Parallelizing Implicit Algorithm for Time-Dependent Problems by Parabolic Domain Decomposition *

appeared in J. of Scientific Computing, Vol. 8, No. 2, pp. 151-166, June 1993

A. Averbuch[‡] , M. Israeli[†], L. Vozovoi[†]

[†] Faculty of Computer Science, Technion, Haifa 32000, Israel

[‡] School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel

Abstract

This paper presents a novel parallel high-order algorithm for the solution of parabolic and elliptic PDE's. The parallelization is achieved by using a domain decomposition technique. For the spatial discretization the trigonometric Fourier basis is employed along with a smoothing procedure in order to preserve spectral accuracy. In the time-discretization scheme the diffusive term is treated implicitly. The matching of the elemental subsolutions which generates the global solution, is performed by using the boundary Green's functions. To avoid the global coupling and data transfer, inherent in the implicitness of the time-stepping algorithm, the localization properties of the boundary Green's functions are exploit in an explicit way. In effect, only local communications between neighbouring processors are necessery so that the parallel algorithm becomes scalable. Another advantage of the present approach is that it allows the implementation of different resolution in each subdomain, which makes it valuable as an adaptive algorithm.

Key-words: Domain decomposition,parabolic and elliptic PDEs, spectral method, lcoal Fourier bases,Green function, parallelization.
AMS(MOS) subject classification: 35K25, 35P05, 35R05
Abbreviated title: Parabolic Domain Decomposition of Time-Dependent Problems.

1 Introduction

The numerical solution of multidimensional non-linear evolution equations (and especially the particularly difficult case of the incompressible Navier-Stokes equations) is a heavy computational task. At the range of parameters, which is of interest in applications (for example

*This research was supported partly by a grant from the French-Israeli Binational Foundation for 1991-1992. The work of the third author was also supported by a grant from Israel Ministry of Science and Technology for 1992-1993

high Reynolds numbers) the resolution requirements in time and space increase enormously. Consequently, a huge number of degrees of freedom must be computed, and memory and CPU requirements increase beyond the capacity of present day supercomputers.

The parallel machines of MIMD (message-passing or shared-memory) type are scalable, i.e. in principle they are able to grow in accordance with the requirements of the task. But such parallel machines can be effective only if efficient *parallelizing algorithms* can be supplied, otherwise, communication and synchronization overheads destroy completely the performance of the best serial methods. It is known that not all problems are amenable to efficient parallelization. For some problems, parallelism cannot help because of essential serial bottlenecks, whereas other problems are completely parallelizable (with suitable algorithms).

The numerical solution of Partial Differential Equations (PDE's) has the potential for efficient parallelization on local memory machines because partial derivatives represent local behaviour. In some cases described by hyperbolic PDE's, the local information is actually transmitted with a finite speed of propagation. These two last properties of the physical solution are essential for straightforward parallelization by geometric domain decomposition where neighbouring pieces of space are allocated to processors with short communication links. On the other hand, bottlenecks are expected where the physical properties dictate a requirement of global communication, or for algorithms which require massive transfer of data. For example, equilibrium processes described by elliptic PDE's result from global interactions and therefore the solution of a corresponding problem requires global communications between subdomains.

Parabolic PDE's describing dissipative evolution processes, which are of most interest to us, hold an intermediate position; they permit an infinite speed of propagation but only for very small changes, while large changes propagate at finite speeds. The Navier-Stokes equations at high Reynolds Numbers are most challenging since they combine finite speeds of propagation manifested in the advection terms (when they dominate), infinite speeds manifested in the diffusion terms and equilibrium processes enforced by the incompressibility condition which results in an elliptic equation for the pressure to be solved at every time step of the evolution process.

It is not surprising to reveal that the physical properties described above have a decisive impact on the numerical algorithms and in particular on the magnitude of the time step required for stability of "explicit algorithms". Such algorithms are most easily programmed and parallelized as they express the values of variables at a later time as a combination of known values computed at certain previous time levels. For space approximations of a low order precision (e.g. finite-difference schemes) the domain of dependence at each location is confined to a small local neighbourhood. Accordingly, only a small number of neighbouring degrees of freedom are involved in computation of local quantities. All explicit methods, however, have the disadvantage of severely restricted time steps at high resolution. Therefore, for dissipative evolution equations explicit time-marching schemes are considered inefficient.

Implicit schemes do not have, in general, time stepping restrictions. but they couple all points at the new time level. This global coupling seems to imply that a parabolic evolution problem requires the solution of a difficult elliptic problem at each time step of the evolution process as the price for avoiding severe stability restrictions.

Another factor causing global dependence (i.e. the requirement of global communication between processors) appears with the use of high-order functional methods. Such methods which employ polynomial or trigonometric basis for discretization must involve the information from the whole computational region. Thus, a massive data transfer is required to compute any local quantity.

The trigonometric Fourier basis has advantages when compared to the polinomial ones. For example, the differential operators are represented in this basis by diagonal matrices which do not couple harmonics with different wave numbers. In [7] this basis was employed for the solution of CFD problems in regions, decomposed into parallel strips. Fourier transform in the periodic direction (along strips) decouples the initial 2-D problem into a set of 1-D problems, which are solved separately by using a Chebychev spectral technique. The straigtforward implementation of a spectral Fourier method for the solution of the elemental problems in the subdomains would give low accuracy. The reason is that Fourier partial sums converge very slowly for the non-periodic functions (Gibbs phenomenon), resulting from the domain decomposition.

In our earlier works [1, 2] a modification of the Fourier method, using a Local Fourier Basis (LFB) technique, was developed for the solution of PDE's in multidomains regions. Such a technique allow us to obtain spectrally accurate solution of non-periodic problems. At the same time, it possesses all usual advantages of the conventional spectral Fourier method. The LFB technique consists in performing a projection procedure, introduced in [3], in order to localize functions in a smooth manner within each subdomain, and the ordinary pseudospectral Fourier method on a local period. The continuity of a global solution is accomplished by a matching procedure, similar to the influence matrix method, but making use of analytical homogeneous solutions (the boundary Green's functions). Thus relations for the unknown matching parameters can be derived as functions of the jumps on the interfaces. In effect, instead of *overall coupling* of collocation points in all subdomains, inherent to the spectral Fourier method, only *interface points* are involved in the matching relations. However, the communication still remains global because all interfaces contribute to the matching parameters in any particular subdomain.

In the current paper we continue the flow of our previous results [1, 2] to decouple matching relations between interfaces, introduced by the implicitness of the time discretization. This is achieved by utilizing the local behaviour of the boundary Green's functions. As a result, the communication is reduced to local interactions only between neighbouring subdomains. We also generalize the matching procedure in [1, 2], which is valid for uniform domain decomposition, onto the case of the arbitrary domain decomposition in order to have variable resolution at different locations in the physical space. This leads us to the possibility of treating unstructered grids.

The structure of the paper is the follows. In section 2 we formulate a problem and outline the mathematical idea, underlying the present local approach. In section 3 we describe the spectral multidomain technique with the Local Fourier Basis and a general matching procedure with the global coupling of the interface unknowns. In section 4 we introduce a Parabolic Domain Decomposition method and demonstrate its accuracy and stability for a one-dimensional time-dependent problem. Solution of two-dimensional parabolic problems with domain decomposition in the form of parallel strips or rectangular cells is given in section

5. In section 6 we show, how the present approach can be applied to elliptic problems to minimize the communication between the processors.

