Sensitivity Analysis of a Two-compartmental Differential Equation Mathematical Model of MS using Parallel Programming

Matheus A M de Paula[1,2], Gustavo G Silva[2], Marcelo Lobosco[1,2 [0000−0002−7205−9509]], and Bárbara M Quintela[1,2 [0000−0002−5320−4109]]

1 Graduate Program in Computational Modelling (PGMC), Federal University of Juiz de Fora (UFJF), Juiz de Fora-MG, Brazil
2 FISIOCOMP, Computational Physiology and High-Performance Computing Laboratory, Federal University of Juiz de Fora (UFJF), Juiz de Fora-MG, Brazil

Abstract. Multiple Sclerosis (MS) is a neurodegenerative disease that involves a complex sequence of events in distinct spatiotemporal scales for which the cause is not completely understood. The representation of such biological phenomena using mathematical models can be useful to gain insights and test hypotheses to improve the understanding of the disease and find new courses of action to either prevent it or treat it with fewer collateral effects. To represent all stages of the disease, such mathematical models are frequently computationally demanding. This work presents a comparison of parallel programming strategies to optimize the execution time of a spatiotemporal two-compartmental mathematical model to represent plaque formation in MS and apply the best strategy found to perform the sensitivity analysis of the model.

Keywords: Computational Biology · Parallel Programming · Multiple Sclerosis.

1 Introduction

Multiple Sclerosis (MS) is a neurodegenerative disease with onset in early adulthood, and the most common form is Relapse-Remitting MS (RRMS), characterized by symptom attacks and remission phases [4]. Mathematical models have been applied to other brain diseases [1, 9, 13], and several models have been proposed for MS, including models of self-tolerance [7], drug effects [16], disease progression [15], and initial damage by the immune system. Our previous paper [3] proposed a new model to represent the damage caused by MS, influenced by the dynamics of immune system activation in the nearest lymph node. The main objective of this work is to identify critical model parameters in each phase of MS using sensitivity analysis. However, sensitivity analysis is a computationally expensive study, and since MS is a long-term disease, the model needs to simulate disease progression over long periods, requiring a parallel implementation to reduce the computation time.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_68

This paper is organized as follows. Section 2 briefly reviews the main characteristics of our mathematical model. The same section also presents the techniques used to develop a parallel version of the code and the sensitivity analysis used in this work. Section 3 presents and discusses the results. Finally, Section 4 presents our conclusions and plans for future works.

2 Methods

2.1 Mathematical Model

MS is characterized by an infiltration of lymphocytes across the blood-brain barrier into the brain. The microglia is stimulated to attack the oligodendrocytes. After being destroyed, brain parenchymal cells are captured by dendritic cells. Thus, the dendritic cells become activated and migrate to the lymph node, acting as presenting antigen cells (APC) and stimulating adaptive immune system [10, 14]. Then, CD8$^+$ T cells and antibodies migrate to the brain parenchyma [12]. CD8$^+$ T cell stimulates microglia and attacks brain cells, while antibodies do the opsonization of brain cells [11, 17].

The mathematical model represented in Figure 1 comprises two distinct compartments: the brain parenchyma (i.e., tissue) and the peripheral lymph node [3].

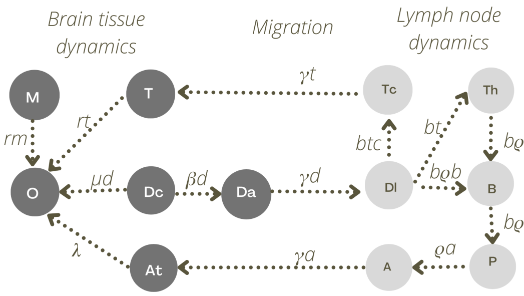

Fig. 1. Two-compartment mathematical model for MS. Dendritic cells migrate to the peripheral lymph node and activate the T and B cells recruiting antibodies. Adapted from [3].

