63

Parallel Domain Decomposition for Reaction-Diffusion Problems

N. VOLFOVSKY [1], M. BERCOVIER [2]

INTRODUCTION

A parallel domain decomposition technique has been applied to construct a solution of 3d non linear reaction-diffusion problems arising in the mathematical modeling of calcium dynamic in neurons. The present method defines the domain decomposition iterations not at each time step of computation but at automatically determined common times for updating the individual subdomains boundary conditions. This method uses over-lapping not necessarily "mesh consistent" geometries. Thus such an approach provides both different space and different time resolution on each subdomain.

The first extension of the classical Schwarz alternative algorithm to a class of parabolic equations appeared in the paper of [Lio88, Lio89]. The basic idea is to divide the domain into several overlapping subdomains and then to solve the parabolic problem in each subdomain alternatively with boundary information from the neighboring subdomains.

There are basically two approaches for which a domain decomposition method can be used to solve a nonlinear problem. The first approach is to locally linearize the nonlinear equation and then to solve the resulting linearized problems at each nonlinear iteration by a domain decomposition method. The second approach is to use domain decomposition, such as the Schwarz alternating method, directly on the nonlinear problems. In this case, a number of smaller nonlinear problems need to be solved in each domain decomposition iteration. In this paper we focus on the last approach. Inside each subdomain we used the standard FEM package, FIDAP 7. 5 [Eng96] which was modified for PVM [Gei94] parallelization [Ah96] and was supplemented by the options of time management introduced in [Ber97]. Convergence

1 Institute of Computer Science, The Hebrew University of Jerusalem, Jerusalem 91904, Israel

2 Address same as above.

Eleventh International Conference on Domain Decomposition Methods
Editors Choi-Hong Lai, Petter E. Bjørstad, Mark Cross and Olof B. Widlund

analysis of the algorithm is based on the convergence theorem [Ran93] and was presented in [Ber97]. The computations were performed on the SP2 machine of High Power Computing Center of Israel.

The parallel implementation of the algorithms greatly reduces the computational time. However the problems of load balancing and synchronization in solving complex nonlinear problems on separate domains become dominant through the inter-processor communication. The present domain decomposition method concentrates on the time step management.

Numerical tests of the algorithm have been undertaken. It took several days to calculate the general problem described here without parallel domain decomposition on an Indy SGI workstation. Parallel computations on a SP2 machine gave us the results in hours. The algorithm presented here is fully parallel relative to the number of domains, that is the total solution time is the same for 4 domains on 4 processors as it is for 32 domains on 32 processors.

THE MODEL PROBLEM

Consider a bounded domain, Ω of R^3 with the boundary $\partial\Omega$. Then, the full problem to be solved is the following : Find u_i (i =1,..n), real valued functions, defined on Ω and satisfying :

$$\begin{cases} \partial_t u_l - \nu_l \triangle u_l = \partial_t f_l \quad \text{in} \quad \Omega \times (0,T] \\ \partial_t f_l = \sum_{m=1}^{n} (-\lambda_m u_l u_m + \mu_m u_{lm}) \quad \text{in} \quad \Omega \times (0,T] \\ \partial_n u_l = g_l \quad \text{on} \quad \partial\Omega \times (0,T] \\ u_l|_{t=0} = u_{l0} \quad \text{in} \quad \Omega \end{cases} \tag{1}$$

where u_l is the concentration of the l-th component of the system, ν_l is the corresponding diffusion coefficient, where $\partial_t f_l$ denotes its reaction with the rest of the system, and where λ_{lm} and μ_{lm} are the rate constants and u_{lm} is the binding complex of components l and m .