2 Localization Propertis of the Helmholtz operator. Preliminary Analysis

In this section we analyse the localization properties of elliptic equations, stemming from the temporal discretization of evolution problems with the diffusive linear operator. We also outline the multidomain technique and show how to exploit the properties of locality in order to reduce the communications between subdomains.

We demonstrate our approach for the simple case of the heat equation in d dimension:

$$\frac{\partial U}{\partial t} = \nu \nabla^2 U, \qquad U = U(\vec{x}),\ \vec{x} \in \Omega \subset R^d$$

$$U(\vec{x},0) = U_0(\vec{x})$$

$$U(\vec{x}_B,t) = U_B(\vec{x}_B,t), \qquad \vec{x}_B \in \partial\Omega. \tag{2.1}$$

The algorithm starts by semi discretization in time of (2.1):

$$\frac{U_{n+1} - U_n}{\Delta t} = \theta \nabla^2 U_{n+1} + (1-\theta)\nabla^2 U_n. \tag{2.2}$$

This is an implicit scheme for $U_{n+1}(x)$ and which does not have a stability restriction on the time step Δt if $\theta \geq \frac{1}{2}$.

Setting $\lambda^2 = 1/(\theta\Delta t)$ we get an elliptic equation of the Helmholtz type:

$$\nabla^2 U_{n+1} - \lambda^2 U_{n+1} = \frac{\theta - 1}{\theta}\nabla^2 U_n - \lambda^2 U_n \tag{2.3}$$

Eq. (2.3) can be simplified by defining:

$$V_n = \theta U_{n+1} + (1-\theta)U_n \tag{2.4}$$

to get

$$\nabla^2 V_n - \lambda^2 V_n = -\lambda^2 U_n. \tag{2.5}$$

With any spatial discretization of (2.5) we can solve for V_n in terms of the known U_n. Then we can solve for U_{n+1} in terms of V_n from equation (2.4). The process is repeated for each time step.

We note, however, that the solution of the elliptic equation (2.5) at any location is influenced by the boundary conditions and also by any change in the RHS. The region of dependence in this case is *global*, that is it spreads all over the domain. It follows that the parabolic equation, that we started with, was replaced by a sequence of global elliptic equations at every time step. Thus, the possible locality of the parabolic equation, discussed in the introduction, seems to have been lost.

For the more detailed analysis that follows we will assume $d = 1$ (one-dimensional case). In this case (2.5) takes the form

$$\frac{d^2V}{dx^2} - \lambda^2 V = f(x), \quad 0 \leq x \leq x_B \tag{2.6}$$

The general solution of this equation can be written as

$$V(x) = V_p(x) + AV_h^+(x) + BV_h^-(x)$$

where $V_p(x)$ is a particular solution satisfying (2.6) and $V_h^\pm$ are the homogeneous solutions (the boundary Green's functions) of (2.6) of the form $e^{\pm\lambda x}$.

Consider now the interval $I = \{x; 0 < x_L \leq x \leq x_R < x_B\}$ and assume

$$V(x_L) = V_p(x_L) \quad V(x_R) = V_p(x_R)$$

Then $A = B = 0$ (which can be easily proven in the present case) and $V(x) = V_p(x)$ for $x \in I$.

Suppose now that we modify $f(x)$ in the region outside the interval I so that $V(x_R)$ changes by δ, then the solution in I will be changed to:

$$V(x) = V_p(x) + \delta e^{-\lambda|x-x_R|} + 0(e^{-\lambda|x_L-x_R|}).$$

It follows that for large λ changes in the boundary conditions have local effect. The smaller the λ is, the larger the region of the boundary influence. The algorithm for parallelizing the solution of eq. (2.6) proceeds by dividing the region Ω into pieces, allocating each piece to a processor and solving in parallel for V_p separately in each processor.

The matching of solutions at the interfaces can be achieved by a proper combination of homogeneous solutions. Exact solutions can be obtained by solving a small number of global equations and communicating the coefficients. This method is particularly appropriate for message-passing distributed memory multiprocessors. For large λ the coefficients can be found locally and are required only at neighbouring processors. For smaller λ more and more processors are coupled by the matching process.

Detailed description of the one-dimensional problem (2.6) is given in section 3. In section 4 we substantiate the local behaviour of the boundary Green's functions for the elliptic operators of the Helmholtz type. In section 5 the domain decomposition technique with the locally interconnected subdomains is extended onto two dimensions.

3 Domain Decomposition Technique in One-Dimension

In this section we describe domain decomposition algorithm incorporating the exact solution of the interface problem by a global matching procedure. Our approach is demonstarted on the Helmholtz equation

$$\frac{d^2U}{dx^2} - \lambda^2 U = f(x), \quad x \in [0, L], \tag{3.1}$$

which is in "canonical" form for semi-implicit time discretization of time-dependent fluid dynamics problems. The boundary conditions may be one of the following types:

$$
\begin{array}{lll}
U(0) = U_0, & U(L) = U_L & \text{Dirichlet} \\
\frac{dU(0)}{dx} = D_0, & \frac{dU(L)}{dx} = D_L & \text{Neumann} \\
U(0) = U(L) & & \text{periodic}
\end{array}
\tag{3.2}
$$

Let the interval L be devided into P pieces (subdomains) of an arbitrary size l_n, $n = 1, 2, ..., P$ (see Fig.1). The original problem (3.1), (3.2) can be represented as a collection of subproblems:

$$
U = \bigcup_{n=1}^{P} U^{(n)}
$$

$$
\frac{d^2U^{(n)}}{dx^2} - \lambda^2 U^{(n)} = f^{(n)}, \qquad x \in [\bar{x}_{n-1}, \bar{x}_n] \tag{3.3}
$$

The boundary conditions on the interfaces $\bar{x}_n$ impose the matching of the adjacent local solutions. The matching conditions for the second order elliptic equation (3.1) are the continuity of the solution and its first derivative:

$$
x = \bar{x}_n \qquad U^{(n)} = U^{(n+1)}, \quad \frac{dU^{(n)}}{dx} = \frac{dU^{(n+1)}}{dx}, \qquad n = 1, ..., P-1, \tag{3.4}
$$

The numerical solution of the multidomain problem (3.3), (3.4) consists of two steps:

Elemental particular solutions - Construction of the elemental solutions (3.3), (3.4) independently in each subdomain $n = 1, ...P$, using a spectral method with the Local Fourier Basis. Because we do not know explicitly the boundary conditions on the interfaces $\bar{x}_n$, $n = 1, 2, ..., P-1$, they are chosen arbitrary at this stage (section 3.1).

Matching step - Matching of the elemental solutions $U^{(n)}$ on the interfaces in order to satisfy the continuity conditions (3.4). The matching procedure makes use of a linear combination of the boundary Green's functions (section 3.2).