The model employs a set of 6 PDEs to depict the dynamics of RRMS in the spatial domain Ω and temporal domain I. These PDEs correspond to the following entities: Microglia (M), Oligodendrocytes (O), Conventional Dendritic Cells (D_C), Activated Dendritic Cells (D_A), Antibodies (A_t), and $CD8^+$ T Cells (T). Additionally, a set of 6 ODEs, defined in the temporal domain I, represents the dynamics in the lymph node, namely: Activated Dendritic cells (D^L), $CD8^+$ T cells (T_C^L), $CD4^+$ T cells (T_H^L), B Cells (B), Plasma cells (P), and Antibodies (A_t^L). Details of the mathematical model can be consulted in our previous work [3].

2.2 Numerical Implementation

The code was implemented in C, utilizing the Finite Difference Method to solve the system of Partial Differential Equations. First-order central difference was used to numerically solve the diffusion, while chemotaxis was resolved using both first-order central difference and up-wind-down-wind. The system of ordinary differential equations was solved using the Explicit Euler method.

For our simulations, we assumed a two-dimensional domain of $20mm \times 20mm$, and the simulation time was set to represent 4 weeks of autoimmune response after the onset of microglia activation. All simulation parameters were obtained from our previous work [3].

2.3 Sensitivity Analysis

Sensitivity analysis (SA) [19] evaluates how changes in input parameters affect model outputs, helping researchers identify the most influential factors and improve model accuracy. SA is essential in assessing the robustness and reliability of mathematical models, especially those involving complex systems with many uncertain input parameters. Various methods are available to perform SA, including the one-factor-at-a-time (OFAT) approach, Sobol method, Latin hypercube sampling, and Fourier Amplitude Sensitivity Test (FAST) [20, 18, 6, 2]. In this work, the second-order Sobol method was used, with bounds set for each parameter, and samples generated to compute the model for each sample.

2.4 Parallel Implementation

While the second-order Sobol method can provide valuable insights into the behavior of complex models, it also comes with a significant computational demand due to the large number of model runs required to estimate the sensitivity indices accurately. More specifically, to evaluate the importance of each input variable and the interactions between them, the second-order Sobol method requires at least $N(2D+2)$ model runs, where N is the number of samples to generate, and D is the number of parameters of the model. This means that the number of model runs increases rapidly as the number of input variables and the maximum order of interaction increase. With $D = 34$ parameters and $N = 2048$, the second-order Sobol method requires the execution of $143,360$ instances of the model, which hurts performance. To overcome this issue, we implemented two parallel versions of the code, one using MPI and the other OpenMP.

The parallel solution for a coupled system of ODEs and PDEs can lead to additional communication and synchronization overheads, data dependency, and memory management issues that can impact the efficiency and accuracy of the parallelized version of the code.

In our mathematical coupled model, the exchange of data between the two compartments requires the use of synchronization primitives, *i.e.*, data must be exchanged, at each time step, between the system of PDEs, which solves the dynamics that occur in the brain tissue, and the system of ODE, which solves

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_68

the dynamics on the lymph node. In other words, before the next time-step starts, it is necessary to calculate the average concentration of those populations of cells in the tissue (PDEs) that migrate to the lymph node, so that the computation of the ODEs system in the next time-step starts with the populations updated. The same occurs with the antigen population, which must be updated before the PDEs system starts its next time-step computation. Finally, the advection and diffusion terms also require synchronization at each time step.

Solving the PDEs with OpenMP (OMP) To solve the model using OMP, the algorithm first creates a team of threads. Next, the ODEs system, which represents the lymph node compartment, is solved sequentially. Then, OMP is utilized to divide the tissue domain into slices and assign each slice to a previously created thread, which will solve the PDEs in parallel in the assigned part of the domain. Since all points of the mesh require the same amount of work, we kept the default static schedule. Finally, we calculate the average concentrations of cell populations in the tissue, which will be used to solve the ODEs system in the next time step.

