é, a quantidade de neutrófilos que entram no tecido infectado por meio dos vasos sanguíneos. Esta quantidade é dependente da permeabilidade do endotélio ( $P_N$ ) e da capacidade do tecido de suportar a entrada de neutrófilos ( $N^{max}$ ), que também demonstra a concentração de neutrófilos no sangue.

As citocinas pró-inflamatórias são descritas pela Equação 3.3:

$$\begin{cases} \frac{dCH}{dt} = -\mu_{CH} CH + (\beta_{CH|N} N + \beta_{CH|MA} MA) \cdot A \cdot (1 - \frac{CH}{I_{CH}}) \\ CH(0) = CH_0 \end{cases} \quad (3.3)$$

O termo  $\mu_{CH} CH$  modela o decaimento das citocinas, onde  $\mu_{CH}$  é a taxa de decaimento. O termo  $(\beta_{CH|N} N + \beta_{CH|MA} MA) \cdot A$  modela a produção de citocinas pró-inflamatórias por neutrófilos e macrófagos ativos, onde  $\beta_{CH|N}$  e  $\beta_{CH|MA}$  são as taxas de produção por neutrófilos e macrófagos, respectivamente. Por fim, o termo  $I_{CH}$  representa a concentração máxima que o tecido suporta de citocinas pró-inflamatórias.

A Equação 3.4 apresenta a dinâmica dos macrófagos em repouso:

$$\begin{cases} P_{MR} = (P_{MR}^{max} - P_{MR}^{min}) \cdot \frac{CH}{CH + K_{CH}} + P_{MR}^{min} \\ S_{MR} = P_{MR} \cdot (M^{max} - (MR + MA)) \\ \frac{dMR}{dt} = -\mu_{MR}MR - \lambda_{MR|A} \cdot MR \cdot A + S_{MR} \\ MR(0) = MR_0 \end{cases} \quad (3.4)$$

O termo  $P_{MR}$  modela como a permeabilidade do endotélio dos vasos sanguíneos em relação aos macrófagos depende da concentração local de citocinas pró-inflamatórias. O termo  $\mu_{MR}MR$  modela a apoptose de macrófagos em repouso, onde  $\mu_{MR}$  é a taxa de apoptose. E  $\lambda_{MR|A}$  é a taxa de ativação em que macrófagos em repouso se tornam ativos após fagocitar LPS.

Por último, temos a dinâmica dos macrófagos ativos apresentada pela Equação 3.5.

$$\begin{cases} \frac{dMA}{dt} = -\mu_{MA}MA + \lambda_{MR|A} \cdot MR \cdot A \\ MA(0) = MA_0 \end{cases} \quad (3.5)$$

O termo  $\mu_{MA}MA$  modela a apoptose de macrófagos ativos, e  $\mu_{MA}$  é a taxa de apoptose.

## 3.2 Modelo EDP

As principais diferenças do modelo EDP em relação ao modelo EDO e à EDP proposta em [3] são:

- acrescentou-se no modelo espaço em forma de hexaedro (3D) ao invés de linha (1D);
- as condições iniciais e de borda foram adaptadas para o modelo 3D; e
- a difusão da quimiotaxia é aplicada em três direções:  $x$ ,  $y$  e  $z$ .

A equação que representa a dinâmica do LPS é dada pela Equação 3.6:

$$\begin{cases} \frac{\partial A}{\partial t} = -\mu_A A - (\lambda_{N|A}N + \lambda_{MA|A}MA + \lambda_{MR|A} \cdot MR) \cdot A + D_A \Delta A \\ A(x, y, z, 0) = A_0, \frac{\partial A(\dots, t)}{\partial n} |_{\partial \Omega} = 0 \end{cases} \quad (3.6)$$

Em relação ao modelo temporal, foi incluído o termo  $D_A \Delta A$ , que modela a difusão do LPS no tecido, onde  $D_A$  é o coeficiente de difusão.  $D_A \Delta A$  é o termo que representa a

difusão da concentração de LPS, onde  $\Delta$  é o operador Laplaciano (divergente do gradiente [28] -  $\nabla \cdot \nabla$ )

Já a Equação 3.7 representa a dinâmica dos neutrófilos no tecido tridimensional:

$$\begin{cases} P_N = (P_N^{max} - P_N^{min}) \cdot CH / (CH + K_{CH}) + P_{min} \\ S_N = P_N \cdot (N^{max} - N) \\ \frac{\partial N}{\partial t} = -\mu_N N - \lambda_{A|N} A \cdot N + S_N + D_N \Delta N - \nabla \cdot (\chi_N N \nabla CH) \\ N(x, y, z, 0) = N_0, \frac{\partial N(\dots, t)}{\partial n} |_{\partial \Omega} = 0 \end{cases} \quad (3.7)$$

Novamente, a novidade nesta equação em relação a equação equivalente da seção anterior é o termo  $D_N \Delta N$  que modela a difusão dos neutrófilos no tecido, onde  $D_N$  é o coeficiente de difusão e  $\Delta$  o operador Laplaciano. Outro novo termo aplicado foi  $\nabla \cdot (\chi_N N \nabla CH)$ , que modela o processo de quimiotaxia dos neutrófilos, onde  $\chi_N$  representa a taxa de quimiotaxia.

As citocinas pró-inflamatórias são descritas pela Equação 3.8:

$$\begin{cases} \frac{\partial CH}{\partial t} = -\mu_{CH} CH + (\beta_{CH|N} \cdot N + \beta_{CH|MA} \cdot MA) \cdot A \cdot (1 - CH / I_{CH}) + D_{CH} \Delta CH \\ CH(x, y, z, 0) = CH_0, \frac{\partial CH(\dots, t)}{\partial n} |_{\partial \Omega} = 0 \end{cases} \quad (3.8)$$

O termo  $D_{CH} \Delta CH$  é a novidade da equação, modelando a difusão das citocinas, onde  $D_{CH}$  é o coeficiente de difusão.

Os macrófagos em repouso são representados pela Equação 3.9:

$$\begin{cases} P_{MR} = (P_{MR}^{max} - P_{MR}^{min}) \cdot \frac{CH}{CH + K_{CH}} + P_{MR}^{min} \\ S_{MR} = P_{MR} \cdot (M^{max} - (MR + MA)) \\ \frac{\partial MR}{\partial t} = -\mu_{MR} MR - \lambda_{MR|A} \cdot MR \cdot A + S_{MR} + D_{MR} \Delta MR \\ \quad - \nabla \cdot (\chi_{MR} MR \nabla CH) \\ MR(x, y, z, 0) = MR_0, \frac{\partial MR(\dots, t)}{\partial n} |_{\partial \Omega} = 0 \end{cases} \quad (3.9)$$

Novamente, as novidades são os termos  $D_{MR} \Delta MR$ , que modela a difusão dos macrófagos em repouso, onde  $D_{MR}$  é o coeficiente de difusão; e  $\nabla \cdot (\chi_{MR} MR \nabla CH)$ , que modela a quimiotaxia dos macrófagos em repouso, onde  $\chi_{MR}$  é a taxa de quimiotaxia.

Por fim, temos a Equação 3.10, que modela os macrófagos ativos:

$$\begin{cases} \frac{\partial MA}{\partial t} = -\mu_{MA}MA + \lambda_{MR|A} \cdot MR \cdot A + D_{MA}\Delta MA - \nabla \cdot (\chi_{MA}MA\nabla CH) \\ MA(x, y, z, 0) = MA_0, \frac{\partial MA(\dots, t)}{\partial n} |_{\partial\Omega} = 0 \end{cases} \quad (3.10)$$

O termo  $D_{MA}\Delta MA$  modela a difusão de macrófagos ativos,  $D_{MA}$  é o coeficiente de difusão. E a expressão  $\nabla \cdot (\chi_{MA}MA\nabla CH)$  modela a quimiotaxia dos macrófagos ativos, onde  $\chi_{MA}$  é a taxa de quimiotaxia. Os demais termos são equivalentes aos termos da seção anterior.

# 4 GPGPUs: ARQUITETURA E PROGRAMAÇÃO

A simulação de modelos matemático-computacionais relacionados ao sistema imunológico normalmente acarreta um grande custo computacional. Implementações sequenciais podem demorar até dias executando determinadas simulações, o que leva a uma enorme queda de produtividade nas pesquisas. Neste contexto, a programação paralela surge como uma ferramenta essencial na redução do tempo necessário para a execução das simulações. Este capítulo apresenta a contextualização relacionada especificamente à programação paralela por meio de unidades de processamento gráfico de propósito geral (GPGPUs - *General-Purpose computation on Graphics Processing Units* ou simplesmente GPUs). As GPUs possuem multi-processadores capazes de realizar uma grande quantidade de processamento de dados simultaneamente. A programação em GPUs era algo difícil de se realizar até há pouco tempo, pois esta era uma unidade voltada apenas para processamento gráfico. Porém, recentemente foram desenvolvidas interfaces para linguagens de programação de alto nível, como C, C++ e Fortran, a fim de facilitar a utilização das GPUs para computação de uso geral; daí a origem do termo "propósito geral", criado por Mark Harris [29], na sigla GPGPUs. Uma das arquitetura atualmente disponíveis para o desenvolvimento de códigos em GPUs é a arquitetura NVIDIA's CUDA.

## 4.1 NVIDIA's CUDA

A arquitetura NVIDIA's CUDA (*Compute Unified Device Architecture*) [2] é uma das plataformas mais populares para desenvolvimento de aplicações em GPUs. CUDA provê ferramentas de desenvolvimento e bibliotecas em diversas linguagens, como C, que conseguem esconder do programador a arquitetura interna das GPUs[30], facilitando assim o desenvolvimento de aplicações.

Para executar uma aplicação em CUDA, o programador deve criar uma função paralela chamada *kernel*. Um *kernel* é uma função que pode ser chamada na CPU mas que é executada simultaneamente por várias *threads*, onde cada *thread* é executada fisicamente

em um dos muitos *stream processors* presentes na GPU. Assim, para aproveitar o máximo do desempenho oferecido pelo hardware, o programador deve subdividir seu problema em várias tarefas mais simples que possam ser executadas em paralelo por distintas *threads*. As *threads* são agrupadas abstratamente em blocos de *threads*, ou apenas blocos. Cada bloco pode ser formado por grupos de *threads* organizadas abstratamente em uma, duas ou três dimensões. Assim como as *threads*, os blocos também podem ser agrupados em grades, que podem ser organizados em uma ou duas dimensões. Uma esquematização desta distribuição de blocos e de *threads* pode ser visualizada na Figura 4.1. A arquitetura CUDA escalona automaticamente os blocos de *threads* entre os núcleos da GPU, como ilustra a Figura 4.2.

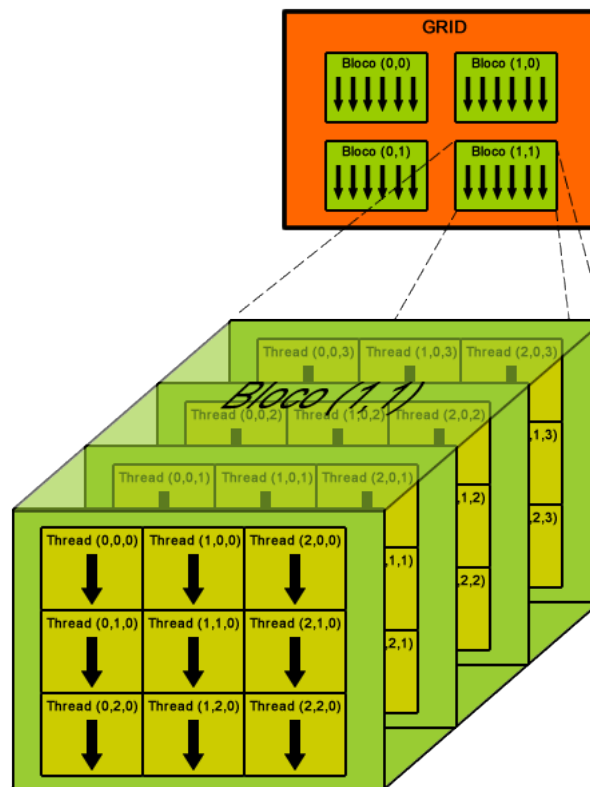


Figura 4.1: Representação de blocos em uma grade de duas dimensões e de *threads* em blocos organizados em três dimensões. Adaptado de [2].

Em virtude da hierarquia de *threads* e de blocos, a memória em CUDA é organizada em três grupos distintos: memória local, memória compartilhada e memória global. A memória local, composta por registradores, é associada a cada *stream processor*, sendo portanto de acesso exclusivo à cada *thread*. A memória compartilhada pode ser acessada por todas as *threads* que pertençam ao mesmo bloco durante todo o ciclo de vida desse bloco. Já a memória global permite a leitura e escrita por qualquer *thread* que executa



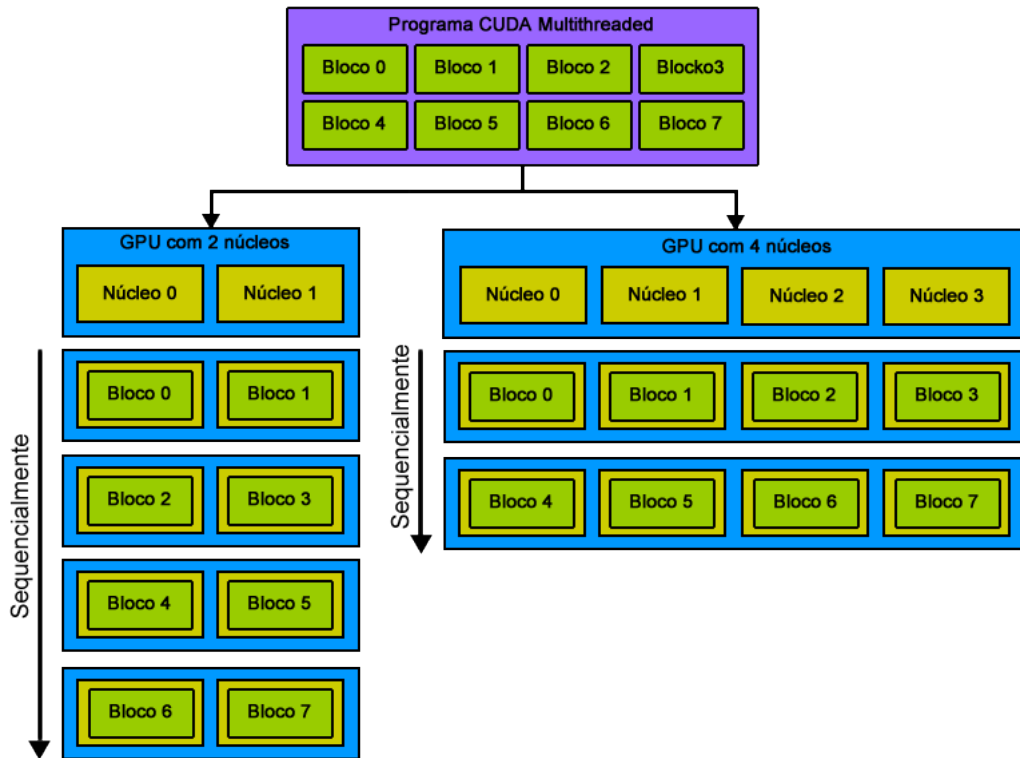


Figura 4.2: Distribuição de blocos de *threads* em núcleos da GPU. Adaptado de [2].

em qualquer bloco. Esta é a memória que pode ser acessada diretamente pela CPU. Do ponto de vista do tempo de acesso, a memória local possui o menor tempo de acesso, seguida pela memória compartilhada e por fim a memória global. A Figura 4.3 ilustra mais claramente essa distribuição de memória.

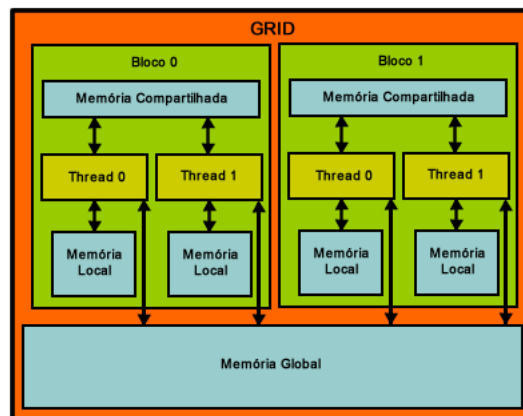


Figura 4.3: Hierarquia de Memória. Adaptado de [2].

## 4.2 Processo de Execução em CUDA

Quando a CPU realiza a chamada do *kernel*, ela deve especificar quantas *threads* deverão ser criadas em tempo de execução. A sintaxe utilizada para a execução do *kernel* é normalmente chamada de **configuração de execução**. Esta configuração é flexível ao ponto de suportar as hierarquias do CUDA: *threads*, blocos de *threads* e grades. Todas as *threads* em um bloco executam sempre o mesmo código, porém, um conjunto de identificadores numéricos é utilizado para distinguir uma *thread* de outra, propiciando que dados distintos sejam computados por *threads* distintas. Este conjunto de identificação é representado por duas únicas coordenadas chamadas *blockId* e *threadId*, associadas automaticamente a cada *thread* pelo sistema de execução CUDA. Estas duas variáveis podem ser acessadas apenas dentro das funções *kernel*; em conjunto, elas compõem um único valor que identifica cada *thread* de maneira unívoca.

O programador deve seguir alguns passos para utilizar a GPU:

- a GPU, também chamada de dispositivo, deve ser inicializada;
- a memória deve ser alocada na GPU e os dados que serão computados devem ser transferidos para ela;
- o *kernel* pode então ser chamado. Depois que o *kernel* termina de executar, os resultados devem ser transferidos de volta para a memória principal, de modo que possam ser utilizados por códigos executando na CPU.

# 5 MÉTODOS

Este capítulo apresenta os principais conceitos utilizados na implementação dos sistemas temporal e espacial. A primeira seção apresenta a análise de sensibilidade e descreve o método adotado neste trabalho para a sua implementação em uma GPU. Já a segunda seção trata do sistema espacial e a sua implementação em GPUs. Por fim, a terceira seção apresenta as técnicas adotadas para se reduzir ainda mais o tempo de execução de ambos os sistemas em GPUs.

## 5.1 Análise de Sensibilidade

Esta seção é dividida em três subseções. A primeira explica o que é a análise de sensibilidade; a segunda aborda o método adotado neste trabalho para realizar a análise de sensibilidade. Por fim, a última subseção trata da sua implementação em GPUs.

### 5.1.1 Principais Conceitos sobre Análise de Sensibilidade

A análise de sensibilidade consiste na avaliação dos impactos causados por variações dos parâmetros e condições iniciais de um modelo em relação às suas variáveis dependentes [31]. Um parâmetro é considerado sensível se ele causar uma drástica mudança na resposta final do problema após sofrer uma pequena mudança em seu valor inicial. Caso contrário, esta variável independente causará pouco impacto no modelo. A análise de sensibilidade é utilizada para prover o conhecimento de quais parâmetros são mais relevantes no modelo matemático, de tal modo que estes parâmetros sensíveis tenham seus valores iniciais cuidadosamente estipulados.

Modelos matemáticos podem oferecer uma vasta e complexa rede de interação entre uma grande quantidade de parâmetros. Deste modo, a análise de sensibilidade tem como objetivo encontrar uma pequena lista de parâmetros que realmente são importantes para o sistema, provocando uma alteração do estado final em questão (chamado de *Occam's razor*). Esta lista é necessária para que o modelo em questão possa ser simplificado por eliminação de partes que aparentem ser irrelevantes para o sistema [31]. Por outro lado, estes parâmetros também podem ser estudados a fim de que se tornem claros os

comportamentos anormais do sistema caso alguns destes, em condições extremamente específicas, sejam alterados. Porém, vale ressaltar que variáveis sensíveis também são importantes para identificar problemas ocultos no modelo, que aparentemente pode estar exibindo resultados realistas mas com dados erroneamente estipulados. Assim sendo, o estudo das variáveis presentes no modelo é importante para que se obtenha um sistema mais preciso e próximo ao sistema real modelado.

Há diferentes métodos para realizar a análise de sensibilidade. O mais utilizado, o OAT (*one-factor-at-a-time*), parcialmente adotado neste trabalho, baseia-se na modificação de apenas um parâmetro do modelo por vez, sendo os demais mantidos em seus valores originais. São então realizados os cálculos e as análises dos impactos ocasionados por esta única alteração. A vantagem de uma análise OAT é que, caso ocorra um erro no modelo, o parâmetro responsável pelo erro é facilmente descoberto [31], sem contar a simplicidade para a sua implementação computacional.

### 5.1.2 Método

Para realizar a análise de sensibilidade das EDOs, o seguinte método foi utilizado. Aplicou-se uma pequena mudança em cada parâmetro, criando, a cada mudança, um novo conjunto de parâmetros. Assim, o resultado das equações foi recomputado para cada novo conjunto. Este processo foi repetido inúmeras vezes, de tal modo que todas as combinações de valores distintos de parâmetros e condições iniciais fossem avaliadas. Os parâmetros foram ajustados de -100% a + 100% de seus valores padrões, em passos de 2%, exceto para alguns parâmetros, que também foram ajustados de -100% a + 100%, porém com passos de 20%. Já para as condições iniciais, foram estipuladas 100 variações com valores independentes para cada variável dependente. A combinação de todos os conjuntos diferentes de parâmetros por cada condição inicial forneceu um total de 469.145 sistemas de EDOs. Esse valor foi obtido da seguinte forma:  $(5 * 101) * (8 * 101 + 11 * 11)$ , onde as variáveis dependentes que foram modificadas, 5 no total, foram multiplicadas pela quantidade de variações avaliadas, 101 (1 valor padrão, mais 50 ajustes que reduziram o valor padrão e outros 50 que o aumentaram). A segunda parte da equação representa os parâmetros que sofreram alterações. Parte deles (8) sofreu alterações em passos de 2%, levando assim a um total de 101 variações em seus valores. Já outra parte dos parâmetros (11) sofreu alterações em passos de 20%, o que leva a um total de 11 variações testadas.

A Tabela A.1, presente no apêndice deste trabalho, apresenta as condições iniciais e os parâmetros padrões usados na simulação de todos os casos.

### 5.1.3 Implementação em GPUs

O código sequencial que implementa a análise de sensibilidade foi elaborado utilizando a linguagem C. O método explícito de Euler foi usado para implementar a solução numérica do sistema de EDOs devido a sua simplicidade. Contudo, como o número de sistemas de EDOs é muito grande, calculá-los sequencialmente é uma tarefa árdua, que demanda um grande tempo de execução. Para resolver esse problema, foi desenvolvido um código paralelo utilizando GPGPUs.

Para compreender a distribuição de cada cenário gerado durante a análise de sensibilidade pelas *threads* CUDA, deve-se relembrar a forma como as *threads* CUDA são organizadas na GPU: *threads* são agrupadas em grupos chamados blocos, e os distintos blocos são agrupados em grades. Blocos de *threads* podem ser organizados em três dimensões distintas, enquanto os blocos podem ser organizados em grupos de duas dimensões.

As dimensões da grade e dos blocos adotadas para realizar a distribuição de cenários são unidimensionais, ou seja, a grade é representada por um vetor de blocos e cada bloco é representado por um vetor de *threads*. A Figura 5.1 apresenta esta estruturação.

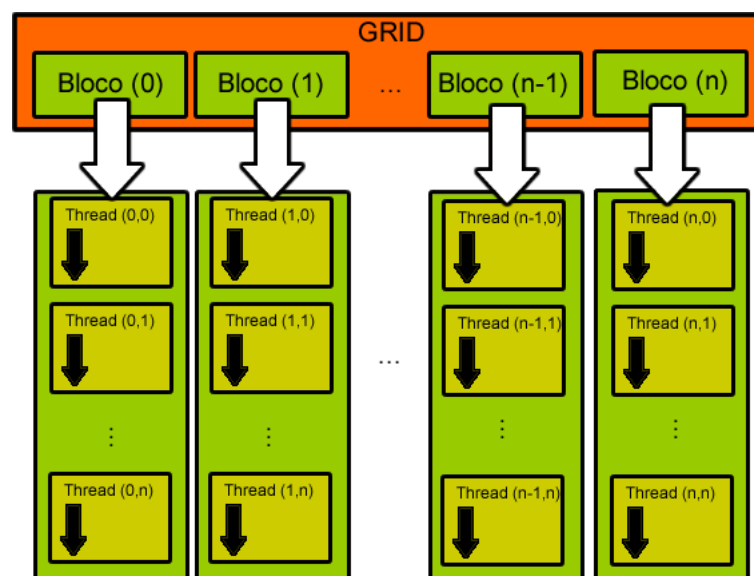


Figura 5.1: Organização adotada para estruturar a grade e os blocos.