The following remarks are appropriate:

- The algorithm outlined above is similar to the influence matrix technique, which is a useful alternative in the domain decomposition approaches (e.g. see [7, 4]). However, an important feature of the present method is that it takes advantage of the localization properties of the boundary Green's functions, analized in previous section, in order to simplfy the matching relations. Eventually, it yields a great reduction in the communication among the processors.
- The matching procedure, which is the key element in our domain decomposition strategy, is *scheme independent*. In other words, any suitable (spectral or finite-difference) method can be employed for the spatial discretization in subdomains instead of currently used the Local Fourier Basis method,

3.1 Solution of Local Subproblems

We construct the elemental particular solutions in each subdomain with any arbitrary boundary conditions on the interfaces. The spatial discretization is performed by using the trigonometric Fourier basis. To avoid Gibbs oscillations, which appear when a non-periodic function is approximated by truncated Fourier series, a preliminary smoothing of the elemental source functions $f^{(n)}(x)$ is carried out in order to bring them to a periodic continuous form. We also need an efficient algorithm which enables us to reconstruct the solutions of the original problems with the primary (not smoothed) source function.

We adopt a smoothing procedure, developed in [3] for the construction of an orthogonal Local Fourier Basis (LFB). The LFB technique is very efficient for the analysis of non-periodic functions since the truncated LFB series converge exponentially fast. However, the straightforward implementation of such a technique for the solution of non-periodic differential equations seems complicated because the differential operators in the transformed LFB space are not local, i.e. the corresponding matrices are not diagonal as they are in the Fourier space.

Nevertheless, we will show that the LFB method can be modified to be efficient and valuable for the solution of non-periodic problems as well. The idea is to combine both the LFB and the classical Fourier techniques in a way such that after tranforming it into the LFB space, the integration and the inverse transform are completed by means of the classical spectral Fourier method. This can be done by using appropriately a constructed bell (window) to perform a smooth projection of a function. In the following, we detail the modified LFB technique.

Let $[\bar{a}, \bar{b}]$ be the range of a subdomain and $[a_1, b_1] \supset [\bar{a}, \bar{b}]$. The bell function $B(x)$ is defined as follows:

$$B(x) = \begin{cases} 0, & x < a_1, x > b_1 \\ s(x), & x \in [a_1, \bar{a}) \\ 1, & x \in [\bar{a}, \bar{b}] \\ c(x), & x \in (\bar{b}, b_1] \end{cases} \tag{3.1.1}$$

where

$$s(x) = sin\theta(x), c(x) = \cos\theta(x);\ s^2(x) + c^2(x) = 1, \quad c(x) = s(-x)$$
$$\theta(x) = \int_{-\infty}^{x} \psi(\xi) d\xi, \quad \theta(x) + \theta(-x) = \frac{\pi}{2};$$
$$\int_{-\infty}^{+\infty} \psi(x)dx = \frac{\pi}{2}, \quad \sup \psi \subset (-\epsilon, \epsilon), \quad \psi(x) = \psi(-x)$$
$$2\epsilon = b_1 - \bar{b} = \bar{a} - a_1 \tag{3.1.2}$$

This function is equal to $B = 1$ inside the subdomain and decays smoothly outwords over the distance 2ϵ (see Fig.2). Some specific forms of $\theta(x)$ were tested in [1].

The smoothing procedure is defined by the following formulas:

$$f_\pm \equiv B \cdot f = \mathcal{F}_a^\pm \mathcal{F}_b^\pm f(x),$$

where

$$\mathcal{F}_\xi^\pm f(x) = B(x)f(x) \pm B(2\xi - x)f(2\xi - x) \tag{3.1.3}$$

(any of the signs "+" or "-" may be chosen, see below). It consists of two independent smoothing operations $\mathcal{F}_a^{\pm}$ and $\mathcal{F}_b^{\pm}$, each one looks like a "folding" of the function $f(x)$ across the lines a and b respectively (we use the values of the function on the interval 2ϵ to furnish the function $f_{\pm}$ with the required properties on a half-interval ϵ).

It can be readily verified, that a "smoothed" function $f_{\pm}$ is supported on the interval (a,b), coincides with the original function f on $x \in [\bar{a}, \bar{b}]$, continuous elsewhere inside (a,b), including two points of conjunction $x = \bar{a}, \bar{b}$, and on the boundaries:

$$x = a, b \qquad \begin{array}{l} f_- = f_-'' = ... = f_-^{(2n)} = ... = 0, \\ f_+' = f_+''' = ... = f_+^{(2n+1)} = ... = 0 \end{array}$$

as in Fig.3.

The smoothing is accomplished by symmetric $(+)$ / antisymmetric $(-)$ extension across the point $x = b$ (or $x = a$) to obtain a smooth periodic function

$$\tilde{f}_{\pm} = \left\{ \begin{array}{lll} & f_{\pm}(x) & x \in [a,b] \\ \pm & f_{\pm}(2b-x) & x \in [b, 2b-a] \end{array} \right. \tag{3.1.4}$$

This function can be represented by exponentially convergent trigonometric series

$$\tilde{f}_+ = \sum_{k=0}^{\infty} \hat{f}_k^+ cos\frac{k\pi x}{l} \qquad \text{or} \qquad \tilde{f}_- = \sum_{k=1}^{\infty} \hat{f}_k^- sin\frac{k\pi x}{l} \tag{3.1.5}$$

where $l = b - a$. There is no preference in the choice of the sign regarding the accuracy and the amount of computation. Below we shall omit the subscripts " $\pm$ ", having in mind that both instances are equally good.

Now we can construct the particular solutions $U_p^{(n)}$ of the elemental problems (3.3) independently in each subdomain. The numerical algorithm consists of several steps.

1. Continuous extension of the local pieces $f^{(n)}(x)$ of the source function beyond the interfaces
$$[\bar{x}_{n-1}, \bar{x}] \longrightarrow [\bar{x}_{n-1} - 2\epsilon_n, \bar{x} + 2\epsilon_n]$$
by communicating the neighbouring subdomains - see Fig.4.

2. Projection of the local functions $f^{(n)}(x)$ by applying the procedure (3.1.1)-(3.1.4) on the extended intervals $[\bar{x}_{n-1} - 2\epsilon_n, \bar{x} + 2\epsilon_n]$. This step results in periodic continuous functions $\tilde{f}^{(n)}$.

3. Solution of the elemental subproblems (3.3) with the smoothed source functions $\tilde{f}^{(n)}$ in the RHS by means of the pseudospectral technique. It consists of the integration in the spectral space and two Fast Fourier Transforms (FFT) from the physical to the

spectral space and v.v. (in our case we can use either Fast Cosinus Transform or Fast Sinus Transform like in (3.1.5)):

$$\begin{array}{lcl} \tilde{f}^{(n)}(x) & \xrightarrow{\text{FFT}} & \hat{f}_k^{(n)}, \\ \hat{U}_k^{(n)} & = & -\ \hat{f}_k^{(n)}/(k^2+\lambda^2), \\ \hat{U}_k^{(n)} & \xrightarrow{\text{FFT}} & \tilde{U}^{(n)}(x) \end{array} \tag{3.1.6}$$

The smoothing procedure does not change the primary source function *inside* the subdomains $[\bar{x}_{n-1}, \bar{x}]$ but only destorts the extra pieces on the *external* intervals $[\bar{x}_{n-1}-\epsilon_n, \bar{x}_{n-1}]$, $[\bar{x}_n, \bar{x}+\epsilon_n]$. Therefore, within each subdomain the solution $\tilde{U}^{(n)}(x)$ of the "smoothed" problem obviously coincides with the particular solution $U_p^{(n)}(x)$ of the original problem (3.3). Hence, our approach does not require any "unfolding" operations to recover the original non-periodic solution. The extra work, which have to be done to replace the solution of the non-periodic local problems in subdomains by the solution of the smooth periodic problems using the spectral Fourier method, results in:

- Overlapping of adjacent subdomains over the diastance $2\epsilon_n$ by means of local communications between neighbouring processors.

