Multidomain Local Fourier Method for PDEs in Complex Geometries *

appeared in J. Computational and Applied Mathematics, Vol. 66, pp. 543-5 55, 1996

L. Vozovoi†, M. Israeli†, A. Averbuch‡
† Faculty of Computer Science, Technion, Haifa 32000, Israel
‡ School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel

Abstract

A low communication parallel algorithm is developed for the solution of time-dependent non-linear PDEs. The parallelization is achieved by domain decomposition. The discretization in time is performed via a third order semi-implicit stiffly stable scheme. The elemental solutions in the subdomains are constructed using a high-order method with the Local Fourier Basis (LFB).

The continuity of the global solution is accomplished by a point-wise matching of the local subsolutions on the interfaces. The matching relations are derived in terms of the jumps on the interfaces. The LFB method enables splitting a 2-D problem with a global coupling of the interface unknowns into a set of uncoupled 1-D differential equations. Localization properties of an elliptic operator, resulting from the discretization in time of a time-dependent problem, are utilized in order to simplify the matching relations. In effect, only local (neighbor-to-neighbor) communication between the processors becomes necessary.

The present method allows the treatment of problems in various complex geometries by the mapping of curvilinear domains into simpler (rectangular or circular) regions with subsequent matching of local solutions. The operator with non-constant coefficients, obtained in the transformed domain, is preconditioned by an appropriate constant coefficient operator, easily inverted by the LFB. The problem is then solved with spectral accuracy by (a rapidly convergent) conjugate gradient iteration.

1 Introduction

Within the last two decades, domain decomposition techniques have become an efficient tool for parallelizing algorithms, particularly due to its ability to provide an equal loading of the computational work into different processors. The last feature is critical to avoid synchronization bottlenecks and thus to achieve high parallel efficiency of a parallel algorithm.

*This research is supported partly by a grant from the French-Israeli Binational Foundation for 1993-1994 and by grant of Israel Ministry of Science for 1992-1993

Domain decomposition can be successfully combined with spectral methods using series expansions into orthogonal bases. The advantage of spectral methods over low-order (finite-difference or finite-element) methods is that they converge exponentially fast with the number of degrees of freedom N. For problems in multidomains, polynomial (like Chebyshev or Legendre) bases are mostly used. Spectral methods with the Fourier basis seem to be of doubtful use for the construction of *non-periodic* local solutions in subdomains. The reason is that the truncated Fourier series of a non-periodic function (having a discontinuous periodic extension at the boundaries) converges very slowly, like $1/N$, inside the region, and gives rise to $O(1)$ spurious oscillations near the boundaries, known as Gibbs phenomenon.

However, the Gibbs phenomenon can be avoided if a smooth decomposition of functions is performed. In [1],a spectrally accurate multidomain Fourier method was developed. This method employs a particular version of the projection technique introduced by R. Coifman and Y. Meyer [4] for the construction of the Local Fourier Basis (LFB). The multidomain local Fourier method (MDLF) of [1] is especially efficient for large problems using high resolutions in space. In this case, the relative amount of operations required to perform the projection is insignificant in comparison with that needed to execute the Fast Fourier Transform (FFT).

An important advantage of this method, when compared to other multidomain spectral techniques, is that it enables a great simplification of the matching relations for the interface unknowns. The matching of local solutions constructed independently in each subdomain is necessary to obtain the continuous global solution. For problems with constant coefficients and in simple domains, the use of the Fourier basis enables us to fulfill the matching of each harmonic separately, and thus to eliminate the global coupling of the interface unknowns.

The matching relations can be simplified furthermore by utilizing localization properties of an elliptic operator resulting from the discretization in time of a time-dependent problem [2]. Eventually, only *local communication* between the neighboring subdomains turns out to be necessary. Thus, the MDLF algorithm is completely scalable and posseses high parallel efficiency [3].

In this paper we generalize our previous multidomain Fourier approach in order to treat problems with non-constant coefficients and complicated geometries. The new algorithm incorporates the MDLF method along with several additional techniques, such as mapping and conjugate gradient iterations. A rapidly convergent (by $O(1)$ iterations) algorithm, using a spectral preconditioner, is constructed for the solution of the full-matrix problem in a mapped domain. In effect, the spectrally accurate solution in complex geometry can be accomplished by a little more work than that required to solve a constant coefficient equation in the simplest rectangular domain using the FFT.

The structure of this paper is the following. In section 2, we describe the MDLF approach in 1-D and in 2-D for the simplest rectangular geometry. In section 3, we set up the problem in complex geometry. The numerical algorithm for solving this problem is introduced in section 4. Finally, in section 5 we test this method on some elliptic and parabolic problems in channels and in confined regions of complicated shape.

2 Multidomain Local Fourier Method (MDLF)

In this section we discuss briefly the MDLF method in 1-D and its extension to 2-D for the simplest rectangular domains decomposed into parallel strips. The detailed analysis of this method can be found in [1], [2], [3].