Pode-se observar na Figura 5.1 que as *threads* possuem um identificador único

composto por dois valores: identificador do bloco ao qual pertence e seu identificador interno dentro do bloco. A Equação 5.1 mostra como um identificador unívoco para cada *thread*, chamado neste trabalho por *tid*, pode ser calculado a partir destes dois valores. Tal identificador único facilita a distribuição de cenários entre as *threads*.

$$tid = idInternoThread + (idDoBloco.tamanhoDoBloco) \quad (5.1)$$

A Figura 5.2 apresenta a mesma distribuição exibida pela Figura 5.1, porém com a numeração das *threads* utilizando o *tid*. Por exemplo, considerando-se que o tamanho dos blocos seja de 512 *threads*, a *thread* identificada pela tupla (1,1), ou seja, a *thread* de identificador 1 do bloco 1, seria representada pelo *tid* 513.

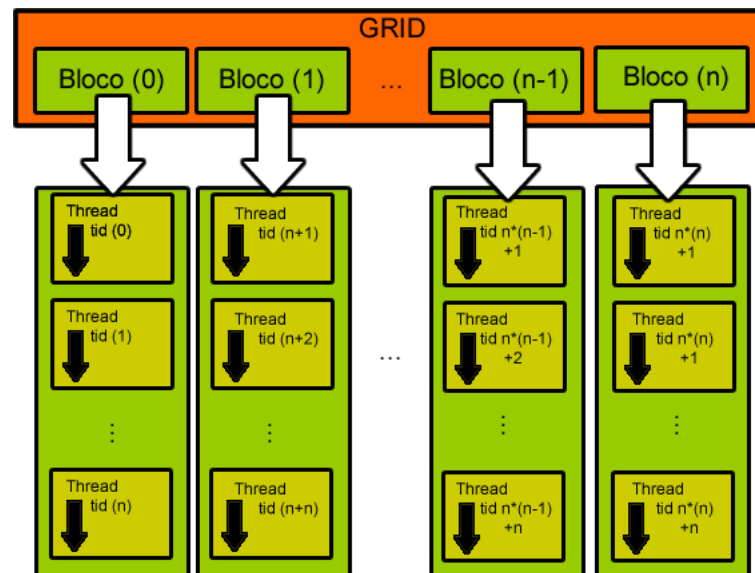


Figura 5.2: Identificadores das *threads*.

A vantagem do uso do identificador único de *thread*, *tid*, é que este permite que se desconsidere a existência dos blocos, de modo que todos os cenários são distribuídos a partir do valor de *tid*. A Figura 5.3 exhibe, simplificadamente, como foram distribuídos os 469.145 cenários entre as *threads*.

A implementação da análise de sensibilidade foi então dividida em duas etapas, implementadas em dois *kernels* distintos. A primeira etapa atribui a cada *thread* a tarefa de ajudar na criação de todas as combinações de condições iniciais e parâmetros para realizar a análise de sensibilidade. Cada *thread* deve, de acordo com o seu identificador, aplicar determinados valores percentuais de ajustes em determinadas condições iniciais ou parâmetros, de modo que todas as combinações de valores a serem avaliados na análise

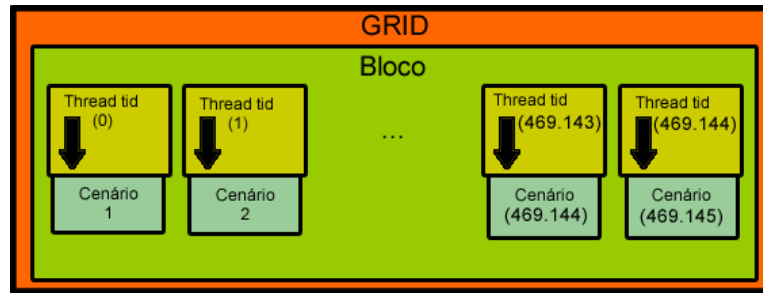


Figura 5.3: Distribuição de cenários por *threads*.

de sensibilidade são criados em paralelo. Um exemplo fictício, apresentado pela Figura 5.4, ilustra como foi elaborada essa organização do vetor na memória global da GPU por cenário/*thread*. Neste exemplo, foi considerado um modelo que possui cinco variáveis dependentes que sofrem 10 ajustes na condição inicial cada e 10 parâmetros que sofrem, também, 10 ajustes cada. Verifica-se que cada cenário é composto por um único ajuste na condição inicial em conjunto com um único ajuste em algum parâmetro.

Todos os cenários criados são então armazenados, ao final da execução da primeira etapa, em um vetor localizado na memória global da GPU. Porém a organização destes cenários no vetor não é uma tarefa trivial, visto que a melhor largura de banda no acesso aos dados localizados na memória global da GPU é obtida apenas quando dados acessados por distintas *threads* em um único multiprocessador estão localizados em endereços consecutivos e alinhados na memória e podem assim ser acessados de modo simultâneo. Neste caso, o acesso é dito agrupado e realizado em uma única transação de memória. Portanto, este aspecto deve ser observado na organização dos dados no vetor para que um melhor desempenho seja obtido na execução do *kernel*.

A segunda etapa da implementação da análise de sensibilidade consiste da resolução das EDOs para cada um dos cenários criados na etapa anterior. Novamente, cada *thread*, baseada em seu identificador, acessa uma posição única do vetor na memória global que armazena os dados do cenário sob sua responsabilidade, calcula o resultado e o armazena em uma posição específica do vetor. Deve-se ressaltar que não é necessária nenhuma comunicação entre as *threads*, tanto para a geração dos cenários quanto para a sua resolução.

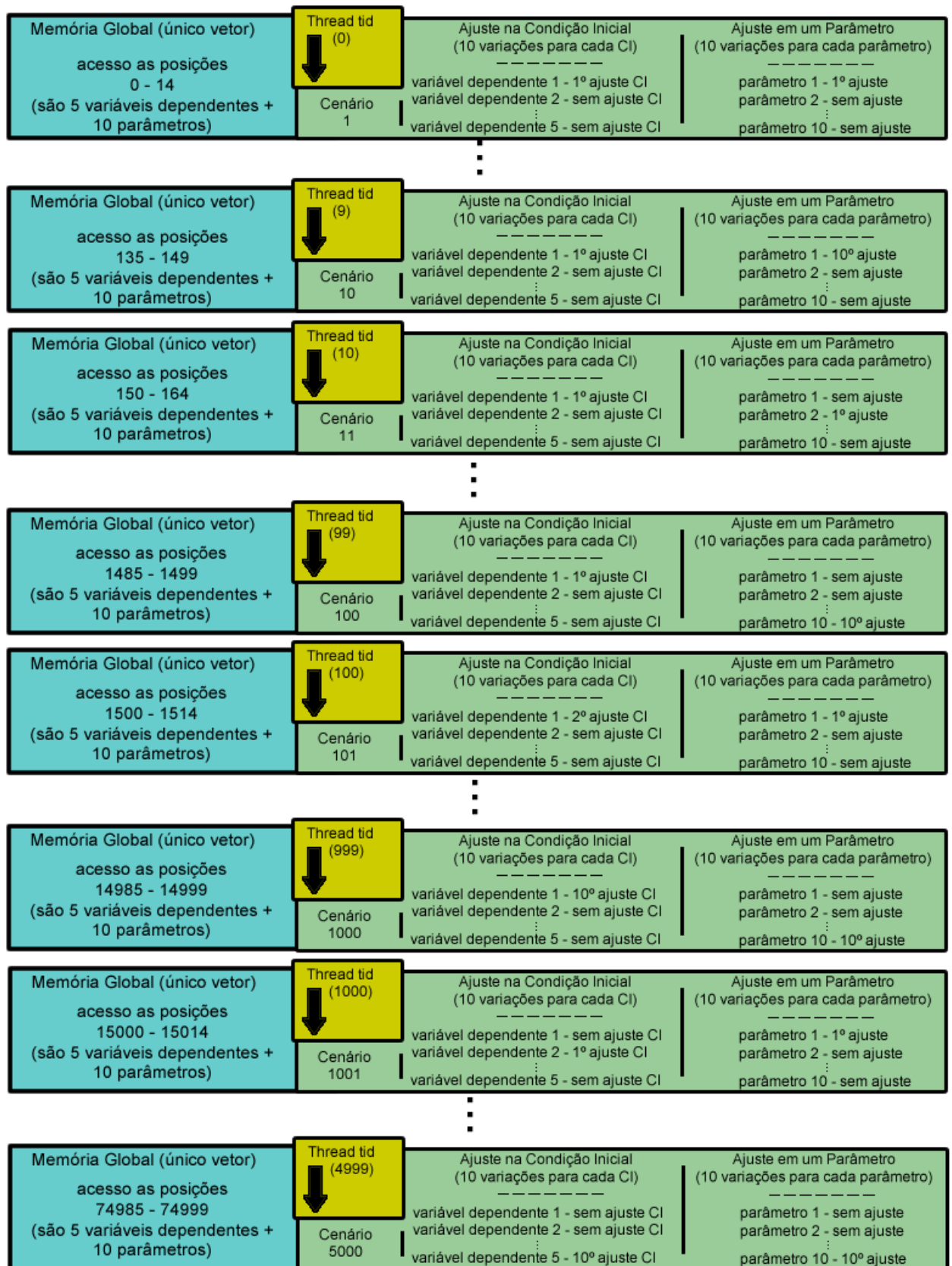


Figura 5.4: Distribuição de cenários e acessos à memória pelas *threads*.



## 5.2 Modelo 3D

Esta seção apresenta os conceitos utilizados na simulação computacional do modelo do SIH inato em uma seção em três dimensões do tecido, bem como a sua implementação paralela.

### 5.2.1 Modelo Computacional

O método numérico adotado na implementação do modelo matemático foi o método de diferenças finitas [32], trivialmente utilizado para discretização numérica de EDPs. O método de diferenças finitas é um processo para resolução de equações diferenciais que se baseia na aproximação de derivadas através de diferenças finitas. Nas simulações realizadas foram usados os valores temporais e espaciais listados na Tabela 5.1, os demais valores de parâmetros e condições iniciais estão explicitadas nas Tabelas A.2 e A.3 no apêndice deste trabalho.

Tabela 5.1: Discretização temporal e espacial

Parâmetro	Tamanho	Discretização
Time	5dias, simbolizada por $5 \cdot 10^6$ iterações	$\Delta T = 0.000001$
X-axis	1mm, simbolizada por 11 pontos	$\Delta X = 0.1$
Y-axis	1mm, simbolizada por 11 pontos	$\Delta Y = 0.1$
Z-axis	1mm, simbolizada por 11 pontos	$\Delta Z = 0.1$

A Equação 5.2 exemplifica como o operador Laplaciano é aproximado por diferenças finitas. O operador Laplaciano simula o fenômeno da difusão no tecido 3D.

$$\begin{aligned}
 D_O \left( \frac{\partial^2 O(x, y, z)}{\partial x^2} + \frac{\partial^2 O(x, y, z)}{\partial y^2} + \frac{\partial^2 O(x, y, z)}{\partial z^2} \right) \approx \\
 D_O \cdot ((o_{x+1,y,z} - 2 \cdot o_{x,y,z} + o_{x-1,y,z}) / \Delta X^2) \\
 + D_O \cdot ((o_{x,y+1,z} - 2 \cdot o_{x,y,z} + o_{x,y-1,z}) / \Delta Y^2) \\
 + D_O \cdot ((o_{x,y,z+1} - 2 \cdot o_{x,y,z} + o_{x,y,z-1}) / \Delta Z^2)
 \end{aligned} \tag{5.2}$$

Na Equação 5.2,  $O$  representa a discretização de algumas células como neutrófilos, macrófagos ativos e em repouso;  $D_O$  é o coeficiente de difusão destas mesmas células;  $x$ ,  $y$  e  $z$  são as posições no espaço; e  $\Delta X$ ,  $\Delta Y$  e  $\Delta Z$  são as discretizações do espaço relativos ao seu eixo.

O fluxo da quimiotaxia em três dimensões foi implementado de acordo com o seguinte código:

---

```

//fluxo no eixo X
if((CH[0][x][y][z]-CH[0][x-1][y][z]) > 0) {
    flux_left = -(CH[0][x][y][z]-CH[0][x-1][y][z]) * O[0][x-1][y][z]/deltaX;
} else {
5   flux_left = -(CH[0][x][y][z]-CH[0][x-1][y][z]) * O[0][x][y][z]/deltaX;
}
if(CH[0][x+1][y][z]-CH[0][x][y][z] > 0) {
    flux_right = (CH[0][x+1][y][z]-CH[0][x][y][z]) * O[0][x][y][z]/deltaX;
} else {
10  flux_right = (CH[0][x+1][y][z]-CH[0][x][y][z]) * O[0][x+1][y][z]/deltaX;
}
fluxX = (flux_left + flux_right)/deltaX;

//fluxo no eixo Y
15 if(CH[0][x][y][z]-CH[0][x][y-1][z] > 0) {
    flux_left = -(CH[0][x][y][z]-CH[0][x][y-1][z]) * O[0][x][y-1][z]/deltaY;
} else {
    flux_left = -(CH[0][x][y][z]-CH[0][x][y-1][z]) * O[0][x][y][z]/deltaY;
}
20 if(CH[0][x][y+1][z]-CH[0][x][y][z] > 0) {
    flux_right = (CH[0][x][y+1][z]-CH[0][x][y][z]) * O[0][x][y][z]/deltaY;
} else {
    flux_right = (CH[0][x][y+1][z]-CH[0][x][y][z]) * O[0][x][y+1][z]/deltaY;
}
25 fluxY = (flux_left + flux_right)/deltaY;

//fluxo no eixo Z
if(CH[0][x][y][z]-CH[0][x][y][z-1] > 0) {
    flux_left = -(CH[0][x][y][z]-CH[0][x][y][z-1]) * O[0][x][y][z-1]/deltaZ;
30 } else {
    flux_left = -(CH[0][x][y][z]-CH[0][x][y][z-1]) * O[0][x][y][z]/deltaZ;
}
if(CH[0][x][y][z+1]-CH[0][x][y][z] > 0) {
    flux_right = (CH[0][x][y][z+1]-CH[0][x][y][z]) * O[0][x][y][z]/deltaZ;
35 } else {
    flux_right = (CH[0][x][y][z+1]-CH[0][x][y][z]) * O[0][x][y][z+1]/deltaZ;
}

```

$$\text{fluxZ} = (\text{flux\_left} + \text{flux\_right})/\text{deltaZ};$$

Neste código,  $ch$  representa a discretização das citocinas pró-inflamatórias;  $O$  representa a discretização de algumas células do sistema; o valor 0 no vetor  $O$  indica tempo anterior;  $x$ ,  $y$  e  $z$  são as posições no espaço; e  $\text{deltaX}$ ,  $\text{deltaY}$  e  $\text{deltaZ}$  são as discretizações do espaço relativos ao seu eixo. A Figura 5.5 representa este processo; porém, foi considerado apenas o eixo X em sua representação. Repara-se que a a imagem demonstra todos os cenários possíveis em relação ao eixo X. Deve-se levar em consideração que a soma dos fluxos é multiplicada pela taxa de quimiotaxia ( $-\chi_O$ ) em negativo (apresentada na Equação 5.3) para que as concentrações de células migrem para o local de maior concentração de citocinas.

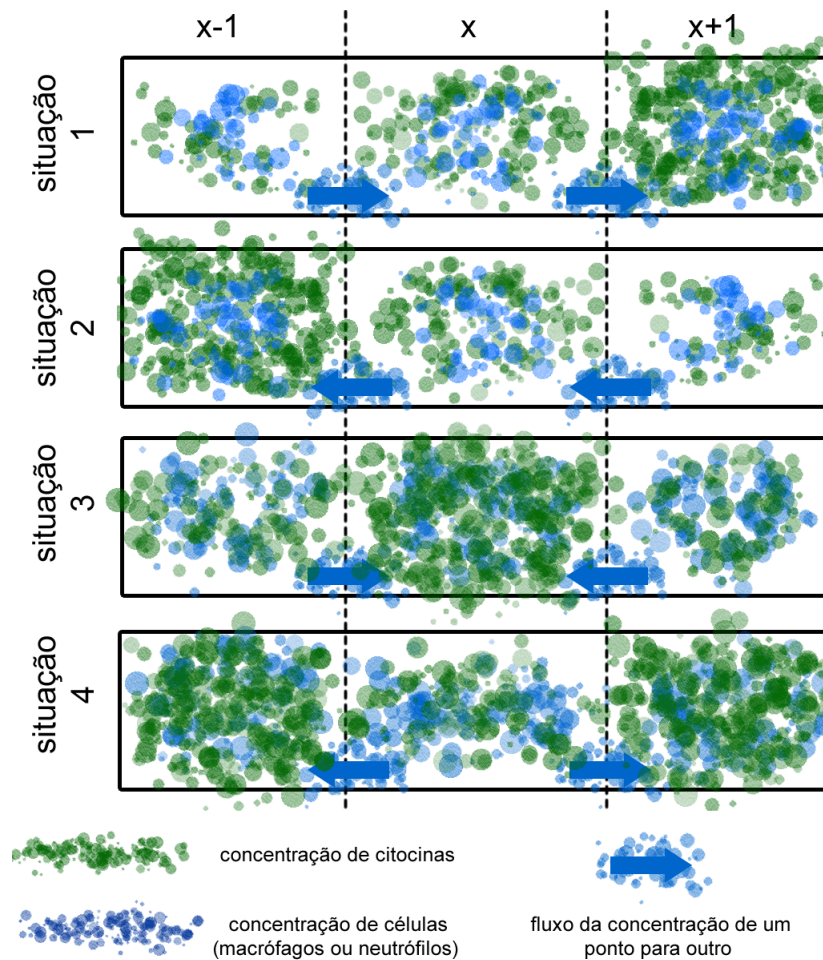


Figura 5.5: Representação do Fluxo exercido pelo processo de quimiotaxia.

$$-\nabla \cdot (\chi_O O \nabla CH) \approx -\chi_O \cdot (\text{fluxX} + \text{fluxY} + \text{fluxZ}) \quad (5.3)$$

O termo  $\nabla CH$  da Equação 5.3 indica a velocidade de movimento e  $\chi_O$  é a taxa de

quimiotaxia correspondente a cada célula. Foi adotado o esquema *upwind* de primeira ordem chamado FOU (*First-Order Upwind scheme*) [33] na implementação, bem como também foi considerado o fluxo como 0 nas bordas do sistema.

### 5.2.2 Implementação em GPUs

A versão paralela do código atribui a cada *thread* a responsabilidade de calcular todo o conjunto de equações do modelo EDP para um único ponto  $(x, y, z)$  do tecido tridimensional. Para a resolução das EDPs por diferenças finitas, cada *thread* deve acessar as informações relativas não só ao ponto sob sua responsabilidade, mas também as informações relativas dos pontos vizinhos calculados por outras *threads*. A Figura 5.6 descreve melhor essa distribuição.

A fim de se evitar sincronizações entre estas *threads*, foi implementado um *buffer* que permite a uma *thread*, no tempo  $t$ , acessar os dados calculados pelos seus vizinhos no tempo  $t-1$ . O *buffer* torna-se necessário já que não há possibilidade em CUDA de sincronizar *threads* que pertençam a blocos diferentes, apenas aquelas pertencentes ao mesmo bloco. O emprego do *buffer* elimina a necessidade de sincronização porque cada dado é escrito em um tempo  $t$  por uma única *thread*, aquela responsável pelo seu cálculo. Cada posição do *buffer* armazena, para cada ponto do espaço  $(x, y, z)$  e para cada variável calculada pela EDP, dois valores: um para os dados produzidos no passo de tempo anterior,  $t - 1$ , e o outro para os dados produzidos no passo de tempo atual,  $t$ . Os dois *buffers* trocam de papéis a cada passo de tempo com objetivo de evitar a cópia desnecessária de dados. O armazenamento das concentrações de cada célula/molécula para cada ponto no espaço é representado por um vetor unidimensional alocado tanto na memória da CPU, quanto na memória da GPU. O vetor possui tamanho igual a 5 (quantidade de variáveis presentes no sistema) vezes o número de posições no tecido vezes 2 (referentes ao tempo  $t - 1$  e  $t$ ).

## 5.3 Otimizações

### 5.3.1 Otimizações Adotadas na Análise de Sensibilidade

A programação via GPU utilizando CUDA permite ao usuário a otimização da execução do *kernel* por meio da definição de alguns parâmetros, como o tamanho ou a disposição

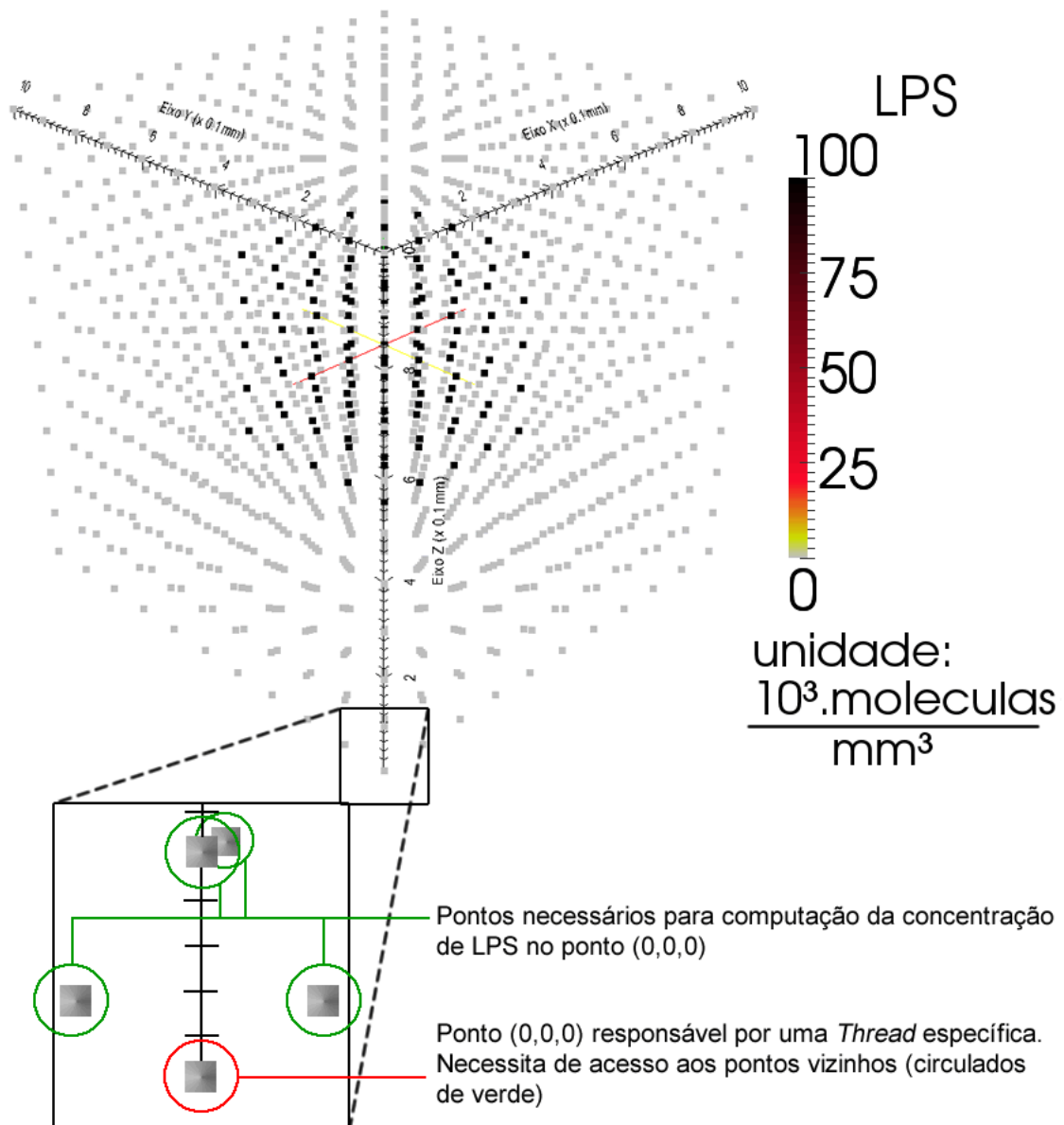


Figura 5.6: Distribuição de pontos por *thread* e acesso a vizinhos.

do bloco. O aumento do tamanho do bloco influencia no aumento da quantidade de blocos pertencentes a um *stream multiprocessor* [34]. Para gerenciar diversas *threads*, o multiprocessador utiliza uma arquitetura chamada SIMT (*Single Instruction Multiple Thread*). A unidade SIMT organiza as *threads* em grupos com tamanho especificado pela configuração de hardware, chamado de *warp*. Hoje em dia, as GPUs possuem uma *warp* de tamanho 32, ou seja, uma unidade SIMT trabalha fornecendo instruções a grupos de 32 *threads*. Se em um código CUDA necessita-se apenas de 30 *threads* para computação paralela, as 2 *threads* restantes, pertencentes a esta mesma *warp*, também serão executadas. Devido a este fator, foi escalonado em um bloco uma

quantidade de *threads* múltipla ao tamanho da *warp* da GPU utilizada, porém respeitando o limite máximo que um bloco suporta. Assim, evita-se que *threads* sejam executadas desnecessariamente.