- Performing the FFT on a larger period $(\bar{x}_{n-1}-\epsilon_n, \bar{x}_n+\epsilon_n)$ instead of $(\bar{x}_{n-1}, \bar{x})$.

- Extra operations (3.1.3) for smoothing the local functions.

It was shown in [1] that the only factor which affects the precision of the smoothing procedure is the number of collocation points n_ϵ on the range of smoothing 2ϵ (it follows from the reasons of scaling). Consequently, the relative amount of overhead due to to overlapping decreases as n_ϵ/n (here n is the number of collocation points in a subdomain) so that it turns to be insignificant for large enough problems.

3.2 Global (exact) Matching Procedure

The particular local solutions $U_p^{(n)}$, constructed *independently* in each subdomain $n = 1, ..., P$, do not necessarily match on the interfaces $\bar{x}_n$. Therefore, the next computational step is to correct those solutions in order to fulfil the continuity conditions (3.4). This can be accomplished by combining the particular solutions with the linearly independent homogeneous solutions $H_\pm^{(n)}$ (boundary Green's functions) in the following way:

$$U = \bigcup_{n=1}^{P} U^{(n)}, \quad U^{(n)} = U_p^{(n)} + A_n H_+^{(n)} + B_n H_-^{(n)} \tag{3.2.1}$$

where $H_\pm$ satisfy equations (3.3) without the source function $f^{(n)}$ in the RHS:

$$\frac{d^2 H_\pm^{(n)}}{dx^2} - \lambda^2 H_\pm^{(n)} = 0 \qquad x \in [\bar{x}_{n-1}, \bar{x}_n] \qquad n = 1, 2, ..., P \tag{3.2.2}$$

The unknown coefficients A_n, B_n have to be determined to enforce the conditions (3.4) on the interfaces.

When λ is a constant number (for example, the case of the temporal discretization of the Navier-Stokes equations for incompressible fluids) the independent solutions of (3.2.2) can be found immediately:

$$H_+^{(n)} = e^{\lambda(x-l_n)}, \quad H_-^{(n)} = e^{-\lambda x}, \qquad l_n = \bar{x}_n - \bar{x}_{n-1} \tag{3.2.3}$$

Note that the analytical form (3.2.3) of the boundary Green's functions is highly general since it is generated by the *diffusive* linear operator in (2.1) and the *implicit* form of the temporal discretization (2.2). Besides, it does not depend on a specific form of nonlinearity and forcing terms in the evolution equation or particular form of an implicit time-marching algorithm. Moreover, the present matching technique can be generalized in a straightforward manner for the operators of the *advection-diffusion* type- see [1]. It can be shown, in a similar way as in section 2, that the corresponding boundary Green's functions also exhibit the local behaviour so that the present local approach is applicable to a wide class of problems with the convection-diffusion type operators as well.

Combining (3.4), (3.2.1) and (3.2.3), we obtain an algebraic system of recurrance relations:

$$A_n = A_{n+1}E_{n+1} - \alpha_n, \qquad B_{n+1} = B_n E_n + \beta_n \qquad n = 1, ..., P-1 \tag{3.2.4}$$

where

$$E_n \equiv e^{-\lambda l_n}$$

$$\alpha_n = \frac{1}{2}\left(\delta_n + \frac{\delta_n'}{\lambda}\right), \qquad \beta_n = \frac{1}{2}\left(\delta_n - \frac{\delta_n'}{\lambda}\right)$$

$$\delta_n = U_p^{(n)}(l_n) - U_p^{(n+1)}(0), \qquad \delta_n' = \frac{dU_p^{(n)}(l_n)}{dx} - \frac{dU_p^{(n+1)}(0)}{dx} \tag{3.2.5}$$

The two extreme subdomains $n = 1, P$ are connected via the global boundary conditions (3.2). For the Dirichlet or Neumann type boundary condition we have

$$A_1 E_1 + B_1 = \alpha_0, \qquad A_P + B_P E_P = -\beta_P,$$

where

$$\begin{aligned} &\alpha_0 = U_0 - U_p^{(1)}(0), && \beta_P = U_p^{(P)}(l_P) - U_L && \text{Dirichlet} \\ &\alpha_0 = \tfrac{1}{\lambda}\left(D_0 - \tfrac{dU_p^{(1)}(0)}{dx}\right), && \beta_P = \tfrac{1}{\lambda}\left(\tfrac{dU_p^{(P)}(l)}{dx} - D_L\right) && \text{Neumann} \end{aligned} \tag{3.2.6}$$

The system (3.2.4)-(3.2.6) can be resolved explicitly with respect to one of coefficients, say B_1:

$$B_1 = \frac{1}{1-\omega_P^2}\sum_{n=0}^{P-1}\left(\alpha_n\omega_n + \beta_{P-n}\frac{\omega_P^2}{\omega_{P-n}}\right) \qquad \text{Dirichlet/Neumann}$$

where

$$\omega_0 = 1; \qquad \omega_n = E_1\ E_2\ ...E_n, \quad n \geq 1 \tag{3.2.7}$$

Then from the recurrance relations (3.2.4) we derive all the other coefficients in terms of the jumps δ_n and δ'_n on the interfaces.

In the case of periodic boundary condition the two extreme subdomains are connected in the following way:

$$A_1 E_1 - A_P = \alpha_0, \qquad B_1 - B_P E_P = \beta_P$$

where

$$\alpha_0 = \frac{1}{2}\left(\delta_0 + \frac{\delta'_0}{\lambda}\right), \qquad \beta_P = \frac{1}{2}\left(\delta_P - \frac{\delta'_P}{\lambda}\right),$$

$$\begin{aligned} \delta_0 = \delta_P &= U_p^{(P)}(l_P) - U_p^{(1)}(0) \\ \delta'_0 = \delta'_P &= \tfrac{dU_p^{(P)}(l_P)}{dx} - \tfrac{dU_p^{(1)}(0)}{dx} \end{aligned} \tag{3.2.8}$$

Solving the system (3.2.4), (3.2.7), we get the following expressions for the coefficients B_1 and A_P:

$$\begin{aligned} B_1 &= \frac{\omega_P}{1-\omega_P}\sum_{n=1}^{P}\frac{\beta_n}{\omega_n} \\ A_P &= -\frac{1}{1-\omega_P}\sum_{n=0}^{P-1}\alpha_n\omega_n \end{aligned} \qquad \text{periodic} \tag{3.2.9}$$

where ω_n are defined in (3.2.7) (in this case both sets of coefficients A_n and B_n become mutually uncoupled).

The computation of matching parameters A_n, B_n in each subdomain, using the relations (3.2.7) or (3.2.9), involves the data from all the interfaces. Thus, the matching of local particular solutions requires the global communication between all subdomains (processors). In the next section we will utilize the localization properties of the homogeneous solutions $H_{\pm}^n(x)$ to simplify the interface problem (3.2.4)-(3.2.7). In effect, only the local communication between the contiguous subdomains appears to be necessary.

