# Application of the Multidomain Local Fourier Method for CFD in Complex Geometries *

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

**Abstract**

A low communication parallel algorithm is presented for the solution of nonlinear time-dependent PDEs. Our particular interest lies in the application of this algorithm to fluid dynamics problems in complex geometries. The parallelization is achieved by domain decomposition. A high-order method with the Local Fourier Basis (LFB) is employed to construct the elemental solutions in the subdomains. The discretization in time is performed via a third order semi-implicit stiffly stable scheme.

The continuity of the global solution is accomplished by using 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 transformation enables to split a 2-D problem with global coupling of the interface unknowns, into a set of uncoupled 1-D differential equations with local matching relations. Localization properties of an elliptic operator, resulting from the discretization in time of time-dependent problem, are utilized in order to simplify the matching relations. In effect, only local (neighbor-to-neighbor) communication between the processors is necessary.

The current 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 the non-constant coefficients, obtained in the transformed domain, is preconditioned by an appropriate constant coefficient operator, easily inverted by the LFB. Then, the problem is solved with spectral accuracy by (a rapidly converged) conjugate gradient iteration.

## 1 Introduction

Within the last two decades, high order methods, in particular, spectral methods, were matured and later became widely used in computational fluid dynamics and computational mechanics. These methods are based on global expansions of the dependent variables in terms of orthogonal (e.g. Chebyshev or Legendre) polynomials or trigonometric functions. The advantage of spectral methods, when compared to low-order finite difference or finite

*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

element schemes, is their fast convergence when the number of degrees of freedom increases. High order schemes are particularly attractive if either fine features of flows are required or a long term integration is performed [1].

Significant difficulties arise when spectral methods are applied to problems in complex geometries. In the last few years considerable progress has been made in this area. Complex geometries are being dealt with by combination of *domain decomposition techniques* and *mappings*. The idea of mapping is to transform the complex geometry into a simpler one (such as rectangular or disk) by a smooth transformation.

Domain decomposition facilitates the problem in complex geometry when the mapping of the full domain into a simple shape is difficult or introduces artificial singularities into the solution in the transformed space. The basic idea of this approach consists of splitting the computation domain into a number of simpler subdomains. The problem is solved independently in each subdomain by applying suitable continuity conditions across the artificial internal boundaries (interfaces). Domain decomposition not only overcomes difficulties resulting from complex geometries but also seem to be amenable to parallel computing.

Among the various spectral methods the Fourier method is the most efficient for the solution of linear differential equations with constant coefficients since the differential operators are represented in the Fourier basis by diagonal matrices, therefore, different Fourier harmonics remain uncoupled. The multi-dimensional Fourier transform is separable, that is it can be expressed as a sequence of one-dimensional transforms. This results in dramatic reduction of the computational complexity for problems of realistic size.

The Fourier method also allows the application Fast Fourier Transform (FFT) on a *uniform grid* of collocation points. For such a grid the CFL stability condition for the time step in an explicit scheme is $\Delta t \sim O(N^{-1})$ (where, $\Delta t$ is the time step and $N$ is the number of collocation points). This restriction is less severe than that which applies for the non-uniform collocation employed by Chebyshev method where $\Delta t \sim O(N^{-2})$ [2]. Another important advantage of a uniform collocation grid, especially in multidimensional problems, is that any numerical dispersion or dissipation error which may affect the shape and speed of propagation and form of wave fronts is much more homogeneous and isotropic. This implies much better representation for hyperbolic problems in medium and low resolutions.

The rate of convergence of any eigenfunction expansion depends on the smoothness of the function being expanded: the convergence is exponential for the smooth (infinitely differentiable) function, and it is algebraic otherwise. There is an additional constraint for the exponential convergence of the Fourier series which requires that the periodic extension of the function must be smooth. When approximating a *non-periodic* function the 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 efficient implementation of the Fourier method is restricted also to problems with constant coefficients. An operator containing *non-constant coefficients* is represented in the Fourier space by a full matrix. In such a case the Fourier method has no preference over the other spectral techniques.

It follows that the Fourier method seems to be not suitable for treating complex geometries in combination with domain decomposition techniques and mappings. Indeed, domain decomposition results in a collection of non-periodic problems to be solved in each sudomain, whereas the coordinate transformation generates non-constant coefficients in the transformed operator.