Solving the PDEs with Message Passing Interface (MPI) To implement a parallel version of the code using MPI, the domain must be divided into smaller subdomains, and each subdomain must be assigned to a separate MPI process. Each process performs local computations for the ODEs system and solves the PDEs within its designated domain using locally available information. Afterwards, the resulting solutions must be communicated to neighboring domains to update the boundary conditions, as depicted in Figure 2. This communication is accomplished using `MPI_Send` and `MPI_Receive` operations. The process of solving the PDEs system and updating the boundary conditions is repeated iteratively until a converged solution is obtained.

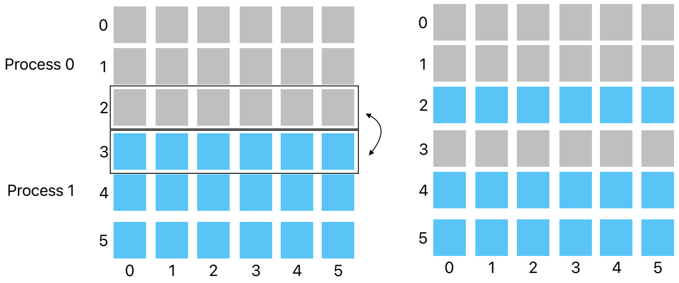

Fig. 2. To properly synchronize the mesh division between two processes, it is essential for each process to exchange borders with its neighboring processes at every time step.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_68

3 Results and Discussion

3.1 Computational Environment

The numerical model presented in our previous work [3] was implemented using the C programming language. The code was compiled using *gcc* version 9.4.0 with the optimization flag $-O3$ enabled. In this work, the second-order Sobol method was implemented with the help of the Python library `SALib` [5, 8] executed with Python 3.9.13. The code was executed in a 3.30 GHz Intel® Core$^{\text{TM}}$ $i5$-12600 CPU with 16GB of main memory. The number of physical cores available is 6, which is the maximum number of threads/processes used during simulations.

3.2 Results for the Parallel Execution

Table 1 presents the average wall clock time obtained for ten executions of each parallel version of the code. Each parallel version was executed using two, four, and six threads/processes. The Table also presents the standard deviation and the 95% confidence interval for each execution. As a reference, the sequential version of the code executes in 734±4.73 seconds. The spatial discretization used in this comparison was $h_x = 0.5$ mm, and the time-step used was $h_t = 2 \times 10^{-6}$ days.

Table 1. Average execution time, standard deviation and 95% confidence interval (in seconds).

	Average		**Standard Deviation**		**Confidence Interval**	
# of Threads	**MPI**	**OMP**	**MPI**	**OMP**	**MPI**	**OMP**
2	420	386	6.13	5.50	$(418, 423)$	$(382, 390)$
4	308	246	7.17	1.92	$(303, 309)$	$(245, 248)$
6	322.2	213	0.87	2.02	$(321, 323)$	$(211, 214)$

The OMP version running on 6 nodes presented the best performance, achieving a speedup close to 3.5. The best MPI version achieved a speedup of 2.4 in 4 nodes. This difference in performance can be attributed to communication costs, such as those depicted in Figure 2.

3.3 Sensitivity Analysis

We have run the SA considering all 34 parameters of the model. The OpenMP version with 6 threads was used to solve each sample generated by the Sobol method. The bounds used for each parameter by the Sobol method are ±10% the baseline values presented in our previous work [3]. The quantity of interest (QoI) is the total concentration of destroyed oligodendrocytes 28 days after the start of the microglia attack. To take advantage of the embarrassingly parallel nature of the SA analysis, where each run with a set of parameters is independent of other runs, we divided the execution into multiple instances that could be run in parallel on multiple machines.

Figure 3 presents the results for the first-order SA. The parameters that have more impact on the destruction of the brain cells are related to the increase in the microglia population: the rate of proliferation (μ_m), diffusion (d_M), and chemotaxis (χ). The decay rate of microglia (c_M) has a lower impact when compared to the other microglia terms, but it is also relevant to prevent/increase the destruction of oligodendrocytes.