4 Parabolic Domain Decomposition (local matching)

Since the problem (3.1) arises from an implicit time-discretization of (2.1), the parameter $\lambda = O(1/\Delta t)$ must be large for Δt small enough. Then the following estimations are valid:

$$\begin{gathered} l = \min_{1\le n\le P} l_n, \qquad E = e^{-\lambda l} \\ E_n \ \le E \ \ll \ 1 \\ \omega_n = E_1\ E_2\ ...E_n \ < \ E^n, \qquad n = 1, ...P \\ \omega_1 \ \ll \omega_2 \ \ll ... \ \ll \omega_P \end{gathered} \tag{4.1}$$

Consequently, we can retain in (3.2.4), (3.2.9) only the low power of the factor E, i.e. the terms ω_n with the low number n.

Then the matching relations take the form:

Dirichlet/Neumann boundarycondition

$$
\begin{array}{llcccc}
B_1 \approx \alpha_0 + \alpha_1 e^{-\lambda l_1} & & & [1] & \longleftarrow & [2] \\
B_n \approx \beta_{n-1} & [n-1] & \longrightarrow & [n] & & \\
A_n \approx -\alpha_n & & & [n] & \longleftarrow & [n+1] \\
A_P \approx -(\beta_P + \beta_{P-1} e^{-\lambda l_P}) & [P-1] & \longrightarrow & [P] & &
\end{array}
\tag{4.2}
$$

periodic boundarycondition

$$
\begin{array}{llcccc}
B_1 \approx \beta_P & [P] & \longrightarrow & [1] & & \\
B_n \approx \beta_{n-1} & [n-1] & \longrightarrow & [n] & & \\
A_n \approx -\alpha_n & & & [n] & \longleftarrow & [n+1] \\
A_P \approx -\alpha_0 & & & [P] & \longleftarrow & [1]
\end{array}
\tag{4.2}
$$

where parameters α_n and β_n are defined in (3.2.5), (3.2.6) and (3.2.8). The communication required for the computation of corresponding matching parameters is described symbolically on the right side of (4.2), (4.3).

The implication of this fact is that the interaction between subdomains become local, i.e. only adjacent subdomains are connected. Thus, only the minimal communication is employed, consistent with the physical (local) nature of the problem (see section 2). This is the key idea of the Parabolic Domain Decomposition approach.

We illustrate this conclusion for the heat-transfer equation

$$
\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + f(x), \qquad x \in [0, 2\pi] \tag{4.2}
$$

with the sinusoidal forcing term, corresponding to the exact solution $U_{ex}(x,t) = (1 - e^{-t})\sin x$. Table 1 gives the maximum error $\varepsilon = |U - U_{ex}|_{max}$ at the asymptotic stage $t \gg 1$ for several numbers of processors P and different time steps Δt; subscripts $_G$ and $_L$ denote global and local matching respectively. In space we use pseudo-spectral discretization with the Local Fourier Basis (see section 3). The number of collocation points in each subdomain $N/P = 32$. In the case $P = 4$ the local matching procedure with small enough

P	Δt	0.005	0.01	0.05	0.1	0.5
4	ε_G	1.9 (-9)	9.1 (-10)	1.9 (-10)	9.6 (-11)	2.3 (-11)
	ε_L	1.9 (-9)	9.1 (-10)	4.3 (-6)	1.9 (-4)	2.8 (-2)
8	ε_G	3.1 (-9)	1.5 (-9)	3.0 (-10)	1.5 (-10)	3.7 (-11)
	ε_L	5.9 (-9)	1.1 (-6)	2.7 (-3)	1.3 (-2)	8.2 (-2)

Table 1: Accuracy of an asymptotic solution of (4.2) for the global (ε_G) and local (ε_L) matching procedure on the interfaces.

$\Delta t \leq 0.01$ ensure practically the same accuracy, as do the global matching. For larger $P = 8$ a smaller $\Delta t < 0.005$ is required to equate the errors ε_L and ε_G. Thus the criterion of "locality" depends not only on Δt, but also on the number of processors due to the factor $e^{-\lambda l}$, $l = 2\pi/P$ in the matching relations (3.5).

The implicit time-discretization, used in (2.2), does not have time-stepping stability restrictions since the global matching duplicates the global solution of the problem. The local approach confines the domain of dependence arguments to the region of the nearest processors. This imposes certain restrictions on the time step in accordance with the CFL theory of stability. Those restrictions, however, are not severe as they are related not to the grid spacing, but to the size of the subdomain itself.

Fig.5 and Fig.6 show the evolution of profile $U(x,t)$ respectively for the stable time step $\Delta t = 0.01 < \Delta t^* = 0.02$ and unstable one $\Delta t = 0.1 > \Delta t^*$; $P = 8$, $N/P = 32$; the parameter of time-discretization scheme (2.2) $\theta = 0.6$. The initial distribution at $t = 0$ contains many harmonics to provoke the instability.

In the case $\Delta t = 0.1$ the jumps across the interfaces are indicative of poor precision of the local matching for the relatively large time step. After the profile approaches the maximum attitude at $t \leq 20$ it is not changed significantly for a long time, while disturbances, localized near the interfaces, grow slowly. The developed stage of instability is shown on Fig.7a (t=200) and Fig.7b (t=250).

We may conclude, that for a time step small enough to provide a reasonable accuracy for the local matching procedure, the numerical algorithm is stable. If one takes into account, that semi-implicit time discretization of non-linear CFD problems with large Reynolds numbers, which are of practical interest, also imposes restrictions on the time step of order $\Delta t = O(1/N)$, then the local approach does not introduce any additional limitations.

5 Parabolic Domain Decomposition for Two-Dimensional Time-Dependent Problems

We discuss the numerical algorithm for the solution of the two-dimensional Helmholtz equation

$$\nabla^2 U - \lambda^2 U = f(x,y), \qquad (x,y) \in \Omega = (0, L_x) \times (0, L_y); \tag{5.1}$$

The computational region is decomposed into parallel strips $\Omega = \bigcup_{n=1}^{P} \omega_n,\ \omega_n = (\bar{x}_{n-1}, \bar{x}_n) \times (0, L_y)$, where $\bar{x}_n = nl,\ n = 1, 2, ..., P$ (case S)- see Fig.8 or rectangular cells $\Omega = \bigcup_{n=1}^{P_x} \bigcup_{m=1}^{P_y} \omega_{mn}$ of the size $l_x \times l_y$; $l_x = L_x/P_x,\ l_y = L_y/P_y$ (case C)- see Fig.9.

Case S: We use pseudo-spectral Fourier approximation and apply the FFT along the strips. Then the problem is reduced to a set of one-dimensional problems for the Fourier coefficients $\hat{U}_k^{(n)}(x)$ of type (3.1):

$$\frac{d^2 \hat{U}_k^{(n)}}{dx^2} - (k^2 + \lambda^2) \hat{U}_k^{(n)} = \hat{f}_k^{(n)}(x) \qquad \text{in } \omega_n, \qquad k = -\frac{N_y}{2}, ..., \frac{N_y}{2}, \tag{5.2}$$

(N_y is the number of collocation points in the y direction), which can be solved by using a routine with the local matching on the interfaces, described in sections 3,4.