### 5.3.2 Otimizações Adotadas no Sistema 3D

Esta seção apresenta as técnicas utilizadas no código paralelo com o objetivo de melhorar o seu desempenho. Inicialmente são descritas as medidas adotadas a fim de se diminuir o uso de ramificações dentro de um *kernel* CUDA. Na sequência são descritas as medidas adotadas para reduzir o acesso à memória global da GPU. Ambas as técnicas aqui descritas foram baseadas no trabalho originalmente publicado por Malits *et alli* [34].

#### 5.3.2.1 Diminuição de usos de Ramificações em um *Kernel*

O hardware que realiza o controle em um multiprocessador de uma GPU opera de maneira extremamente simples. Basicamente ele encaminha, para cada *warp* que executa em um multiprocessador, uma única instrução, que deve ser executada simultaneamente por todas estas *threads*. Contudo, durante a execução de cada *thread*, podem ocorrer divergências em seu fluxo de execução, ou seja, cada *thread*, dependendo do seu estado atual, pode optar por executar uma sequência diferente de código. Por exemplo, em uma sequência de código simples do tipo *if-else*, algumas *threads* podem avaliar a condição do *if* como verdadeira, levando a execução de um bloco de código, enquanto outras *threads* podem avaliar a mesma condição como falsa, levando-as a executar uma outra sequência de código. Contudo, devido às restrições SIMD (*Single Instruction, Multiple Data*), somente um caminho dentro de uma sequência de código pode ser executada por vez. Deste modo, algumas *threads* podem ter sua execução suspensa enquanto as outras *threads* estivessem ativas executando a sua sequência de código, e vice-versa. A Figura 5.7 ilustra esta situação. Por isto, o uso de ramificações em um *kernel* pode acarretar em uma grande queda de desempenho pelo fato de parte dos processadores ser desativada durante o tempo em que instruções que não deveriam executar estivessem sendo emitidas pelo controle do multiprocessador.

A fim de se evitar o uso excessivo de ramificações, foram realizadas algumas modificações na primeira versão do código do modelo tridimensional. O código abaixo apresenta um trecho desta primeira versão, responsável pela computação da difusão das

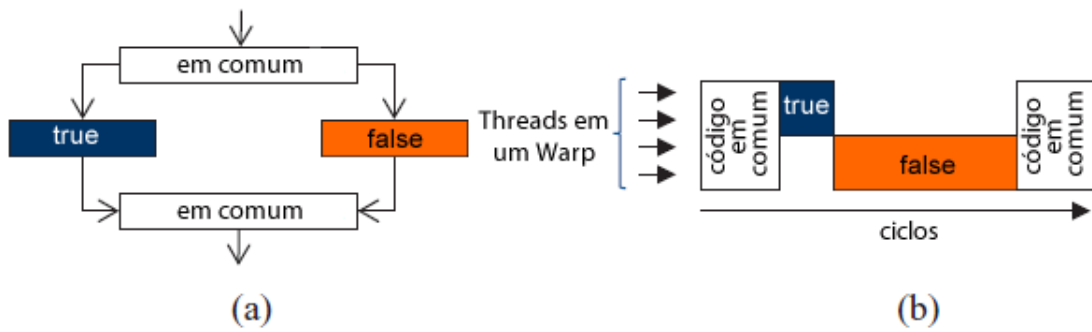


Figura 5.7: (a) divergências por ramificações e (b) queda de desempenho devido à serialização do código por causa da ramificação. Adaptado de [34].

células pelo tecido. Observa-se que foi necessário o uso de *IFs* para avaliar a condição de contorno em relação ao eixo X. Neste código, as variáveis  $x$ ,  $y$  e  $z$  representam a posição espacial sob a responsabilidade da *thread*,  $ESPACO\_X$  indica a quantidade de pontos presentes no eixo X (11 pontos),  $vec$  é o vetor global que armazena os valores no passo de tempo atual e anterior ( $tAntigo$ ) para todas as variáveis do modelo,  $equacao$  é uma variável que representa a concentração que está sofrendo a difusão,  $deltaX$  é a discretização e  $resX$  é o resultado final referente a difusão apenas do eixo X.

---

```

if (x == 0) {
    resX = (vec(tAntigo, equacao, x+1, y, z) - vec(tAntigo, equacao, x, y, z))
           / (deltaX * deltaX);
} else if (x == ESPACO_X - 1) {
    resX = (vec(tAntigo, equacao, x-1, y, z) - vec(tAntigo, equacao, x, y, z))
           / (deltaX * deltaX);
5 } else {
    resX = (vec(tAntigo, equacao, x+1, y, z) - 2 * vec(tAntigo, equacao, x, y, z)
           + vec(tAntigo, equacao, x-1, y, z)) / (deltaX * deltaX);
}

```

---

Apesar de correto, o processamento deste código pode levar a três ramificações distintas. Para melhorar o desempenho foram feitas algumas modificações no código de forma a fim de se eliminar o uso dos *IFs*. Como pode ser observado na primeira linha do próximo código, foi utilizada apenas uma única instrução que substitui os dois primeiros *IFs* do código anterior (referentes ao contorno na borda  $x == 0$  e  $x == ESPACO\_X - 1$ ). Todas as *threads* executam simultaneamente esta instrução.

---

```

resX = (((ESPACO_X - 1) - x) / (ESPACO_X - 1)) * ((vec(tAntigo, equacao, x+1, y, z) -
           vec(tAntigo, equacao, x, y, z)) / (deltaX * deltaX)) + (x / (ESPACO_X - 1)) * ((vec(

```

```

    tAntigo , equacao , x-1, y, z) - vec(tAntigo , equacao , x, y, z))/(deltaX*deltaX))
;
resX = (x > 0 && x < (ESPACO_X-1)) ? ((vec(tAntigo , equacao , x+1, y, z) -2 *
    vec(tAntigo , equacao , x, y, z) + (vec(tAntigo , equacao , x-1, y, z))/(deltaX*
    deltaX) : resX;

```

---

A segunda instrução do código avalia se, para aquela *thread* em particular, o valor de contorno na borda se aplica ou não. Caso se aplique, o valor calculado pela instrução anterior é mantido. Caso contrário, o novo valor é calculado da mesma forma que no código original. Contudo, deve-se ressaltar que a forma de avaliação neste segundo caso difere da avaliação e execução da instrução *IF*. Em particular, com o uso da expressão  $? :$ , a expressão  $(x > 0 \&\& x < (ESPACO\_X - 1))$  é avaliada simultaneamente por todas as *threads* em execução, sendo o valor de *resX* também atualizado simultaneamente por todas as *threads em execução*.

Por fim, podemos reestruturá-lo para que todo o código seja representado por uma única instrução. Para isso basta substituir *resX* após os  $:$  (que representa um *else*) na última linha do código anterior pela equação da primeira linha:

```

resX = (x > 0 && x < (ESPACO_X-1)) ? ((vec(tAntigo , equacao , x+1, y, z) -2 *
    vec(tAntigo , equacao , x, y, z) + (vec(tAntigo , equacao , x-1, y, z))/(deltaX*
    deltaX) : (((ESPACO_X-1)-x)/(ESPACO_X-1)) * ((vec(tAntigo , equacao , x+1, y,
    z) - vec(tAntigo , equacao , x, y, z))/(deltaX*deltaX)) + (x/(ESPACO_X-1)) *
    ((vec(tAntigo , equacao , x-1, y, z) - vec(tAntigo , equacao , x, y, z))/(deltaX*
    deltaX));

```

---

### 5.3.2.2 Redução do Número de Acessos à Memória Global

A única maneira de trocar informações entre CPU e GPU é através do armazenamento de variáveis globais nas GPUs. Porém, manter dados na memória global pode ocasionar um impacto negativo no desempenho da aplicação, uma vez que a latência no acesso aos dados localizados nesta memória é alta. Para que se reduza este custo, optou-se por reduzir o acesso à memória global através do armazenamento dos dados mais constantemente utilizados em registradores. Pode-se observar no código abaixo que o acesso ao vetor global *vec* nas posições  $(tAntigo, equacao, x, y, z)$  (linhas 2,4,6),  $(tAntigo, equacao, x + 1, y, z)$  (linhas 2 e 6) e  $(tAntigo, equacao, x - 1, y, z)$  (linhas 4 e 6) é realizado mais de uma vez ao longo da execução.



---

```

if(x == 0) {
    resX = (vec(tAntigo, equacao, x+1, y, z) - vec(tAntigo, equacao, x, y, z))
           /(deltaX*deltaX);
} else if(x == ESPACO_X-1) {
    resX = (vec(tAntigo, equacao, x-1, y, z) - vec(tAntigo, equacao, x, y, z))
           /(deltaX*deltaX);
5 } else {
    resX = (vec(tAntigo, equacao, x+1, y, z) -2 * vec(tAntigo, equacao, x, y, z)
           + vec(tAntigo, equacao, x-1, y, z))/(deltaX*deltaX);
}

```

---

A fim de se evitar o acesso repetido à memória global, os valores dessas posições do vetor foram solicitados apenas uma vez ao longo da execução do código, sendo armazenados em registradores, um espaço de memória de baixa latência de acesso exclusivo a cada *thread*. Os demais acessos às mesmas posições do vetor foram então substituídos por acessos aos registradores. O trecho de código abaixo apresenta a reestruturação proposta:

---

```

float _ori = vec(tAntigo, equacao, x, y, z);
float _xp1 = vec(tAntigo, equacao, x+1, y, z);
float _xm1 = vec(tAntigo, equacao, x-1, y, z);

5 if(x == 0) {
    resX = _xp1 - _ori)/(deltaX*deltaX);
} else if(x == ESPACO_X-1) {
    resX = (_xm1 - _ori)/(deltaX*deltaX);
} else {
10    resX = _xp1 -2 * _ori + _xm1)/(deltaX*deltaX);
}

```

---

## 6 RESULTADOS

Este capítulo apresenta os resultados obtidos pela análise de sensibilidade do modelo, bem como os oriundos da simulação do modelo 3D. Por fim, também são apresentados os ganhos, em termos de desempenho, obtidos pelo emprego de GPUs para realizar ambas as computações.

### 6.1 Análise de Sensibilidade

Para estudar a importância de algumas células, moléculas e processos da dinâmica do SIH inato, um conjunto de simulações foi realizado para distintos valores de parâmetros e condições iniciais. A Tabela A.1 presente no Apêndice apresenta as condições iniciais e os parâmetros padrões usados na simulação de todos os casos. Deve-se levar em consideração que o parâmetro estudado em cada caso de análise de sensibilidade irá sofrer um ajuste relacionado ao passo estipulado. Como já descrito, para realizar a análise de sensibilidade deve-se resolver um mínimo de 469.145 sistemas EDOs. Devido a quantidade considerável de resultados, este trabalho irá tratar apenas daqueles mais relevantes.

Deve-se também ressaltar que todos os casos analisados nesta seção utilizam dois conjuntos distintos de condições iniciais: um em que  $A_0 = 20 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$  e outro que  $A_0 = 40 \frac{10^3 \text{.moleculas}}{\text{mm}^3}$ . A análise de sensibilidade demonstra que dois parâmetros são de extrema importância para o modelo: a capacidade do tecido suportar novas entradas de neutrófilos ( $N^{max}$ ) e a taxa de fagocitose de LPS pelos neutrófilos ( $\lambda_{N|A}$ ). Também será apresentado um cenário em que a mudança dos valores do parâmetro não acarretou grandes alterações nos resultados das simulações: a taxa de fagocitose de LPS por macrófagos ativos ( $\lambda_{MA|A}$ ). O conjunto de EDOs da Equação 6.1 será considerado em todos os cenários descritos ao

longo desta seção.

$$\left\{ \begin{array}{l}
 \frac{dA}{dt} = -\mu_A A - (\lambda_{N|A} \cdot N + \lambda_{MA|A} \cdot MA + \lambda_{MR|A} \cdot MR) \cdot A \\
 A(0) = A_0 \\
 permeability_N = (P_N^{max} - P_N^{min}) \cdot \frac{CH}{CH + k_{eqch}} + P_N^{min} \\
 source_N = permeability_N \cdot (N^{max} - N) \\
 \frac{dN}{dt} = -\mu_N N - \lambda_{A|N} A \cdot N + source_N \\
 N(0) = N_0 \\
 permeability_{MR} = (P_{MR}^{max} - P_{MR}^{min}) \cdot \frac{CH}{CH + k_{eqch}} + P_{MR}^{min} \\
 source_{MR} = permeability_{MR} \cdot (M^{max} - (MR + MA)) \\
 \frac{dMR}{dt} = -\mu_{MR} MR - \lambda_{MR|A} \cdot MR \cdot A + source_{MR} \\
 MR(0) = MR_0 \\
 \frac{dMA}{dt} = -\mu_{MA} MA + \lambda_{MR|A} \cdot MR \cdot A \\
 MA(0) = MA_0 \\
 \frac{dCH}{dt} = -\mu_{CH} CH + (\beta_{CH|N} N + \beta_{CH|MA} MA) \cdot A \cdot (1 - \frac{CH}{ch_{Inf}}) \\
 CH(0) = CH_0
 \end{array} \right. \quad (6.1)$$

### 6.1.1 Parâmetro $N^{max}$

A capacidade do tecido de suportar a entrada de novos neutrófilos está diretamente relacionada à permeabilidade das células endoteliais (que revestem os vasos sanguíneos), tornando  $N^{max}$  um dos parâmetros mais sensíveis do modelo. Se uma mudança é feita para tornar o tecido mais permeável, então há uma maior migração de neutrófilos para o tecido inflamado. Um aumento na quantidade de neutrófilos no tecido tem muitas consequências. Primeiramente, muito mais LPS é fagocitado, ou seja, a quantidade de LPS cai mais rapidamente. Além disso, esta redução na quantidade de LPS também leva a uma redução na quantidade de macrófagos que se torna ativa. Por último, com o aumento no número de neutrófilos, citocinas pró-inflamatórias são produzidas mais rapidamente. Se a permeabilidade do tecido é reduzida, pode-se observar um comportamento inverso: o LPS demora mais tempo para ser eliminado, uma maior quantidade de macrófagos se torna ativa, e a produção de citocinas é mais suave ao longo dos dias, visto que o LPS é

eliminado em torno do quinto dia. As Figuras 6.1 a 6.5 ilustram esta situação. Pode-se observar que o LPS cai em uma velocidade maior quando há um aumento do valor do parâmetro  $N^{max}$ .

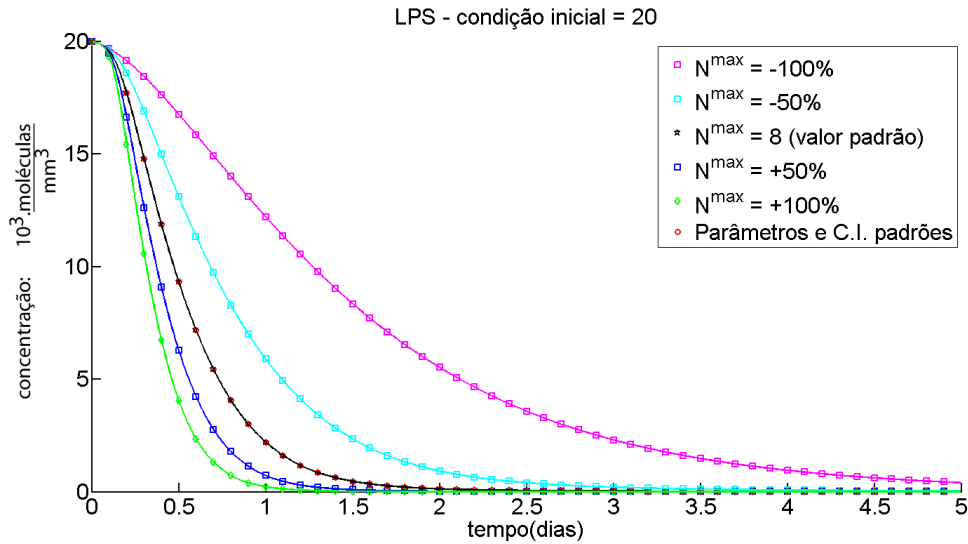


Figura 6.1: Evolução temporal de LPS quando  $A(0) = 20 \frac{10^3 \text{ moléculas}}{\text{mm}^3}$  para distintos valores de  $N^{max}$ .

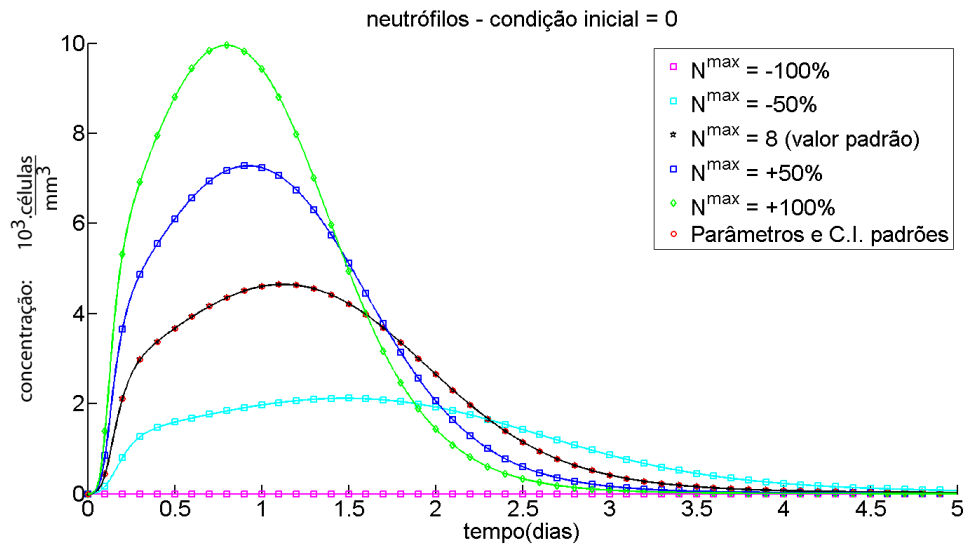


Figura 6.2: Evolução temporal de neutrófilos quando  $A(0) = 20 \frac{10^3 \text{ moléculas}}{\text{mm}^3}$  para distintos valores de  $N^{max}$ .

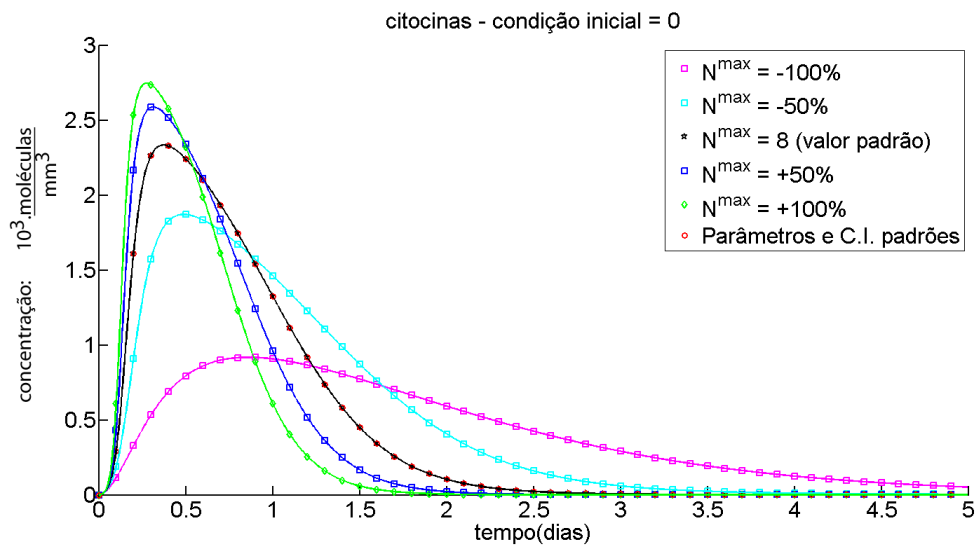


Figura 6.3: Evolução temporal de citocinas pró-inflamatórias quando  $A(0) = 20 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$  para distintos valores de  $N^{\max}$ .

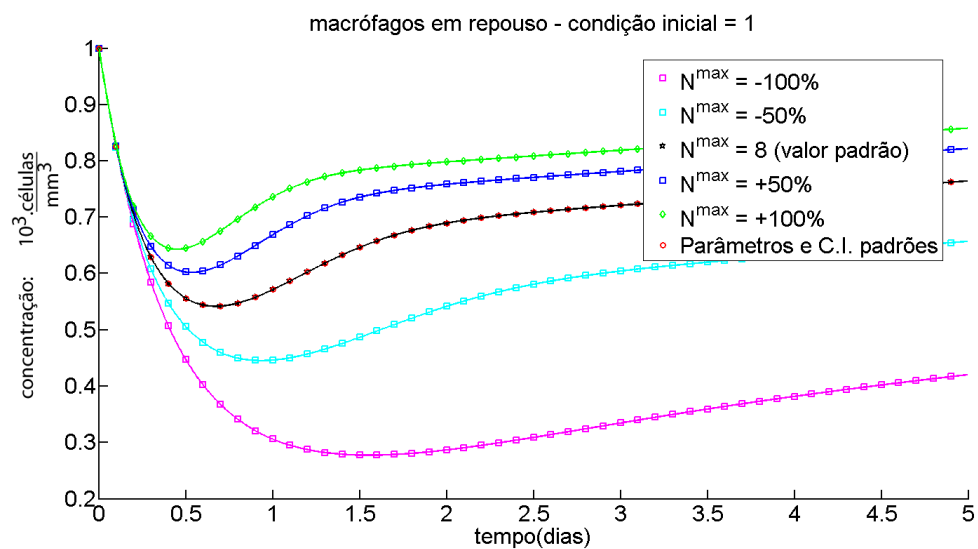


Figura 6.4: Evolução temporal de macrófagos em repouso quando  $A(0) = 20 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$  para distintos valores de  $N^{\max}$ .

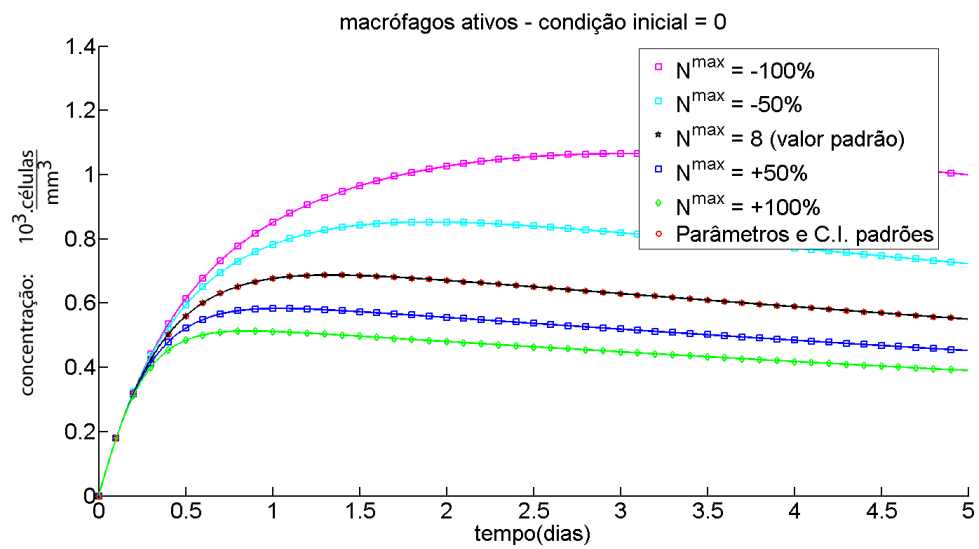


Figura 6.5: Evolução temporal de macrófagos ativos quando  $A(0) = 20 \frac{10^3 \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $N^{\max}$ .

