On a model of three-dimensional bursting and its parallel implementation

S. Tabik [a], L.F. Romero [b], E.M. Garzón [a], J.I. Ramos [c,*]

[a] *Departamento de Arquitectura de Computadores y Electrónica, Universidad de Almería, 04120 Almería, Spain*
[b] *Departamento de Arquitectura de Computadores, Complejo Politécnico, Universidad de Málaga, Campus de Teatinos, Apartado 4114, 29080 Málaga, Spain*
[c] *Room I-320-D, E.T.S. Ingenieros Industriales, Universidad de Málaga, Plaza El Ejido, s/n, 29013 Málaga, Spain*

Received 7 April 2006; received in revised form 18 September 2007; accepted 31 October 2007

Abstract

A mathematical model for the simulation of three-dimensional bursting phenomena and its parallel implementation are presented. The model consists of four nonlinearly coupled partial differential equations that include fast and slow variables, and exhibits bursting in the absence of diffusion. The differential equations have been discretized by means of a second-order accurate in both space and time, linearly-implicit finite difference method in equally-spaced grids. The resulting system of linear algebraic equations at each time level has been solved by means of the Preconditioned Conjugate Gradient (PCG) method. Three different parallel implementations of the proposed mathematical model have been developed; two of these implementations, i.e., the MPI and the PETSc codes, are based on a message passing paradigm, while the third one, i.e., the OpenMP code, is based on a shared space address paradigm. These three implementations are evaluated on two current high performance parallel architectures, i.e., a dual-processor cluster and a Shared Distributed Memory (SDM) system. A novel representation of the results that emphasizes the most relevant factors that affect the performance of the paralled implementations, is proposed. The comparative analysis of the computational results shows that the MPI and the OpenMP implementations are about twice more efficient than the PETSc code on the SDM system. It is also shown that, for the conditions reported here, the nonlinear dynamics of the three-dimensional bursting phenomena exhibits three stages characterized by asynchronous, synchronous and then asynchronous oscillations, before a quiescent state is reached. It is also shown that the fast system reaches steady state in much less time than the slow variables.

PACS: 82.40.Ck; 02.60.Lj; 07.05.Tp

Keywords: Reaction–diffusion equations; Bursting; Three-dimensional model; MPI; OpenMP; PETSc; Performance representation

1. Introduction

Many processes in nature are characterized by transitions between different modes of activity, such as quiescent, periodic, quasiperiodic and chaotic states. Bursting [1] is a common phenomenon in many natural systems that arises in many fields, e.g., boundary layers and turbulence in hydrodynamics [2], free and forced convection [3], magnetohydrodynamics [4], plasma confinement [5], X-ray pulsar emission [6], neuronal processing [7], physiology [8], biology [9], electronics [10], optics, lasers [11], nonlinear mechanics, chemical reactions/kinetics [12], control [13], etc.

Bursting events are usually characterized by sudden, short-lived, high-amplitude deviations of a nonlinear dynamical system from its otherwise quiescent state; in neuroscience and physiology, these events are called spikes and the term burst is usually employed for repetitive spiking. For example, neurons in the brain communicate with each other by firing and transmitting sequences of electrical spikes or action potentials. Usually, action potentials occur in a periodic fashion, as in response to a constant applied current of sufficiently large magnitude. In addition, many cell types, e.g., pancreatic β-cells, exhibit more complex behavior characterized by brief bursts of oscillatory activity interspersed with quiescent periods during which the cell membrane potential changes slowly.

One of the major challenges in neuroscience is to understand the basic physiological mechanisms underlying the complex

* Corresponding author. Tel.: +34 95 2131402; fax: +34 95 2132816.
E-mail address: jirs@lcc.uma.es (J.I. Ramos).

spatio-temporal patterns of spiking activity observed during normal brain functioning, and to determine the origins of pathological dynamical states such as epileptic seizures and Parkinsonian tremors. A second major challenge is to understand how the patterns of spiking activity provide a substrate for the encoding and transmission of information, that is, how do neurons compute with spikes? It is likely that an important element of both the dynamical and computational properties of neurons is that they can exhibit bursting, which is a relatively slow rhythmic alternation between an active phase of rapid spiking and a quiescent phase without spiking.

Due to its ubiquity and importance, understanding the mechanisms that give rise to bursting is a research topic of significant current interest. Indeed there have been many analytical and numerical studies of the nonlinear dynamics of models governed by nonlinear ordinary differential equations that exhibit bursting phenomena. For example, models of bursting electrical activity in physiology can be classified into two main groups. The first and earliest one was based on the assumption that bursting was caused by an underlying slow oscillation in the intracellular Ca^{2+} concentration [14,15]; however, recent experiments indicate that this assumption is not entirely correct and, as a consequence, models relying on alternative mechanisms have been developed [16].

The different known bursting mechanisms can be classified into three main groups. In type I, bursts arise from hysteresis and bistability as in the pancreatic β-cell model. In type II, bursts arise from an underlying slow oscillation, while, in type III, bursting arises from a subcritical Hopf bifurcation [8, 14,15,17]. This classification is by no means complete [18], for it is based on both knowledge acquired with nonlinear ordinary differential equations that usually consist of slow and fast variables, and the analysis of their local bifurcations by means of, for example, singular perturbation methods. Thus, for example, Rinzel and Troy [19] employed a simplified three-equation model of the Belousov–Zhabotinskii reaction in a continuous flow, stirred tank reactor (CSTR) with a steady-state assumption for one of the dependent variables and deduced the existence of a periodic solution from the Poincaré–Bendixson theorem; they also determined the bursts of oxidation pulses and the periods of quiescence. These authors also showed that the solution alternates between the model's stable periodic solution during the oscillatory phase and the model's stable steady state of low oxidation during the quiescence intervals.

Periodic bursting in slow-fast systems governed by nonlinear ordinary differential equations can be viewed as closed paths through the unfolding parameters of degenerate singularities. Using this approach, it can be shown that Hopf–Hopf mode interactions can lead to bursting between in-phase and out-of-phase periodic solutions, and Takens–Bogdanov singularities can lead to bursting that randomly chooses between two symmetrically related limit cycles.

There have been very few studies on the effect of diffusion on mathematical models that exhibit bursting under homogeneous conditions. Exception is made to the paper by Carpenter [20] who, using singular perturbation methods in phase space, studied the traveling wave solutions of a generalized Hodgkin–Huxley model governed by a one-dimensional nonlinear reaction–diffusion equation and three nonlinear ordinary differential equations, and presented a simple classification that determines whether a system exhibits finite wave trains and periodic bursting behavior or only single pulse and regular periodic behavior. However, Carpenter's study as well as other studies which have considered diffusion have been concerned with traveling waves in one-dimensional systems. In extended two- and three-dimensional systems subject to homogeneous boundary conditions, the geometry of the domain and the diffusion introduce new time scales that may result in new bifurcations and completely different dynamics than that observed under homogeneous conditions.

The objective of this paper is several-fold. First, a three-dimensional reaction–diffusion model consisting of four nonlinearly coupled partial differential equations is presented. The model exhibits bursting under homogeneous conditions, and both bursting and extinction/quiescence in two dimensions provided that the initial conditions are not homogeneous. The source or reaction terms of these partial differential equations can be classified into two subsystems exhibiting fast and slow behavior. The accurate simulation of the fast processes demands the use of small time steps, while the extended system considered here requires the use of sufficiently small spatial step sizes in order to accurately resolve the steep gradients of the dependent variables. The finite difference equations resulting from the discretization of the three-dimensional bursting model equations is solved by means of the Preconditioned Conjugate Gradient (PCG) method. Second, three parallel implementations of the discretized bursting model are presented; two of these implementations, i.e., the MPI code and the PETSc code, are based on a message passing paradigm, while the third one, i.e., the OpenMP code, is based on a shared space address paradigm. The three implementations are then evaluated on two parallel architectures, a dual-processor cluster and a Shared Distributed Memory (SMD) system. Third, the representation of the performance results is usually given in terms of runtimes, speedup or efficiency but lacks information about the communication, load unbalance and cache effect factors. In this work, we propose a novel and clear way to present the performance results that comprises all the important factors that affect the parallel efficiency.

This paper is organized as follows. In Section 2, the proposed mathematical model and its finite difference discretization in equally-spaced grids are described. In Section 3, we present two sequential implementations of the model in order to illustrate the storage of the matrix that will be used in the parallel implementations. Section 4 presents a detailed description of the three parallel implementations developed in this paper. In Section 5, a comparative analysis of the three parallel implementations in terms of their performance is presented, while, in Section 6, some sample results illustrating the nonlinear dynamics of three-dimensional bursting phenomena are shown. A final section on conclusions summarizes the most important findings of the paper.

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

2. Mathematical model and discretization

In this paper, bursting in extended, i.e., spatio-temporal, systems has been simulated by means of the following nonlinearly coupled system of partial differential equations of the reaction–diffusion type

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{D}\left(\frac{\partial^2 \mathbf{U}}{\partial x^2} + \frac{\partial^2 \mathbf{U}}{\partial y^2} + \frac{\partial^2 \mathbf{U}}{\partial z^2}\right) + \mathbf{S}(\mathbf{U}), \tag{1}$$

where $\mathbf{U} = (u, v, w, p)^{\mathrm{T}}$, the superscript T denotes transpose, $\mathbf{D}$ is a diagonal matrix with components equal to D_u, D_v, D_w and D_p, t is time, x, y and z are Cartesian coordinates, $\mathbf{S} = (S_u, S_v, S_w, S_p)^{\mathrm{T}}$ is the vector of the source/reaction terms given by

$$\begin{cases} S_u = f(u) - v - gw(u - u_0), \\ S_v = \frac{1}{5}(v_\infty(u) - v), \\ S_w = f_w(w) + \alpha_w(p - 0.3), \\ S_p = \beta_p((1 - p)H(u) - p), \end{cases} \tag{2}$$

and

$$\begin{cases} f(u) = 1.35u(1 - \frac{u^2}{3}), \\ f_w(w) = -0.2(w - 0.2)(w - 0.135)(w - 0.21), \\ v_\infty(u) = \tanh(5u), \\ H(u) = \frac{3}{2}(1 + \tanh(5u - 2.5)). \end{cases} \tag{3}$$

The functions $f(u)$ and $f_w(w)$ have three different zeros each, whereas $v_\infty(u)$ and $H(u)$ are monotonically increasing functions of their arguments, and u_0, α_w, β_p and g are constants.