Case C: The direct pointwise matching by using the corresponding two-dimensional homogeneous solutions of the form $e^{(\lambda_1 x + \lambda_2 y)}$, $\lambda_1^2 + \lambda_2^2 = \lambda^2$ would involve all interface points and results in $O(P_x N_x + P_y N_y)$ matching relations (where $P = P_x \times P_y$ is the number of cells, $N = N_x \times N_y$ is the total number of collocation points). It is clear, that such approach is able to cope only with a few number of subdomains [4].

The application of the LFB technique to the present C-type multidomain problem enables the drastic simplification of the matching relations. In effect, the solution of a two-dimensional problem with the overall coupling of interface unknowns is reduced, like in the previous case S, to repeated implementation of the one-dimensional routine with the simplest matching relations for only few interface points at a time.

The idea is as follows. Instead of *simulteneous* matching at all collocations points we rather match *successively* in x and y directions. In the first step we take a horizontal row of cells $\Omega_\nu = \bigcup_{m=1}^{P_x} \omega_{\nu m}, \nu = 1, ..., P_y$ and transform the source function $f(x, y)$ into the LFB in the y-direction (the procedure includes an ϵ-extention, smooting across the boundaries $\bar{y} = (\nu - 1)l_y - \epsilon$ and $\bar{y} = \nu l_y + \epsilon$, antisymmetric reflection and FFT on the local period $\tau = 2(l_y + 2\epsilon)$- see section 3).

In the second step we solve the system of one-dimensional problems of the type (5.2) for the Fourier coefficients with the local matching on the vertical intefaces, after which we transform back into the physical space. The resulting particular solution $U^{(I)}(x, y)$ will be smooth in each horizontal band exept for the interfaces $\bar{y}_n = nl_y,\ n = 1, .., P_x$.

To remove discontinuities at the horizontal interfaces we use again our one-dimensional matching routine (third step). We take a vertical column $\Omega^\mu = \bigcup_{n=1}^{P_y} \omega_{n\mu},\ \mu = 1, ..., P_x$, transform $U^{(I)}$ into the LFB (by analogy with the source function on the first step) and implement matching on the lines $\bar{y}_n$ for each Fourier coefficient. After inverse transform into the physical space we get a vertically smooth solution $U^{(II)}$.

Smoothing in the y- direction induces discontinuities on the vertical lines $\bar{x}_n = nl_x,\ n = 1, ..., P_x$, because each Fourier coefficient is matched independently. Those jumps can be removed by another matching (in the x- direction). Each new "ironing" alternatively along x and y will result in a smoother solution.

The convergence of an alternate smoothing procedure with the local matching on the interfaces is illustrated on Fig.10 in the case $f(x, y) = k^2\ cos(k_x x)cos(k_y y),\ \ k^2 = k_x^2 + k_y^2$ for the parameters $P_x = P_y = 4,\ \ N_x = N_y = 128,\ \ k_x = k_y = 1$ at several values of λ. The iterations converge very fast and after 2-3 smoothing steps the maximum precision is achieved. We would like to emphasize the fact that the fast rate of convergence is ensured by the large value of λ in (5.1) for any particular form of the source function $f(x, y)$.

For evolutionary problem (5.1) the alternate smoothing procedure should be performed, naturally, at each time step. We illustrate the accuracy of the LFB local approach on two-dimensional heat equation

$$\frac{\partial U}{\partial t} = \Delta U + f(x, y)$$

with the sinusoidal forcing term, the parameters are the same as in the previous example. Table 2 gives the maximum numerical error for the local matching procedure (ε_L) compared to the exact global matching (ε_G); $P = 4 \times 4,\ N = 128 \times 128$. We can see, that for a time

Δt	0.01	0.05	0.1
ε_G	2.4 (-9)	1.4 (-8)	8.3 (-7)
ε_L	2.6 (-9)	4.1 (-6)	1.6 (-3)

Table 2: Numerical error for the local and the global matching procedure in the case C (cells)

step smaller than $\Delta t < 0.01$ the local matching gives practically the same error as the exact global procedure.

6 Optimized Algorithm for Elliptic Problems

The solutions of elliptic equations which describe equilibrium processes are determined, in principle, at each location by the combined effect of all nonhomogeneous forcing terms and boundary conditions. One may conclude, therefore, that these solutions must be obtained by global procedures requiring repeated global communications.

However, as explained above (section 4), certain elliptic operators have fast decaying Green's functions which can be used to localize the computations. Unfortunately, this is not the case for the Poisson equation:

$$\Delta U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x, y), \qquad (x, y) \in \Omega \tag{6.1}$$

which must be solved for the pressure at each time step when a splitting algorithm is employed for the solution of the Navier-Stokes equations (e.g. see [5]).

The homogeneous solutions of (6.1) decay slowly as a function of the distance, so that the contribution of all remote interfaces seems to be necessary for the computation of the matching parameters in each location. Nevertheless, even in this "tough" case, a substantial reduction can be achieved in the magnitude of the inter-processor communication while preserving good accuracy.

We presume, that the computational region: $\Omega = 0 \leq x \leq L_x, 0 \leq y \leq L_y$, is decomposed into parallel strips $\Omega = \bigcup_{n=1}^{P} \omega_n$, $\omega_n = (\bar{x}_{n-1}, \bar{x}_n) \times (0, L_y)$, where $\bar{x}_n = nl$, $n = 1, 2, ..., P$ and the boundary conditions are periodic in both directions. Fourier transforming along the strips results in:

$$\frac{d^2 \hat{U}_k^{(n)}}{dx^2} - k^2 \hat{U}_k^{(n)} = \hat{f}_k^{(n)}(x), \quad \text{in } \omega_n \quad k = -\frac{N_y}{2}, ..., \frac{N_y}{2}, \tag{6.2}$$

where N_y is the number of collocation points in the y direction.

For $k \neq 0$, the homogeneous solutions of this equation decay exponentially as function of x. The decay becomes faster as k increases. Thus, a global matching procedure is required only for the long waves $k \leq k_*$, whereas the short waves $k \geq k^*$ can be treated by using local matching on the interfaces. For the intermediate wavenumbers $k_* < k < k^*$ the number of connected subdomains (processors) can be controlled in accordance with the prescribed accuracy.