Em um segundo cenário a quantidade inicial de LPS foi dobrada ( $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$ ). Pode-se observar neste cenário efeitos similares aos do cenário anterior. O aumento de neutrófilos no tecido continua levando a uma menor ativação dos macrófagos, porém, em comparação com o caso anterior, observa-se uma maior ativação dos macrófagos. Isto basicamente ocorreu devido ao aumento na quantidade de LPS. Como há uma maior concentração de macrófagos ativos, também há uma maior produção de citocinas quando comparado ao cenário anterior. As Figuras 6.6 a 6.10 apresentam os resultados.

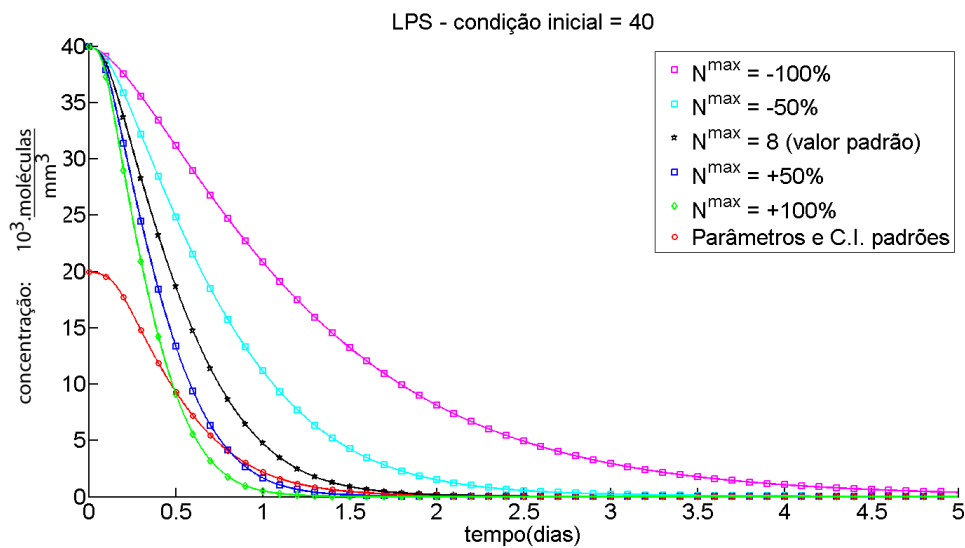


Figura 6.6: Evolução temporal de LPS quando  $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $N^{\max}$ .

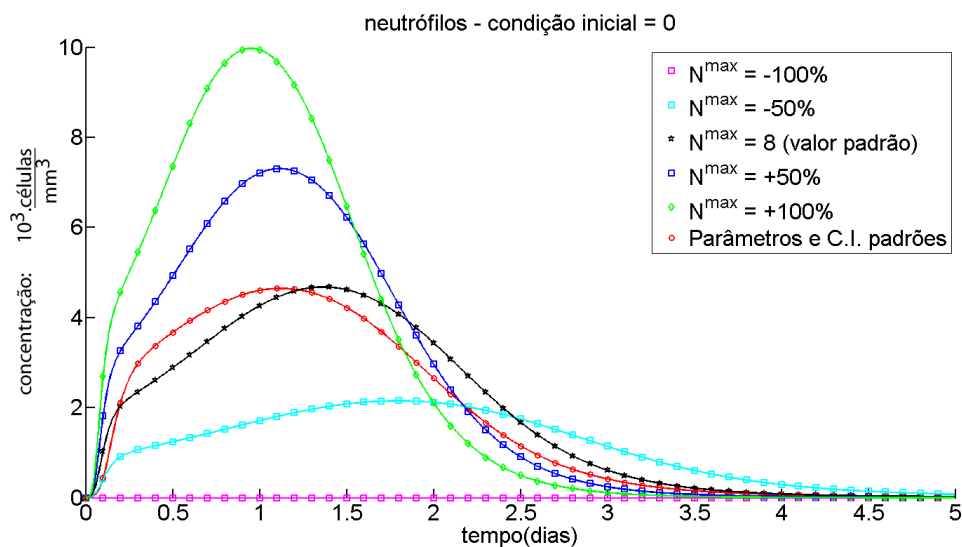


Figura 6.7: Evolução temporal de neutrófilos quando  $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $N^{\max}$ .

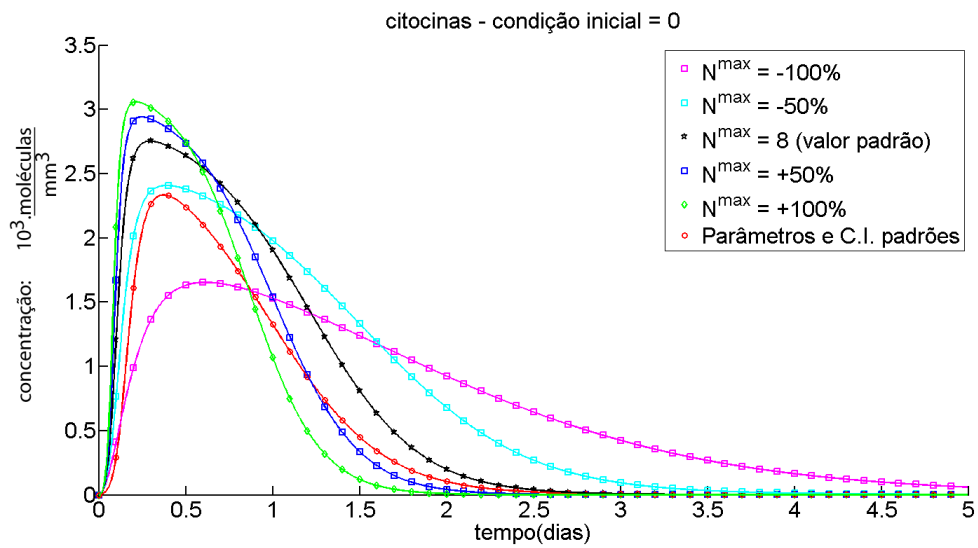


Figura 6.8: Evolução temporal de citocinas pró-inflamatórias quando  $A(0) = 40 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$  para distintos valores de  $N^{\max}$ .

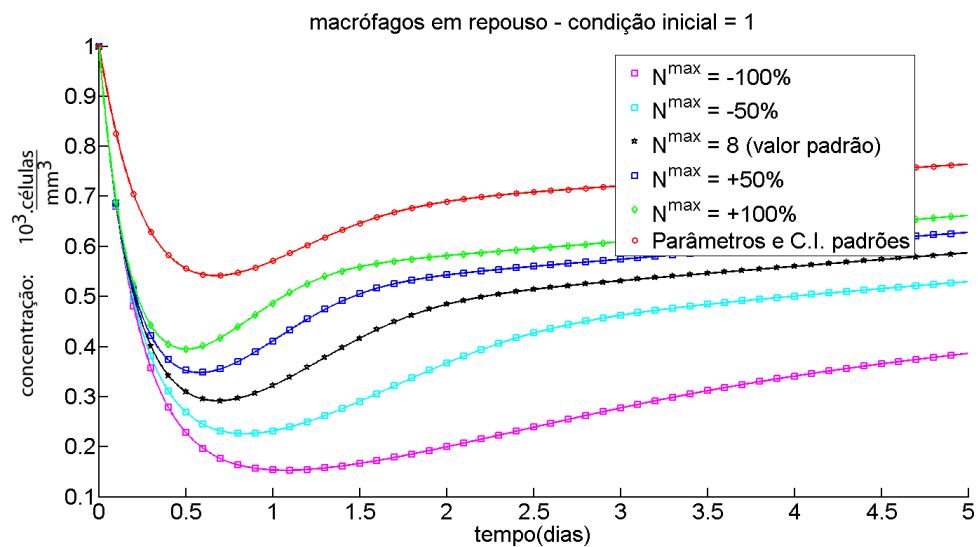


Figura 6.9: Evolução temporal de macrófagos em repouso quando  $A(0) = 40 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$  para distintos valores de  $N^{\max}$ .



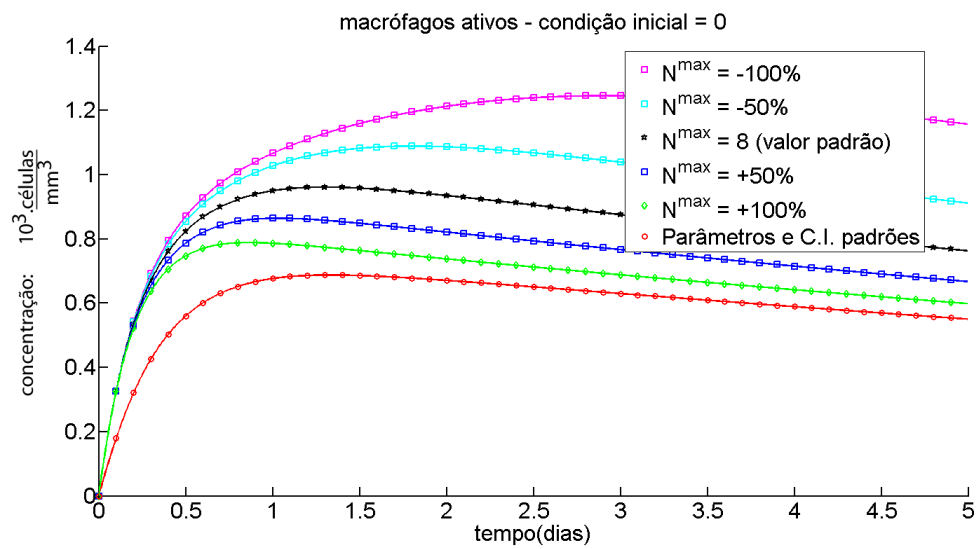


Figura 6.10: Evolução temporal de macrófagos ativos quando  $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $N^{\max}$ .

### 6.1.2 Parâmetro $\lambda_{N|A}$

Nesta seção, considera-se o parâmetro  $\lambda_{N|A}$ , responsável por determinar o quão efetiva é a fagocitose realizada pelos neutrófilos. Este parâmetro também mostra-se um parâmetro bastante sensível para este modelo.

Por meio das Figuras 6.11 a 6.15 observa-se que um ajuste negativo em  $\lambda_{N|A}$  muda o comportamento dos neutrófilos, tornando-os menos efetivos na fagocitose de LPS, sendo assim necessária uma maior quantidade de macrófagos para auxiliar em sua eliminação. Já um aumento em  $\lambda_{N|A}$  provoca uma menor produção de citocinas, uma vez que o LPS é eliminado em um menor espaço de tempo e que é justamente o contato de neutrófilos e macrófagos com LPS o responsável pela produção de TNF- $\alpha$  e IL-8 [16, 17, 18]).

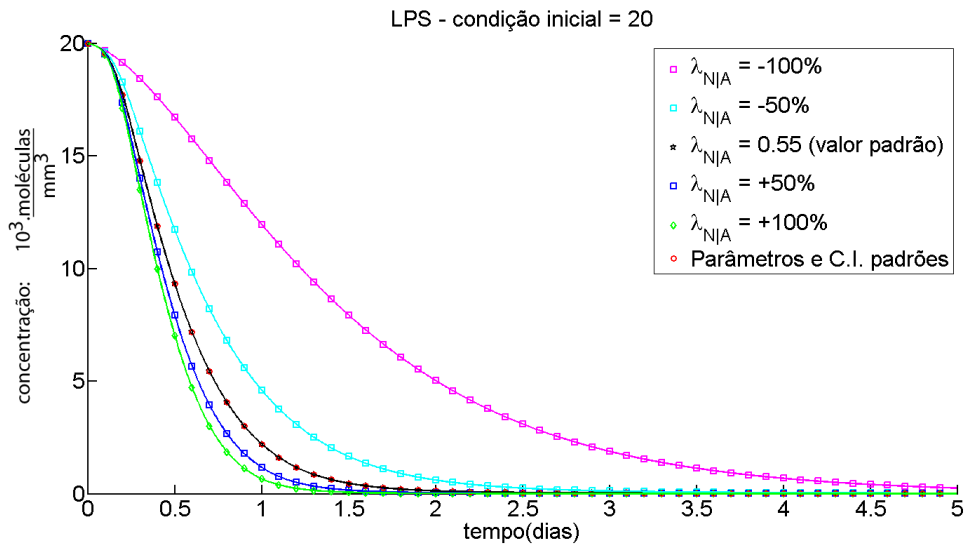


Figura 6.11: Evolução temporal de LPS quando  $A(0) = 20 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{N|A}$ .

Com a alteração da condição inicial do LPS ( $A(0) = 40 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$ ), pode-se observar efeitos similares aos do cenário anterior. O aumento da taxa de fagocitose por neutrófilos provoca igualmente uma redução na produção geral de citocinas pró-inflamatórias. Porém, como há uma maior quantidade de LPS no tecido, a produção de citocinas é mais elevada do que a do caso anterior. Além disso, nota-se que há um aumento considerável da migração de neutrófilos para o tecido, devido basicamente ao aumento de citocinas pró-inflamatórias e outrossim há um aumento na quantidade de ativações de macrófagos. Repara-se que mesmo com a quantidade inicial de LPS mais elevada, o SIH inato foi capaz de eliminar o LPS com a mesma eficiência, porém para isto foi necessário o

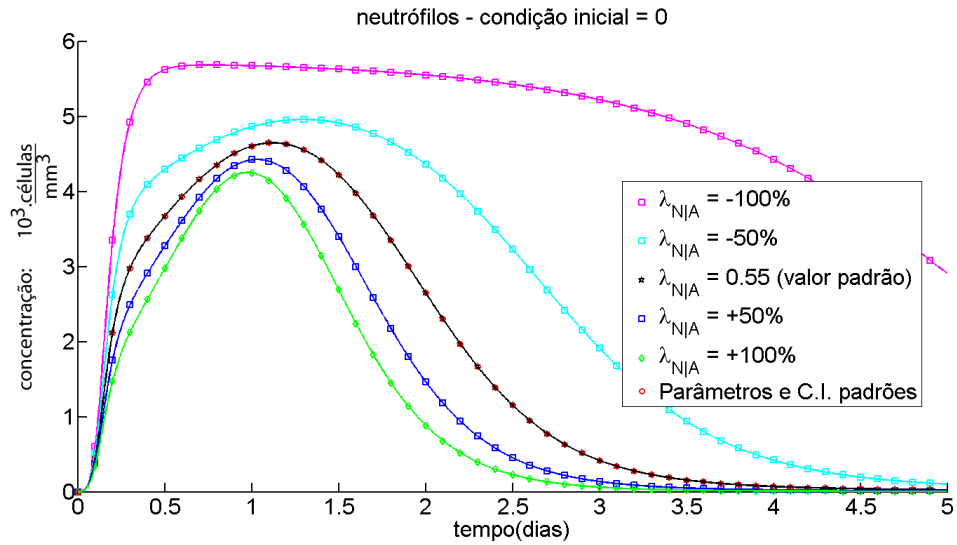


Figura 6.12: Evolução temporal de neutrófilos quando  $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{N|A}$ .

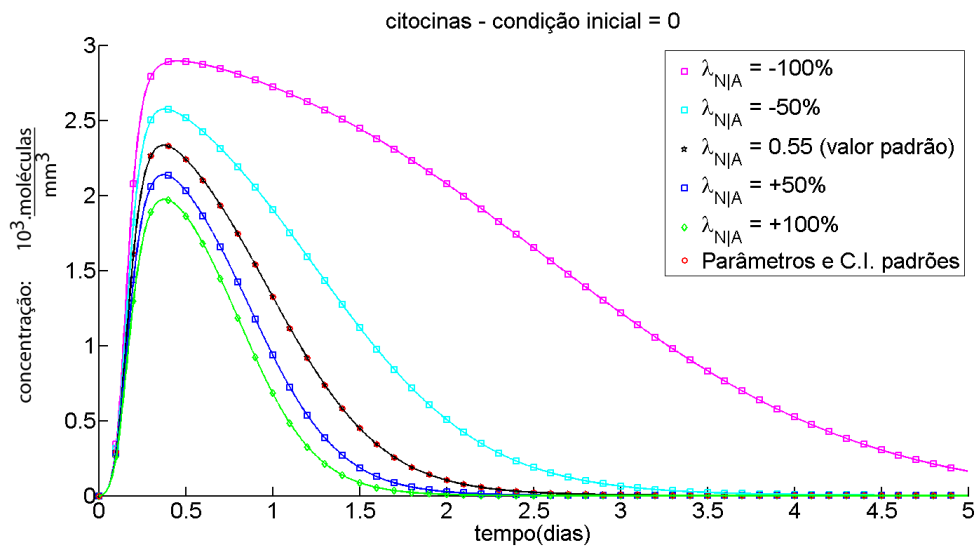


Figura 6.13: Evolução temporal de citocinas pró-inflamatórias quando  $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{N|A}$ .

recrutamento em maior escala de novas células advindas dos vasos sanguíneos. Este cenário é representado pelas Figuras 6.16 até 6.20.

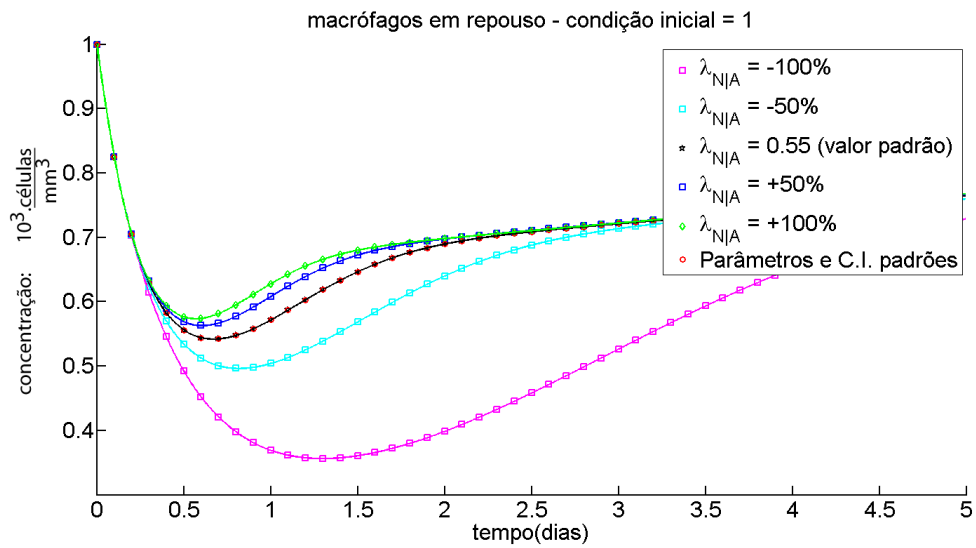


Figura 6.14: Evolução temporal de macrófagos em repouso quando  $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{N|A}$ .

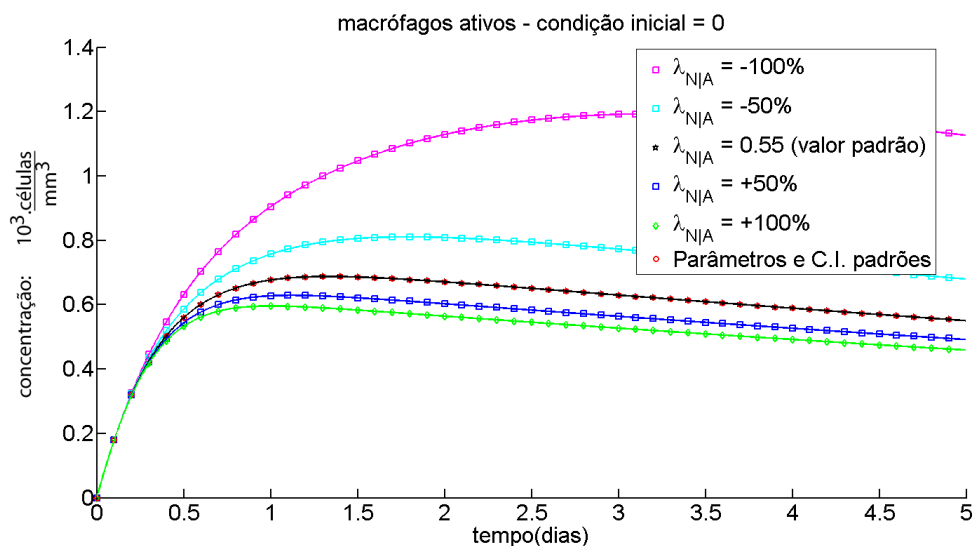


Figura 6.15: Evolução temporal de macrófagos ativos quando  $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{N|A}$ .

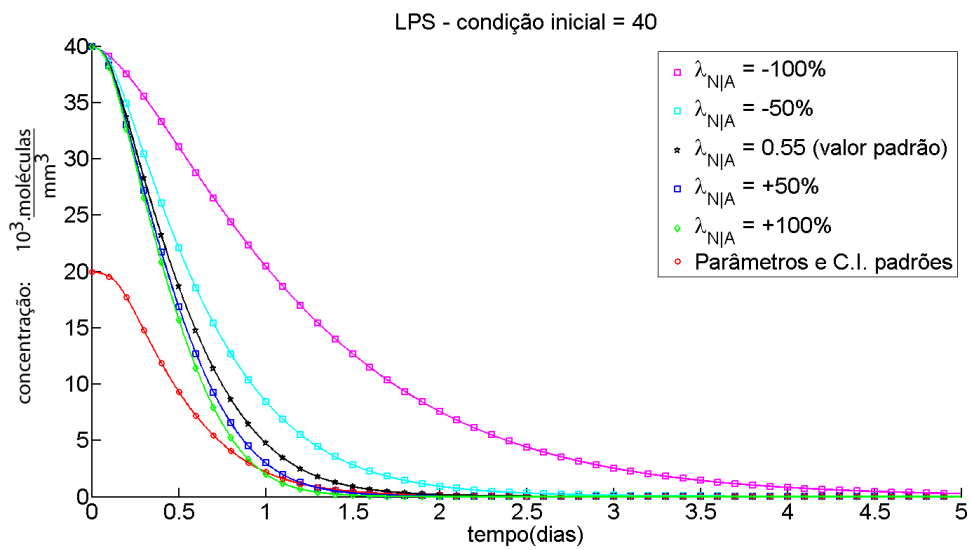


Figura 6.16: Evolução temporal de LPS quando  $A(0) = 40 \frac{10^3 \text{ moléculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{NIA}$ .

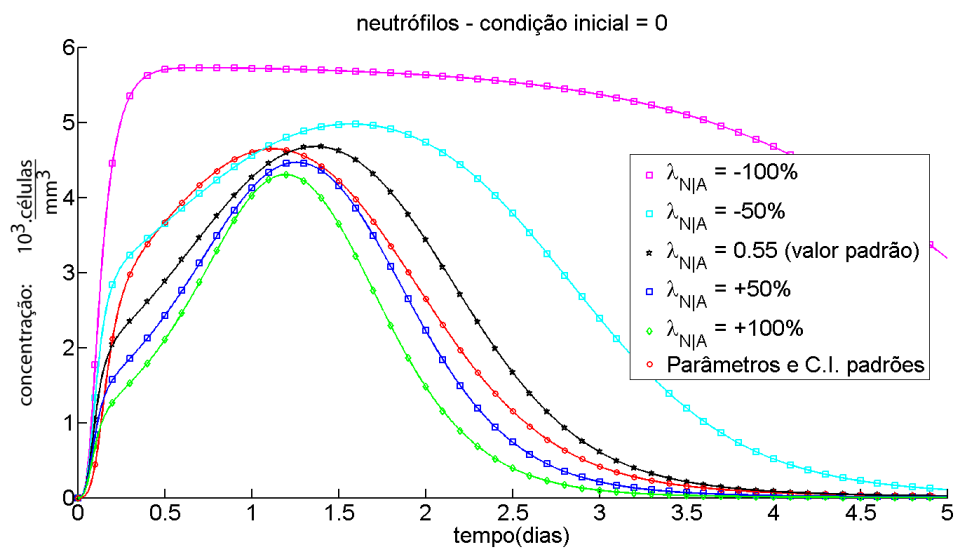


Figura 6.17: Evolução temporal de neutrófilos quando  $A(0) = 40 \frac{10^3 \text{ moléculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{NIA}$ .