In [3]-[6] a domain decomposition spectral method was developed, using the Local Fourier Basis (LFB)[9]. It consists of a set of smooth windowed trigonometric functions, well localized both in physical and spectral domains, which form an orthonormal basis in $L_2$ ($R^N$ in the discrete case). The expansion of a smooth function in this basis converges exponentially fast. Besides, LFB possesses all advantages of the usual (global) Fourier basis such as fast and separable transformation from the physical to spectral domain and vice versa, equispaced discrete sets of collocation points and so on. Another advantage of this basis is that it allows a different resolution in each sub-domain, and can be used in an adaptive setting.

The Multidomain Local Fourier (MDLF) method of [3]-[6] was applied to the solution of elliptic (Poisson and Helmholtz) and dissipative evolution equations in one and two dimensions. The domain decomposition was implemented in the form of parallel strips or rectangular cells. The separability of the local Fourier transform results in great simplification of the local problems in subdomains. As a matter of fact, the solution of 2-D equations can be reduced to repeated solutions of 1-D equations for the Fourier coefficients with simple matching relations, which couple only few interface points at a time. Another simplification of the matching relations can be gained by using the localization properties of the boundary Green functions, employed for the matching of local subsolutions. In effect, only local (neighbor-to-neighbor) communication between the processors is utilized. The latter feature is critical to keeping the communication overheads within reasonable bounds in the case of massively parallel processor.

This paper presents a further development of the local Fourier approach. The MDLF method is extended to treat complex domains using an appropriate coordinate transformation. The domain decomposition is carried out for the *transformed* (rectangular) domain. The resulting full matrix problem is then solved with spectral accuracy by a preconditioned conjugate gradient (CG) iteration. An appropriate *constant coefficient* operator is employed as a preconditioner, amenable to efficient solution by the MDLF technique. Such a *spectral preconditioner* provides faster convergence of the CG iterations than any low-order finite-difference preconditioner. Thus, the solution of the problem in complex geometry can be accomplished by an additional little work which is required to solve the problem in the simplest geometries.

# 2 Multidomain Local Fourier Method (MDLF)

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 bases such as Chebyshev or Legendre are used frequently. For the construction of *non-periodic* local solutions in subdomains spectral methods with the Fourier basis seem to be of no use. 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 [3],a spectrally accurate multidomain Fourier method was developed. This method employs a particular version of the projection technique [9] which is the foundation for the construction of the Local Fourier Basis (LFB). The multidomain local Fourier method (MDLF)[3] 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 among 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 [5]. Eventually, only *local communication* between the neighboring subdomains turns out to be necessary.

Another advantage of the the MDLF method is that its decomposition strategy can easily be changed to reflect different resolutions in each subdomain, which makes it valuable as an adaptive algorithm[7]. Arbitrary domain decomposition is possible in order to have variable resolution at different locations in the physical space. This leads us to the possibility of treating unstructured grids. In any case, in uniform or nonuniform grid distribution, the communication is reduced to local interactions only between neighbouring subdomains.

Thus, the MDLF algorithm is completely scalable and possesses high parallel efficiency [7].

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 [3]-[7].

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 [9], 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 decay smoothly outwards over the intervals $2\epsilon$. 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 $2\epsilon$ (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 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 [5] 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 $\lambda$ in Eq. (2.1) is much greater than 1 (see section 7.2), 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 current 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 [8].

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 Extension of the Function Beyond its Boundaries

Assume

$$u'' + u' - \lambda^2 u = f(x), \qquad x \in [-1, 1], \tag{3.1}$$

with Dirichlet boundary conditions

$$u(-1) = c_1, \qquad u(1) = c_2.$$

Extension of $f(x)$ beyond the boundaries $x = \pm 1$ is performed by applying the finite Taylor series expansion where its derivatives are computed on a stencil with $I$ points. For the boundary $x = 1$ we compute first the derivatives by solving the linear system:

$$f(1 + j\Delta x) = \sum_{n=0}^{I} f^{(n)}(1)\frac{(j\Delta x)^n}{n!}, \qquad j = 1, ..., I. \tag{3.2}$$

Then, we construct an extrapolation of $f(x)$ in the form of a $K$-order Taylor expansion ($K \leq I$):

$$f(1 + l\Delta x) = \sum_{n=0}^{K} f^{(n)}(1)\frac{(l\Delta x)^n}{n!}, \qquad l = 1, ..., N_\epsilon. \tag{3.3}$$

The following examples demonstrate the accuracy of this algorithm.