In order to assess the potential of this approach we present the following numerical tests. We solve (6.2) with RHS corresponding to the solution $U_{ex} = cosky \sum_{m=1}^{M} cos(mx + \alpha_m)/m$. Table 3 gives the maximum relative error $\varepsilon_n = |U - U_{ex}|_{max}/U_{max}$ at $P = 8$, $N_l = N_x/P = 64$, $M = 64$ while phases α_m are chosen randomly; the subsrcipt n in ε_n is the number of neighbouring subdomains (from each side) contributing to the matching relations (3.5) and (3.6) and thus creating clusters of $2n + 1$ connected subdomains (see Fig. 11). For $n = P/2$ all the processors are connected so that the global information is available to each processor.

k	$\varepsilon_L = \varepsilon_1$	ε_2	ε_3	$\varepsilon_4 = \varepsilon_G$
1	5.5 (-2)	3.2 (-2)	2.2 (-2)	3.5 (-12)
2	1.6 (-2)	3.9 (-3)	1.0 (-3)	1.4 (-12)
3	5.4 (-3)	5.4 (-4)	5.8 (-5)	9.1 (-13)
4	1.8 (-3)	7.8 (-5)	3.6 (-6)	6.8 (-13)
5	5.7 (-4)	1.1 (-5)	2.3 (-7)	5.4 (-13)
6	1.8 (-4)	1.7 (-6)	1.5 (-8)	4.6 (-13)
7	6.0 (-5)	2.4 (-7)	1.0 (-9)	4.0 (-13)
8	1.9 (-5)	3.6 (-8)	6.8 (-11)	3.5 (-13)
9	6.3 (-6)	5.4 (-9)	4.6 (-12)	3.2 (-13)
10	2.1 (-6)	8.1 (-10)	3.2 (-13)	2.9 (-13)

Table 3: Dependence of numerical error on the number of connected processors for the Poisson's equation

If the cut-off condition is, for example, $\varepsilon \leq 10^{-5}$, then global matching is needed only for $k = 1, 2, 3$; for $k = 4, 5$ we can restrict the communication to $n = 3$ neighbouring processors; for $k = 6, 7, 8$ the communication can be reduced to $n = 2$ and, at last, for all $k \geq k^* = 9$ local communication ensures the desired accuracy. If the number of collocation points in the y direction is $N_y = N_x = 64 \times 8 = 512$ then the percentage $\eta = 2k^*/N_y \cdot 100\%$ of the equations (6.2), which are treated locally, amounts to more than 96% (here we take into account that each harmonic k is represented by one couple of Fourier coefficients). The results in Table 3 are appropriate for the case $P = 8$ and depend slightly on the particular form of the source function $f(x)$ in (6.1).

The decomposition into "local" and "global' waves depends on the number of subdomains P, which determines the local length $l = L_x/P$, and thus, affects the factor of locality $\mathcal{L} = e^{-kL_x/P}$ (see section 4). We now estimate the dependence of η on the number of processors P for a given accuracy ε. The processors are assumed to have a fixed memory size, so that the total number of collocation points in each subdomain $N_s = (N_x/P) \times N_y$ is independent of P. We define γ by $N_y/N_x = P^\gamma$. Then the following relations hold:

$$N_s = \frac{N_x}{P} \cdot P^\gamma N_x = N_x^2 P^{\gamma-1},$$

$$N_x \propto P^{\frac{1-\gamma}{2}}, \quad N_y \propto P^{\frac{1+\gamma}{2}}.$$

In order to keep the parameter of locality $\mathcal{L}$ constant (to ensure the prescribed accuracy ε), the cut-off wavenumber k^* must be proportional to P, so that

$$\eta \propto k^*/N_y \propto P/P^{\frac{1+\gamma}{2}} = P^{\frac{1-\gamma}{2}}.$$

For $\gamma = 0$ we have $N_x = N_y \propto \sqrt{P}$ and $\eta \propto \sqrt{P}$ (e.g., if the number of processors $P = 128$, then, according to Table 3, $\eta = 4\% \times \sqrt{128/8} = 16\%$). Another limiting case, $\gamma = 1$, corresponds to constant N_x and $N_y \propto P$ (the resolution is changed only in the direction of strips, the case of a long channel, for example.) In this case η is independent of P, i.e. algorithm is fully scalable.

We would like to emphasize the fact that even in the case when global communication is required, a careful choice of method may results in a small total communication.

7 Conclusions

A spectral domain decomposition method with Local Fourier Basis is used for the solution of parabolic and elliptic PDE's with linear operator of the diffusive type. The domain decomposition is considered in the form of parallel strips or rectangular cells. The matching of the local subsolutions is performed by using boundary Green's functions. The method exploits the localization properties of the Green's functions to simplify substantially the matching relations between different domains.

For parabolic problems the communication may be reduced to interactions only between neighbouring subdomains. For elliptic problems a flexible parallel strategy is employed to reduce the amount of communications between the processors. As a result, global communication is used only for the first few modes while the higher harmonics can be treated either locally or by using a minimum number of connections dictated by the required accuracy. The relative amount of global communications η is scalable (independent of the number of processors P) provided that the resolution is changed only in one direction (along strips) and is fixed in the other direction. If the resolution is the same in both directions, then η grows as $\sqrt{P}$. For relatively small $P \sim 10 - 10^2$ the number of modes, treated globally, is a small fraction of the whole number of modes.

When the present algorithm is employed for the solution of the Navier-Stokes system along with an appropriate splitting method for time discretization, the time-dependent equations for the velocity components can be solved by using only local communication between neighbouring subdomains, whereas the computation of the pressure can be performed by using the optimized algorithm with minimal number of connected subdomains. The data to be transfered globally are the vectors of the interface unknowns which constitute a very small percentage of the total data.