GENERAL ALGORITHM

We consider the domain decomposition of the parent domain Ω into M subdomains ($\{\Omega_i\}_{i=1}^{M}$) in such a way that each pair of adjacent subdomains sharing a common boarder will have a non-empty interior intersection. Thus,

$$\begin{aligned} &\Omega = \bigcup_{i=1}^{M} \Omega_i \\ &\partial\Omega_i \cap \partial\Omega_j \neq \emptyset \quad \Rightarrow \quad \Omega_i \cap \Omega_j \neq \emptyset \quad \text{for all} \quad i \neq j \end{aligned} \tag{2}$$

where $\Omega_i \subset R^3$ is the i-th open subdomain with the boundary $\partial\Omega_i$. The steps followed to define the domain decomposition scheme are as follows

Step 1: Initialization and definitions

- set dt_i the time step associated with the various reactions in the i th subdomain. Note that this " internal " time step can vary from subdomain to subdomain.
- set T - the global time over all subdomains and let $dT_n = T_{n+1} - T_n$ be the time step for updating the information between the subdomains.
- set $u_i(0) = u_0 \quad \text{in} \quad \Omega_i$
- let k be the current number of the domain decomposition iterations
- let j be the current number of of the internal iterations on each subdomain.

Step 2: Domain-decomposition iteration

- for k,n solve M subproblems:
  Find u_i^k, a real valued functions, defined on Ω_i and satisfying :

$$\begin{cases} \partial_t u_i^k - \nu_i \triangle u_i^k = \partial_t f_i \quad \text{in} \quad \Omega \times (T_n, T_{n+1}] \\ \partial_n u_i^k = g_i \quad \text{on} \quad (\partial\Omega_i \cap \partial\Omega) \times (T_n, T_{n+1}] \\ \partial_n u_i^k = u_*^k \quad \text{on} \quad (\partial\Omega_i \backslash \partial\Omega) \times (T_n, T_{n+1}], \end{cases} \tag{3}$$

  where

$$u_*^k = \begin{cases} u_{j_1}^{k-1} & \text{on} \quad \partial\Omega_i \cap \Omega_j \cap [\cup_{j \neq i, J_1} \Omega_{j_1}^c] \quad \text{for all} \quad j_1 \neq i \\ u_{j_1 j_2}^{k-1} & \text{on} \quad \partial\Omega_i \cap \Omega_{j_1} \cap \Omega_{j_2} \cap [\cup_{j \neq i, j_1, j_2} \Omega_j^c] \quad \text{for all} \\ \vdots & \qquad\qquad j_1 \neq j_2 \neq i \\ u_{j_1 \ldots j_{M-1}}^{k-1} & \text{on} \quad \partial\Omega_i \cap \Omega_{j_1} \cap \Omega_{j_2} \cap \ldots \cap \Omega_{j_{M-1}}, \end{cases} \tag{4}$$

  and $u_{j_1 \ldots j_m}^{k-1}$ $(m \leq M-1)$ is selected to satisfy $Min(u_{j_1}^{k-1}, \ldots u_{j_m}^{k-1}) \leq u_{j_1 \ldots j_m}^{k-1} \leq Max(u_{j_1}, \ldots u_{j_m}^{k-1})$.
- a segregated algorithm [Eng96] is used to solve these problems; internal iterations for the current local time step of the implicit Euler integration scheme are stopped when the following convergence criterion is satisfied

$$\frac{\| u_i^{k(j+1)} - u_i^{k(j)} \|}{\| u_i^{k(j)} \|} \leq \epsilon_i \tag{5}$$

  where $u_i^{k(j)}$ is the solution of subprocess i at domain decomposition iteration k at iteration j of the segregated algorithm, and ϵ_i is the tolerance for subdomain computations.
- *management of time steps*
  if the convergence (5) is not attained then
  (i) the solution restarts with the smallest dt_i obtained on the current local step, these dt_i being estimated directly in FIDAP.
  (ii) if (i) is not sufficient for convergence, the solution restarts from the last global time step T_n with a decreased dt_i;
  (iii) if (ii) does not converge, the solution restart s from the same global time step T_n, but with a decreased global time step $dT_n \leq T - T_n$.