We describe the numerical algorithm as applied to the boundary value problem of the diffusion type

$$u'' - \lambda^2 u = f(x), \qquad x \in [0, L], \tag{2.1}$$

with periodic boundary conditions

$$u(0) = u(L), \qquad u'(0) = u'(L). \tag{2.2}$$

We divide the computational interval $[0, L]$ into P pieces (subdomains) of an arbitrary size and discretize the local subproblems on the uniform grid of collocation points.

Following [4], we cover the interval with the collection of smooth overlapping bell functions $b_n(x)$, which satisfy certain properties to allow the projection in L_2. In our particular version of the LFB method the bell functions $b_n = 1$ inside the subdomains $l_n = [x_{n-1},\ x_n]$ and smoothly decay outwards over the intervals 2ϵ. This choice of $b_n(x)$ is particularly convenient for the solution of differential equations using the LFB technique. The algorithm consists of two steps.

1.**Construction of the local solutions**. For each subdomain we perform:

• Extension of a local source function $f^{(n)}(x)$ beyond the interfaces over the distance 2ϵ (from each side). It can be done by exchanging data with two neighboring subdomains. For periodic problems, Eq.(2.2), the continuous extension beyond the boundaries, $x < 0$ and $x > L$, can be obtained by overlapping the first and the last subdomains.

• Projection of $f^{(n)}(x)$ with the help of the bell functions b_n. The projection procedure results in a smooth function $\tilde{f}^{(n)}(x)$, localized on the extended interval $\bar{l}_n = [\bar{x}_{n-1} - \epsilon, \bar{x}_n + \epsilon]$ which can be represented by a rapidly convergent Fourier series. Due to the choice of b_n, this function coincides with the original one, $f^{(n)}(x)$, on the range of a subdomain l_n.

• Integration of Eq.(2.1) with the smooth source function using the Discrete Fast Fourier Transform (DFFT) on the extended interval $\bar{l}_n$.

The resulting solutions $\tilde{u}^{(n)}(x)$ will be the true particular solutions $u_p^{(n)}(x)$ to Eq.(2.1) on the intervals l_n, where the projection procedure does not change the source function $f_n(x)$. As for the "extra" intervals outside subdomains $[\bar{x}_{n-1} - \epsilon, \bar{x}_{n-1}]$ and $[\bar{x}_n, \bar{x}_n + \epsilon]$, where the source function was distorted by projection, the most economic way is not to reconstruct solutions on these intervals but rather not to utilize them (it will be extra work for avoiding the Gibbs phenomenon when the Fourier method is applied to non-periodic problems). It can be shown [2] that for large problems using high resolution in space this extra work is a small percentage of the whole computational work.

The global solution $u_p = \bigcup_{n=1}^{P} u_p^{(n)}$ will be piecewise continuous since the local usually solutions have jumps at the interfaces.

2. **Matching step**. In order to obtain the continuous global solution, we correct the particular solutions, constructed at the previous step, by properly weighted local homogeneous

solutions of Eq.(2.1):

$$u = \bigcup_{n=1}^{P} u^{(n)}, \quad u^{(n)} = u_p^{(n)} + A_n h_+^{(n)} + B_n h_-^{(n)}. \tag{2.3}$$

These solutions are two exponential functions $h_-^{(n)} = e^{-\lambda x}$, $h_+^{(n)} = e^{\lambda(x-l_n)}$, $0 < x < l_n$, which decay inward in the subdomain. The coefficients A_n, B_n have to be found in order to enforce the continuity of $u(x)$ and $u'(x)$ at the interfaces. They can be expressed, in closed form, in terms of jumps of particular solutions u_p, u_p' at the interfaces.

It seems that computation of matching coefficients requires the global communication between the processors since all the interfaces contribute to the matching relations. We observe, however, that if the parameter λ in Eq.(2.1) is much greater than 1 (see section 5.2), then the functions $h_\pm(x)$ decay rapidly away from the interfaces. Thus, for each particular location x, the influence of the remote interfaces becomes negligible. As a result, the matching relations decouple in such a way that the computation of each pair of matching parameters A_n, B_n requires only the knowledge of jumps at two nearest interfaces as follows:

$$A_n = -\alpha_n, \qquad B_n = -\beta_{n-1}, \qquad n = 1, 2, ..., P$$

where

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

$$\delta_n = u_p^{(n)}(l) - u_p^{(n+1)}(0), \qquad \delta_n' = \frac{d}{dx}u_p^{(n)}(l) - \frac{d}{dx}u_p^{(n+1)}(0).$$

To conclude, the computation of particular local solutions and the matching procedure requires only local, neighbor-to-neighbor, inter-processor communication that makes the whole algorithm completely scalable.

The present algorithm can be extended to solve non-periodic problems. For such problems the continuous extension of functions beyond the boundaries is not available. Therefore, the boundary subdomains should be treated differently. An alternative approach consists in re-expanding the Fourier partial sums, burdened by the Gibbs phenomenon, into the rapidly converging Gegenbauer series [7].