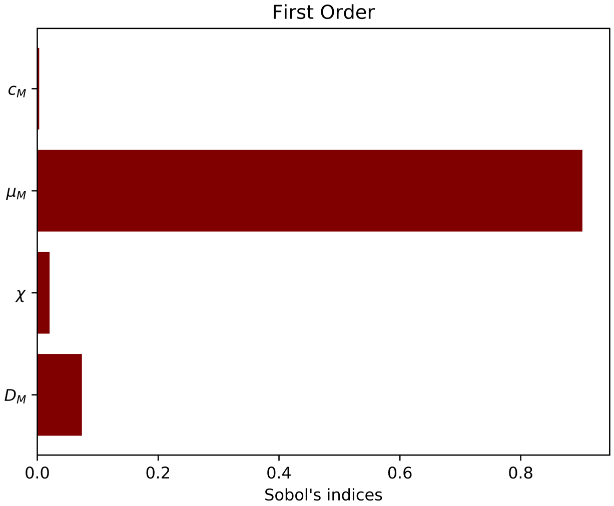

Fig. 3. Results from the first-order sensitivity analysis show that the parameters most relevant to oligodendrocyte destruction are those related to microglia proliferation and diffusion, while the other parameters appear to have no significant relevance.

Figure 4 shows the covariance of parameters obtained using the second-order Sobol method. As we can observe, there is a high covariance between μ_m and d_M. Additionally, the decay rate of microglia (c_M) is also relevant when combined with the parameters of the equations for lymph nodes and the decay rate of dendritic cells in the tissue.

The first and second-order Sobol's indices suggest a significant dependence of oligodendrocyte destruction on the concentration of microglia. Our current model configuration indicates a limited impact of other immune system cells on the disease dynamic. Therefore, it can be concluded that microglia' concentration plays a crucial role in determining the destruction of oligodendrocytes.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_68

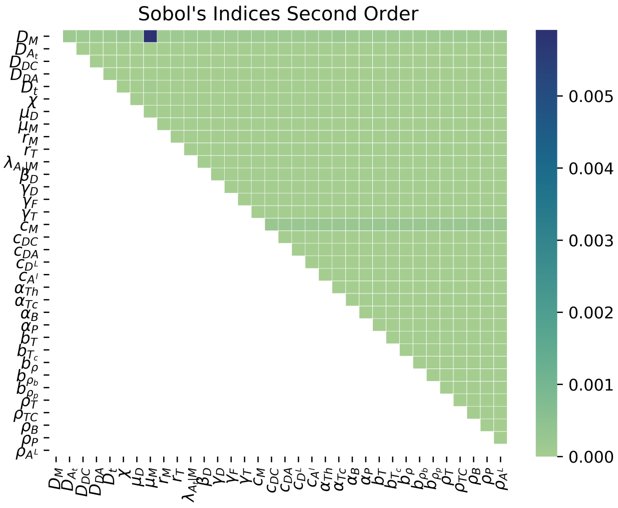

Fig. 4. Parameters covariance.

4 Conclusions

Our study compared two strategies for enhancing the performance of a differential equation model for Multiple Sclerosis (MS), namely using OpenMP and MPI. Based on the size of our problem, we found that the former strategy demonstrated better performance, enabling us to execute the model up to 3.5 times faster. As a result, we were able to conduct a Sensitivity Analysis (SA) for the model over a 28-day simulation period that represents the acute phase of MS.

We believe that conducting an SA to assess the impact of the parameters on chronic MS, which can last for years, would be a valuable next step. Our study contributes to the current understanding of MS dynamics by providing insight into the potential impact of different strategies for enhancing computational performance.

Acknowledgements This work has been supported by UFJF, by CAPES - Finance Code 001; by CNPq - Grant number 308745/2021-3; and by FAPEMIG Grant number APQ 02830/17 and APQ-02513-22; by FINEP SOS Equipamentos 2021 AV02 0062/22.