Step 3: Global convergence

- the domain decomposition iterations are stopped when the following convergence criterion is met in all subdomains

$$\frac{\| u_i^k - u_i^{k-1} \|}{\| u_i^k \|} \leq \epsilon_T \tag{6}$$

  where ϵ_T is the selected tolerance
- *management of time step*
  if criterion (6) is not satisfied then
  (i) the solution restarts from the last global time step T_n but with the last evaluated internal boundaries conditions (u_*^k).
  (ii) if the number of iteration exceeds the limit, the solution restarts from the last global time step T_n with a decreased global time step dT_n.

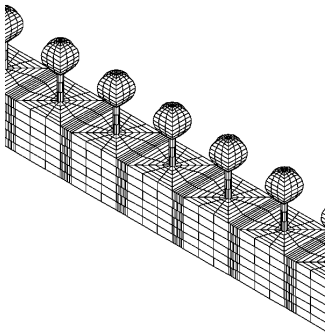

Figure 1 Fragment of the dendrite with eight spines

MODEL OF Ca^{2+} DYNAMICS IN THE DENDRITIC SPINES

The reaction-diffusion problem of Ca^{2+} dynamic in the dendritic spines includes the following processes associated with Ca^{2+} distribution [Gam87, Hol90] : 1) diffusion; 2) Ca^{2+} buffering, 3) Ca^{2+} influx and 4) Ca^{2+} extrusion.

1) The diffusion and 2) Ca^{2+} buffering is modeled by

$$\begin{array}{l} \frac{\partial C}{\partial t} = D \triangle C - k_+ C \cdot B + k_-(BC) \\ \frac{\partial B}{\partial t} = -\frac{\partial (BC)}{\partial t} = -k_+ C \cdot B + k_-(BC) \end{array} \tag{7}$$

where D is the Ca^{2+} diffusion coefficient, C is the Ca^{2+} concentration, B is the free buffer and $BC$is the Ca^{2+} bounded buffer. k_+ and k_- denote the binding and dissociation rate constants, respectively (for parameter values see Table 1).

Boundary conditions of the problem are determined by the following equations; a description of the physical context and real values of the parameters are given in the Appendix.

3) Ca^{2+}influx

$$J_{in} \mid_{\Gamma_{in}} = \alpha_1 \cdot [1 - \alpha_2 ln(\frac{\alpha_3}{C(t)})] \tag{8}$$

4) Ca^{2+} extrusion

$$J_p \mid_{\Gamma} = \frac{\beta_1 \cdot C}{C + \beta_2} \tag{9}$$

where Γ is the spine surface, Γ_{in} is the surface of head of spine.

NUMERICAL TESTS

The numerical example represents a three dimensional model of second-messenger dynamics in the dendrite (Figure 1) . The physical domain is a long wide cylinder with any number of

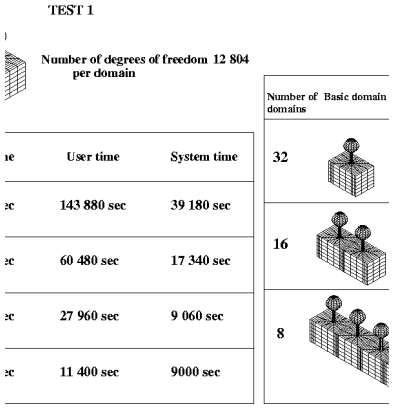

Figure 2 TEST1: A reaction-diffusion problem was solved for a domain with 4, 8, 16, and 32 spines. The DD algorithm was applied for partitions on 4, 8, 16, and 32 domains. TEST2: A reaction-diffusion problem was solved for a domain with 32 spines. The DD algorithm was applied for partitions on 32, 16, and 8 domains

spines. Each spine is the union of a thin cylinder and a ball.
To implement the proposed Domain Decomposition (DD) algorithm, the initial dendritic domain was divided in basic ones, describing a spine head and a fragment of the dendrite.

The numerical tests concerned the numerical efficiency of the present algorithm as well as an original 3d model of Ca^{2+} dynamics in the dendrite. To evaluate the parallel properties of the DD algorithm two different tests were performed and the following computational parameters were discussed: real time of computations, user time - total time of the computation and the system time, which is the communications and I/O time.