Eq. (1) may be considered as a very simplified model of bursting electrical activity in cells, where u and v represent the currents of activated and voltage-dependent channels, respectively, and are the fast system, whereas w and p represent a current and its activation, respectively, and constitute the slow subsystem.

For $\mathbf{D} = \mathbf{0}$, i.e., in the absence of diffusion, Eq. (1) exhibits bursting phenomena provided that the values of the u_0, α_w, β_p and g are properly chosen, and simulations performed in the absence of diffusion indicate that u and v are fast variables, while w and p are slow ones. Computations carried out in one and two dimensions with Eq. (1) and homogeneous Neumann conditions indicate that, depending on the values of u_0, α_w, β_p and g, the initial conditions, the diffusion coefficients and the size of the domain, the solution of Eq. (1) exhibits a rich dynamic behavior including periodicity, quasiperiodicity, chaos, bursting and extinction, where the latter is used here to denote quiescence.

Eq. (1) has been solved in a parallelepiped $\mathbf{\Omega} \equiv [-L_x, L_x] \times [-L_y, L_y] \times [-L_z, L_z]$, and homogeneous Neumann boundary conditions have been used on all the boundaries. The initial conditions used in the computations are

$$\begin{cases} u = -2.5, v = -0.2, \omega = -0.5, p = 0.5 & \text{in } \mathbf{\Omega}, \\ u = 2.5, v = -0.2, \omega = 0.5, p = -0.5 & \text{in } \mathbf{\Omega} - \mathbf{\Gamma}, \end{cases} \tag{4}$$

where $\mathbf{\Gamma} = [-\frac{L_x}{2}, \frac{L_x}{2}] \times [-\frac{L_y}{2}, \frac{L_y}{2}] \times [-\frac{L_z}{2}, \frac{L_z}{2}]$. These initial conditions were selected so that, under homogeneous conditions, Eq. (1) exhibits bursting oscillations for $u_0 = 2$, $\alpha_w = 0.002$, $\beta_p = 0.00025$ and $g = 0.73$.

By discretizing only the time variable in Eq. (1) by means of the (second-order accurate) Crank–Nicolson technique, one can obtain a system of nonlinear elliptic equations at each time step. These elliptic equations can be linearized with respect to time in order to obtain a system of linear elliptic equations at each time level [21], i.e., the nonlinear terms $\mathbf{S}^{n+1}$ are approximated by means of the second-order accurate terms $\mathbf{S}^n + \mathbf{J}^n \Delta\mathbf{U}$, where $\mathbf{J} \equiv \frac{\partial \mathbf{J}}{\partial \mathbf{U}}$ denotes the Jacobian of the source terms, the superscript n denotes the nth time level, i.e., $t^n = n\Delta t$, $n = 0, 1, 2, 3, \ldots$, Δt, is the time step, and $\Delta\mathbf{U} \equiv \mathbf{U}^{n+1} - \mathbf{U}^n$. The resulting system of linear elliptic equations was discretized by means of second-order accurate central finite difference formulae in an equally-spaced grid of $N_x \times N_y \times N_z$ points, and the resulting system of linear algebraic equations can be written as

$$\begin{aligned} &\left(\mathbf{I} - \frac{k}{2}\mathbf{J}^n_{i,j,k}\right)\Delta\mathbf{U}_{i,j,k} \\ &\quad - \alpha_x(\Delta\mathbf{U}_{i+1,j,k} - 2\Delta\mathbf{U}_{i,j,k} + \Delta\mathbf{U}_{i-1,j,k}) \\ &\quad - \alpha_y(\Delta\mathbf{U}_{i,j+1,k} - 2\Delta\mathbf{U}_{i,j,k} + \Delta\mathbf{U}_{i,j-1,k}) \\ &\quad - \alpha_z(\Delta\mathbf{U}_{i,j,k+1} - 2\Delta\mathbf{U}_{i,j,k} + \Delta\mathbf{U}_{i,j,k-1}) = \mathbf{T}^n_{i,j,k}, \end{aligned} \tag{5}$$

where $\mathbf{I}$ is the identity or unit matrix, $\alpha_x = \frac{k}{2\Delta x^2}\mathbf{D}$, $\alpha_y = \frac{k}{2\Delta y^2}\mathbf{D}$, $\alpha_z = \frac{k}{2\Delta z^2}\mathbf{D}$, Δx, Δy and Δz are the grid spacings in the x, y and z directions, respectively, and

$$\begin{aligned} \mathbf{T}_{i,j,k} = {} & 2\alpha_x(\mathbf{U}_{i+1,j,k} - 2\mathbf{U}_{i,j,k} + \mathbf{U}_{i-1,j,k}) \\ & + 2\alpha_y(\mathbf{U}_{i,j+1,k} - 2\mathbf{U}_{i,j,k} + \mathbf{U}_{i,j-1,k}) \\ & + 2\alpha_z(\mathbf{U}_{i,j,k+1} - 2\mathbf{U}_{i,j,k} + \mathbf{U}_{i,j,k-1}) + \mathbf{S}_{i,j,k}. \end{aligned} \tag{6}$$

In this paper, a natural ordering of the grid points and blocking of nodal variables have been chosen to obtain both a well structured matrix and a good cache behavior. For this ordering, Eq. (5) can be expressed in matrix form as $\mathbf{A}\Delta\mathbf{U} = \mathbf{b}$, where $\mathbf{A}$ is an heptadiagonal block matrix consisting of $(N_x \times N_y \times N_z)$ 4×4 blocks. Fig. 1 shows the pattern of the matrix $\mathbf{A}$ for an equally-spaced mesh of $5 \times 5 \times 5$ points.

3. Sequential implementations

The sequential algorithm for the solution of the system of linear algebraic equations $\mathbf{A}\Delta\mathbf{U} = \mathbf{b}$ can be described as follows:

1—Initialize the source/reaction terms according to Eqs. (2) and (3).
2—Initialize the solution $\mathbf{U}^0$ according to the initial conditions considered.
3—$t = t_0 = 0$
do while ($t < t_{\text{end}}$)
- Update the matrix $\mathbf{A}$ and right-hand side $\mathbf{b}$ according to Eqs. (5) and (6), respectively.
- Solve the system of equations $\mathbf{A}\Delta\mathbf{U} = \mathbf{b}$.
- Update the solution $\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta\mathbf{U}$.

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

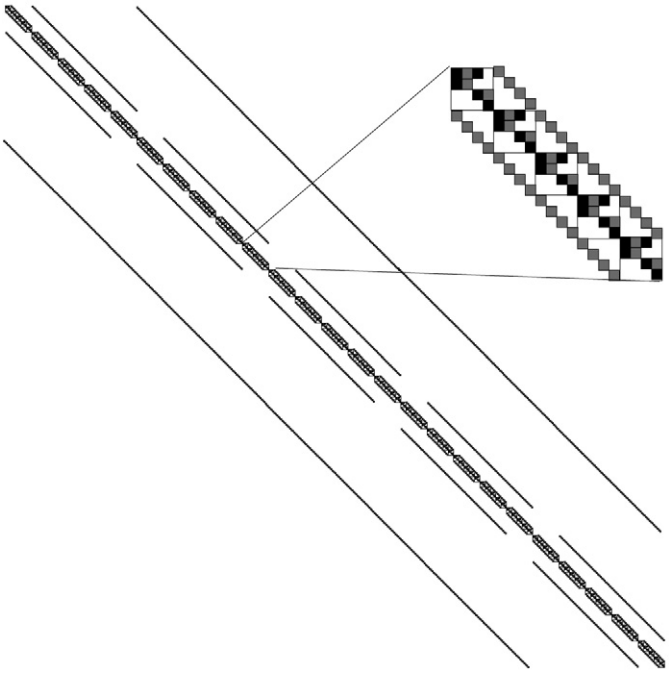

Fig. 1. Matrix pattern corresponding to a $5 \times 5 \times 5$-point mesh, where the grey squares represent nonzero time-independent elements and the black squares represent nonzero elements that have to be updated at each time level.

In this paper, two different sequential implementations have been developed; both implementations employ the conjugate gradient (CG) method with Jacobi preconditioner. In the first one, the system of linear algebraic equations at each time level has been solved by means of an optimized Preconditioned Conjugate Gradient (PCG) method for banded matrices [22] based on a maximum exploitation of the data locality. Furthermore, the matrix has been stored using a compressed diagonal format which takes into account the sparsity pattern of the Jacobian term, and the implementation of the sparse matrix–vector product includes a consequent hand-unrolled product which exploits the pipelined floating point units [23].

In the second sequential implementation, the system of equations is solved by the Conjugate Gradient solver preconditioned by the Jacobi method; both the solver and the preconditioner are included in the free portable extensible toolkit for scientific computation (PETSc) library [24]. PETSc is a suite of data structures and routines for the scalable solution of scientific applications modeled by partial differential equations. Some PETSc routines have been adapted to the characteristics of the problem in order to optimize their efficiency. In addition, the matrix $\mathbf{A}$ has been stored in Compressed Sparse Row (CSR) format.

These two types of storing the matrix $\mathbf{A}$ have also been used in the parallel implementations described in the next section.

4. Parallel implementations

With the availability of (1) a wide range of parallel platforms (from clusters of workstations to Shared Distributed Memory platforms), (2) several parallel paradigms, i.e., shared-address space paradigms and message-passing paradigms, and (3) a wide variety of software, from low-level to high-level software, the physicist/programmer is faced with the selection of portable paradigms and software which will provide the best performance, since both the time and the effort that should be spent in developing and tuning a parallel implementation using one of the above mentioned paradigms is approximately the same. Note that the use of high level software usually requires less programming effort.

In this work, three parallel implementations based on MPI, PETSc and OpenMP have been evaluated, analyzed and compared on two parallel architectures, i.e., a dual-Xeon cluster and a SGI Altix 3700 Bx2 system. Hereon, these implementations are also referred to as the MPI-code, the PETSc-code and the OpenMP-code, respectively.

From a parallel computational point of view, this paper makes three contributions. First, it describes the different computational strategies that have been used to obtain the best performance for each implementation. Second, it provides an exhaustive comparison of the performance of the three codes/implementations. And, finally, it proposes a novel and clear way to represent the performance of each implementation as a function of the three most important factors that affect it, i.e., communications, load unbalance and cache effect.

In the three parallel implementations, at each time step, each processor updates its local sub-matrix and local right-hand side. Then, the system of equations is solved in parallel using the PCG method. In the following subsections, a brief description of the three parallel implementations/codes is given. In addition, a block distribution of the grid in the z-direction and, therefore, block distributions of the matrix rows and vectors, have been used in the parallel implementations.