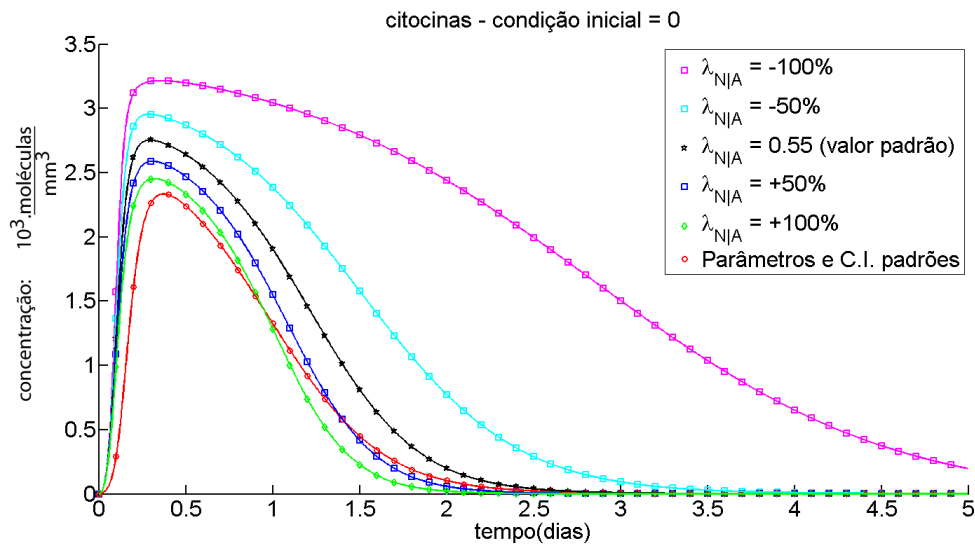


Figura 6.18: Evolução temporal de citocinas pró-inflamatórias quando  $A(0) = 40 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{N|A}$ .

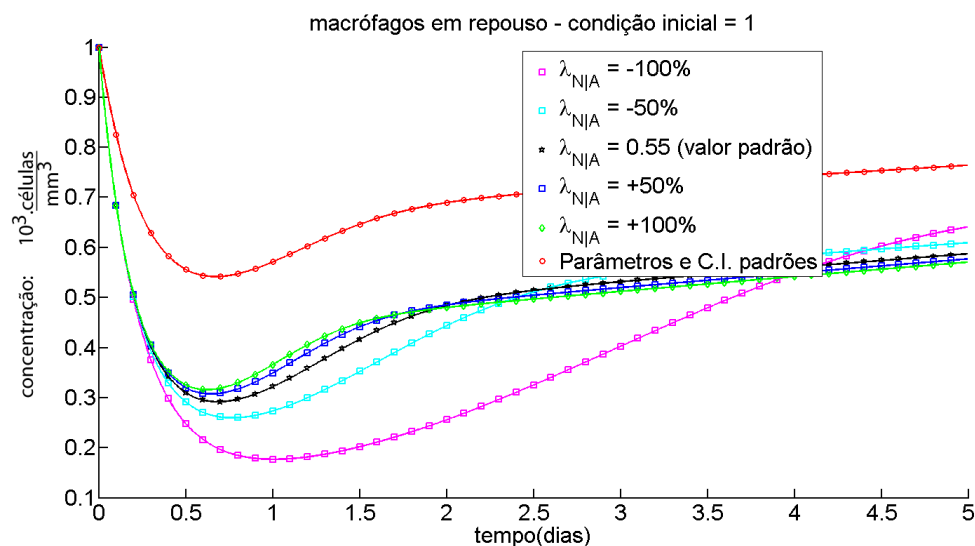


Figura 6.19: Evolução temporal de macrófagos em repouso quando  $A(0) = 40 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{N|A}$ .

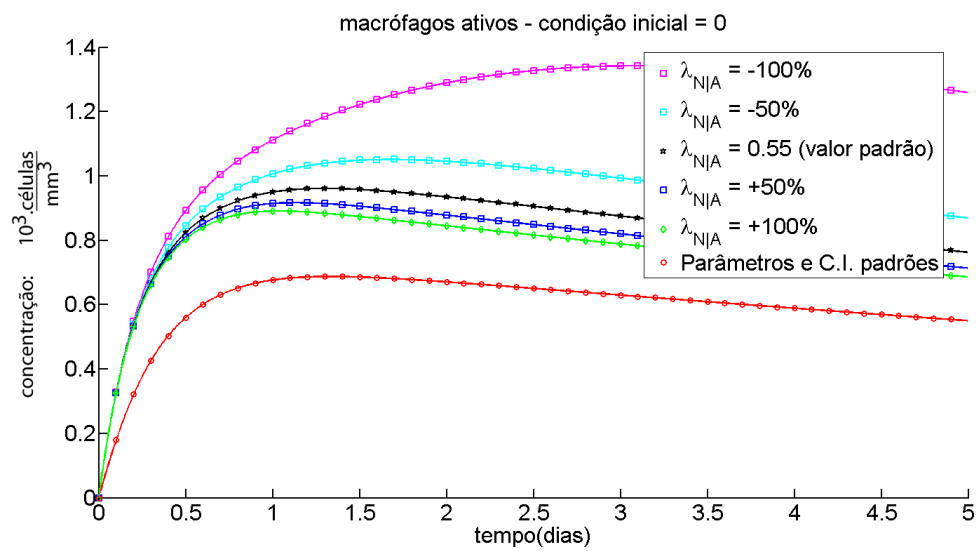


Figura 6.20: Evolução temporal de macrófagos ativos quando  $A(0) = 40 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{NIA}$ .

### 6.1.3 Parâmetro $\lambda_{MA|A}$

Por fim, esta seção faz uma análise de um parâmetro cujas alterações em seus valores acarretaram poucas variações nos resultados dos sistemas de EDOs: a taxa de fagocitose de LPS por macrófagos ativos. As Figuras 6.21 até 6.25 apresentam os resultados para este cenário, onde a condição inicial de LPS é igual a  $20 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$  ( $A(0) = 20 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$ ). Observa-se em todas as figuras alterações menores nos resultados das simulações. A quantidade de citocinas pró-inflamatórias produzida foi levemente reduzida quando se aumentou a taxa de fagocitose por macrófagos ativos. Isto ocorre em virtude de um leve aumento na eliminação do LPS. Em um cenário com redução de 100% no valor de  $\lambda_{MA|A}$  (situação em que macrófagos ativos não são capazes de promover a fagocitose) e de baixa taxa de fagocitose de LPS pelos macrófagos em repouso, o neutrófilo se torna a principal célula responsável pela eliminação de LPS. Neste cenário a eliminação de LPS ocorre em um intervalo de tempo ligeiramente maior do que no caso em que os macrófagos ativos também fagocitam.

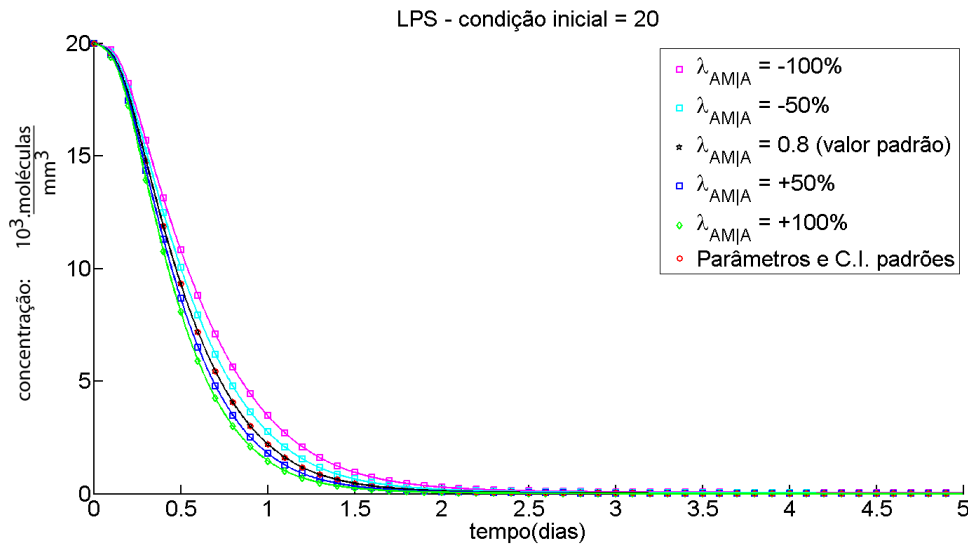


Figura 6.21: Evolução temporal de LPS quando  $A(0) = 20 \frac{10^3 \cdot \text{moléculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{MA|A}$ .

### 6.1.4 Observações Finais sobre a Análise de Sensibilidade

É importante salientar que não foram estudados todos as combinações de cenários elaboradas para a análise de sensibilidade. Houve um maior foco, primordialmente, nos parâmetros para os quais se esperava que ocorressem maiores variações nos resultados.



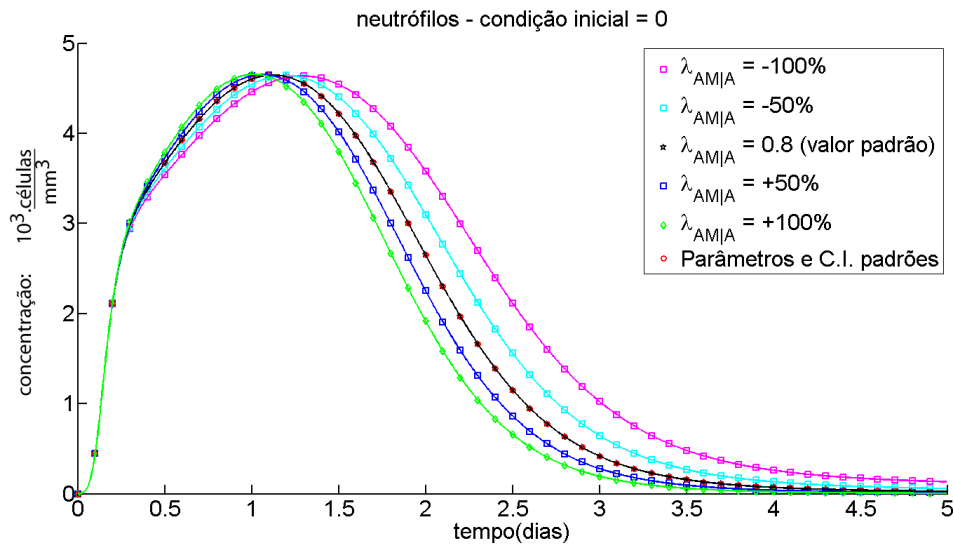


Figura 6.22: Evolução temporal de neutrófilos quando  $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{MA|A}$ .

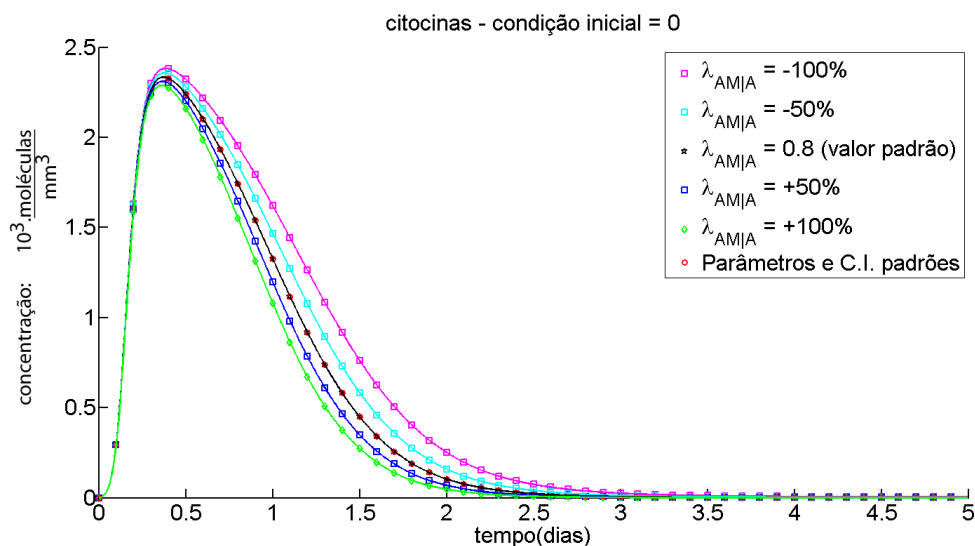


Figura 6.23: Evolução temporal de citocinas pró-inflamatórias quando  $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{MA|A}$ .

As análises efetuadas indicaram que os dois parâmetros  $N^{max}$  e  $\lambda_{N|A}$  provocam, quando alteradas, grandes mudanças no comportamento das simulações. Já outras variáveis, como  $\lambda_{MA|A}$ , não provocaram mudanças significativas. Contudo, isto não significa que tais mudanças não sejam importantes para o SIH inato. Necessário se faz ressaltar que o sistema imunológico é composto por diversos comportamentos redundantes, o que o torna excepcional em sua tarefa de defender o organismo.

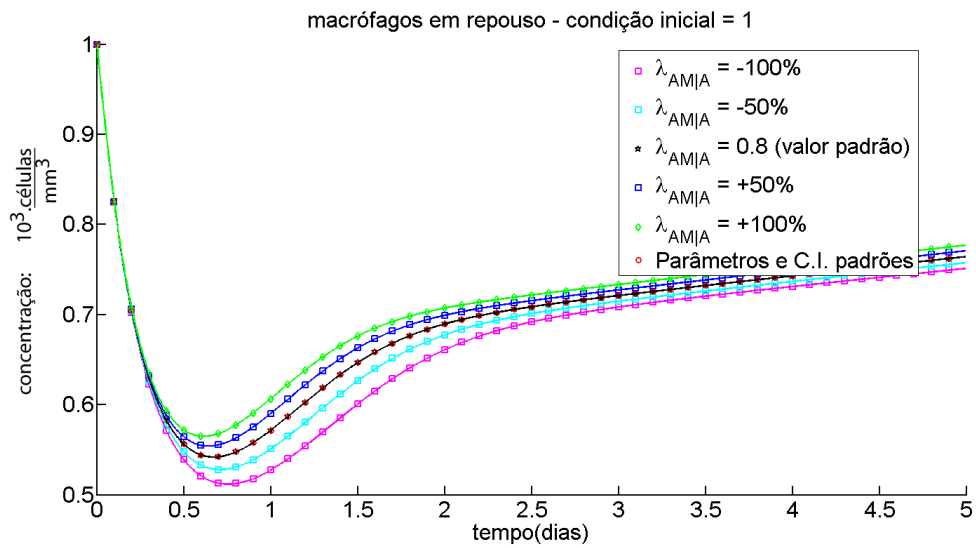


Figura 6.24: Evolução temporal de macrófagos em repouso quando  $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{MA|A}$ .

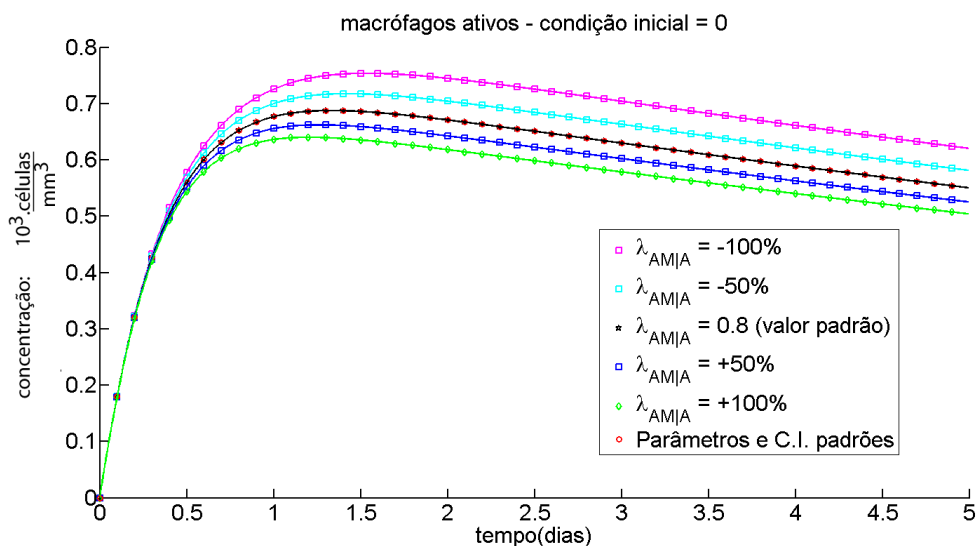


Figura 6.25: Evolução temporal de macrófagos ativos quando  $A(0) = 20 \frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$  para distintos valores de  $\lambda_{MA|A}$ .

## 6.2 Simulação do Modelo 3D

Esta seção apresenta os resultados obtidos na simulação computacional do modelo do SIH inato em uma seção de tecido em três dimensões. Os parâmetros e condições iniciais utilizados nas simulações desta seção são apresentados no Apêndice A. O modelo de EDPs será avaliado usando 3 casos distintos e incrementais.

- Caso 1: Considera-se no modelo apenas LPS e sua difusão pelo tecido;
- Case 2: São considerados no modelo também macrófagos (em repouso e ativos) e a produção de citocinas;
- Case 3: O modelo completo é considerado.

### 6.2.1 *Caso 1*

O caso 1 considera apenas a presença de LPS no tecido e sua difusão, conforme ilustrado pela Equação 6.2. Neste cenário, o sistema imunológico não realiza nenhuma ação em resposta a presença de LPS. O conjunto de Figuras 6.26 apresenta os resultados este cenário.

$$\begin{cases} \frac{\partial A}{\partial t} = -\mu_A A + D_A \Delta A \\ A(x, y, z, 0) = A_0, \frac{\partial A(\dots, t)}{\partial n} |_{\partial\Omega} = 0 \end{cases} \quad (6.2)$$

Observa-se que, sem a intervenção do SIH inato, o LPS injetado em uma pequena seção do tecido logo se difunde por todo o domínio.

### 6.2.2 *Caso 2*

No caso 2 o sistema foi incrementado com concentração de macrófagos (em repouso e ativos), bem como pela produção de citocinas pró-inflamatórias, responsáveis pelo controle

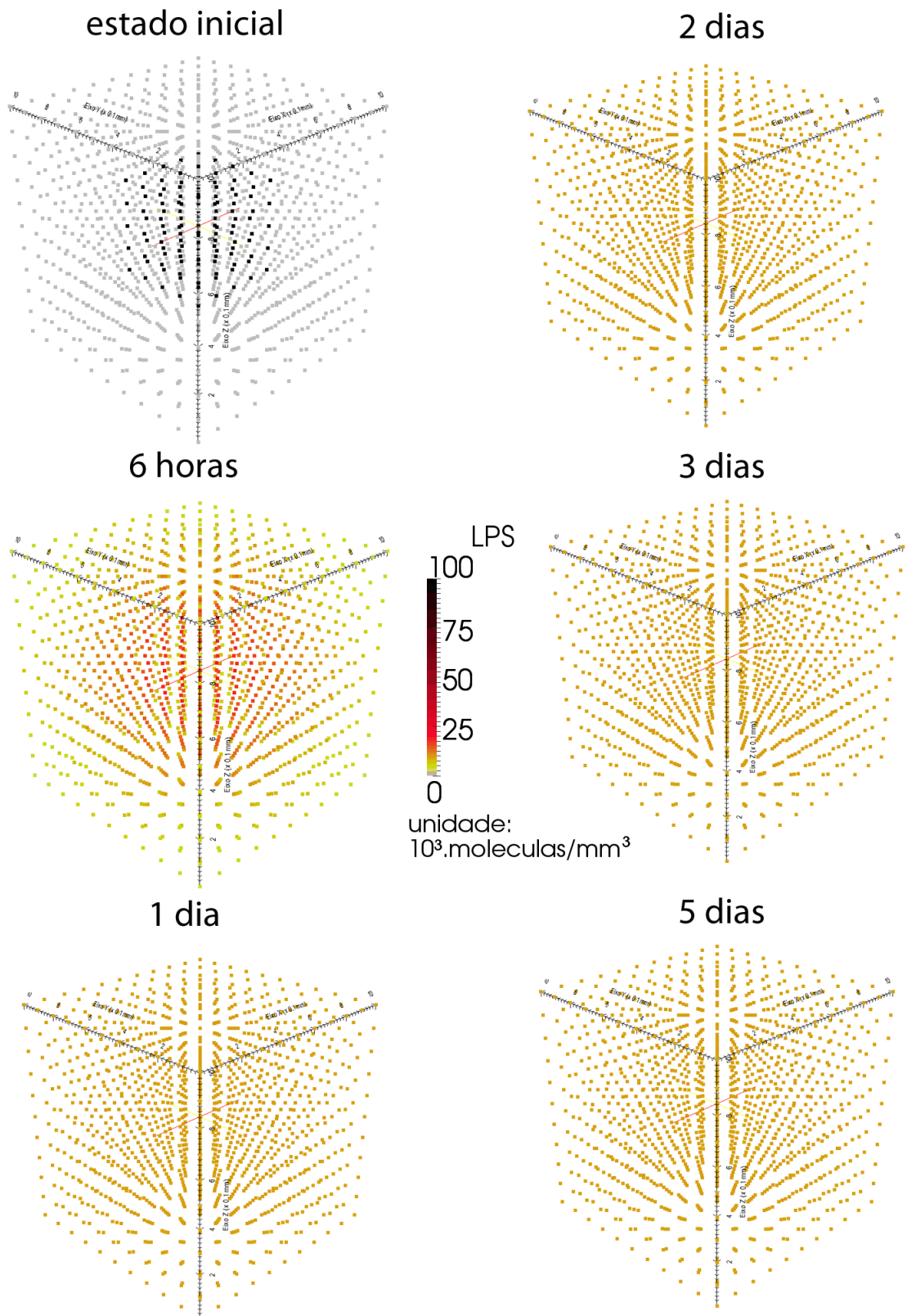


Figura 6.26: Distribuição espacial e temporal de antígenos.

da permeabilidade endotelial. O conjunto de Equações 6.3 sintetiza este cenário.

$$\left\{ \begin{array}{l}
 \frac{\partial A}{\partial t} = -\mu_A A - (\lambda_{MA|A} MA + \lambda_{MR|A} MR).A + D_A \Delta A \\
 A(x, y, z, 0) = A_0, \frac{\partial A(\dots, t)}{\partial n} |_{\partial \Omega} = 0 \\
 permeability_{MR} = (P_{MR}^{max} - P_{MR}^{min}) \cdot \frac{CH}{CH + k_{eqch}} + P_{MR}^{min} \\
 source_{MR} = permeability_{MR} \cdot (M^{max} - (MR + MA)) \\
 \frac{\partial MR}{\partial t} = -\mu_{MR} MR - \lambda_{MR|A} MR.A + source_{MR} + D_{MR} \Delta MR \\
 - \nabla \cdot (\chi_{MR} MR \nabla CH) \\
 MR(x, y, z, 0) = MR_0, \frac{\partial MR(\dots, t)}{\partial n} |_{\partial \Omega} = 0 \\
 \frac{\partial MA}{\partial t} = -\mu_{MA} MA + \lambda_{MR|A} MR.A + D_{MA} \Delta MA - \nabla \cdot (\chi_{MA} MA \nabla CH) \\
 MA(x, y, z, 0) = MA_0, \frac{\partial MA(\dots, t)}{\partial n} |_{\partial \Omega} = 0 \\
 \frac{\partial CH}{\partial t} = -\mu_{CH} CH + \beta_{CH|MA} MA.A.(1 - CH/chInf) + D_{CH} \Delta CH \\
 CH(x, y, z, 0) = CH_0, \frac{\partial CH(\dots, t)}{\partial n} |_{\partial \Omega} = 0
 \end{array} \right. \quad (6.3)$$

Como pode-se observar pelos resultados apresentados pela Figura 6.27, inicialmente o LPS se distribui ao longo do tecido. Porém, com a ação dos macrófagos, o LPS é eliminado após 5 dias de infecção. Como o LPS, em seu estado inicial, está aglomerado em apenas uma parte do tecido, espera-se que a quantidade de macrófagos em repouso que se tornam ativos seja maior nesta região do que nas demais regiões do tecido. A Figura 6.28 confirma este cenário. Ademais, observa-se que a quantidade de macrófagos em repouso volta a crescer no local de maior infecção após o primeiro dia. Isto ocorre devido à atração de mais macrófagos da corrente sanguínea para o tecido em virtude da intensa produção de citocinas pró-inflamatórias. A Figura 6.29 apresenta os locais de maior produção de citocinas ao longo do tempo, podendo ser verificado que a quantidade é consideravelmente maior em pontos onde há maior concentração de LPS. A produção de citocinas diminui na medida em que a quantidade de LPS no tecido se reduz pois, além da diminuição na quantidade produzida por macrófagos, as citocinas se difundem e desfazem-se em razão do tempo. Por fim, é notório (Figura 6.30) o fato de que os macrófagos se ativam nos locais onde é maior a concentração de LPS no tecido. Após o terceiro dia, os macrófagos ativos decaem levemente por causa da diminuição na quantidade de macrófagos se tornando

ativos e da morte das células já existentes por conta do fim do seu ciclo de vida.