The extension of this method to rectangular domains decomposed into parallel strips, is straightforward. After applying the DFFT along the strips, we arrive at a collection of *uncoupled* 1-D ODEs for the Fourier coefficients. These equations are of the same type as Eq.(2.1). Therefore, they can be solved by using the 1-D MDLF routine.

3 Problems in Complex Geometries

Consider a second-order elliptic equation

$$\nabla^2 u - \lambda^2 u = F(x, y) \tag{3.1}$$

in a region $\Omega = \{0 \le x \le L,\ y_1(x) \le y \le y_2(x)\}$. Here $y_1(x)$, $y_2(x)$ are smooth L- periodic functions of x, having the mean values $\bar{y}_1 = 0$, $\bar{y}_2 = 1$ on the period L. The boundary conditions are:

$$y = y_{1,2}, \qquad u = \phi_{1,2}(x),$$
$$u(0,y) = u(L,y). \tag{3.2}$$

The curvilinear region Ω can be mapped onto the rectangle $\omega = \{0 \le \xi \le L,\ 0 \le \eta \le 1\}$ by means of a simple stretching transformation:

$$\xi = x, \qquad \eta = \frac{y - y_1(x)}{y_2(x) - y_1(x)}. \tag{3.3}$$

The complication of using a coordinate transformation appears in the coefficients of the differential operator in the transformed domain ω. After defining

$$\alpha = y_1, \quad \beta = y_2 - y_1, \qquad \eta_y = 1/\beta, \quad \eta_x = -(\alpha' + \eta\beta')/\beta,$$
$$\eta_{xx} = -(\alpha'' + \beta'' + 2\eta_x \beta')/\beta, \qquad J(\xi,\eta) = \eta_x^2 + \eta_y^2$$
$$a = 2J^{-1}\eta_x, \quad b = J^{-1}, \quad c = J^{-1}\eta_{xx}, \quad d = -J^{-1}\lambda^2, \quad f = J^{-1}F, \tag{3.4}$$

(here subscripts "x" and "y" denote the derivatives with respect to corresponding variables, whereas $'$ is the derivative of a function of one variable) we can rewrite Eq.(3.1) as follows:

$$\mathcal{L}u = f \qquad \text{in } \omega \tag{3.5}$$

where

$$\mathcal{L}u = \frac{\partial^2 u}{\partial \eta^2} + a(\xi,\eta)\frac{\partial^2 u}{\partial \xi \partial \eta} + b(\xi,\eta)\frac{\partial^2 u}{\partial \xi^2} + c(\xi,\eta)\frac{\partial u}{\partial \eta} + d(\xi,\eta)u. \tag{3.5.1}$$

The boundary conditions (3.2) take the form:

$$\eta = 0, 1 \qquad u = \phi_{1,2}(\xi),$$
$$u(0,\eta) = u(L,\eta). \tag{3.6}$$

Similar consideration is applicable to confined regions. Consider e.g. an annular region $\Omega = \{r_1(\theta) \le r \le r_2(\theta),\ 0 \le \theta \le 2\pi\}$ with smooth boundaries $r_1(\theta)$, $r_2(\theta)$, where r, θ are polar coordinates. We introduce a stretching transformation:
$\Omega \to \omega = \{0 \le \xi \le 2\pi,\ 0 \le \eta \le 1\}$, similar to Eq.(3.3), and apply it to Eq.(3.1) with Laplacian $\nabla^2_{r,\theta}$ written in polar coordinates. In the transformed domain, Eq.(3.1) takes the form of Eq.(3.5) with non-constant coefficients resulting from a coordinate transformation and also due to the Laplacian form in polar coordinates, e.g.

$$a = \frac{2\eta_\theta}{r^2}J^{-1}, \qquad J(\xi,\eta) = \eta_r^2 + \frac{\eta_\theta^2}{r^2}, \quad r = r_1(\xi) + \eta\left(r_2(\xi) - r_1(\xi)\right),$$

and so forth.

Spectral approximations to the boundary value problem, Eqs. (3.5), (3.6), lead to full $N \times N$ matrix equations for the N expansion coefficients. The solution of this algebraic

system requires $O(N^2)$ storage locations and $O(N^3)$ operations to invert the matrix (using the Gauss elimination or other direct methods). Obviously, for multi-dimensional problems of realistic size (typically, with $N \sim 10^6$), the direct solution of such a system is prohibitively expensive.

The straightforward implementation of iteration methods to the solution of large linear algebraic systems is also inefficient. The reason is that the typical number of iterations required to reduce the error by an order of magnitude (it can be characterized by the *condition number* κ) grows with the size N of a system. For the second order operator (3.5), $\kappa \propto N^2$ no matter whatever discretization method is used.