4.1. The MPI implementation

In this implementation, the parallel solution of the system $\mathbf{A}\Delta\mathbf{U} = \mathbf{b}$ is performed by the PCG solver as in [22]. Using a Jacobi preconditioner, each iteration of the PCG method includes two inner products and one sparse matrix–vector product; therefore, three communications and their corresponding synchronizations are needed. In the MPI implementation, computations and communications are overlapped using asynchronous messages in order to minimize the communications overhead, specially in the matrix–vector product. In this manner, after the iterations of the PCG are completed, a message with the boundary values of $\mathbf{U}$ is sent to the neighboring processors and this is overlapped with the last update of $\Delta\mathbf{U}$.

4.2. The PETSc implementation

In this implementation/code, structures of the free Parallel Extensible Toolkit for scientific computing (PETSc) library have been used. The solution of the linear system of algebraic equations has been carried out in parallel using the parallel solver CG, which employs the MPI standard for all message passing communications. In order to accelerate the convergence of the CG method, the Jacobi preconditioner has been used. Both the parallel solver and the preconditioner are included in

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

PETSc. The PETSc routines employed here have been adapted to the characteristics of the problem by eliminating all the subroutines arguments that could penalize their performance. After solving the system of linear algebraic equations, each processor communicates to its adjacent ones its local boundaries by explicit MPI communications.

4.3. The OpenMP implementation

In this implementation, we have chosen the SPMD style instead of the loop level one because it reduces overhead and results in better scalability [27]. In addition, in this implementation, only one parallel section covers the whole dynamic extent of the code and, therefore, only OpenMP synchronization directives have been used. The ordering of the computations employed in the MPI implementation to minimize the communication overhead is also employed here to reduce the waiting times at the synchronization points.

For the inner products, the synchronization events have been implemented by using counters protected by lock variables. Additional flags have been included in order to grant permission for accessing shared data. As soon as a processor has computed the data on its borders, it enables a flag. If other processor requires these data, it waits for this flag to be enabled. Reset of both counters and flags has been carefully implemented by means of odd and even sense-reversing flags to enhance performance and avoid data race conditions.

In both, the MPI and the OpenMP implementations, global barriers have been avoided in order to minimize waiting times.

5. Comparison of the three parallel implementations

In this section, the performance of the three parallel implementations described in the previous section is assessed.

5.1. Architectures

We have assessed the performance of the three implementations/codes with $L_x = L_y = L_z = 15$, $\mathbf{D} = 0.001\mathbf{I}$ where $\mathbf{I}$ denotes the 4×4 unit matrix, and a time step equal to 0.001 (a.u.), on the following computer architectures

- A dual-Xeon$^{(TM)}$ cluster at 3.06 GHz with 2 GB RAM and 512 KB cache per processor. The nodes of the cluster are interconnected via two gigabit Ethernet networks; one for input–output and the other for computation.
- An SGI Altix 3700 Bx2 system with Intel Itanium 2 processors running at 1600 MHz with 32 KB, 256 KB and 6 MB of L1, L2 and L3 cache, respectively, and 2 GB per processor and a total RAM of 128 GB. The Altix 3700 computer system is based on a Distributed Shared Memory (DSM) architecture [25] and uses a cache-coherent Non-Uniform Memory Access (NUMA) where the latency of the processors to access the local memory is lower than the latency to access the global (or remote) memory [26].

5.2. Evaluation decisions

In order to compare the three implementations on an equal ground, some decisions had to be made as described below.

In the parallel algorithm, after the initialization of all the parameters, the solution process includes four stages per time step: (1) the update of both the local matrix $\mathbf{A}$ and the right-hand side $\mathbf{b}$, (2) the solution of the (local) system of equations $\mathbf{A}\Delta\mathbf{U} = \mathbf{b}$, (3) the update of the solution $\Delta\mathbf{U}$, and (4) interchanging the boundary values of $\mathbf{U}$. In addition, a complete simulation of bursting phenomena in the spatio-temporal systems considered here requires well over 2×10^6 time steps to simulate only one bursting cycle.

In this paper, the performance of the codes was based on that per time step. To this end, the codes were executed 100 time steps to determine the average execution time per time step.

The number of iterations required by the PCG method to converge within each time step is about ten; therefore, we have fixed the number of the iterations of the PCG method to ten for the evaluation of the three parallel implementations. Note that, as it will be shown in the next section, the solution of Eq. (1) exhibits rapid oscillations, and the number of iterations required by the CG method depends on the amplitude and frequency of these oscillations. When the solution has a smooth behavior in time, the number of iterations decreases drastically.

The compilation of the three codes and the library PETSc on both computer architectures was carried out with the Intel Fortran compiler ifort, 8-byte precision arithmetic and some optimizations (−O3 −ipo). Optimized BLAS and Lapack libraries for each architecture have been employed; in particular, the Mathematical Kernel Library (MKL) in the cluster and the Scientific Computing Software Library (SCSL) for the Altix 3700 Bx2 system were used.

The comparison of the three codes is based on two analyses; the first analysis employs a new method (described below) to represent the efficiency and all the factors that affect the performance, for two problem sizes, i.e., a coarse $51 \times 51 \times 51$-point grid and a fine $101 \times 101 \times 101$-point grid; hereon, we shall refer to these grids as simply the coarse and fine grids, respectively. The second analysis deals with the scalability of the codes, and the evaluation in this case includes a third $75 \times 75 \times 75$-point grid, hereon, referred to as the medium grid.

The codes were evaluated on, at most, 16 processors of the dual-Xeon cluster and only 8 processors on the Altix 3700 Bx2 system which were available/allowed to the authors.

The performance of a given parallel implementation on a given parallel computer is influenced mainly by three factors: (1) the cost of the communications, (2) the load unbalance and (3) the cache effect. In this paper, a novel and clear representation of the performance results of the three parallel codes is described as illustrated in Figs. 2–6 where the thicker curve represents the real efficiency of the parallel implementation, the green region presents the benefit from the cache effect, the orange region represents the load unbalance effect and the red region represents the communications effect.

In Figs. 2–6, reading from the bottom to the top, the first curve represents the efficiency of the parallel system that would

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

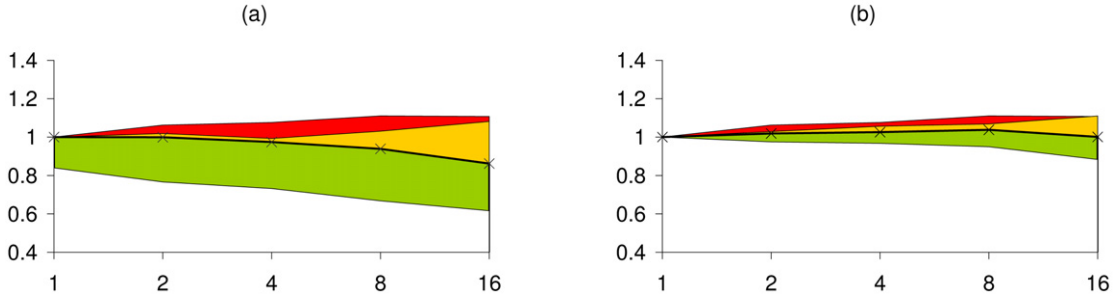

Fig. 2. Efficiency of the MPI-code on the dual-Xeon cluster versus the number of processors for the coarse (a) and fine (b) grids.

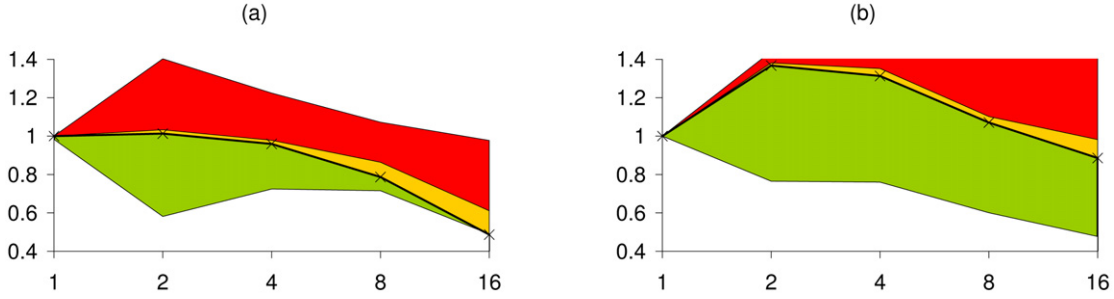

Fig. 3. Efficiency of the PETSc-code on the dual-Xeon cluster versus the number of processors for the coarse (a) and fine (b) grids.

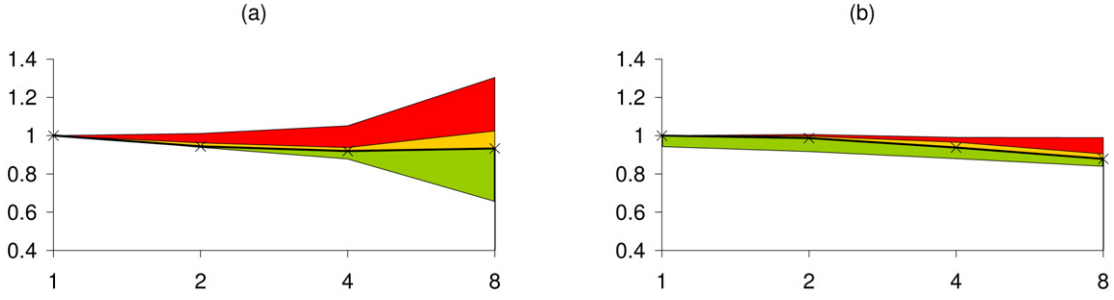

Fig. 4. Efficiency of the MPI-code on the SG Altix 3700 Bx2 system versus the number of processors for the coarse (a) and fine (b) grids.