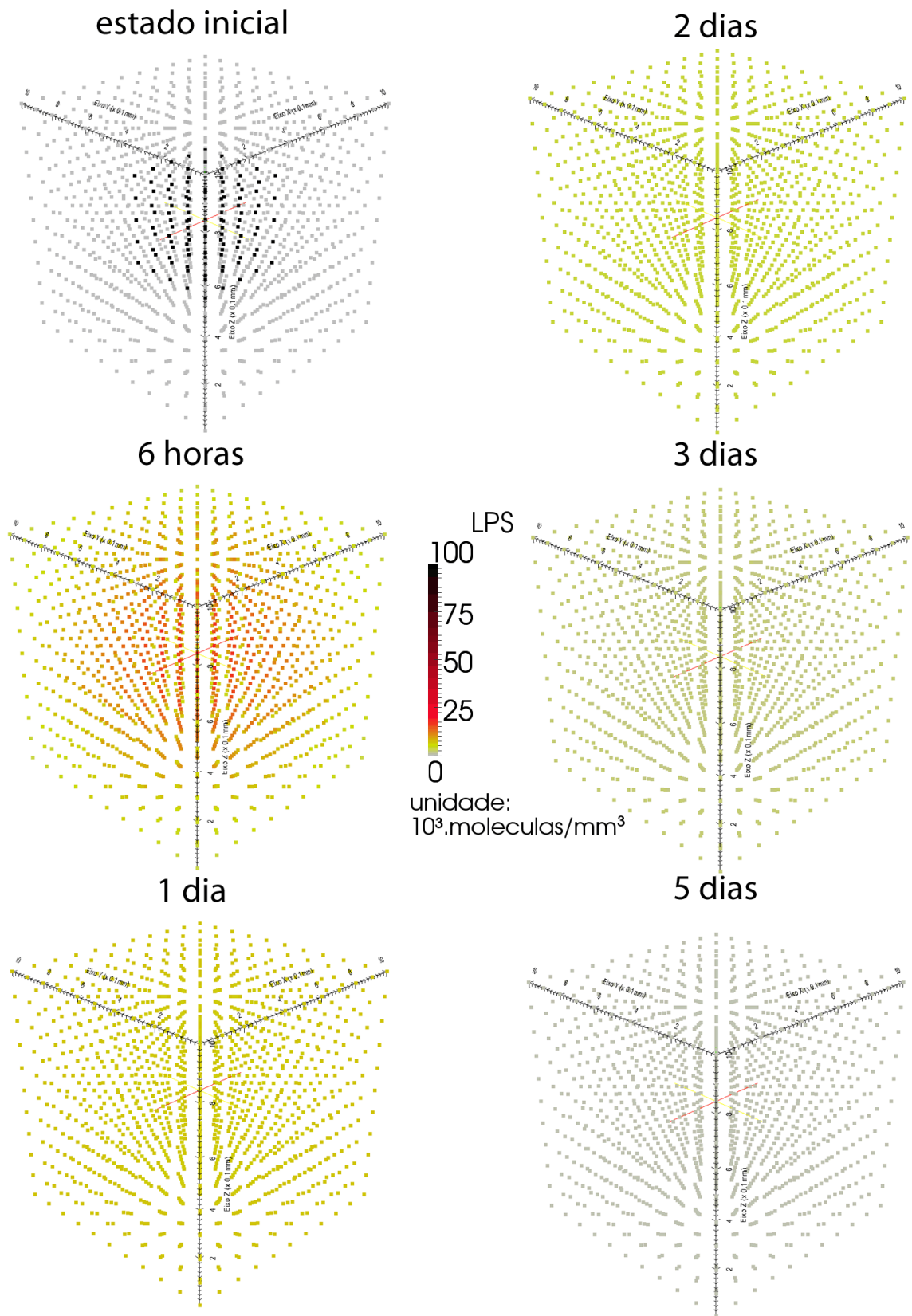


Figura 6.27: Distribuição espacial e temporal do LPS.

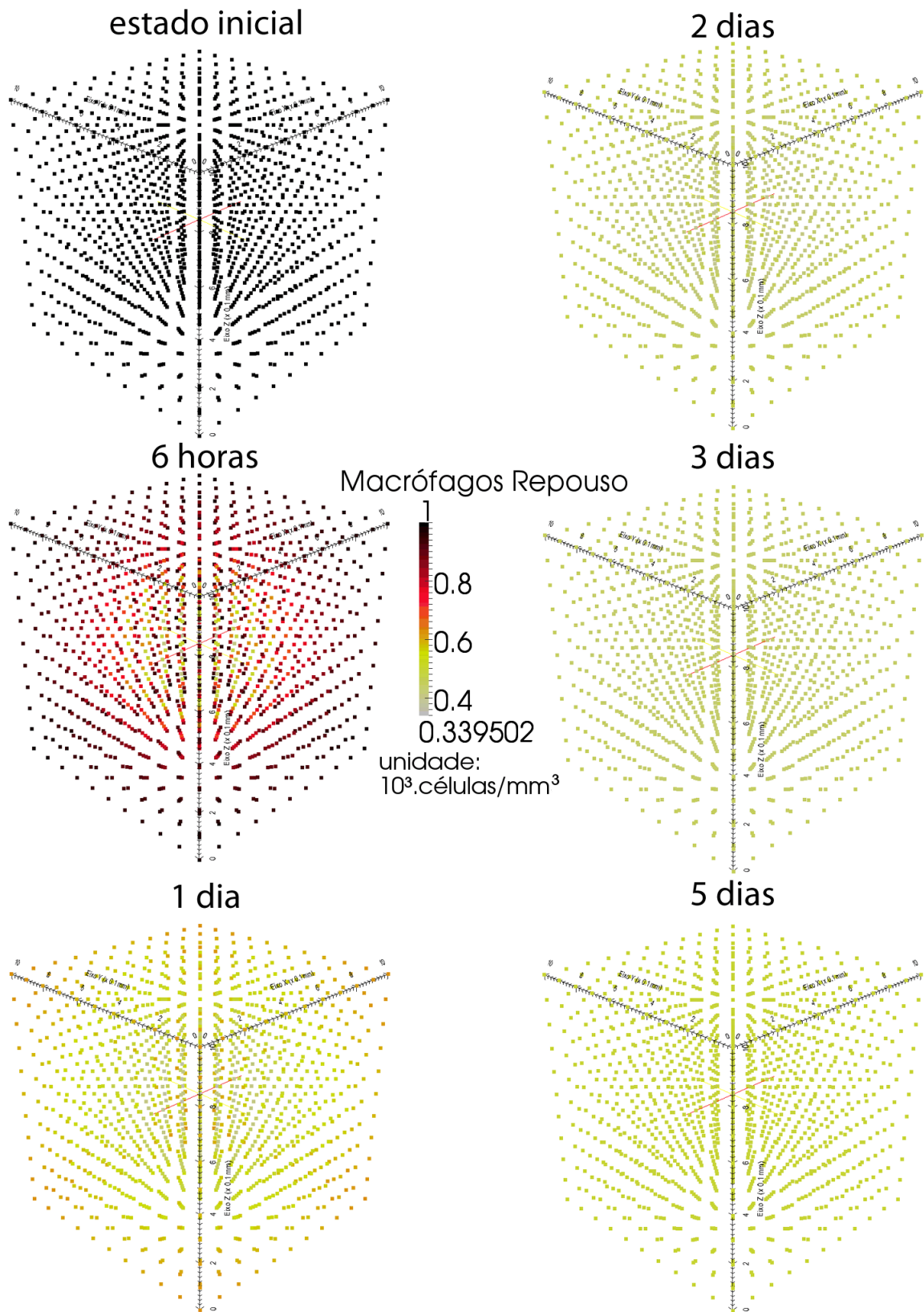


Figura 6.28: Distribuição espacial e temporal de macrófagos em repouso.



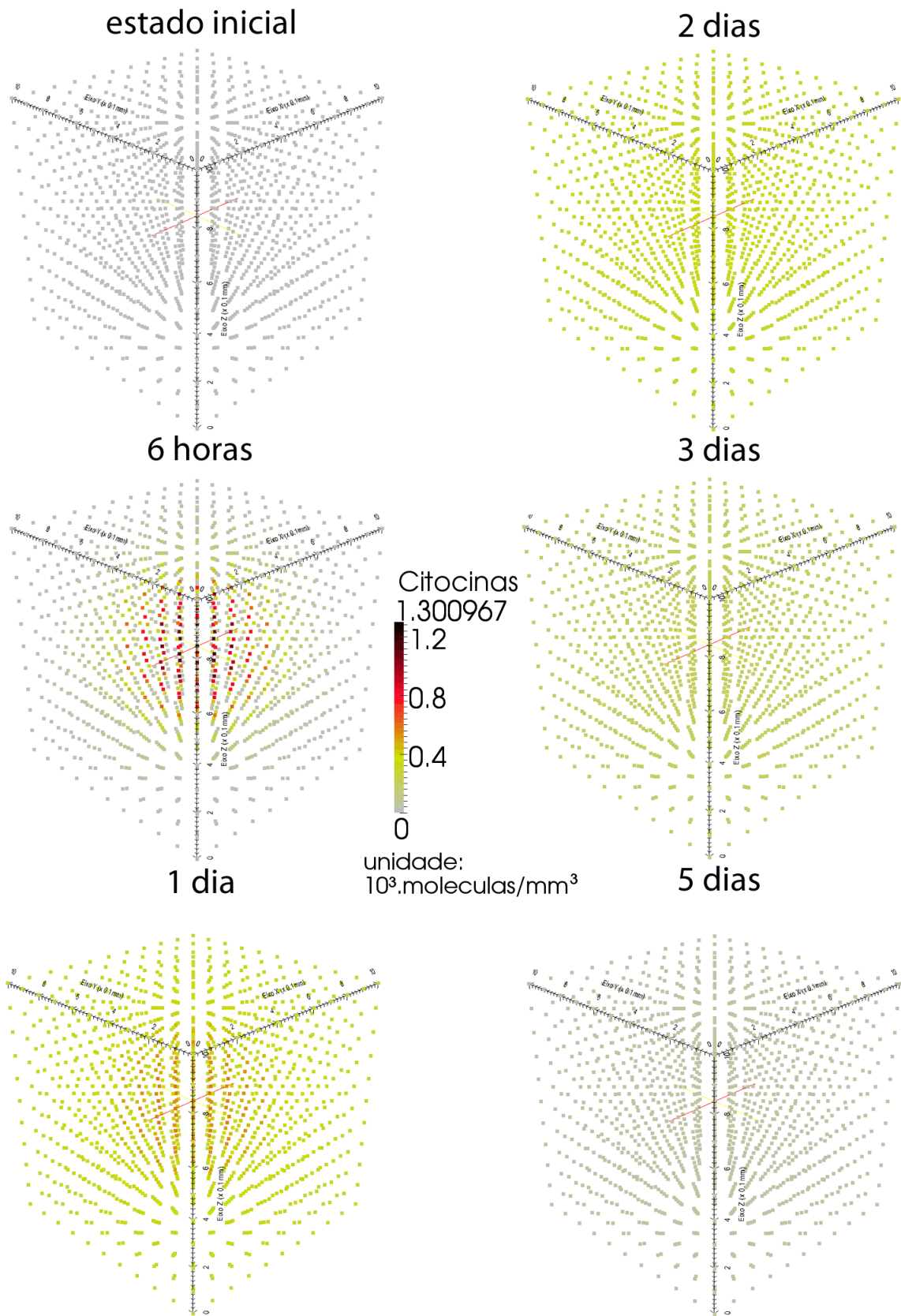


Figura 6.29: Distribuição espacial e temporal de citocinas pró-inflamatórias.



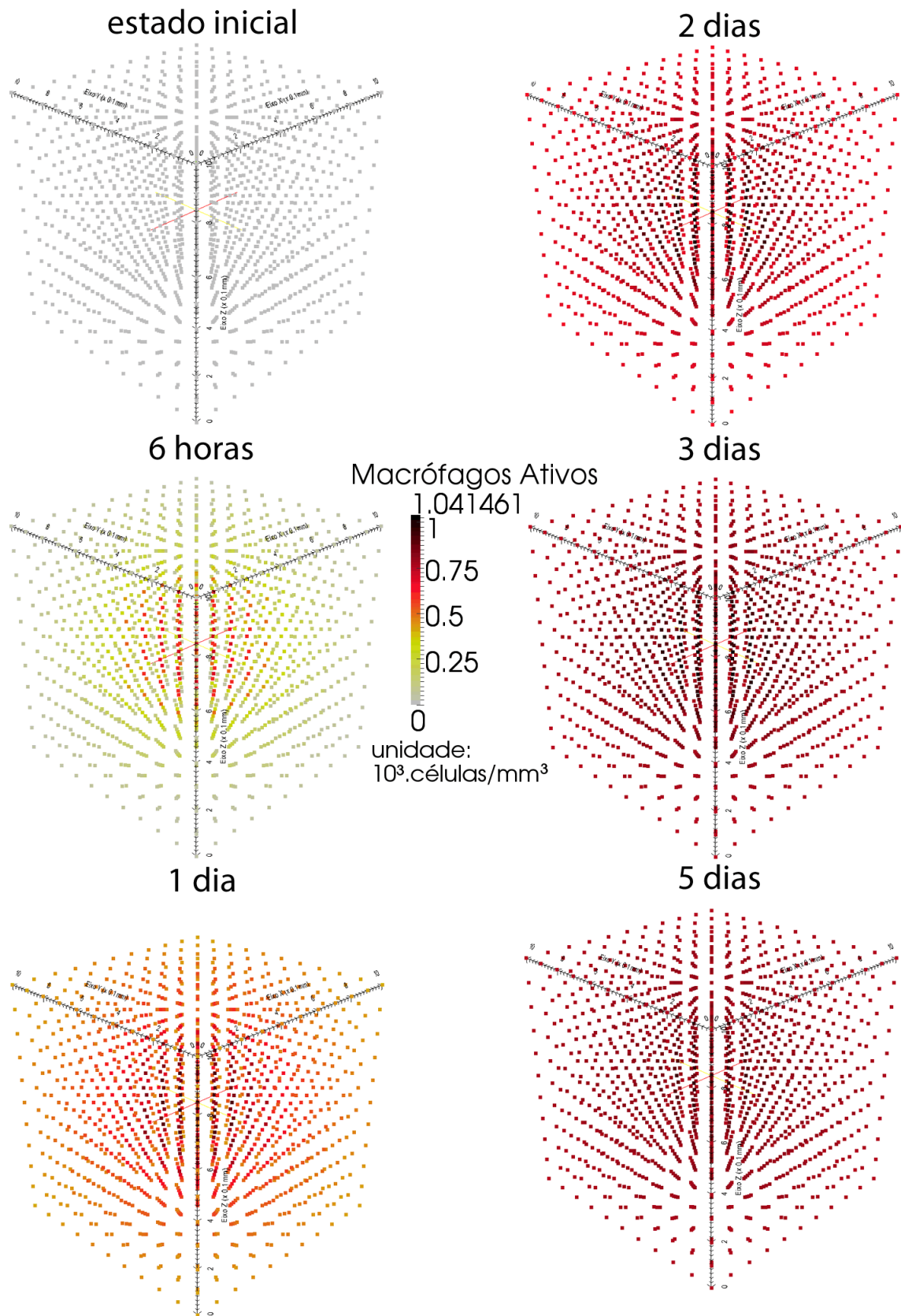


Figura 6.30: Distribuição espacial e temporal de macrófagos ativos.

### **6.2.3 Caso 3**

Para o caso 3, representado pela Equação 6.4, o sistema completo, composto por antígenos, macrófagos em repouso, macrófagos ativos, citocinas pró-inflamatórias e neutrófilos, foi analisado. Como pode-se observar na Figura 6.31, os neutrófilos exercem um papel importante na defesa do organismo, visto que com sua presença o LPS é eliminado em um prazo bem menor (aproximadamente 2 dias após infecção) do que no caso anterior. Com a adição de neutrófilos, a quantidade de macrófagos em repouso que se ativam torna-se menor, o que pode ser observado na Figura 6.32. Apesar desta diminuição, os neutrófilos e os macrófagos ativos produzem uma quantidade de citocina muito maior e em menor tempo do que no caso 2 (Figura 6.33), o que pode ser facilmente observado 6 horas após o início da infecção. Além disso, percebe-se que as citocinas, mesmo com uma concentração maior, decaem em um menor espaço de tempo, pois a eliminação de LPS ocorreu com 3 dias de antecedência em relação ao caso anterior. Como há uma menor ativação de macrófagos em repouso, espera-se uma menor concentração de macrófagos ativos, o que pode ser notado pela Figura 6.34. Por fim, observa-se na Figura 6.35 o crescimento e a dispersão da concentração de neutrófilos. Os neutrófilos adentram no tecido em grande escala no local de maior concentração de LPS em virtude da alta concentração de citocinas neste ponto, que por sua vez aumentam a permeabilidade do endotélio na região. Ao entrar no tecido, observa-se que os neutrófilos realizam a fagocitose do LPS e se difundem. Após o primeiro dia de infecção os neutrófilos são encontrados praticamente em todo tecido com igual concentração. Contudo, os neutrófilos se tornam apoptóticos, sendo eliminados

do tecido entre o terceiro e quinto dia após o início da infecção.

$$\left\{ \begin{array}{l}
 \frac{\partial A}{\partial t} = -\mu_A A - (\lambda_{N|A} N + \lambda_{MA|A} MA + \lambda_{MR|A} MR) \cdot A + D_A \Delta A \\
 A(x, y, z, 0) = A_0, \frac{\partial A(\dots, t)}{\partial n} \Big|_{\partial \Omega} = 0 \\
 \\
 permeability = (P_N^{max} - P_N^{min}) \cdot CH / (CH + keqch) + Pmin \\
 sourceN = permeability \cdot (N^{max} - N) \\
 \frac{\partial N}{\partial t} = -\mu_N N - \lambda_{A|N} A \cdot N + sourceN + D_N \Delta N - \nabla \cdot (\chi_N N \nabla CH) \\
 N(x, y, z, 0) = N_0, \frac{\partial N(\dots, t)}{\partial n} \Big|_{\partial \Omega} = 0 \\
 \\
 \frac{\partial CH}{\partial t} = -\mu_{CH} CH + (\beta_{CH|N} \cdot N + \beta_{CH|MA} \cdot MA) \cdot A \cdot (1 - CH/chInf) + D_{CH} \Delta CH \\
 CH(x, y, z, 0) = CH_0, \frac{\partial CH(\dots, t)}{\partial n} \Big|_{\partial \Omega} = 0 \\
 \\
 permeability_{MR} = (P_{MR}^{max} - P_{MR}^{min}) \cdot \frac{CH}{CH + keqch} + P_{MR}^{min} \\
 source_{MR} = permeability_{MR} \cdot (M^{max} - (MR + MA)) \\
 \frac{\partial MR}{\partial t} = -\mu_{MR} MR - \lambda_{MR|A} MR \cdot A + source_{MR} + D_{MR} \Delta MR \\
 \quad - \nabla \cdot (\chi_{MR} MR \nabla CH) \\
 MR(x, y, z, 0) = MR_0, \frac{\partial MR(\dots, t)}{\partial n} \Big|_{\partial \Omega} = 0 \\
 \\
 \frac{\partial MA}{\partial t} = -\mu_{MA} MA + \lambda_{MR|A} MR \cdot A + D_{MA} \Delta MA - \nabla \cdot (\chi_{MA} MA \nabla CH) \\
 MA(x, y, z, 0) = MA_0, \frac{\partial MA(\dots, t)}{\partial n} \Big|_{\partial \Omega} = 0
 \end{array} \right. \tag{6.4}$$

#### 6.2.4 Comparação entre casos

Esta subseção apresenta as comparações entre os casos 2 e 3, já que o caso 1 é bastante simples por haver apenas LPS e a sua difusão. Primeiramente, percebe-se a importância que os neutrófilos possuem para o SIH inato. A Figura 6.36 apresenta a comparação entre o caso em que os neutrófilos não estão presentes no tecido com o caso em que eles estão

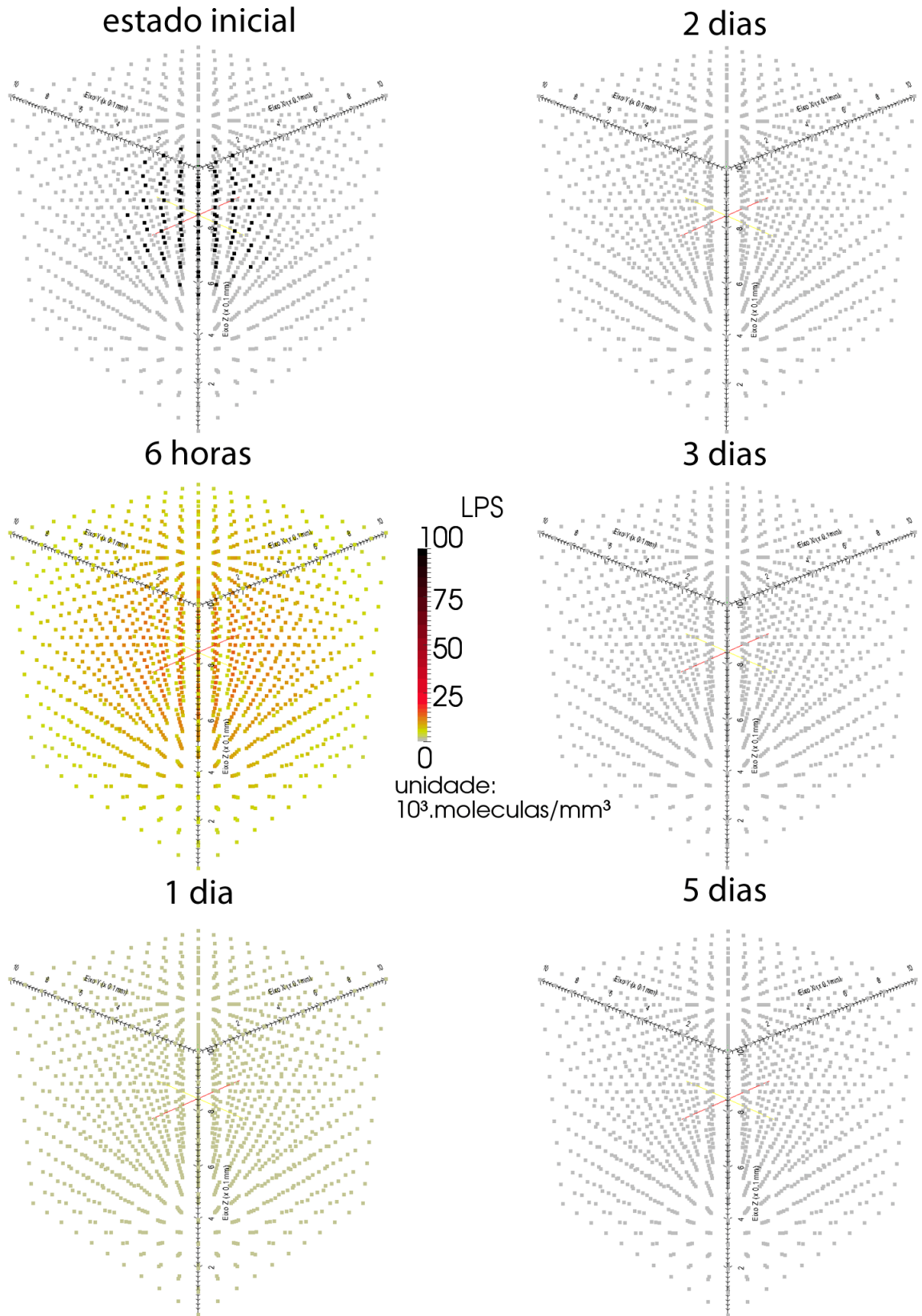


Figura 6.31: Distribuição espacial e temporal do LPS.