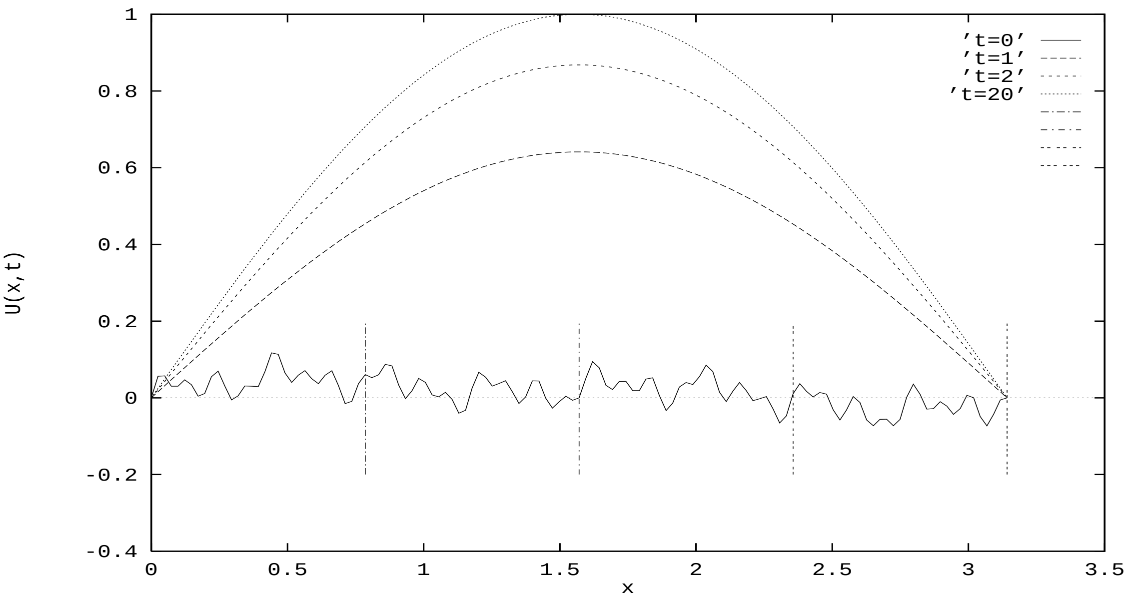

Figure 1: Stable evolution with $\Delta t = 0.01$

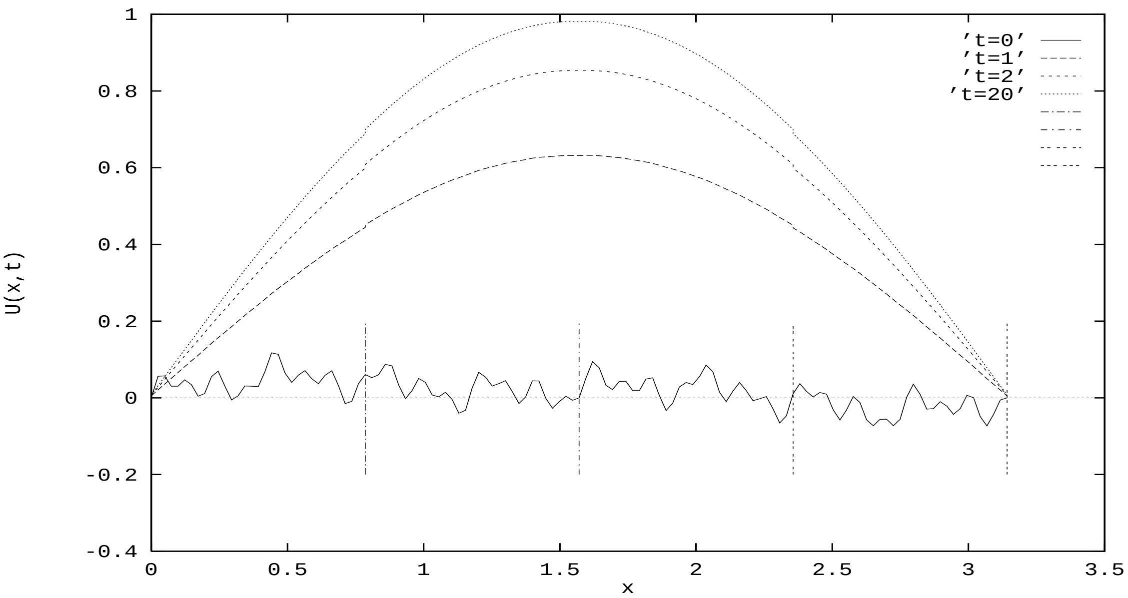

Figure 2: Unstable evolution with $\Delta t = 0.1$

References

[1] A, Averbuch, M. Israeli, L. Vozovoi, Spectral multi-domain technique with Local Fourier Basis, Journal of Scientific Computing, Vol. 8, No. 2, pp. 135-149, June 1993.

[2] L. Vozovoi, M. Israeli, A. Averbuch, Spectral multi-domain technique with Local Fourier Basis II: decomposition into cells, Journal of Scientific Computing, Vol. 9, No. 3, pp. 311-326, 1994.

[3] R. Coifman, Y. Meyer - Remarques sur l'analyse de Fourier à fenêtre, série, *C.R.Acad.Sci.* Paris **312** (1991), p. 259-261.

[4] D.Gottlib, R.S. Hirsh - Parallel pseudospectral domain decomposition

[5] G.E. Karniadakis, M. Israeli, S. Orszag - High-order splitting methods for the incompressible Navier-Stokes equations, *J.Comp.Phys.*, **97**, No. 2, (1991), p. 414-443

[6] M.G.Macaraeg, C.L. Street - Improvements in spectral collocation discretization through a multidomain technique, *Appl. Numeric. Mathem.*

[7] R. Peyret - The Chebyshev multidomain approach to stiff problems in fluid mechanics, *Comp. Meth. in Appl. Mech. and Engin.* **80**, (1990) p. 129-145

Development of the instability at t=200 (Fig.4a) and t=250 (Fig.4b)

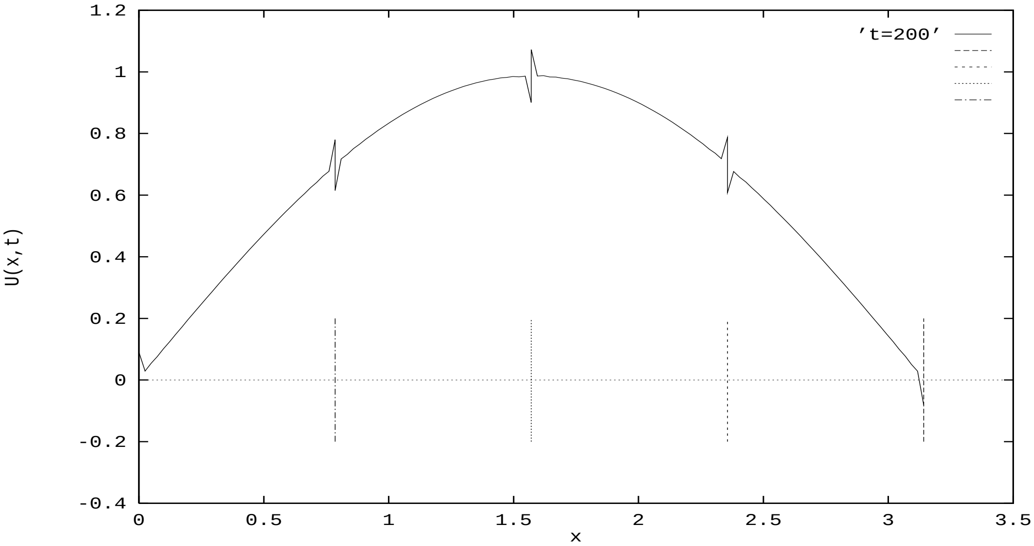

Fig.4a

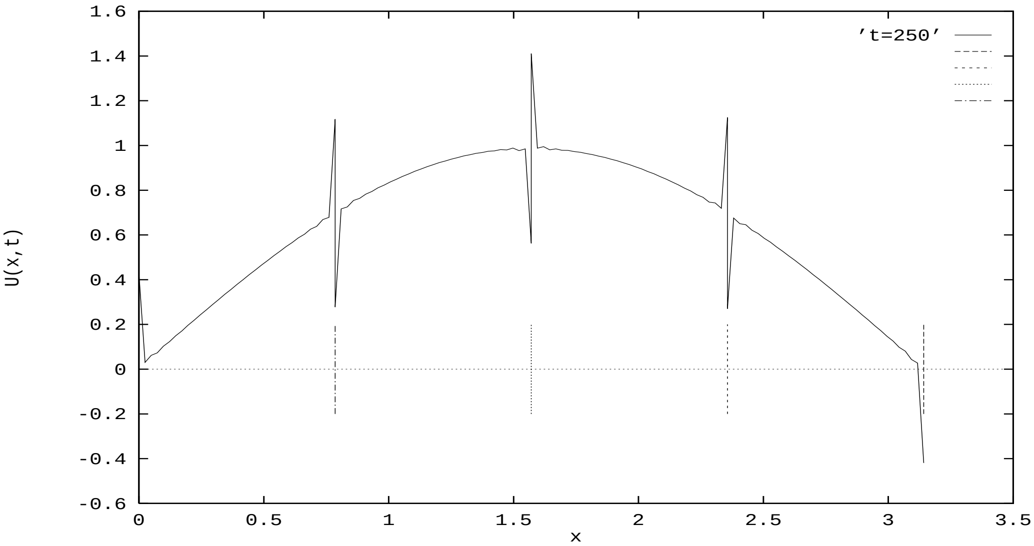

Fig.4b