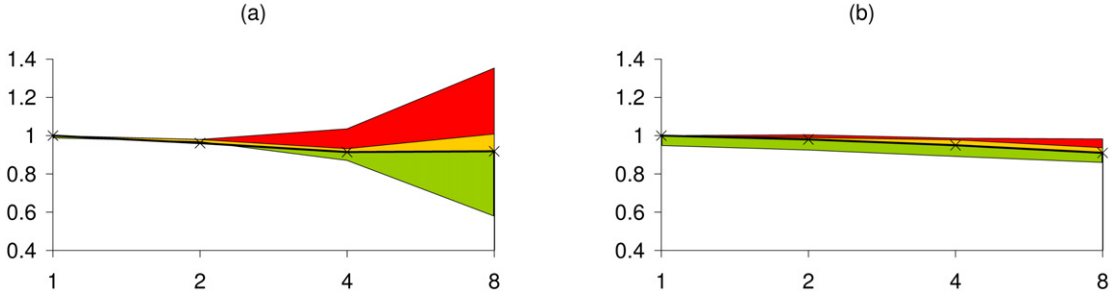

Fig. 5. Efficiency of the OpenMP-code on the SG Altix 3700 Bx2 system versus the number of processors for the coarse (a) and fine (b) grids.

result from eliminating the cache effect; the second curve represents the real efficiency including the cache effect; the third curve represents the efficiency that could be obtained without load unbalance; and, finally, the last curve represents the efficiency that could be obtained by eliminating both the load unbalance and the communication cost. The cache effect which indeed includes any other intraprocessor optimization related with the problem size, has been calculated by using as a reference the runtime of the sequential implementations (cf. Section 3) with the fine grid, i.e., the $101 \times 101 \times 101$-point grid, determining the number of floating point operations in each processor, and employing a proportionality rule for determining the time that the most loaded processor would take to perform its allocated work.

The load unbalance in this problem is due to the distribution of the matrix $\mathbf{A}$ and vector $\mathbf{b}$ amongst the processors (recall that a z-distribution was employed), and has been determined analytically by extrapolation from an unbalanced computation

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

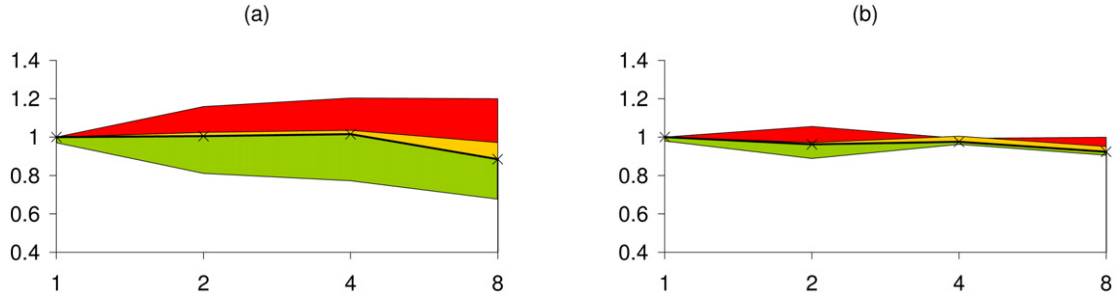

Fig. 6. Efficiency of the PETSc code on the SG Altix 3700 Bx2 system versus the number of processors for the coarse (a) and fine (b) grids.

to a balanced or equilibrated one, because assigning the same workload to each processor would interfere with the other effects mentioned above. It must, however, be pointed out that an analytical determination of the load unbalance may not be feasible if the data from the same z-plane is assigned to different processors, for this would require slight changes in the codes and would result in a slightly higher cost and, most importantly, a greater interference amongst data interchanges which would, in turn, result in a narrower band for the load unbalance effect illustrated in Figs. 2–6, but which would not affect the width of the communications effect. In our case, the interest in the load balance is merely informative.

The elimination of the communication cost has been achieved by suppressing the inter-processor communications; in this case, the simulation has been executed using outdated or old values, which do affect the solution accuracy but do not affect the performance measurement, since the number of the CG iterations was fixed to ten, as discussed above.

The comparative analysis of the performance of the MPI, OpenMP and PETSc implementations has been carried out on the dual-Xeon cluster and on the Altix 3700 Bx2 system as described in the next two sections.

5.3. Results on the dual-Xeon cluster

Figs. 2 and 3 show the efficiency of the MPI and PETSc implementations, respectively, on the dual-Xeon cluster versus the number of processors for the coarse (a) and fine (b) grids. These figures show that the efficiency of the MPI-code is, in general, higher than that of the PETSc implementation for both grids, with better behavior for the fine grid, especially when the number of processors is larger than 8. This is due to the fact that the influence of the communications and load unbalance decreases as the number of mesh points is increased. In addition, there is a remarkable benefit from the cache effect when the number of processors increases, and this benefit is more considerable for the coarse grid, even in the sequential implementation.

The performance of the PETSc-code is also good for the fine grid, with super-linear efficiency from two to eight processors. This is a consequence of the improvement in the memory management, as it can be clearly observed in Fig. 3(b). For the coarse grid, the decrease in the efficiency starts earlier when the communications and load unbalance become important, since the improvement in the cache exploitation does not compensate them. The OpenMP-code cannot be evaluated on the dual-Xeon cluster due to the limitations of this platform as a shared memory device.

Table 1
Runtimes and speedup (SPU) of the MPI and PETSc implementations per time step on the dual-Xeon cluster for three grid sizes $N = N_x = N_y = N_z$

N	MPI-code			PETSc-code		
	1 proc.	8 proc.	SPU	1 proc.	8 proc.	SPU
51	6.23	0.787	7.904	8.71	1.383	6.695
75	20.55	2.466	8.331	28.38	3.391	8.370
101	56.65	6.663	8.501	68.76	8.029	8.564

Table 2
Runtimes and speedup (SPU) of the MPI and PETSc implementations on the SGI Altix 3700 Bx2 system for the three grid sizes $N = N_x = N_y = N_z$

N	MPI-code			PETSc-code			OpenMP-code		
	1 proc.	8 proc.	SPU	1 proc.	8 proc.	SPU	1 proc.	8 proc.	SPU
51	2.57	0.345	7.465	5.77	0.815	7.079	2.58	0.346	7.446
75	8.18	1.215	6.730	18.85	2.415	7.802	8.42	1.208	6.965
101	19.87	3.151	6.960	46.19	6.245	7.280	20.11	2.762	7.281

Table 1 shows the runtimes of the MPI and the PETSc implementations on one and eight processors, and their corresponding speedups, for the coarse, medium and fine grids. This table shows that both codes scale very well on the dual-Xeon cluster, showing super-linear speedup for the fine grids. However, in terms of runtimes, the MPI code is about 130% better than the PETSc one; this is due to the fact that the communications and load unbalance are very optimized in the MPI-code; in addition, there is an improvement in the cache factor. It must be noted that the CRS storage for the sparse matrix in PETSc does not take into account the sparsity pattern of the 4×4 blocks (cf. Section 2) which can penalize even further the performance of the PETSc-code.

5.4. Results on the Altix 3700 Bx2 system

Figs. 4, 5 and 6 show the performance of the MPI, OpenMP and PETSc implementations, respectively, on the SGI Altix 3700 Bx2 system for the coarse (a) and fine (b) grids. Good efficiencies have been obtained for both the MPI and OpenMP implementations, with a very slight decrease due to the eventual increase in the communications and load unbalance factors. The efficiency of the PETSc-code is also good, especially for the

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

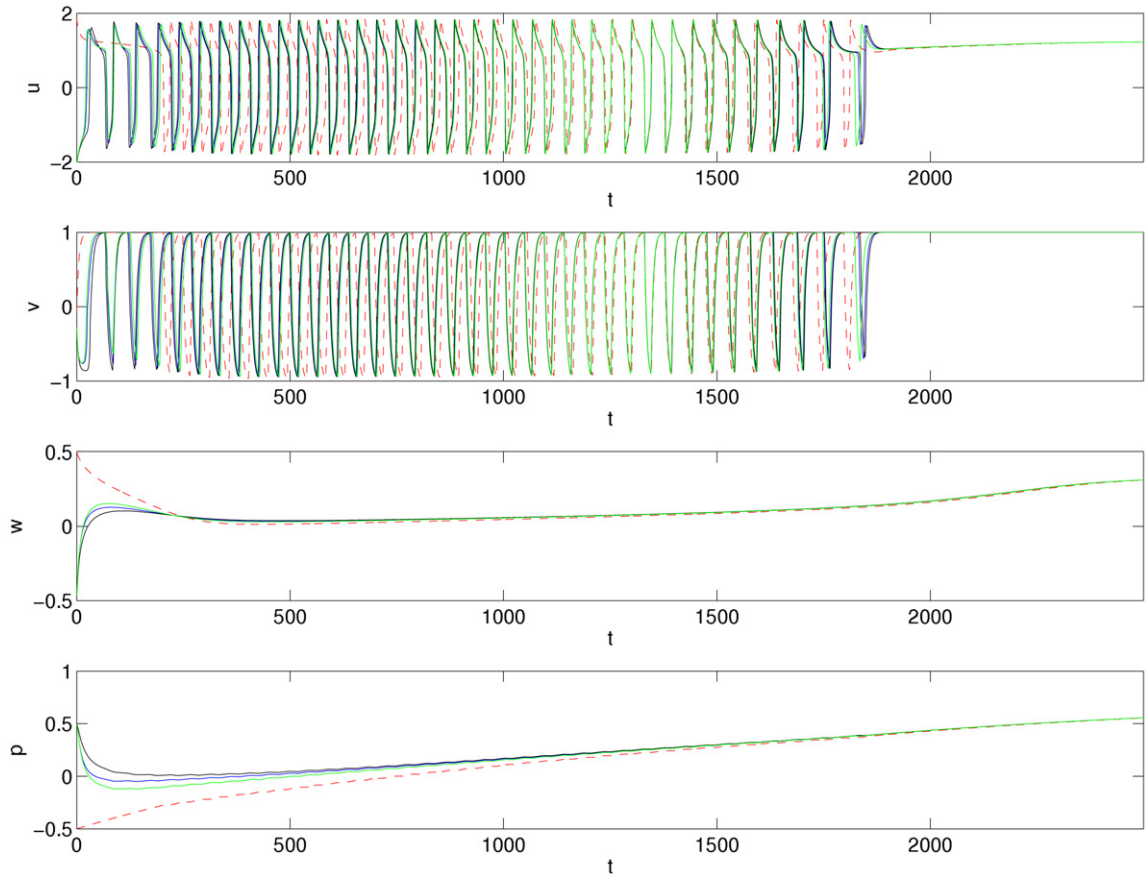

Fig. 7. u, v, w and p as functions of t (a.u.) at the monitoring locations. The red, green, blue and black curves correspond to the monitoring locations 1 and 2, 3, 4 and 5, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

fine grid due to the decrease of the influence of the communications and load unbalance factors.

Table 2 shows the runtimes of the MPI, PETSc and OpenMP implementations on the Altix 3700 Bx2 system for one and eight processors and their corresponding speedups for the coarse, medium and fine grids. This table shows that the scalability of all the codes is good and similar, independently of the grid size. However, in terms of runtimes, the PETSc-code is approximately 230% slower than both the MPI and the OpenMP implementations. This can be explained mainly by the bad behavior of the CSR storage on Itanium for matrices of small rows [23].