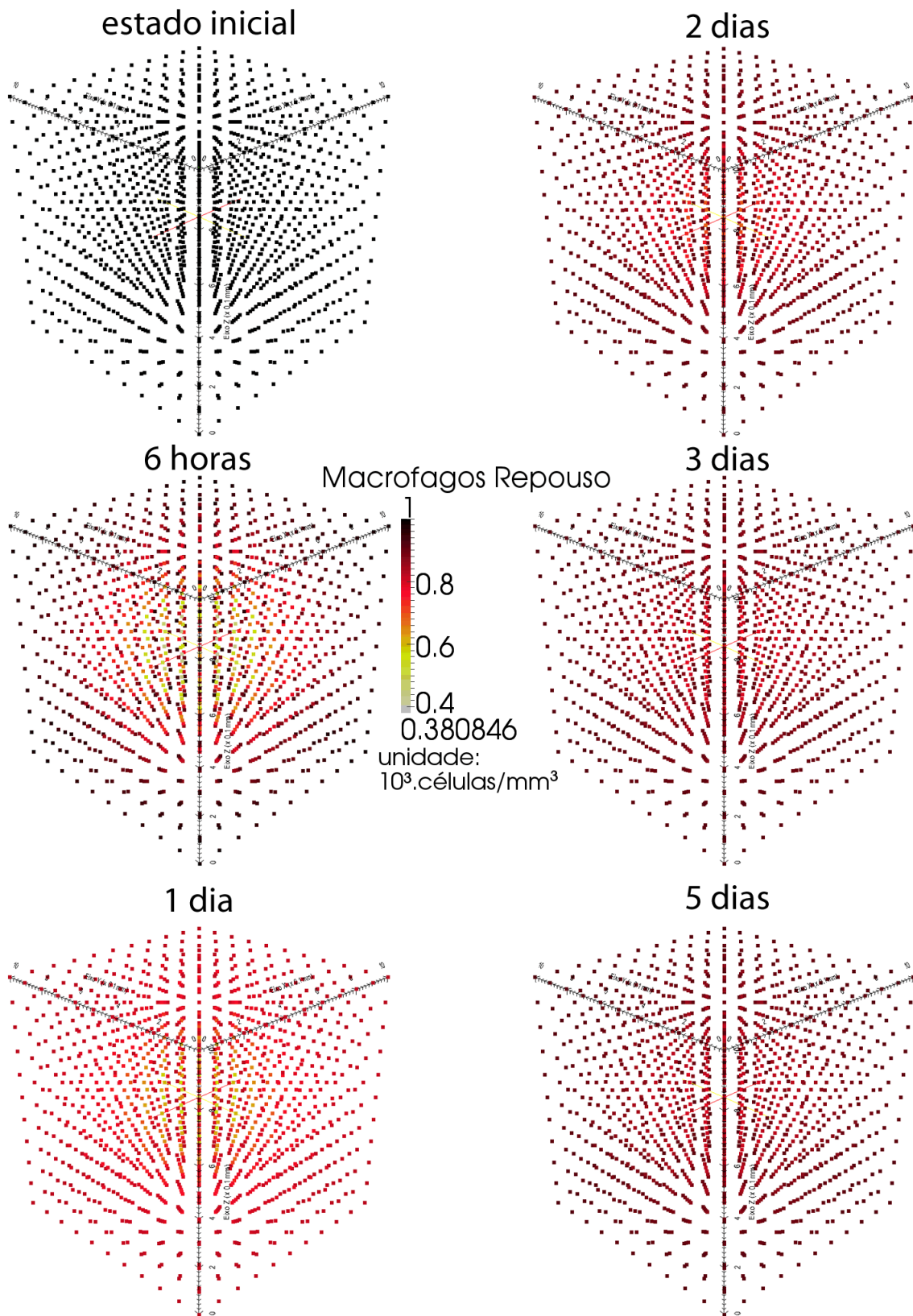


Figura 6.32: Distribuição espacial e temporal de macrófagos em repouso.

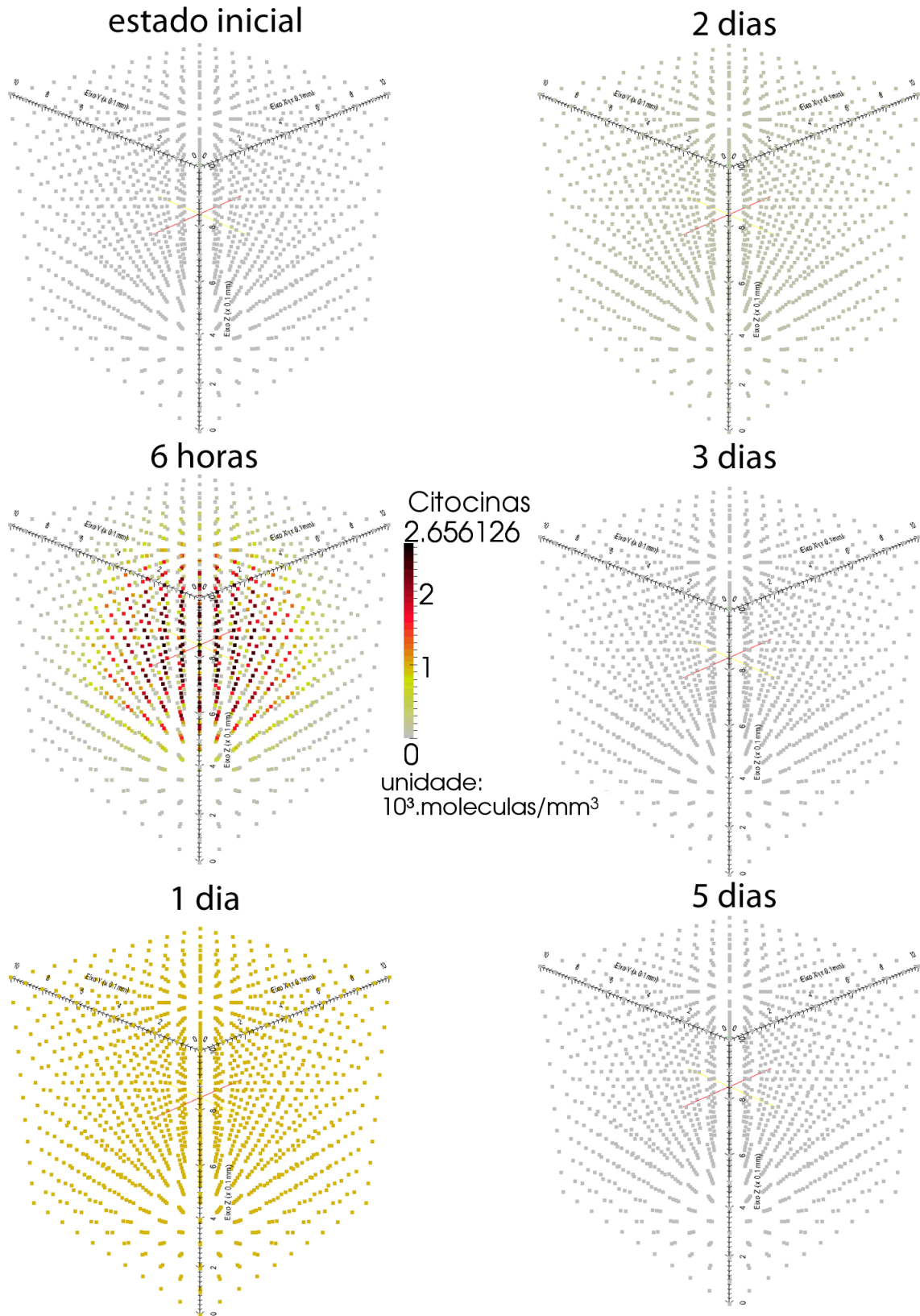


Figura 6.33: Distribuição espacial e temporal de citocinas pró-inflamatórias.



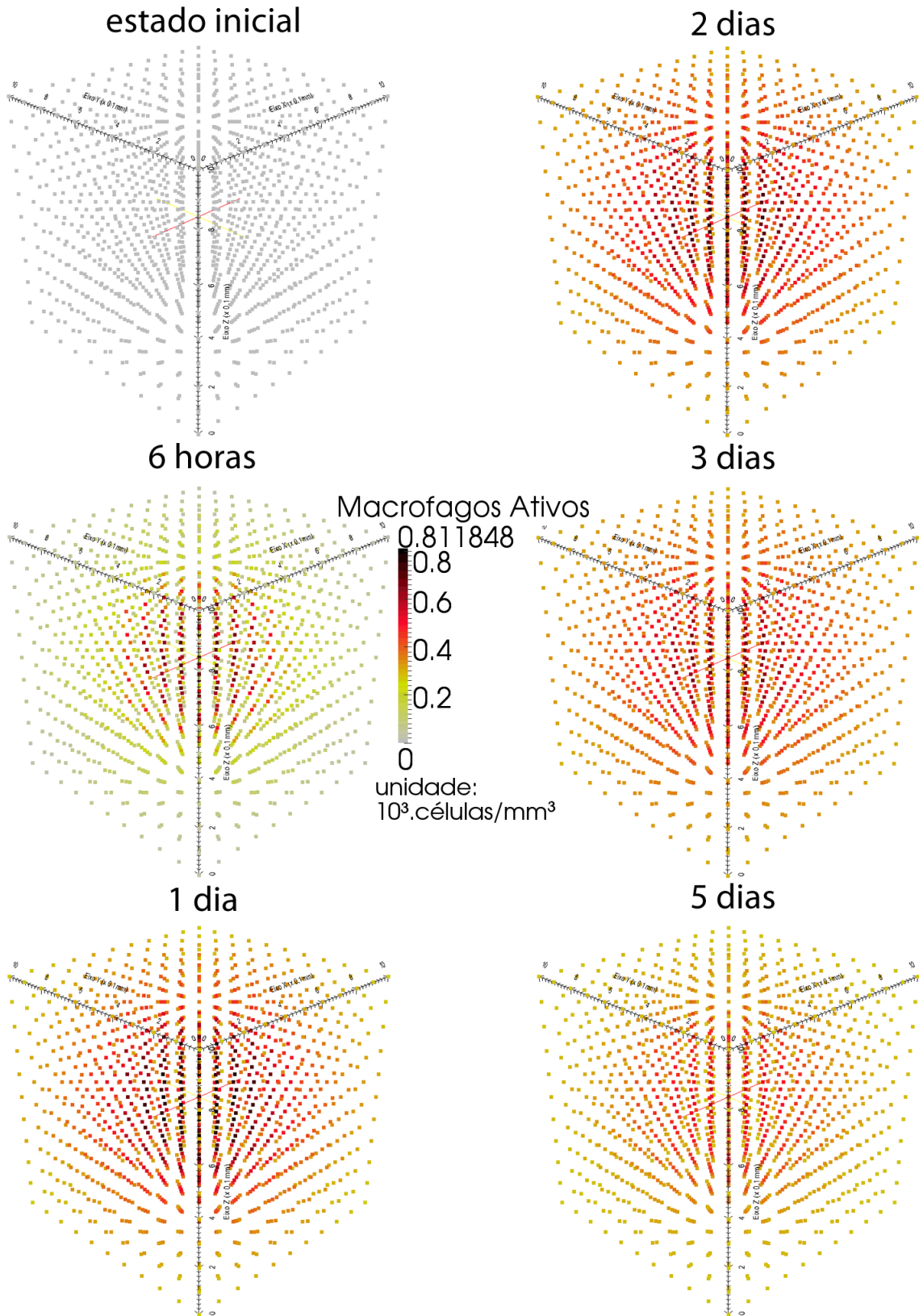


Figura 6.34: Distribuição espacial e temporal de macrófagos ativos.

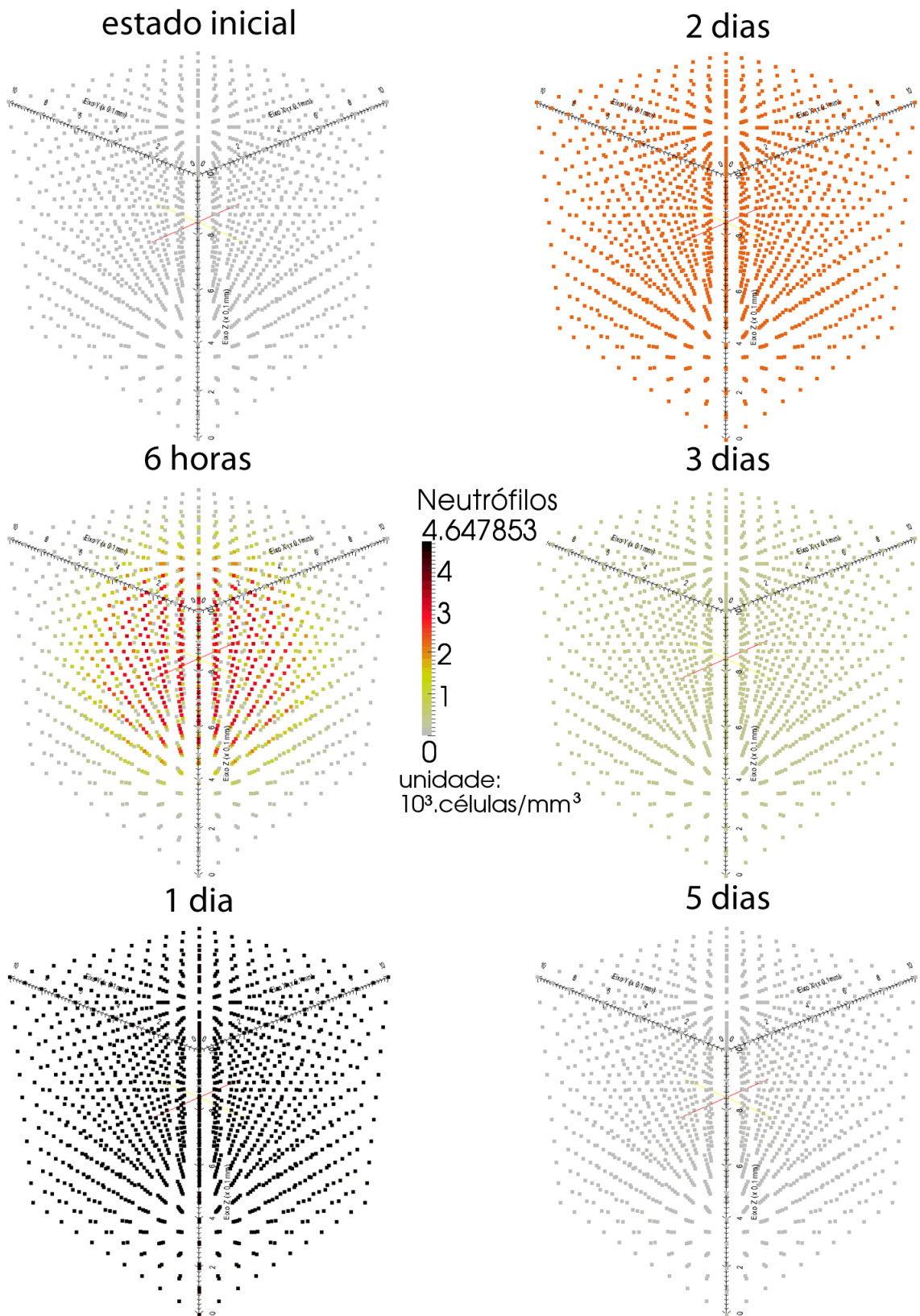


Figura 6.35: Distribuição espacial e temporal de neutrófilos.



presentes. Com sua presença, o LPS é removido em apenas 2 dias, enquanto que na sua ausência 5 dias são necessários para sua completa eliminação.

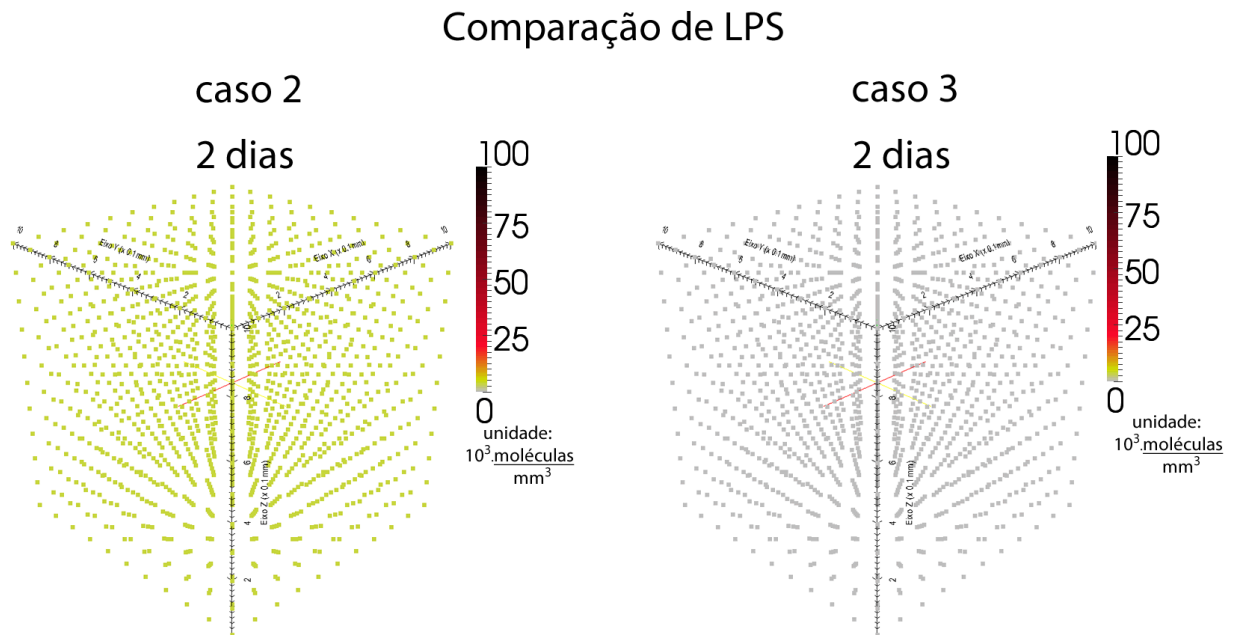


Figura 6.36: Distribuição espacial e temporal de LPS.

Outro ponto a ser ressaltado é que as células mais numerosas na resposta do sistema inato são os neutrófilos [7], extremamente importantes na produção de IL-8 (citocina pró-inflamatória) [16, 17]. A Figura 6.37 ilustra esta importância: a produção de citocinas é duplicada com a inclusão de neutrófilos no tecido, que contribui para a eliminação do LPS no tecido. Afinal, com uma maior concentração de IL-8, há um aumento da permeabilidade do tecido, o que provoca também um aumento do recrutamento de novos monócitos e neutrófilos.

Por fim, a Figura 6.38 compara a quantidade de macrófagos em repouso. Observa-se que, apesar do aumento da permeabilidade, há uma alta concentração de macrófagos em repouso por todo o tecido. Como os neutrófilos são os grandes responsáveis pela eliminação do LPS, uma menor quantidade de macrófagos em repouso se torna ativa. Além disso, por haver uma maior produção de citocinas, há um aumento na quantidade de monócitos que entram no tecido. Assim, a quantidade de macrófagos em repouso se mantém mais alta no caso 3 do que no caso 2.

## Comparação de citocinas

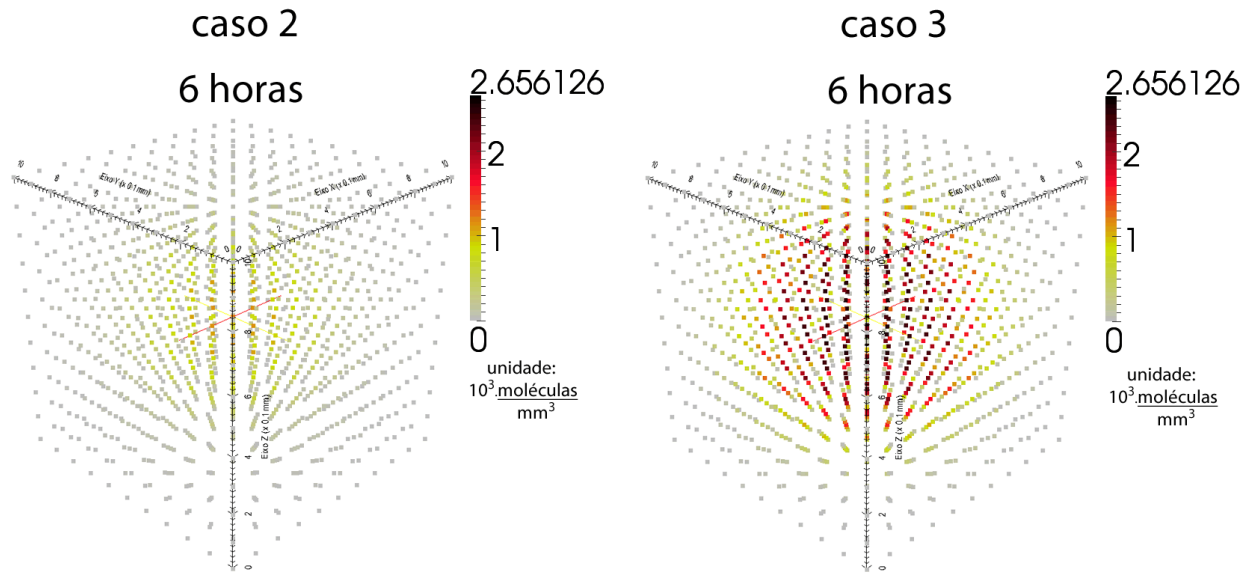


Figura 6.37: Distribuição espacial e temporal de neutrófilos.

## Comparação de macrófagos em repouso

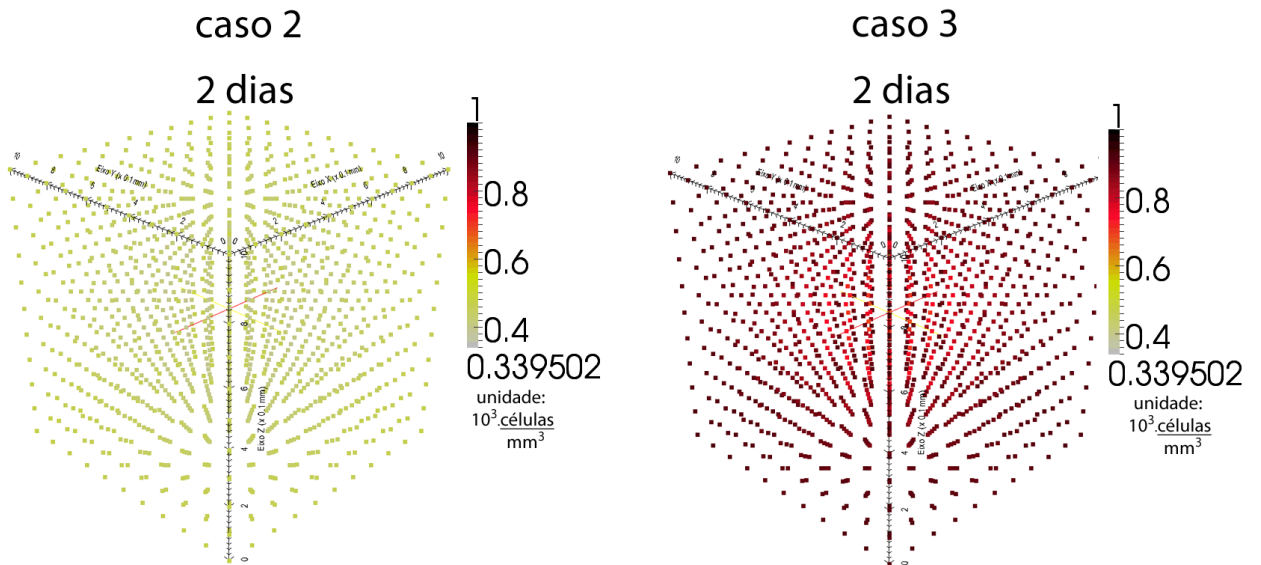


Figura 6.38: Distribuição espacial e temporal de macrófagos em repouso.

## 6.3 Aceleração Obtida

Esta seção apresenta os resultados, em termos de aceleração, obtidos pelo uso de GPUs nas execuções de ambos os sistemas. Novamente divide-se o texto em duas partes: as acelerações para a análise de sensibilidade e para a execução do modelo 3D.

### 6.3.1 Aceleração da Análise de Sensibilidade

Para fins de comparação de desempenho, foi criada também uma versão sequencial do código da análise de sensibilidade. Tanto o código sequencial quanto o código paralelo foram implementados utilizando a linguagem C. O método explícito de Euler foi usado para implementar a solução numérica do sistema de EDOs com passos de tempo igual a 0,0001 dias. O modelo simulado representa um sistema que descreve a dinâmica das células em um período equivalente a 5 dias após a infecção inicial.

Os experimentos foram realizados em um processador Intel Core i7-860 2.8 GHz, com 8 GB RAM, 32 KB de cache L1, 8 MB de cache L2 e com a placa de vídeo NVIDIA GeForce GTX 285. O sistema utiliza a versão 64-bits do Linux kernel 2.6.31 e CUDA toolkit versão 3.0. Utilizou-se também a versão 4.4.2 do *gcc* para compilar todas as versões do código. A placa de vídeo NVIDIA GeForce GTX 285 possui 240 *stream processors*, 30 multi-processadores, cada um com 16KB de memória compartilhada, e 1GB de memória global. A quantidade de blocos utilizados foi de 879, e cada bloco utilizou 512 *threads*. O código foi executado três vezes em todas as versões simuladas, então a média do tempo de execução foi calculada. O desvio padrão obtido foi insignificante. A aceleração (*speedup*) foi obtida dividindo o tempo de execução sequencial do simulador por sua versão em paralelo. Os resultados são apresentados na Tabela .