The convergence of iterations can be improved substantially, if an appropriate *preconditioner* to the original differential operator is used [6]. In the next section we construct an efficient *spectral* preconditioner to $\mathcal{L}$ to obtain a rapidly convergent iteration algorithm.

4 Preconditioned Iteration Method with Spectral Preconditioner

The idea of preconditioning is to solve the equation

$$H^{-1}\mathcal{L}u = H^{-1}f \tag{4.1}$$

rather than Eq. (3.5). The preconditioned iterations (for the particularly simple Richardson scheme) look as follows:

$$Hu^{n+1} = f - (\mathcal{L} - H)u^n. \tag{4.2}$$

Both Eqs (4.1) and (3.5) are equivalent in view of the solution but the iteration method may converge much faster for Eq.(4.1) if the preconditioner H approximates $\mathcal{L}$ in the sense that the condition number of the "compound" operator $H^{-1}\mathcal{L}$ be bounded (independent of N).

Starting with the work by S. Orszag [5], a low order (finite-difference or finite-element) approximation to $\mathcal{L}$ is commonly used in the capacity of H. The use of a higher order finite-difference preconditioner improves the rate of convergence but is partially off set by the additional cost of inverting a denser matrix (representing an operator, discretized on a larger stencil). Therefore, high-order finite-difference preconditioners are not useful.

We propose to construct H in the same form as $\mathcal{L}$ but with *constant* coefficients in place of variable coefficients in $\mathcal{L}$. For instance, the preconditioner to the operator Eq.(3.5) should be

$$Hu = \frac{\partial^2 u}{\partial \eta^2} + \bar{a}\frac{\partial^2 u}{\partial \xi \partial \eta} + \bar{b}\frac{\partial^2 u}{\partial \xi^2} + \bar{c}\frac{\partial u}{\partial \eta} + \bar{d}u. \tag{4.3}$$

The coefficients $\bar{a}, \bar{b}, \bar{c}$ and $\bar{d}$ can be obtained, for example, by the averaging of the corresponding non-constant coefficients in Eq.(5.5) over the region ω:

$$\bar{b} = \frac{1}{L}\int_{\omega} J^{-1}(\xi, \eta)d\xi d\eta. \tag{4.4}$$

The operator H, defined in a rectangular domain ω, is trivially inverted by the Fourier method.

If the coefficients in $\mathcal{L}$ vary in small ranges, then the *spectral* preconditioner Eq.(4.3) obviously provides a better approximation to $\mathcal{L}$ than any low-order preconditioner. However, if the coefficients vary substantially on the computational region, then Eq.(4.3) is going to be a poor approximation to $\mathcal{L}$.

The situation will change if the preconditioned iteration method is combined with the domain decomposition technique. Domain decomposition provides us with the new possibility of using *different* preconditoners in subdomains, each one being a good constant-coefficient approximation to $\mathcal{L}$ on a small enough patch of the region. As a result, the convergence rate of iterations can be improved considerably.

We illustrate this conclusion by the following 1-D example

$$u'' + a(x)u' - \lambda^2 u = f(x), \qquad x \in [0,1]$$

$$u(0) = u(1) = 0, \tag{4.5}$$

where $a(x) = 10\ (1 + \tan\ 20(x - 0.5))$. Function $a(x)$ changes rapidly from 0 to 20 on a small interval $\Delta x \sim 1/20$ near the point $x = 0.5$. The forcing function has the form $f(x) = 2 - a + (2a + \lambda^2)x - \lambda^2 x^2$ which corresponds to the exact solution $u_{ex}(x) = x(x-1)$.

Following our approach, we divide the computational interval $x \in [0,1]$ into a number of subintervals $[x_{n-1}, x_n],\ n = 1, ..., P$ and define a set of constant coefficient operators H_n as follows:

$$H_n = u'' + \bar{a}_n u' - \lambda^2 u, \qquad x \in [x_{n-1}, x_n], \tag{4.6}$$

where

$$\bar{a}_n = \frac{1}{x_n - x_{n-1}} \int_{x_{n-1}}^{x_n} a(x)dx.$$

Fig.1 shows the variance of the coefficient $a(x)$ on the interval $x \in [0,1]$ and its step-wise approximation $\bar{a}_n$ for $P = 15$.

The convergence history of the preconditioned conjugate gradient (PCG) iterations with the spectral preconditioners, Eq.(4.6), is given in Fig.2 for several domain decompositions (parameter $\lambda = 1$). When the number of subdomains varies from $P = 11$ to $P = 15$, the same asymptotic accuracy is eventually attained. However, the convergence is much *faster* for larger P, i.e., for smaller deviations $|a(x) - a_n|$ within the intervals $[x_{n-1}, x_n]$. If the deviations $|a(x) - a_n|$ are too large $(P = 9)$, then a *smaller* accuracy is achieved after an affordable (some tens) number of iterations.

5 2-D Examples

In this section we demonstrate the capability of the Multidomain Local Fourier method to solve elliptic and parabolic problems in 2-D domains with curvilinear boundaries.