6. Presentation of results

As stated above, calculations were performed on the coarse and fine grids mentioned above with $L_x = L_y = L_z = 15$, $\mathbf{D} = 0.001\mathbf{I}$ where $\mathbf{I}$ denotes the 4×4 unit matrix, and a time step equal to 0.001 (a.u.). The coarse mesh represents the largest spatial step size for which accurate results could be obtained for the problem considered here, and, therefore, the results presented in this section were obtained with the fine grid. The calculations were performed up to $t = 5000$ which corresponds to 5×10^6 time steps in order to capture the bursting as well as the extinction or quiescence of the solution.

Before proceeding with the presentation of some sample results, it is worth mentioning that a huge amount of data was generated and, therefore, the results presented here only show some of the most important features of the computations. Even though, only time histories of the four dependent variables at five monitoring points and some snapshots of the solutions are presented at selected times and selected z-planes in this section, the discussion that follows is also based on three-dimensional, time-dependent visualizations of the isocontours of the dependent variables.

For the initial conditions described above, the time histories of the four dependent variables at five monitoring points exhibit the trends illustrated in Fig. 7. The monitoring points are numbered 1–5 and their locations correspond to $(x, y, z) = (-14.4, -14.4, -14.4)$, $(13.8, 13.8, 13.8)$, $(6, -6, 6)$, $(-6, -6, 0)$ and $(-3, 0, 0)$, respectively; therefore, the monitoring points 1 and 2 which are identified with the red color in the figures are located in $\boldsymbol{\Omega} - \boldsymbol{\Gamma}$, whereas the monitoring points 3, 4 and 5 which are identified with red, blue and black, respectively, are located in $\boldsymbol{\Gamma}$ (cf. Section 2).

Fig. 7 indicates that the time history of u shows oscillations characterized by steep temporal gradients at the monitoring points 3–5, and that these oscillations are not in-phase for most of the integration time considered in this paper. Fig. 7 also shows that, at the location of the monitoring points 1 and 2, u first decreases from its initial value in a smooth manner

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

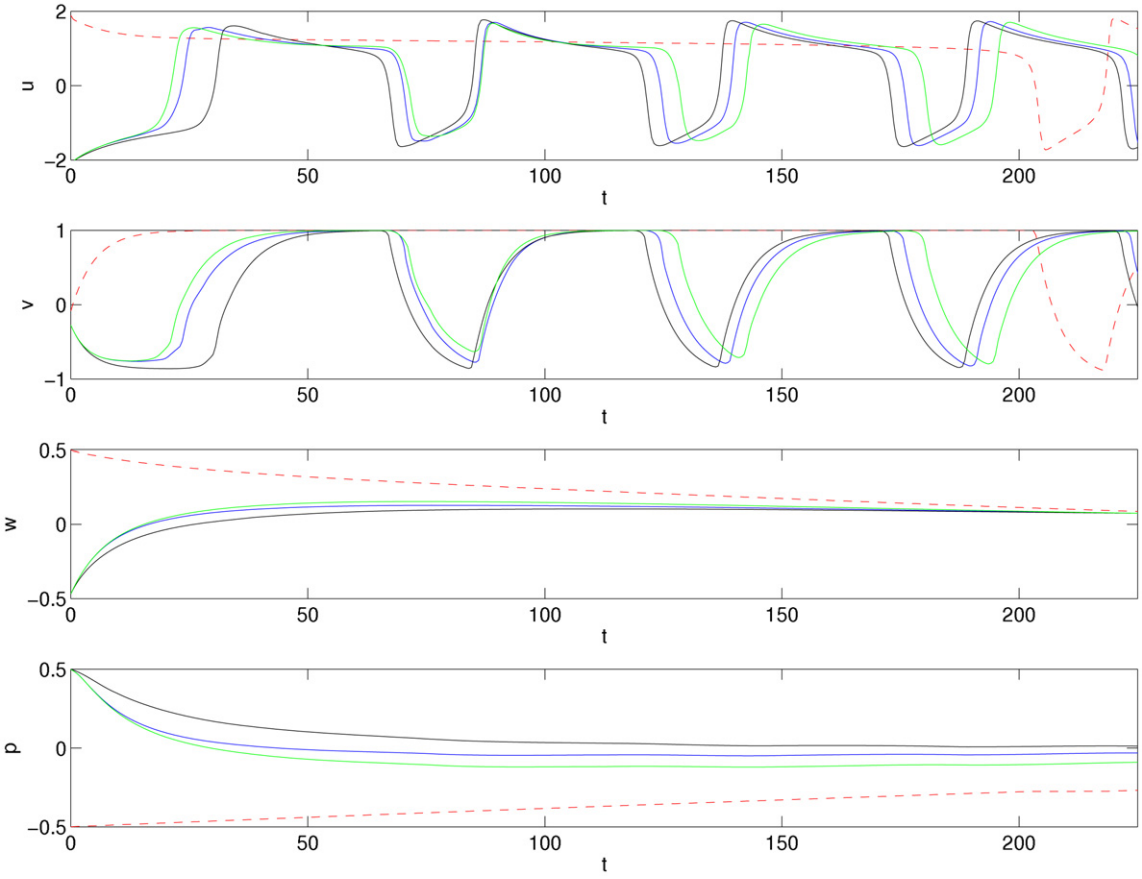

Fig. 8. u, v, w and p as functions of t (a.u.) in the first regime at the monitoring locations. The red, green, blue and black curves correspond to the monitoring locations 1 and 2, 3, 4 and 5, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

prior to oscillating in a vigorous fashion; a similar behavior can be observed in the time history of v which shows that the time histories at the monitoring locations 3–5 exhibit oscillations from $t = 0$, whereas those at the locations 1 and 2 first increase rapidly and then reach an almost constant value before they start oscillating in a fast manner.

Fig. 7 also shows that there is an interval of time from about $t = 1200$ to 1450 when the time histories of u and v at the five monitoring locations are nearly identical. At larger times, the time histories differ from each other, and, about $t = 1900$, both u and v reach stationary values that correspond to a steady state or quiescence. On the other hand, the time histories of w and p initially decrease and increase, respectively, at the monitoring locations 1 and 2, increase and decrease, respectively, at the monitoring locations 3–5, and asymptotically tend to each other in a slightly increasing fashion.

Fig. 7 also illustrates that there are very few differences between the time histories of the four dependent variables at the monitoring location 1 and those at the location 2.

The results illustrated in Fig. 7 indicate that one can clearly distinguish four main regimes in the solution. These regimes correspond to roughly $0 \leqslant t < 250$, $250 \leqslant t < 800$, $800 \leqslant t < 1850$, and $t \geqslant 1850$. The first corresponds to the initial response and its time history is illustrated in Fig. 8 which clearly illustrates the initially smooth decreasing and increasing, respectively, behavior of w and p at the monitoring locations 1 and 2 and the smoothly increasing and decreasing, respectively, character of these two variables at the monitoring locations in $\mathbf{\Gamma}$. This figure also illustrates the initial complex behavior of both u and v and, in particular, the fact that the time histories of these two variables at the monitoring points located in $\mathbf{\Omega} - \mathbf{\Gamma}$ decrease and increase smoothly before they start oscillating in an analogous manner to those of the other three monitoring locations, i.e., those located in $\mathbf{\Gamma}$. It is worthy stating that it takes about $t = 200$ for the monitoring points located in $\mathbf{\Omega} - \mathbf{\Gamma}$ to begin oscillating in a similar manner to those in $\mathbf{\Gamma}$.

The results of the time histories presented in Fig. 8 as well as of those based on the visualization of the time-dependent isocontours of the four dependent variables indicate that, until about $t = 200$, most of the interesting dynamics takes place in $\mathbf{\Gamma}$. This can be appreciated in a better manner in the snapshots of the solutions presented in Figs. 9 and 10 that correspond to $z = 0$ and indicate that u and v oscillate in $\mathbf{\Gamma}$ as illustrated in Fig. 9 at $t = 47.5$ and 57.5; on the other hand, the results presented in Fig. 10 indicate that w and p evolve smoothly without oscillations as indicated previously in Fig. 8. Figs. 9 and 10 also indicate that the values of u and v at $(x, y) = (-15, 15)$ are 1.00 and 1.00, respectively, at both $t = 47.5$ and 57.5, whereas the values of w and p at the same location are equal to about 0.33 and -0.42, respectively, at $t = 47.5$, and 0.30 and -0.45, respectively, at $t = 57.5$.

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

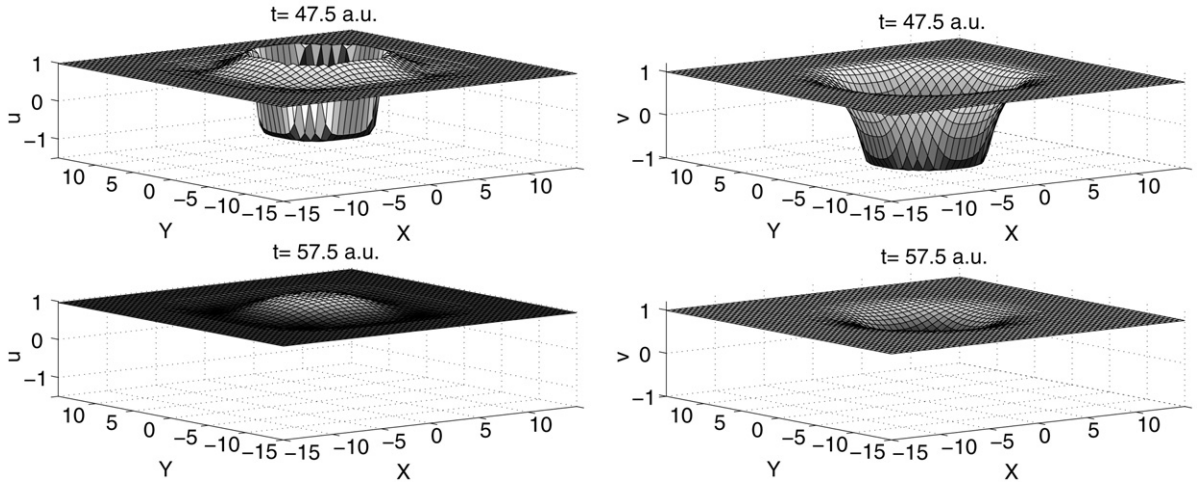

Fig. 9. u (left) and v (right) as functions of x and y at $z = 0$ in the first regime at $t = 47.5$ (top) and 57.5 (bottom).