References

1. Broome, T.M., Coleman, R.A.: A mathematical model of cell death in multiple sclerosis. J. Neurosci. Methods **201**(2), 420–425 (Oct 2011)

2. Cukier, R.I., Fortuin, C.M., Shuler, K.E., et al.: Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. i theory. The Journal of Chemical Physics **59**(8), 3873–3878 (Oct 1973). https://doi.org/10.1063/1.1680571
3. de Paula, M.A.M., de Melo Quintela, B., Lobosco, M.: On the use of a coupled mathematical model for understanding the dynamics of multiple sclerosis. Journal of Computational and Applied Mathematics **428**, 115163 (2023). https://doi.org/10.1016/j.cam.2023.115163
4. Giovannoni, G., Butzkueven, H., Dhib-Jalbut, S., et al: Brain health: time matters in multiple sclerosis. Multiple Sclerosis and Related Disorders **9**, S5–S48 (Sep 2016)
5. Herman, J., Usher, W.: SALib: An open-source python library for sensitivity analysis. The Journal of Open Source Software **2**(9) (jan 2017). https://doi.org/10.21105/joss.00097
6. Iman, R.L.: Uncertainty and sensitivity analysis for computer modeling applications. ASME AEROSP DIV PUBL AD., ASME, NEW YORK, NY(USA), 1992, **28**, 153–168 (1992)
7. Iwami, S., Takeuchi, Y., Miura, Y., et al: Dynamical properties of autoimmune disease models: Tolerance, flare-up, dormancy. J. Theor. Biol. **246**(4), 646–659 (Jun 2007)
8. Iwanaga, T., Usher, W., Herman, J.: Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses. Socio-Environmental Systems Modelling **4**, 18155 (May 2022). https://doi.org/10.18174/sesmo.18155
9. Khonsari, R.H., Calvez, V.: The Origins of Concentric Demyelination: Self-Organization in the Human Brain. PLoS ONE **2**(1), e150 (Jan 2007)
10. Kinzel, S., Weber, M.S.: B Cell-Directed Therapeutics in Multiple Sclerosis: Rationale and Clinical Evidence. CNS Drugs **30**(12), 1137–1148 (Dec 2016)
11. Lassmann, H.: Multiple Sclerosis Pathology. Cold Spring Harb. Perspect. Med. **8**(3), a028936 (Mar 2018)
12. Lazibat, I.: Multiple Sclerosis: New Aspects of Immunopathogenesis. Acta Clin. Croat. **57**(2) (2018)
13. Luca, M.: Chemotactic signaling, microglia, and alzheimer's disease senile plaques: Is there a connection. Bull. Math. Biol. **65**(4), 693–730 (2003)
14. Ludewig, P., Gallizioli, M., Urra, X., et al.: Dendritic cells in brain diseases. Biochim Biophys Acta Mol Basis Dis **1862**(3), 352–367 (Mar 2016)
15. Pappalardo, F., Russo, G., Pennisi, M., et al.: The Potential of Computational Modeling to Predict Disease Course and Treatment Response in Patients with Relapsing Multiple Sclerosis. Cells **9**(3), 586 (Mar 2020)
16. Pernice, S., Follia, L., Maglione, A., et al: Computational modeling of the immune response in multiple sclerosis using epimod framework. BMC Bioinform **21**(S17), 550 (Dec 2020)
17. Salou, M., Nicol, B., Garcia, A., et al.: Involvement of CD8+ T Cells in Multiple Sclerosis. Front. immunol. **6** (Nov 2015)
18. Saltelli, A., Annoni, P., Azzini, I., et al.: Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. Computer Physics Communications **181**(2), 259–270 (2010). https://doi.org/10.1016/j.cpc.2009.09.018
19. Saltelli, A., Tarantola, S., Campolongo, F., et al.: Sensitivity analysis in practice: a guide to assessing scientific models, vol. 1. Wiley Online Library (2004)
20. Sobol, I.: Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. Mathematics and Computers in Simulation **55**(1), 271–280 (2001). https://doi.org/10.1016/S0378-4754(00)00270-6, the Second IMACS Seminar on Monte Carlo Methods