5.1 Stationary Problems

Consider the solution of the Poisson equation

$$\Delta u = 1, \qquad \text{in } \Omega \tag{5.1}$$

in a channel-like domain $\Omega = \{0 \le x \le L,\ y_1(x) \le y \le y_2(x)\}$ with the boundaries $y_{1,2}(x)$ having the form of sinusoidal waves:

$$y_1(y) = -(1 + \delta cos\ kx), \qquad y_2(y) = 1 + \delta cos\ kx. \tag{5.2}$$

The following boundary conditions are applied:

$$y = y_{1,2}, \qquad u = 0,$$
$$u(0, y) = u(L, y). \tag{5.3}$$

This problem is solved by the generalized MDLF method. The stretching transformation Eq.(3.3) is used to transform Ω into the rectangular domain ω which is decomposed afterwards into parallel strips:

$$\omega = \bigcup_{n=1}^{P} \omega_n, \qquad \omega_n = \{0 \le \xi \le L,\ \bar{\eta}_{n-1} \le \eta \le \bar{\eta}_n\}. \tag{5.4}$$

In each subdomain ω_n a spectral preconditioner H_n is constructed in the form of Eq.(4.3) with constant coefficients, computed by the averaging of the corresponding variable coefficients in Eq.(3.5) within the strip ω_n.

The computed solution is plotted in Fig.3 for $\delta = 0.1, k_w = 3, L = 2\pi$. The convergence history of the PCG iterations in this case is shown in Fig.4 at several resolutions in the lengthwise direction, $N_x = 16, 32, 64$, and a fixed resolution in the transverse direction, $N_y = 64$. The number of subdomains $P = 2$. The "exact" solution (solution of reference) u_{ex} is computed on the fine grid 256×64. Evidently the rapid convergence of the spectral solution with N_x is preserved in this case.

Another example of complicated geometry is a channel with grooves. The boundaries are given in the form:

$$\begin{array}{ll} 0 \le x \le L/2, & y_1(x) = 1 - 0.5h(1 + \tan\ \alpha(x - x_0)), \\ L/2 < x \le L, & y_1(x) = y_1(L - x), \\ 0 < x < L & y_2 = 1. \end{array}$$

In Fig.5 we plot the solution for $x_0 = L/8,\ h = 0.3,\ \alpha = 6,\ L = 2\pi$. In Fig.5 we plot the convergence history at $N_y = 64$ and different N_x. The maximum pointwise error in the asymptotic solution is listed in Table 1; the solution of the reference is computed on the grid 512×64.

N_x	32	64	128	256
$\log \lvert u - u_{ex} \rvert_{max}$	-3.6	-5.2	-8.6	-9.4

Table 1. Errors in the solution of the Poisson equation (5.1) in a grooved channel, Fig.5 ($\alpha = 6$, $N_y = 64$, $P = 2$)

The impact of the profile of grooves (parameter α) on the accuracy is given in Table 2 (other geometric parameters remain as in Table 1). The accuracy decreases as the profile becomes steeper.

α	6	7	8	9
$\log \lvert u - u_{ex} \rvert_{max}$	-8.6	-7.8	-6.6	-5.7

Table 2. Dependence of accuracy on the steepness of the groove's profile ($N_x = 128$, $N_y = 64$, $P = 2$)

Unlike the previous case (a wavy channel with *smoothly* deformed boundaries) for a channel with *sharp* grooves the error does not decay exponentially fast at high resolution ($N_x = 256$). The use of larger P or a larger resolution within each subdomain does not improve the accuracy. The reason is that steep gradients in the profile of the groove, $y_1(x)$, generate large coefficients Eq.(3.4) in the transformed operator Eq.(3.5). Domain decomposition in the form of parallel strips eliminates the large variance of the coefficients only in the transverse direction. In the lengthwise direction the difference between the mean and the local values may be large. In such a case the error behaves like in Fig.2 for $n = 9$ (relatively low accuracy sustains after some tens iterations) or, most fortunately, like for $n = 11$ (slow convergence to the exact solution). This difficulty can be overcome by applying domain decomposition in the form of rectangular boxes. The results will be reported in a future paper.

5.2 Time-Dependent Problems

Consider the solution of the heat equation

$$\frac{\partial u}{\partial t} = \nabla^2 u - f(x,y), \qquad \text{in } \Omega \tag{5.5}$$

with Dirichlet boundary conditions

$$u(x,y) = \phi(x,y), \qquad \text{on } \partial\Omega \tag{5.6}$$

where $\Omega = \{r_1(\theta) \le r \le r_2(\theta),\ 0 \le \theta \le 2\pi\}$, r and θ are polar coordinates.