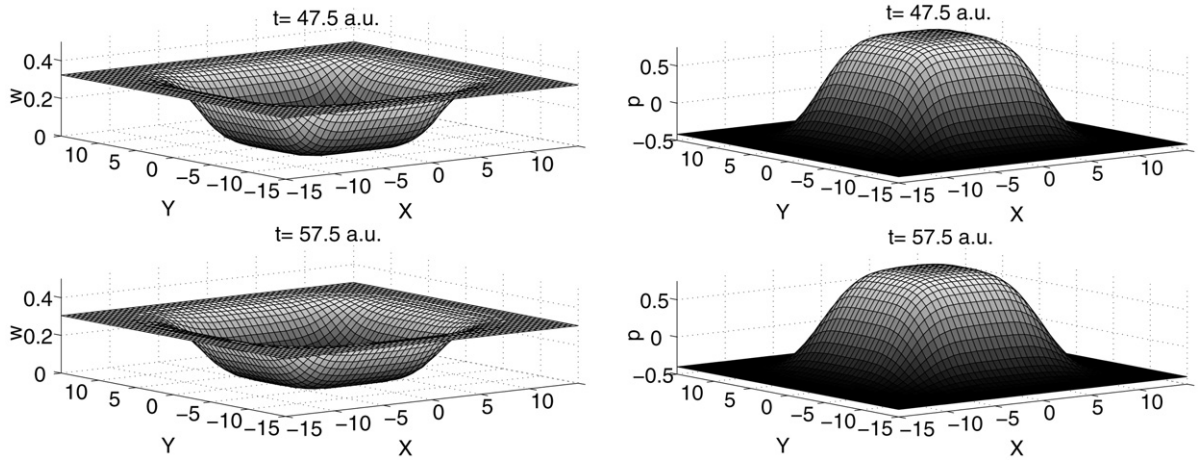

Fig. 10. w (left) and p (right) as functions of x and y at $z = 0$ in the first regime at $t = 47.5$ (top) and 57.5 (bottom).

In the second regime, the time histories of u and v at the five monitoring locations illustrated in Fig. 11 indicate that the monitoring points located in $\mathbf{\Gamma}$ and $\mathbf{\Omega} - \mathbf{\Gamma}$ exhibit oscillations of the same characteristics. The phase difference between these oscillations decreases as time increases. On the other hand, w and p show a slight decrease and a slight increase, respectively, as functions of time. The differences in w at the five monitoring locations are much smaller than the differences in p.

The fact that, in the second regime, the monitoring points located in $\mathbf{\Omega} - \mathbf{\Gamma}$ exhibit an oscillatory pattern can also be observed in Figs. 12 and 13 that correspond to two different times and clearly indicate that there is propagation from $\mathbf{\Gamma}$ to $\mathbf{\Omega} - \mathbf{\Gamma}$ and then from $\mathbf{\Omega} - \mathbf{\Gamma}$ to $\mathbf{\Gamma}$. This behavior is analogous to breathing or heat waves. On the other hand, w and p exhibit an uninteresting result, for the evolution of these variables is very slow in the second regime as shown in Fig. 13, in agreement with the time histories illustrated in Fig. 11.

Figs. 12 and 13 also indicate that the values of u and v at $(x, y) = (-15, 15)$ are 0.80 and 0.90, respectively, at $t = 685$, and -1.40 and 0.47, respectively, at 695, whereas the values of w and p at the same location are equal to about 0.33 and 0.26, respectively, at $t = 685$, and 0.32 and 0.26, respectively, at $t = 695$. Therefore, u and v exhibit a translational breathing motion.

In the third regime, the phase differences between the time histories of u and w measured in $\mathbf{\Gamma}$ and $\mathbf{\Omega} - \mathbf{\Gamma}$ first decrease, then become nil, and then increase as time increases as illustrated in Fig. 14, whereas the time histories of w and p indicate that w increases extremely slowly with time and p increases in an almost linear manner with t.

The oscillatory behavior of u and v in the third regime is shown in the snapshots presented in Fig. 15 that once again have the same characteristics as those of a translational breathing motion. Note that even though the nonuniformities of u and v are relatively small at the times shown in Fig. 15, the values of these two variables at $(x, y) = (-15, 15)$ are about 0.75 and 1.00, respectively, at $t = 1002.5$, and -1.35 and -0.25, respectively, at $t = 1012.5$. On the other hand, w and p exhibit the snapshots illustrated in Fig. 16; although these two variables are non-homogeneous in $z = 0$, the level of non-homogeneity is very small.

In the fourth regime, the difference in phase between the time histories of u and v in $\mathbf{\Gamma}$ and $\mathbf{\Omega} - \mathbf{\Gamma}$ increase, while the amplitude of these two variables decreases until they reach con-

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

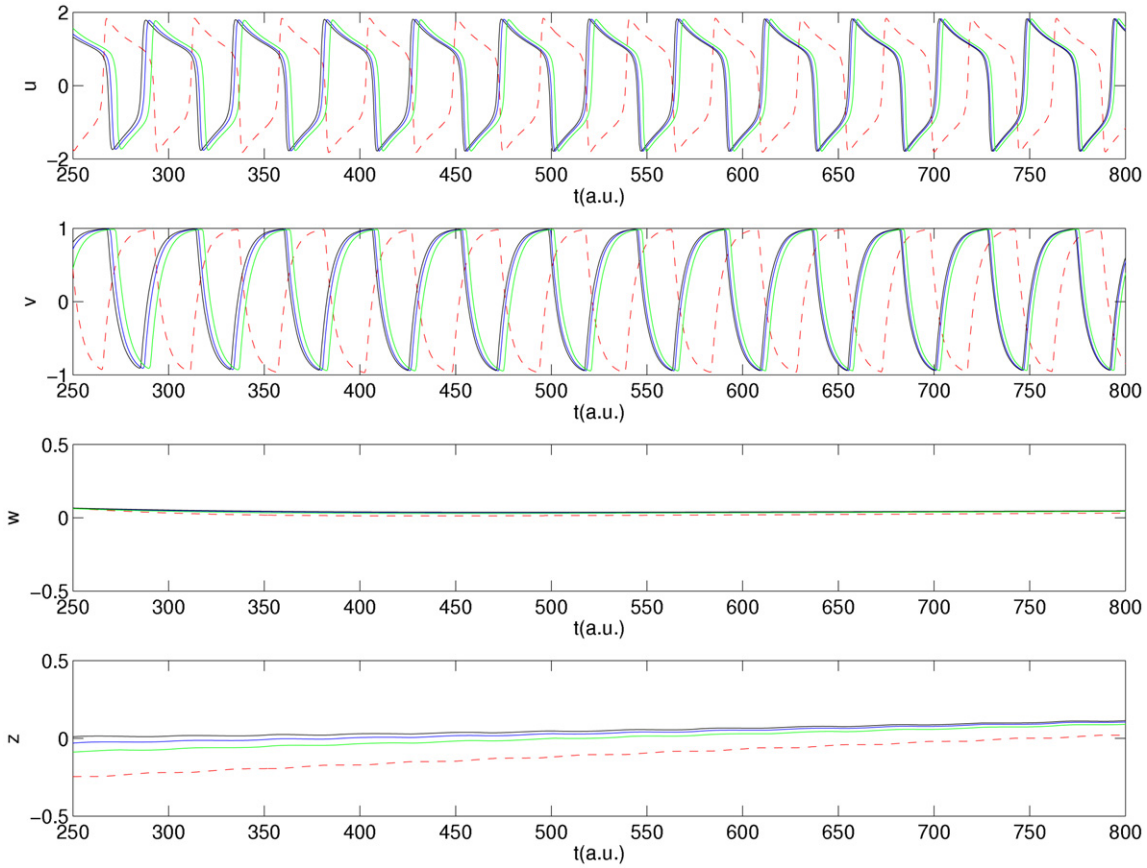

Fig. 11. u, v, w and $p \equiv z$ as functions of t (a.u.) in the second regime at the monitoring locations. The red, green, blue and black curves correspond to the monitoring locations 1 and 2, 3, 4 and 5, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

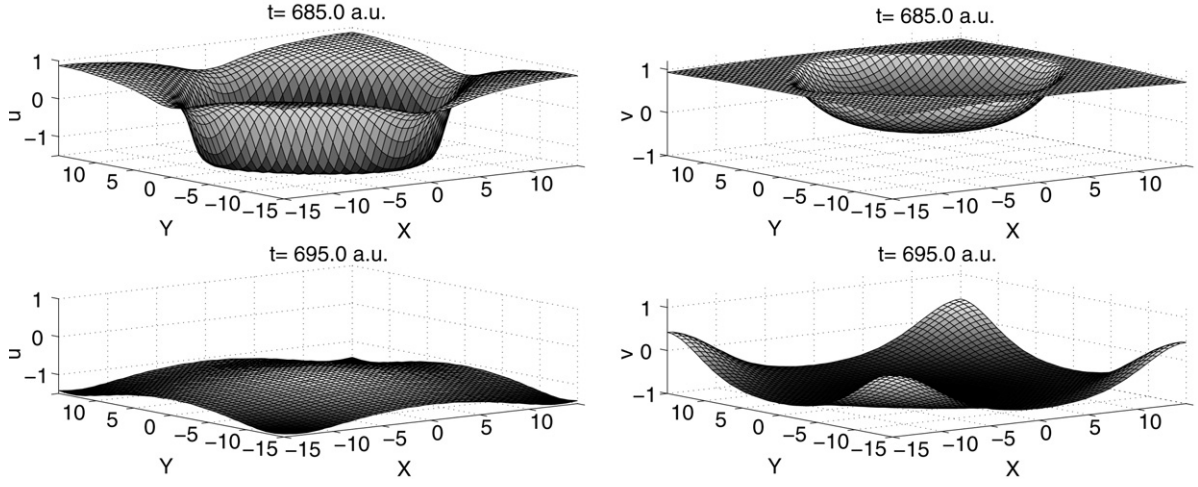

Fig. 12. u (left) and v (right) as functions of x and y at $z = 0$ in the second regime at $t = 685$ (top) and 695 (bottom).

stant values as illustrated in Fig. 17. On the other hand, w increases very slightly with time and p does so at a slightly faster pace.