First, problems of different sizes were considered. Fragments of dendrite with 4, 8, 16, and 32 spines were divided in 4, 8, 16, and 32 unitary subdomains respectively. Results of the first series of the tests are presented in Figure 2. This indicates that, as expected, an increase in the number of unitary domains (Test1) leads to a proportional increase in user and system time of the simulation , but the "real time" i.e. the actual clock time a user has to wait for an answer is growing significantly less than the computer resources usage.

In the second series of computations a domain with 32 spines (Figure 2) was studied. Three cases of partition were selected on 32, 16, and 8 subdomains, each with one, 2 and 4 spines correspondingly. For all simulations , convergence was defined by requiring that $\epsilon_T = 10^{-3}$.

The second test is significantly different than the first one. Real time of simulation declined proportionally to the increase in the number of subdomains. User time increased with the largest number of subdomains. However, relative changes are small. In these tests, unlike the first ones, the system time decreased with the number of subdomains, which suggests that this parameter is insensitive to the size of the problem on each CPU.

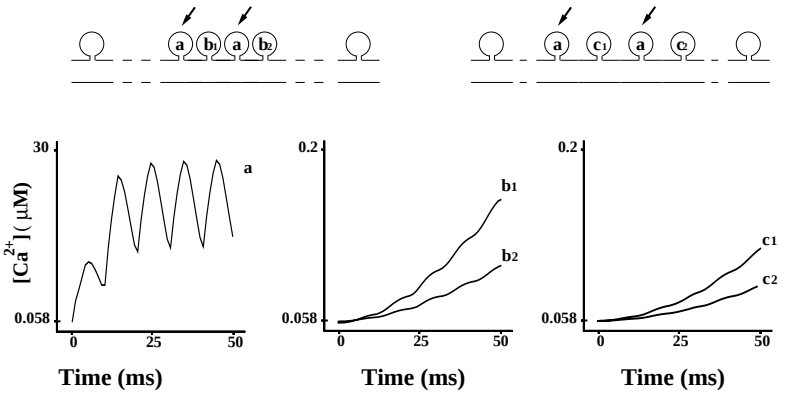

Figure 3 Interaction of the spines with different activation protocol; line **a.** denotes distribution of calcium in the stimulated spine, lines $\mathbf{b}_1$, $\mathbf{b}_2$ denote distribution of calcium in the instimulated spines (distance between spines 0.65μm) and lines $\mathbf{c}_1$, $\mathbf{c}_2$ -the same but with a distance of 0.95μm between the spines

A specific characteristic of the DD algorithm presented here is that it accepts different time steps on separate subdomains, such as in the problem of chemical interactions of dendritic spines with different activation protocols. The results of simulations are shown on Figure 3.

The dynamics of calcium (Ca^{2+}) was monitored in a spine which was not stimulated at all and only two neighbors were stimulated (an index 1 denotes a spine whose two neighbors are stimulated and an index 2 when only one neighbor is stimulated.) Five depolarizing pulses of 5msec were given at 100 Hz. The accumulation of Ca^{2+} in nonstimulated spines shows interspine interaction, which is one of experimentally unsolved problems of Ca^{2+} dynamics in neurons. As expected this interaction is more profound at the high density of 0.65 μm interval between spines ($\mathbf{b}_1$ and $\mathbf{b}_2$). At the lower density of 0.95 μ m ($\mathbf{c}_1$ and $\mathbf{c}_2$) interval the interaction is less profound, but clearly apparent.

The main conclusions of this interdisciplinary research are : (1) the present DD algorithm is fully parallel relative to the number of domains and (2) parallel domain decomposition methods provide an efficient procedure for solution of reaction-diffusion problems arising in the modeling of complex biological systems.

APPENDIX

Boundary conditions and description of parameters.

Boundary conditions are determined by the influx equation on the spine head boundary and extrusion equation on the all boundary of the spine.