Time discretization of Eq.(5.5), using an implicit scheme, results in an elliptic equation of the Eq.(3.1) type. For example the simplest two-level Crank-Nikolson scheme reads:

$$\frac{u_{n+1} - u_n}{\tau} = \nabla^2 \left(\frac{u_{n+1} + u_n}{2} \right) - f(x,y) \tag{5.7}$$

where n refers to the times $t_n = n\tau$. After defining

$$\lambda^2 = 2/\tau, \qquad F_{n+1} = 2f - \nabla^2 u_n - \lambda^2 u_n \tag{5.8}$$

we have

$$\nabla^2_{r,\theta} u_{n+1} - \lambda^2 u_{n+1} = F_{n+1}(r,\theta) \tag{5.9}$$

where Laplacian $\nabla^2_{r,\theta}$ is written in polar coordinates.

This equation is solved repeatedly at each time step by the generalized MDLF algorithm as described in previous sections. The strip-wise decomposition Eq.(5.4) of the transformed

(rectangular) domain ω is carried out. For typical τ used in computations, $\lambda \gg 1$ that justifies the implementation of the local matching procedure for the Helmholtz type equation (5.9), was discussed in section 2.

As an example we consider the annular region plotted in Fig.7. The inner and outer boundaries are given, respectively, by

$$\begin{aligned} r_1(\theta) &= 0.3 + 0.1\ \sin\theta + 0.12\ \sin 2\theta, \\ r_2(\theta) &= 1 + 0.1\ \cos\theta + 0.1\ \sin 3\theta. \end{aligned} \tag{5.7}$$

The source function in Eq.(5.5) is $f(x,y) = e^x + e^y$ and the boundary values are $\phi(x,y) = e^x + e^y$, $(x,y) \in \partial\Omega$.

As $t \to \infty$, the solution $u(x,y,t)$ approaches the steady state regime $u_\infty(x,y) = e^x + e^y$, $(x,y) \in \Omega$. The maximum relative error in the steady solution is given in Table 3. It decays exponentially fast as the resolution in the azymuthal direction increases.

N_θ	16	32	64
$\log \frac{\lvert u-u_\infty \rvert_{max}}{\lvert u_\infty \rvert_{max}}$	-4.0	-8.1	-12.4

Table 3. Errors in steady state solution ($t \to \infty$) of Eqs. (5.5),(5.6) in the region Fig.7; $N_r = 64$, $P = 2$

However, the PCG iterations converge rather slowly in this case (at the resolution $N_\theta = 64$ about 45 iterations are needed to converge). Moreover, for more complicated geometries than the one in Fig.7 (e.g. annular regions described by Eq.(5.7) but with higher frequencies and amplitudes of the oscillating components), only a low accuracy can be achieved after an affordable number of PCG iterations.

The reason is that a strip-wise domain decomposition cannot handle large changes in non-constant coefficients in the lengthwise direction (see explanation at the end of section (5.1)). More general domain decompositions in the form of rectangular boxes will be able to treat efficiently any complicated geometries.

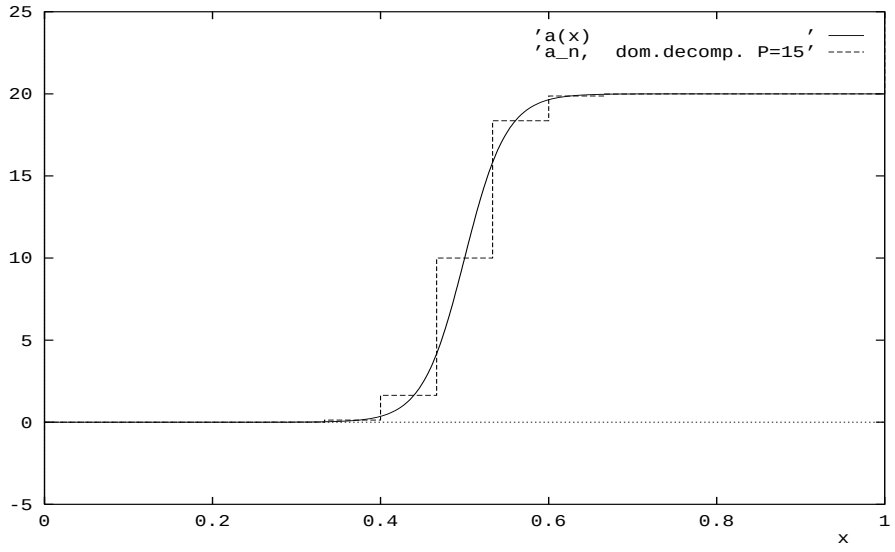

Figure 1: Step-wise approximation of the variable coefficient $a(x) = 10\ (1 + \tan\ 20(x - 0.5))$ in Eq.(4.5)