Sequencial	Paralelo	<i>Speedup</i>
4.315,47s	15,63s	276,12

Tabela 6.1: Tempo de execução sequencial e paralelo (utilizando CUDA em execução na placa GTX 285). Tempo representado em segundos.

Os resultados revelam que a versão CUDA foi responsável por acelerar a execução da aplicação em 276 vezes. O código sequencial demorava em torno de 1 hora e 12 minutos para executar, enquanto o código paralelo executou em apenas 15,63 segundos. Em particular, o ganho computacional deve-se à quantidade enorme de dados e ao fato de não haver nenhuma dependência e/ou comunicação entre as tarefas paralelas.

### 6.3.2 Aceleração do Modelo Tridimensional

Nesta subseção são descritos os ganhos de desempenho decorrentes do emprego de GPUs em comparação a versão sequencial do código. Tanto o código sequencial como o paralelo

foram executados em uma máquina com dois processadores Intel Xeon E5620, cada um com 4 núcleos, ou seja, 8 núcleos no total. Cada núcleo possui 64 KB de cache L1 e 256 KB de cache L2. A placa de vídeo utilizada em conjunto com este computador foi a GPU Tesla M2050 que tem 448 núcleos e 2,6 GB de memória global. O *gcc* 4.1.2 foi utilizado para compilar o código sequencial e, para a versão paralela, o *nvcc* versão 3.2. O tempo de execução foi obtido por meio de uma média entre 5 execuções, e o desvio padrão teve valor menor do que 0.95%. Mediu-se cada execução do código utilizando a aplicação Linux *time*. Para realizar as simulações foram usados os valores temporais e espaciais listados na Tabela já exposta 5.1.

Novamente a aceleração foi calculada através da divisão do tempo médio de execução sequencial sobre o tempo médio de execução paralelo. Para avaliar o impacto das otimizações implementadas, os tempos de execução paralelos apresentados na Tabela 6.2 são divididos em dois tempos: os obtidos sem o uso de otimizações e com uso de otimizações.

Tabela 6.2: Aceleração obtida pelo uso de GPUs. O tempo médio de execução é expresso em segundos.

<b>Versão</b>	<b>Tempo médio de execução</b>	<b><i>Speedup</i></b>
Sequencial	5.362,38s	-
Paralelo Inicial	278,16s	19,28
Paralelo Otimizado	61,87s	86,67

A primeira versão do código, ainda sem as otimizações apresentadas anteriormente, obteve uma aceleração de cerca de 19 vezes no tempo de execução. Já com o uso das otimizações, o ganho no desempenho chegou a aproximadamente 87 vezes. Em termos de horas/minutos/segundos, o código sequencial demorou em torno de 1 hora e 30 minutos para executar, enquanto o código paralelo (sem otimizações) gastou em torno de 4 minutos e 38 segundos e o código otimizado gastou apenas 1 minuto e 2 segundos.

# 7 CONCLUSÕES E PERSPECTIVAS FUTURAS

Este trabalho desenvolveu uma modelagem matemática-computacional da dinâmica das principais células e moléculas encontradas no SIH inato. Foram realizados estudos relacionados à resposta imune ao lipopolissacarídeo (LPS), tanto em um caso temporal quanto em uma seção de tecido tridimensional. Para o sistema temporal, atingiu-se o objetivo de analisar e capturar os principais parâmetros envolvidos no sistema por meio de uma massiva análise de sensibilidade. Já para o sistema tridimensional, o objetivo de simular a dinâmica espacial das principais células e moléculas envolvidas na resposta do SIH inato também foi atingido. Contudo, deve-se ressaltar que, em ambos os casos a versão sequencial do código apresentou problemas de desempenho devido ao grande conjunto de interações entre os componentes do modelo. Neste sentido, uma das principais contribuições do trabalho foi a implementação de ambos os sistemas para execução paralela em GPUs. Além do mais, os códigos foram otimizados para que o desempenho obtido fosse ainda melhor.

O resultados obtidos pela análise de sensibilidade mostraram a importância de dois parâmetros, permeabilidade do tecido e taxa de fagocitose realizada pelos neutrófilos, no modelo computacional da resposta do SIH inato. Pode-se ainda observar, pelos resultados da simulação do modelo 3D, que neutrófilos e citocinas exercem um papel fundamental no estabelecimento de uma resposta imune mais vigorosa contra o LPS.

Em resumo, este trabalho conseguiu identificar, através de simulações, várias características importantes para resposta imune inata, como:

- Os parâmetros que devem ser ajustados de maneira mais cuidadosa nas simulações do SIH inato;
- O importante papel de eliminação de LPS exercido pelos neutrófilos;
- A importância das citocinas no processo de aumento de permeabilidade do endotélio;
- A dinâmica das células e moléculas em um sistema temporal tridimensional, o que

torna as simulações mais próximas de um sistema real.

Como trabalhos futuros, a princípio, pode-se citar o incremento do modelo deste trabalho, de modo a incorporar mais células e moléculas presentes na resposta imune, como células NK e células dendríticas. Também podem ser incorporadas outras células e moléculas presentes no modelo proposto por Pigozzo [3]: grânulos proteicos e citocinas anti-inflamatórias.

Em relação à análise de sensibilidade, nota-se que muitos cenários foram produzidos em um pequeno intervalo de tempo. Porém, o estudo por meio da análise de cada cenário demanda bastante tempo e dedicação. Neste sentido pode-se vislumbrar o desenvolvimento de uma ferramenta computacional que identifique automaticamente quais são os parâmetros mais relevantes do sistema, baseado nas variações sofridas pelas variáveis dependentes, em relação ao seus valores padrões, ao longo do tempo.

Outro trabalho, de equivalente importância, seria o aumento na quantidade de pontos no espaço discretizado, de modo a aumentar o tamanho do tecido simulado e/ou a precisão dos resultados obtidos. Por fim, pode ser desenvolvido um estudo que une os dois aspectos abordados neste trabalho. Ou seja, aplicar uma análise de sensibilidade massiva em um modelo tridimensional do SIH inato, o que também acarretará um grande custo computacional. As otimizações descritas ao longo deste trabalho podem ser utilizadas e aperfeiçoadas neste novo e desafiador contexto. Em particular, pode-se empregar múltiplas GPUs, de modo simultâneo, para realizar tais simulações. Contudo, o custo de comunicação entre os múltiplos dispositivos pode se mostrar proibitivo, exigindo o desenvolvimento de novas técnicas para contornar este problema.

## REFERÊNCIAS

- [1] PIGOZZO, A., *Implementação Computacional de um Modelo Matemático do Sistema Imune Inato*, Master's Thesis, Universidade Federal de Juiz de Fora, 2011.
- [2] NVIDIA, *Nvidia cuda programming guide*, Technical report, VIDIA Corporation, 2007.
- [3] PIGOZZO, A. B., MACEDO, G. C., DOS SANTOS, R. W., LOBOSCO, M., “Implementation of a Computational Model of the Innate Immune System”. In: *ICARIS*, pp. 95–107, 2011.
- [4] PIGOZZO, A. B., LOBOSCO, M., DOS SANTOS, R. W., “Parallel Implementation of a Computational Model of the HIS Using OpenMP and MPI”. In: *International Symposium on Computer Architecture and High Performance Computing Workshops*, pp. 67–72, IEEE Computer Society, 2010.
- [5] PACHECO, P. S., *Parallel programming with MPI*. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1996.
- [6] CHANDRA, R., DAGUM, L., KOHR, D., MAYDAN, D., MACDONALD, J., MENON, R., *Parallel Programming in OpenMP*. 1st ed. Morgan Kaufmann Publishers, 2001.
- [7] MURPHY, K. M., TRAVERS, P., WALPORT, M., *Imunobiologia*. 7th ed. Garland Science, November 2010.
- [8] KINDT, T. J., OSBORNE, B. A., GOLDSBY, R. A., *Kuby Immunology*. W. H. Freeman, July 2006.
- [9] SOMPAYRAC, L. M., *How the Immune System Works*. Wiley, John & Sons, Incorporated, 2008.
- [10] MCCORMICK, T. S., STEVENS, S. R., KANG, K., “Macrophages and cutaneous inflammation”, *Nat Biotech*, v. 18, n. 1, pp. 25–26, Jan. 2000.
- [11] N, F., K., K., “Macrophages in inflammation”, *Current Drug Targets - Inflammation & Allergy*, v. 4, n. 3, pp. 281–286(6), June 2005.

- [12] LJ., O., “Tumor necrosis factor”, *Scientific American*, v. 258, 1988.
- [13] FILBET, F., “A finite volume scheme for the Patlak–Keller–Segel chemotaxis model”, *Numerische Mathematik*, v. 104, pp. 457–488, 2006.
- [14] MALLICK, A., ISHIZAKA, A., STEPHENS, K., HATHERILL, J., TAZELAAR, H., RAFFIN, T., “Multiple organ damage caused by tumor necrosis factor and prevented by prior neutrophil depletion”, *Chest*, v. 95, 1989.
- [15] HORGAN, M. J., PALACE, G. P., EVERITT, J. E., MALIK, A. B., “TNF-alpha release in endotoxemia contributes to neutrophil-dependent pulmonary edema”, *American Journal of Physiology - Heart and Circulatory Physiology*, v. 264, n. 4, pp. H1161–H1165, April 1993.
- [16] BAZZONI, F., CASSATELLA, M. A., ROSSI, F., CESKA, M., DEWALD, B., BAGGIOLINI, M., “Phagocytosing neutrophils produce and release high amounts of the neutrophil-activating peptide 1/interleukin 8.” *The Journal of Experimental Medicine*, v. 173, n. 3, pp. 771–774, March 1991.
- [17] STRIETER., R. M., KASAHARA., K., ALLEN., R. M., STANDIFORD., T. J., ROLFE., M. W., BECKER., F. S., CHENSUE., S. W., KUNKEL, S. L., “Cytokine-induced neutrophil-derived interleukin-8”, *Am J Pathol*, v. 141, pp. 397–407, 1992.
- [18] M, B., B, D., B, M., “Human chemokines: an update”, *Annu Rev Immunol*, v. 15, pp. 675–705, 1997.
- [19] SAKLATVALA, J., DAVIS, W., GUESDON, F., KARIN, M., MARSHALL, C. J., “Interleukin 1 (IL1) and Tumour Necrosis Factor (TNF) Signal Transduction”, *Philosophical Transactions: Biological Sciences*, v. 351, n. 1336, pp. 151–157, Feb. 1996.
- [20] MULLIGAN, M., JONES, M., BOLANOWSKI, M., BAGANOFF, M., DEPPELER, C., MEYERS, D., RYAN, U., WARD, P., “Inhibition of lung inflammatory reactions in rats by an anti-human IL-8 antibody”, *The Journal of Immunology*, v. 150, n. 12, pp. 5585–5595, June 1993.



- [21] SEKIDO, N., MUKAIDA, N., HARADA, A., NAKANISHI, I., WATANABE, Y., MATSUSHIMA, K., “Prevention of lung reperfusion injury in rabbits by a monoclonal antibody against interleukin-8”, *Nature*, v. 365, n. 6447, pp. 654–657, Oct. 1993.
- [22] BROADDUS, V., BOYLAN, A., HOEFFEL, J., KIM, K., SADICK, M., CHUNTHARAPAI, A., HEBERT, C., “Neutralization of IL-8 inhibits neutrophil influx in a rabbit model of endotoxin-induced pleurisy”, *The Journal of Immunology*, v. 152, n. 6, pp. 2960–2967, March 1994.
- [23] ENG, R. H., SMITH, S. M., FAN-HAVARD, P., OGBARA, T., “Effect of antibiotics on endotoxin release from gram-negative bacteria”, *Diagnostic Microbiology and Infectious Disease*, v. 16, n. 3, pp. 185–189, 1993.
- [24] ZHANG, B., HIRAHASHI, J., CULLERE, X., MAYADAS, T. N., “Elucidation of Molecular Events Leading to Neutrophil Apoptosis following Phagocytosis”, *The Journal of biological chemistry*, v. 278, pp. 28443–28454, 2003.
- [25] GOUTELLE, S., MAURIN, M., ROUGIER, F., BARBAUT, X., BOURGUIGNON, L., DUCHER, M., MAIRE, P., “The Hill equation: a review of its capabilities in pharmacological modelling”, *Fundamental & clinical pharmacology*, v. 22, n. 6, pp. 633–648, December 2008.
- [26] WAGNER, J. G., “Kinetics of pharmacologic response I. Proposed relationships between response and drug concentration in the intact animal and man”, *Journal of Theoretical Biology*, v. 20, n. 2, pp. 173 – 201, 1968.
- [27] VOLL, R. E., HERRMANN, M., ROTH, E. A., STACH, C., KALDEN, J. R., GIRKONTAITE, I., “Immunosuppressive effects of apoptotic cells”, *Nature*, v. 390, n. 6658, pp. 350–351, Nov. 1997.
- [28] REITZ, J. R., MILFORD, F. J., CHRISTY, R. W., *Fundamentos da Teoria Eletromagnética*. 3rd ed. Editora Campus.
- [29] GPGPU.ORG, “About GPGPU.org”, 201, [Online; accessed 1-Agosto-2012].
- [30] KIRK, D., HWU, W., *Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 2010.

- [31] SALTELLI, A., RATTO, M., ANDRES, T., CAMPOLONGO, F., CARIBONI, J., GATELLI, D., SAISANA, M., TARANTOLA, S., *Global Sensitivity Analysis: The Primer*. Wiley, 2008.
- [32] LEVEQUE, R. J., *Finite Difference Methods for Ordinary and Partial Differential Equations*. Society for Industrial and Applied Mathematics, 2007.
- [33] HAFEZ, M. M., CHATTOT, J. J., *Innovative Methods for Numerical Solution of Partial Differential Equations*. World Scientific Publishing Company, 2002.
- [34] MALITS, R., BOLOTIN, E., KOLODNY, A., MENDELSON, A., “Exploring the Limits of GPGPU Scheduling In Control Flow Bound Applications”, *ACM Transactions on Architecture and Code Optimization*, v. 8, n. 4, pp. 267–276, 2012.
- [35] SU, B., ZHOU, W., DORMAN, K. S., JONES, D. E., “Mathematical modelling of immune response in tissues”, *Computational and Mathematical Methods in Medicine: An Interdisciplinary Journal of Mathematical, Theoretical and Clinical Aspects of Medicine*, v. 10, pp. 1748–6718, 2009.
- [36] PRICE, T., OCHS, H., GERSHONI-BARUCH, R., HARLAN, J., ETZIONI, A., “In vivo neutrophil and lymphocyte function studies in a patient with leukocyte adhesion deficiency type II”, *Blood*, v. 84, n. 5, pp. 1635–1639, Sept. 1994.
- [37] DE WAAL MALEFYT, R., ABRAMS, J., BENNETT, B., FIGDOR, C., DE VRIES, J., “Interleukin 10(IL-10) inhibits cytokine synthesis by human monocytes: an autoregulatory role of IL-10 produced by monocytes.” *J Exp Med*, v. 174, n. 5, pp. 1209–20–, Nov. 1991.
- [38] FELDER, S., KAM, Z., “Human neutrophil motility: Time-dependent three-dimensional shape and granule diffusion”, *Cell Motility and the Cytoskeleton*, v. 28, n. 4, pp. 285–302, 1994.
- [39] CHETTIBI, S., LAWRENCE, A., YOUNG, J., LAWRENCE, P., STEVENSON, R., “Dispersive locomotion of human neutrophils in response to a steroid-induced factor from monocytes”, *J Cell Sci*, v. 107, n. 11, pp. 3173–3181, Nov. 1994.

# APÊNDICE A - CONDIÇÕES INICIAIS E PARÂMETROS

Tabela A.1: Condições iniciais, parâmetros e unidades da Análise de Sensibilidade.

Parâmetro	Valor padrão	Unidade	Referência	Ajuste
$N_0$	$N(0) = 0$	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	baseado em [1]	0 a 4, passo:0.04
$CH_0$	$CH(0) = 0$	$\frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$	baseado em [1]	0 a 4, passo:0.04
$A_0$	$A(0) = 20$	$\frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$	baseado em [1]	0 a 40, passo:0.4
$MR_0$	$MR(0) = 1$	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	baseado em [1]	0 a 4, passo:0.04
$MA_0$	$MA(0) = 0$	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	baseado em [1]	0 a 4, passo:0.04
$\mu_{CH}$	7	$\frac{1}{\text{dia}}$	[1]	passo de 2%
$\mu_N$	3.43	$\frac{1}{\text{dia}}$	[1]	passos de 20%
$\mu_A$	0	$\frac{1}{\text{dia}}$	[35]	passos de 20%
$\mu_{MR}$	0.033	$\frac{1}{\text{dia}}$	[35]	passos de 20%
$\mu_{MA}$	0.07	$\frac{1}{\text{dia}}$	[35]	passos de 20%
$\lambda_{N A}$	0.55	$\frac{1}{\text{dia}}$	[35]	passo:2%
$\lambda_{A N}$	0.55	$\frac{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}{1}$	[35]	passo:2%
$\lambda_{MR A}$	0.1	$\frac{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}{1}$	[1]	passo:2%
$\lambda_{MA A}$	0.8	$\frac{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}{1}$	[35]	passo:2%
$\beta_{CH N}$	1	$\frac{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}{1}$	[1]	passo:2%
$\beta_{CH MA}$	0.8	$\frac{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}{1}$	[1]	passo:20%
$N^{max}$	8	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	[1]	passo de 2%
$M^{max}$	6	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	[1]	passo de 2%
$P_N^{max}$	11.4	$\frac{1}{\text{dia}}$	baseado em [36]	passo:20%
$P_N^{min}$	0.0001	$\frac{1}{\text{dia}}$	[1]	passo:20%
$P_{MR}^{max}$	0.1	$\frac{1}{\text{dia}}$	[1]	passo:20%
$P_{MR}^{min}$	0.01	$\frac{1}{\text{dia}}$	[1]	passo:20%
$I_{CH}$	3.6	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	baseado em [37]	passo:20%
$K_{CH}$	1	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	[1]	passo:20%

Tabela A.2: Condições iniciais do modelo tridimensional.

Parâmetro	Valor	Unidade	Referência
$N_0$	$N(x, y, z, 0) = 0, (x, y, z) \in [0, 1]$	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	[1]
$CH_0$	$CH(x, y, z, 0) = 0, (x, y, z) \in [0, 1]$	$\frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$	[1]
$A_0$	$A(x, y, z, 0) = 100, se(x, y, z) \in [0.3, 0.7]$	$\frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$	baseado em [1]
$A_0$	$A(x, y, z, 0) = 0, \text{caso contrário}$	$\frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}$	baseado em [1]
$MR_0$	$MR(x, y, z, 0) = 1, (x, y, z) \in [0, 1]$	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	[1]
$MA_0$	$MA(x, y, z, 0) = 0, (x, y, z) \in [0, 1]$	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	[1]

Tabela A.3: Parâmetros do modelo tridimensional.

Parâmetro	Valor	Unidade	Referência
$\mu_N$	3.43	$\frac{1}{\text{dia}}$	[1]
$\mu_{CH}$	7	$\frac{1}{\text{dia}}$	[1]
$\mu_A$	0	$\frac{1}{\text{dia}}$	[35]
$\mu_{MR}$	0.033	$\frac{1}{\text{dia}}$	[35]
$\mu_{MA}$	0.07	$\frac{1}{\text{dia}}$	[35]
$\lambda_{N A}$	0.55	$\frac{1}{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}$	[35]
$\lambda_{A N}$	0.55	$\frac{1}{(\frac{10^3 \cdot \text{moleculas}}{\text{mm}^3}) \cdot \text{dia}}$	[35]
$\lambda_{MR A}$	0.1	$\frac{1}{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}$	[1]
$\lambda_{MA A}$	0.8	$\frac{1}{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}$	[35]
$\beta_{CH N}$	1	$\frac{1}{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}$	[1]
$\beta_{CH MA}$	0.8	$\frac{1}{(\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}) \cdot \text{dia}}$	[1]
$N^{max}$	8	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	[1]
$M^{max}$	6	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	[1]
$P_N^{max}$	11.4	$\frac{1}{\text{dia}}$	baseado em [36]
$P_N^{min}$	0.0001	$\frac{1}{\text{dia}}$	[1]
$P_{MR}^{max}$	0.1	$\frac{1}{\text{dia}}$	[1]
$P_{MR}^{min}$	0.01	$\frac{1}{\text{dia}}$	[1]
$I_{CH}$	3.6	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	baseado em [37]
$K_{CH}$	1	$\frac{10^3 \cdot \text{celulas}}{\text{mm}^3}$	[1]
$D_N$	0.012096	$\frac{\text{mm}^2}{\text{dia}}$	[38]
$D_{CH}$	0.009216	$\frac{\text{mm}^2}{\text{dia}}$	[35]
$D_A$	0.2	$\frac{\text{mm}^2}{\text{dia}}$	[1]
$D_{MR}$	0.00432	$\frac{\text{mm}^2}{\text{dia}}$	[1]
$D_{MA}$	0.003	$\frac{\text{mm}^2}{\text{dia}}$	[1]
$\chi_N$	0.0144	$\frac{\text{mm}^2}{\text{dia}}$	[39]
$\chi_{MR}$	0.0036	$\frac{\text{mm}^2}{\text{dia}}$	[1]
$\chi_{MA}$	0.00432	$\frac{\text{mm}^2}{\text{dia}}$	[1]

Tabela A.4: Lista de Símbolos 1

Símbolo	Descrição
$N$	concentração de neutrófilos
$CH$	concentração de citocinas
$A$	concentração de antígenos - LPS
$MR$	concentração de macrófagos em repouso
$MA$	concentração de macrófagos ativados
$\mu_N$	taxa de apoptose de neutrófilos
$\mu_{CH}$	taxa de decaimento de citocinas
$\mu_A$	taxa de decaimento de antígenos
$\mu_{MR}$	taxa de apoptose de macrófagos em repouso
$\mu_{MA}$	taxa de apoptose de macrófagos ativados
$\lambda_{N A}$	taxa de fagocitose de LPS realizada por neutrófilos
$\lambda_{A N}$	taxa de apoptose de neutrófilos devido a fagocitose
$\lambda_{MR A}$	taxa de fagocitose de LPS realizada por macrófagos em repouso
$\lambda_{MA A}$	taxa de fagocitose de LPS realizada por macrófagos ativados
$\beta_{CH N}$	taxa de produção de citocinas por neutrófilos
$\beta_{CH MA}$	taxa de produção de citocinas por macrófagos ativados
$N^{max}$	quantidade máxima de neutrófilos suportado pelo tecido
$M^{max}$	quantidade máxima de macrófagos(repouso e ativados) suportado pelo tecido
$P_N^{max}$	taxa máxima de aumento de permeabilidade para neutrófilos
$P_N^{min}$	taxa mínima de aumento de permeabilidade para neutrófilos
$P_{MR}^{max}$	taxa máxima de aumento de permeabilidade para macrófagos em repouso
$P_{MR}^{min}$	taxa mínima de aumento de permeabilidade para macrófagos em repouso
$I_{CH}$	concentração máxima que o tecido suporta de citocinas pró-inflamatórias
$K_{CH}$	concentração de citocinas pró-inflamatórias que exercem 50% do efeito máximo no aumento da permeabilidade
$D_N$	coeficiente de difusão de neutrófilos
$D_{CH}$	coeficiente de difusão de citocinas
$D_A$	coeficiente de difusão de antígenos - LPS
$D_{MR}$	coeficiente de difusão de macrófagos em repouso
$D_{MA}$	coeficiente de difusão de macrófagos ativados
$\chi_N$	taxa de quimiotaxia para neutrófilos
$\chi_{MR}$	taxa de quimiotaxia para macrófagos em repouso
$\chi_{MA}$	taxa de quimiotaxia para macrófagos ativados

Tabela A.5: Lista de Símbolos 2

<b>Símbolo</b>	<b>Descrição</b>
$\Delta$	laplaciano
$\nabla$	divergente e gradiente
$\Delta T$	discretização temporal
$\Delta X$	discretização eixo X
$\Delta Y$	discretização eixo Y
$\Delta Z$	discretização eixo Z
$O$	concentração de neutrófilos, antígenos, citocinas ou macrófagos
$D_O$	coeficiente de difusão da concentração de neutrófilos, antígenos, citocinas ou macrófagos