The Ca^{2+} *flux* through voltage dependent channels situated in the membrane of the spine head and dendrite is described by

$$J_{in} \mid_{\Gamma_{in}} = g_{max} \cdot [V - \frac{RT}{ZF} ln(\frac{C_{out}}{C(t)})] \tag{10}$$

where Γ_{in} is the segment of the membrane on the spines head and on the dendrite which include voltage-dependent channels, V is the membrane potential, C_{out} is the extracellular concentration of Ca^{2+}, g_{max} is the maximal conductance, R is the gas constant, T is the absolute temperature (in degrees Kelvin), Z is the valence of Ca^{2+}, F is the Faraday constant, and $C(t)$ is the concentration of intracellular Ca^{2+} beneath the head membrane at time t.

The Ca^{2+} *extrusion* which is conducted by pumps located throughout the entire spine membrane is given by

$$J_p \mid_\Gamma = \frac{x_p \cdot C}{C + k_p} \tag{11}$$

where Γ is the spine surface.

Table 1 Parameters of the model

Parameter	Description	Value
D	Diffusion coefficient	$4 \cdot 10^{-6}\ cm^2 s^{-1}$
C_0	Initial concentration of Ca^{2+}	$0.06\ \mu M$
B_{cl0}	Initial concentration of calcineurin	$10\ \mu M$
B_{cn0}	Initial concentration of calmodulin	$25\ \mu M$
k_{cl+}	on rate constant of calcineurin	$50\ \mu M^{-1} s^{-1}$
k_{cl-}	off rate constant of calcineurin	$25\ s^{-1}$
k_{cn+}	on rate constant of calmodulin	$50\ \mu M^{-1} s^{-1}$
k_{cn-}	off rate constant of calmodulin	$500\ s^{-1}$
x_p	maximal rate of extrusion	$15\ \mu M s^{-1}$
k_p	half saturation constant	$0.9\ \mu M$

REFERENCES

[Ah96] Aharon S., Bercovier M., and Parnas H. (1996) Parallel computation enables precise description of Ca^{2+} distribution in nerve terminals. *Bull. Math. Biol.* 58: 1075–1097.

[Ber97] Bercovier, M., Volfovsky, N., and Parnas, H. (1997) Parallel domain decomposition for reaction-diffusion problems: theory and experiments. In Bristeau, M-O., Etgen, G., Fitzgibbon, W., Lions, P.L., and Wheeler, M.F (ed) *Computational Science for the 21st century.* INRIA, University of Houston and GAMNI/SMAI.

[Eng96] Engelman, M.S. (1996) *FIDAP Theoretical Manual, version 7.5.* FDI, Evanston, IL.

[Gam87] Gamble, E. & Koch, C. (1987) The dynamics of free calcium in dendritic spines in response to repetitive synaptic input. *Science* 236: 1311–1315.

[Gei94] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V. (1994) *PVM 3 user's guide and reference manual.* Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831.

[Hol90] Holmes W.R. & Levy W.B. (1990) Insights into long-term potentiation from computational models of nmda receptors-mediated calcium influx and intracellular calcium concentration changes. *J Neurophysiol* 63: 1148–1168.

[Lio88] Lions, P.L. (1988) On the Schwarz alternating method. i. In Glowinski, R., Golub, G.H., Meurant, G.A., and Périaux J. (ed) *1st Int. Symp. on Dom. Decomp. Meth. for PDE.* SIAM, Philadelphia.

[Lio89] Lions, P.L. (1989) On the Schwarz alternating method. ii: Stochastic interpretation and order properties. In Chan, T., Glowinski, R., Périaux, J., and Widlund, O. (ed) *Domain Decomposition Methods*. SIAM, Philadelphia.

[Ran93] Rannacher, R. (1993) Domain decomposition in the nonstationary streamline diffusion finite element method. In Neitaanmåki P. (ed) *Proc FEM 50 Fifty Years Finite Element Method,*.