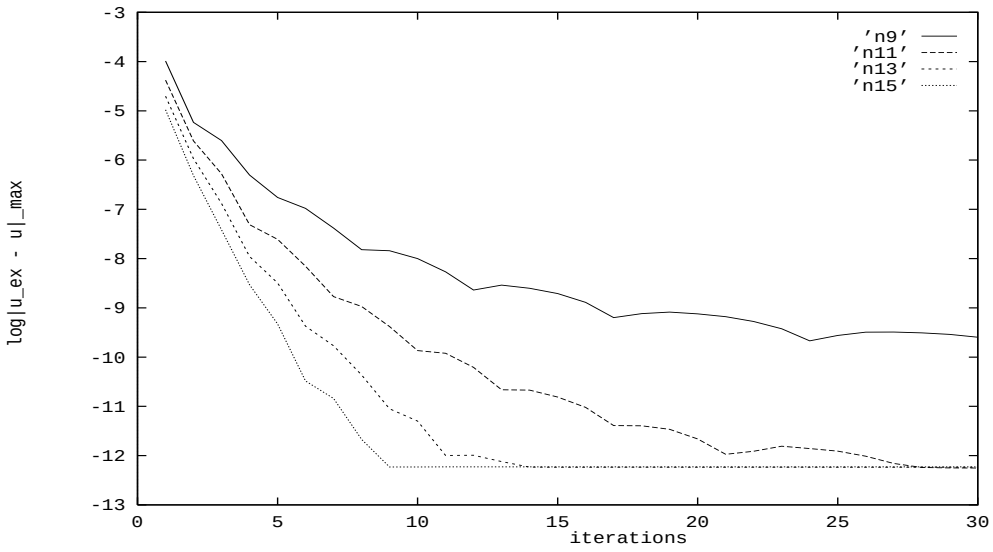

Figure 2: Accelerating convergence by splitting the computational region into an increasing number of subdomains P.

References

[1] M. Israeli, L. Vozovoi, A. Averbuch - Spectral multi-domain technique with Local Fourier Basis, *Journ. Scient. Comp.*, **8**, N 2, p. 135-149, (1993).

[2] M. Israeli, L. Vozovoi, A. Averbuch - Parallelizing implicit algorithm for time-dependent problems by parabolic domain decomposition, *Journ. Scient. Comp.*, **8**, N 2, p. 151-165, (1993).

[3] A. Averbuch, M. Israeli, L. Vozovoi - Parallel implementation of non-linear evolution problems using parabolic domain decomposition, to appear.

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

[5] S. A. Orszag - Spectral methods for problems in complex geometries. *Numerical methods for PDE's*, 1979.

[6] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang - Spectral methods in fluid dynamics, *Series in Computational Physics*, Springer-Verlag, (1989)

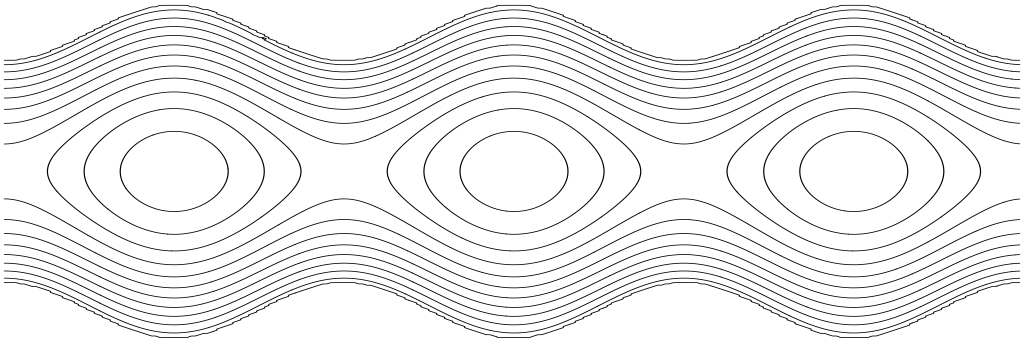

Figure 3: Solution of the Poisson equation (5.1) in a channel with wavy boundaries
. (lines of constant values $u = const$)

[7] L. Vozovoi, M. Israeli, A. Averbuch - Analysis and application of Fourier-Gegenbauer method to stiff differential equations, to appear in *SIAM, J.Numer.Anal.*

[8] 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

[9] L. Vozovoi - Convection in a vertical layer with undulating boundaries, *Fluid Dynamics*, **11**, N 2, (1976), p. 202-206