As stated before, the computations were performed until $t = 5000$, but very few differences were found between the results presented in Fig. 17 and those at $t = 5000$, thus indicating that, for the initial conditions considered in this paper, the fate of the solution is a state of quiescence, at least, until $t = 5000$, despite the fact that the results shown in Fig. 17 indicate that p seems to increase very slowly with time, for this variable is a slow one. In fact, $\mathbf{U}_i(t = 2000) - \mathbf{U}_i(t = 1999) = (0, 0, 2.5 \times 10^{-5}, 3.9 \times 10^{-5})^{\mathrm{T}}$, for $i = 1, 2$, and $(0, 0, 2.3 \times 10^{-5}, 3.8 \times 10^{-5})^{\mathrm{T}}$, for $i = 3, 4, 5$, where i denotes the monitoring locations.

Calculations not reported here were also performed in the fine grid with a time step equal to 0.002 until $t = 25000$ and their results indicate that a quiescence state is eventually

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

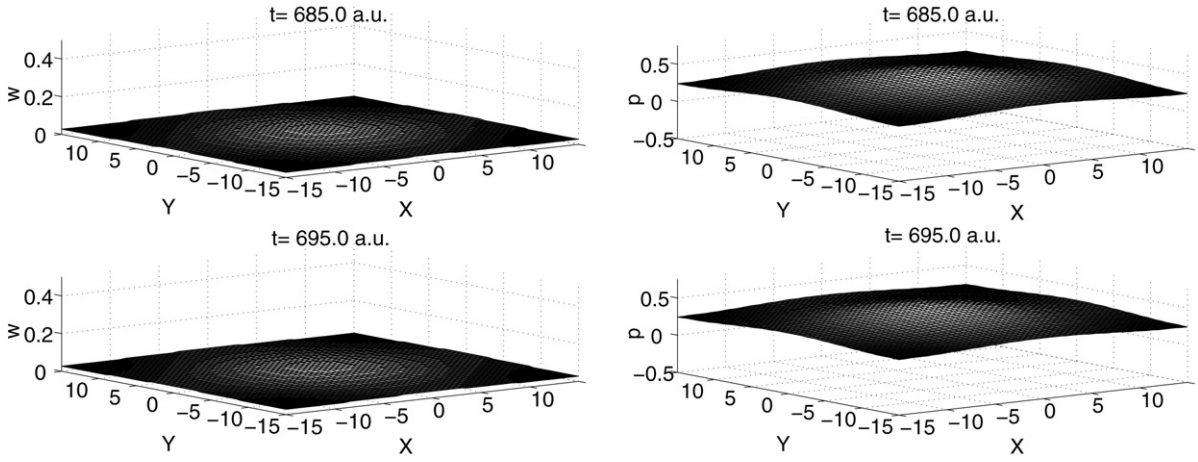

Fig. 13. w (left) and p (right) as functions of x and y at $z = 0$ in the second regime at $t = 685$ (top) and 695 (bottom).

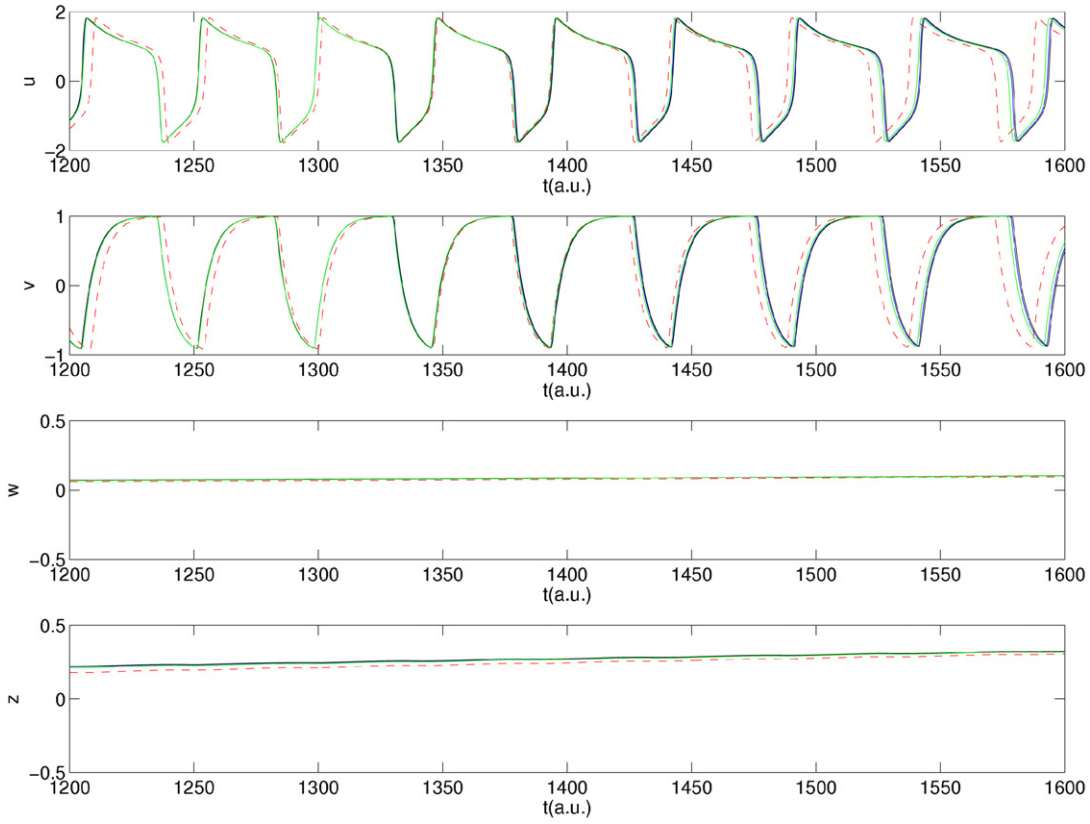

Fig. 14. u, v, w and $p \equiv z$ as functions of t (a.u.) in the third regime at the monitoring locations. The red, green, blue and black curves correspond to the monitoring locations 1 and 2, 3, 4 and 5, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

reached, for there were differences only in the sixth decimal digit for all the four components of **U** between the results at $t = 25000$ and those at $t = 5000$. These calculations also indicate that w and p evolve very slowly towards their final equilibrium or steady state values as indicated in Fig. 17.

It must be emphasized that the four regimes describe above and which include asynchronous, then synchronous and finally synchronous oscillations before a finally quiescent state is reached, have only been observed for the initial conditions, diffusion coefficients and domain considered in this paper. In two dimensions, results not shown here indicate that the dynamics of the four-equation model considered here exhibit a rich dynamic behavior including quiescent, periodicity, quasi-periodicity and chaos depending on the values of u_0, α_w, β_p and g (cf. Section 2).

7. Conclusions

A mathematical model of three-dimensional bursting phenomena and three different parallel implementations of it have

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

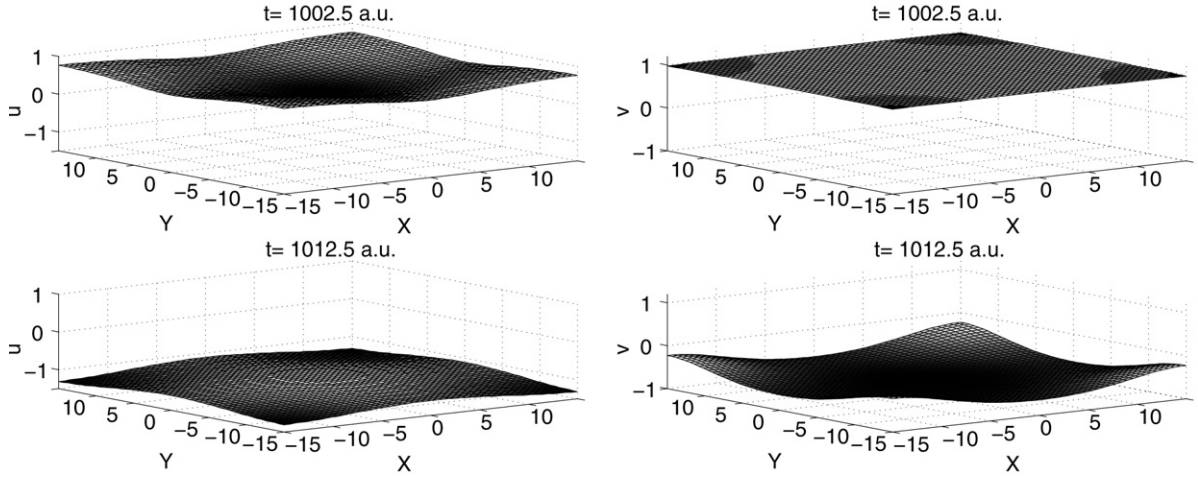

Fig. 15. u (left) and v (right) as functions of x and y at $z = 0$ in the third regime at $t = 1002.5$ (top) and 1012.5 (bottom).

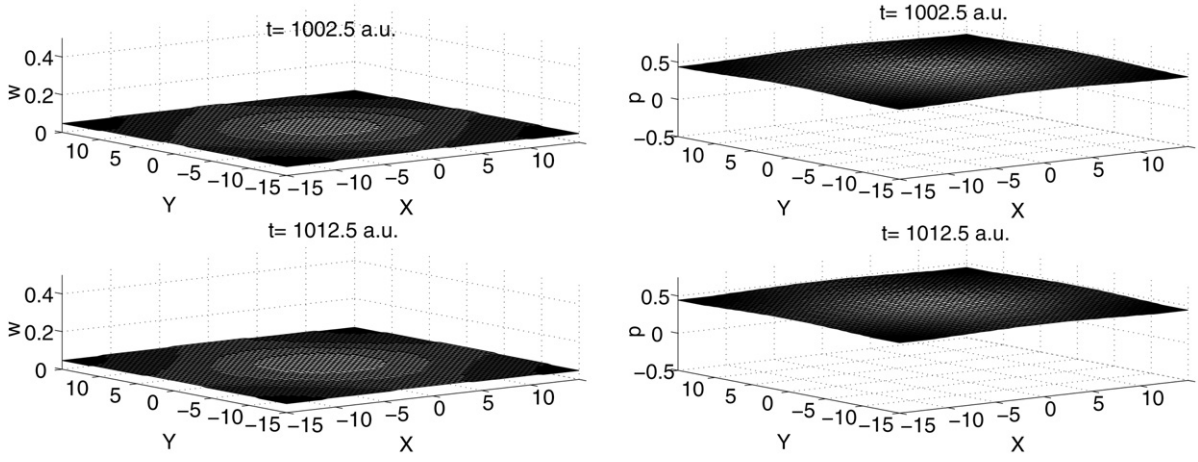

Fig. 16. w (left) and p (right) as functions function of x and y at $z = 0$ in the third regime at $t = 1002.5$ (top) and 1012.5 (bottom).

been presented. The model is described by four, nonlinearly coupled partial differential equations which have been discretized by means of a second-order accurate, linearly-implicit finite difference method in equally-spaced grids. The resulting system of linear algebraic equations at each time level has been solved by means of the Preconditioned Conjugate Gradient technique.

Three different parallel implementations of the proposed mathematical model have been developed. Two of these implementations, i.e., the MPI and the PETSc codes, are based on a message passing paradigm, while the third one, i.e., the OpenMP code, is based on a shared space address paradigm. These three codes have been evaluated on two parallel architectures, i.e., a cluster of dual-processor nodes and a Shared Distributed Memory system. The efficiency results of these implementations have been represented by means of a new method that clearly illustrates the effects of the communications, load unbalance and cache on the parallel performance.

It has been shown that the use of PETSc on symmetric multiprocessor (SMP) clusters can be a good choice, for it requires low programming effort. However, on Shared Distributed Memory systems, more time and programming effort should be spent to reach good performance. It has also been shown that, on the SDM system, the proposed MPI and OpenMP codes are about 230% more efficient than the PETSc code; on the cluster of dual-processor nodes, the MPI code is about 130% more efficient than the PETSc code. These results can be extrapolated to other scientific applications that are based on the numerical solution of partial differential equations where the discretization yields a banded matrix.

In terms of the physics of the bursting problem considered in the paper, it has been shown by means of visualization studies, time histories at five monitoring locations, and snapshots of three-dimensional isosurfaces at selected times that, for the initial conditions, diffusion coefficients, nonlinear reaction terms and geometry considered here, the nonlinear dynamics exhibits four clearly different regimes. In the first regime, the main activity takes place initially in the inner cube and the variables in the outer one exhibit initially very small variations. After some time, waves propagate from the inner to the outer cube and this results in a decreasing phase difference between the time histories of the fast system in the inner and outer cubes, in the second regime. In the third regime, the phase difference first decreases, then becomes zero and then increases as a function

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

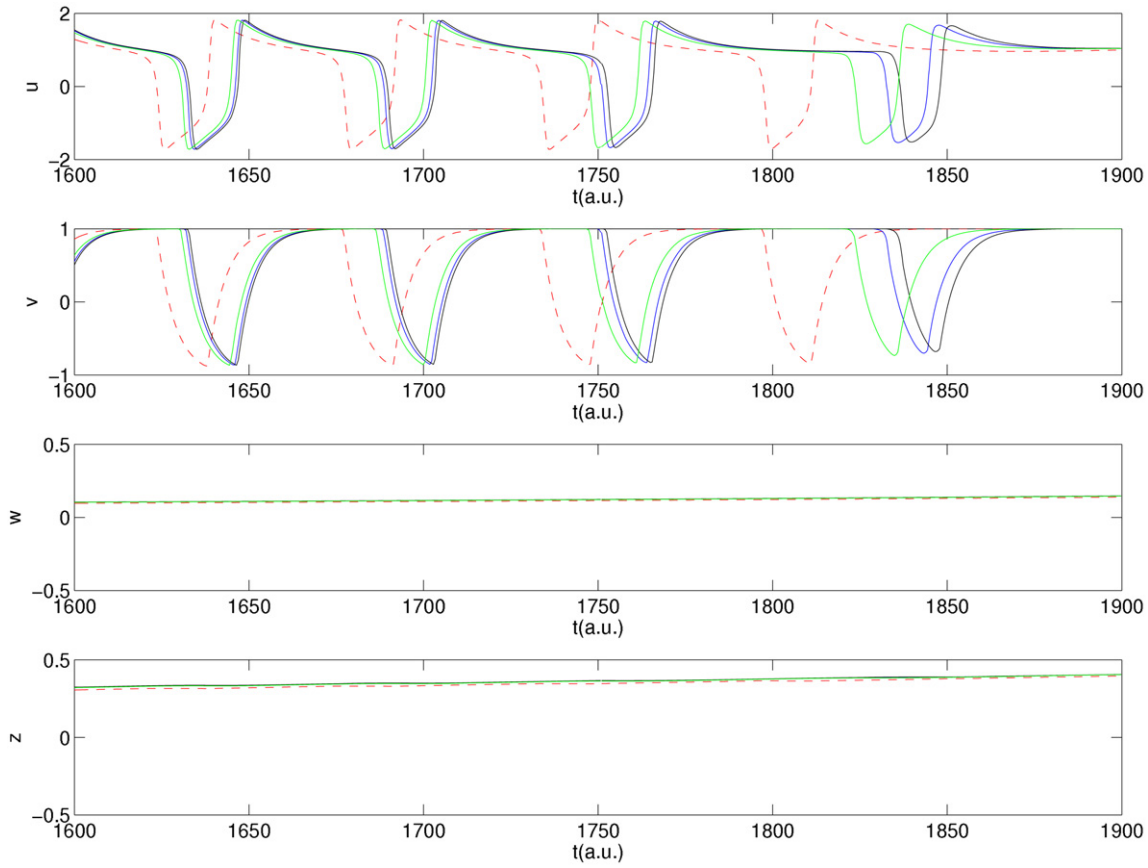

Fig. 17. u, v, w and $p \equiv z$ as functions of t (a.u.) in the fourth regime at the monitoring locations. The red, green, blue and black curves correspond to the monitoring locations 1 and 2, 3, 4 and 5, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of time, and the dynamics of the fast variables is characterized by a translational breathing motion associated with the propagation of reacting waves from the inner to the outer cube and vice versa. In the fourth regime, the fast variables settle to a steady state whereas the slow ones increase very slightly with time before complete quiescence is achieved through the domain.

Acknowledgements

This work was partially supported by the Ministerio de Educación y Ciencia of Spain under Projects TIN2005-00447 and FIS2005-03191 (JIR) and Fondos FEDER

References

[1] S. Coombes, P.C. Bressloff (Eds.), Bursting: The Genesis of Rhythm in the Nervous System, World Scientific, Singapore, 2005.
[2] S.T. Fleischmann, J.M. Wallace, Mean streamwise spacing of organized structures in transitional and developed bounded turbulent flows, AIAA J. 22 (1984) 766–769.
[3] J.M. López, J.E. Hart, F. Marques, S. Kittelman, J. Shen, Instability and mode interactions in a differentially driven rotating cylinder, J. Fluid Mech. 462 (2002) 383–409.
[4] E. Ott, T.M. Antonsen, D.P. Lathrop, J.M. Finn, Blowout bifurcations and the onset of magnetic dynamo action, Physics of Plasmas 8 (2001) 1944–1952.
[5] T. Ozeki, JT-60 team, Studies of MHD behaviour in JT-60U, Plasma Phys. Control. Fusion 45 (2003) 645–655.
[6] R.A. Ong, Very high-energy gamma-ray astronomy, Physics Reports 305 (1998) 93–202.
[7] N.F. Rulkov, Modeling of spiking-bursting neural behavior using two-dimensional map, Physical Review E 65 (2002) 041922.
[8] J. Keener, J. Sneyd, Mathematical Physiology, Springer, New York, 1998.
[9] J.D. Murray, Mathematical Biology, Springer, New York, 1989.
[10] R. Meucci, A. Di Garbo, E. Allaria, F.T. Arrecchi, Autonomous bursting in a homoclinic systems, Physical Review Letters 88 (2002) 144101.
[11] D.J. DeShazer, J. García-Ojalvo, R. Roy, Bursting dynamics of a fiber laser with an injected signal, Physical Review E 67 (2003) 036602.
[12] P. Gray, G. Nicolis, F. Baras, P. Borckmans, S.K. Scott (Eds.), Spatial Inhomogeneities and Transient Behaviour in Chemical Kinetics, John Wiley & Sons, New York, 1992.
[13] L. Hsu, R. Costa, Bursting phenomena in continuous-time adaptive control systems with σ-modification, IEEE Trans. Automat. Contr. AC-32 (1987) 84–86.
[14] J. Rinzel, Y.S. Lee, Dissection of a model for neuronal parabolic bursting, J. Math. Biol. 25 (1987) 653–675.
[15] J. Rinzel, Electrical excitability of cells, theory and experiment: Review of the Hodgkin–Huxley foundation and an update, Bull. Math. Biol. 52 (1990) 5–23.
[16] P. Smolen, J. Keizer, Slow voltage inactivation of Ca^{+2} currents and bursting mechanisms for the mouse pancreatic β-cell, J. Membr. Biol. 127 (1992) 9–19.
[17] R. Bertram, M.J. Buttle, T. Kiemel, A. Sherman, Topological and phenomenological classification of bursting oscillations, Bull. Math. Biol. 57 (1995) 413–439.
[18] S. Wiggins, Normally Hyperbolic Invariant Manifolds in Dynamical Systems, Springer, New York, 1994.
[19] J. Rinzel, W.C. Troy, Busting phenomena in a simplified Oregonator flow system model, J. Chem. Phys. 76 (1982) 1775–1789.

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005

[20] G.A. Carpenter, Bursting phenomena in excitable membranes, SIAM J. Appl. Math. 36 (1979) 334–372.
[21] J.I. Ramos, Linearization methods for reaction–diffusion equations: Multidimensional problems, Appl. Math. Comput. 88 (1997) 225–254.
[22] E.M. Ortigosa, L.F. Romero, J.I. Ramos, Parallel scheduling of the PCG method for banded matrices rising from FDM/FEM, J. Parall. Distr. Comput. 63 (2003) 1243–1256.
[23] J. Mellor-Crummey, J. Garvin, Optimizing sparse matrix–vector product computations using unroll and jam, Int. J. High Perfor. Comput. Appl. 18 (2004) 225–236.
[24] S. Balay, K. Buschelman, W. Gropp, D. Kaushik, M. Knepley, L.C. McInnes, B. Smith, H. Zhang, PETSc Users Manual, Revision 2.1.5, ANL-95/11, Argonne National Laboratory, Chicago, 2002.
[25] J. Protic, M. Tomasevic, V. Milutinovic, Distributed Shared Memory: Concepts and Systems, John Wiley & Sons, New York, 1997.
[26] T. Dunigan, J. Vetter, P. Worley, Performance evaluation of the SGI Altix 3700, in: Proc. of the IEEE Int. Conf. Parallel Proc., ICPP (2005) 231–240.
[27] G. Krawezik, F. Cappello, Performance comparison of MPI and OpenMP on shared memory multiprocessors, Concurr. Comput.: Practice and Experience 18 (2006) 29–61.

Please cite this article in press as: S. Tabik et al., On a model of three-dimensional bursting and its parallel implementation, Computer Physics Communications (2007), doi:10.1016/j.cpc.2007.10.005