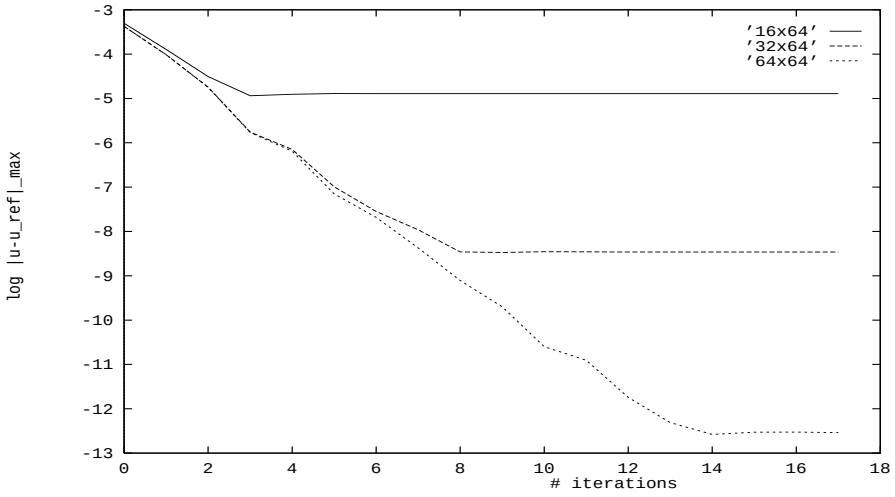

Figure 4: Convergence of the PCG iterations for the geometry in Fig.3

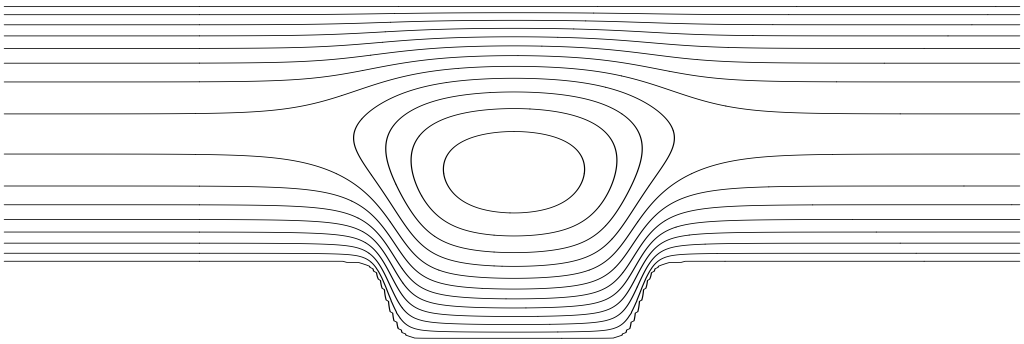

Figure 5: Computed solution of Eq.(5.1) in a channel with grooves

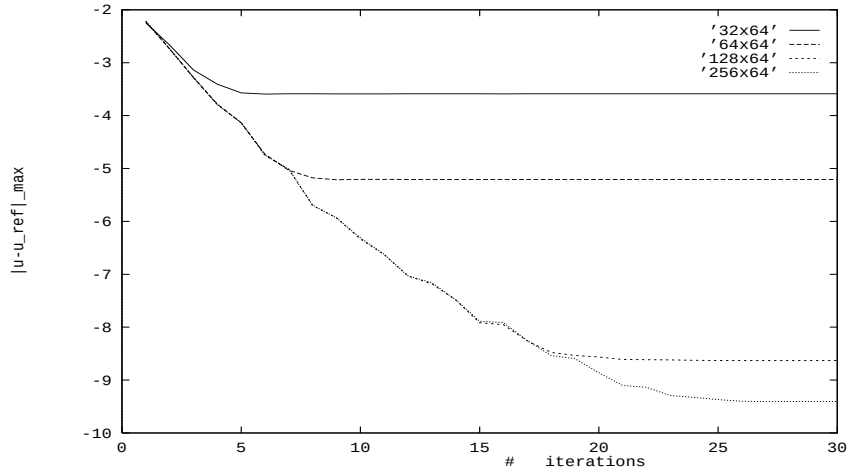

Figure 6: Convergence of the PCG iterations for the geometry in Fig.5

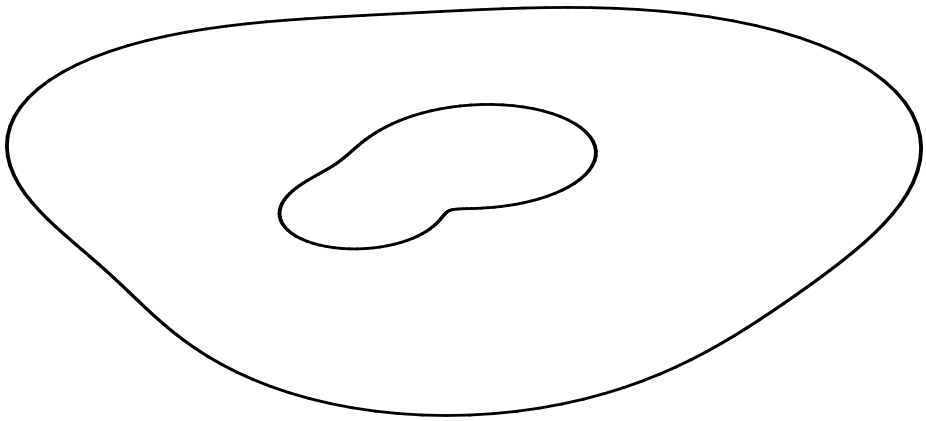

Figure 7: Annular region defined by Eq.(5.7)