1. The exact solution of (3.1) is $u_{ex} = \sin(\omega x)$ (the source function $f(x)$ is computed accordingly). Results are shown in Table 1.

| $\omega$ | $\varepsilon_m ax = \|u - u_{ex}\|_{max}$ | | | |
|---|---|---|---|---|
| | $P = 4$ | $P = 8$ | $P = 16$ | $P = 32$ |
| 1.4 | 1.3 (-11) | 3.4 (-12) | 3.2 (-12) | 3.3 (-12) |
| 5.4 | 6.7 (-7) | 2.3 (-11) | 9.9 (-13) | 3.8 (-12) |
| 12.4 | 2.6 (-6) | 1.2 (-9) | 3.3 (-11) | 9.0 (-12) |
| 22.4 | 1.8 (-3) | 9.4 (-7) | 2.1 (-9) | 2.9 (-11) |
| 32.4 | 2.0 (-2) | 4.4 (-6) | 3.3 (-8) | 1.6 (-10) |

Table 1: Maximum error for the solution $u_{ex} = \sin(\omega x)$, $x \in [-1, 1]$. $\lambda = 1$, $N/P = 32$, $I = 8$, $K = 4$.

Numerical tests also show that the error remains almost the same when the local resolution $n = N/P$ increases by factor $k$ while the number of subdomains $P$ is lessen by the same factor, so that the total number of points $N$ does not changes. This result testifies to the high accuracy of the smoothing procedure.

The rate of convergence $\varepsilon = O(N^{-k})$ is rather high: for $\omega = 22.4$ $k$ varies from $k = 7$ at high resolution $P = 16,\ 32$ up to $k = 12$ at lower resolution $P = 4,\ 8$; for $\omega = 32.4$ $k$ varies from $k = 6$ at $P = 16,\ 32$ up to $k = 10$ at $P = 4,\ 8$.

2. The exact solution of (3.1) is $u_{ex} = \sinh(\alpha x)/\sinh(\alpha)$, which exhibit a steep distribution near the boundaries when $\alpha$ is large enough. Results are shown in Table 2.

| $\alpha$ | $\varepsilon_m ax = \|u - u_{ex}\|_{max}$ | | |
|---|---|---|---|
| | $P = 1$ | $P = 2$ | $P = 4$ |
| 1 | 1.6 (-12) | 7.0 (-13) | 3.5 (-13) |
| 5 | 3.2 (-11) | 2.4 (-13) | 2.9 (-13) |
| 20 | 4.9 (-7) | 1.6 (-9) | 2.2 (-11) |

Table 2: The maximum error for the solution $u_{ex} = \sinh(\alpha x)/\sinh(\alpha)$, $x \in [-1, 1]$. $\lambda = 1, \alpha = 20,\ N/P = 32,\ K = 4,\ I = 6$.

In this case, the exponential factor $k$, that is related to $\varepsilon = O(N^{-k})$, varies in the range $8 < k < 10$.

The LFB method (section 2), together with the extrapolation procedure (3.3) on the boundaries, converges rather rapidly even in these tough cases that are demonstrated in Tables 1,2, but the rate of convergence is not exponential. However, we can foster the accuracy of the performance in the boundary subdomains by increasing the resolution with respect to that in the interior subdomains. Then, the lower rate of convergence near the boundaries will not effect the accuracy of the global solution in the whole region.

# 4 Solutions with Non-Constant Coefficients

## 4.1 Finite Difference High-Order Preconditioner

In this section we describe an algorithm that utilizes the pseudo-spectral Fourier approximation, for the solution of differential equations with *non-constant* coefficients. The computational load and the memory it consumes are of the same order of magnitude as required for the solution of equations with constant coefficient having the same resolution.

We illustrate the technique for the second-order differential equation

$$\mathcal{L}u = -(a(x)u')' + \mu(x)u = f(x), \qquad x \in [0, L] \tag{4.1}$$

with periodic boundary conditions $u(x+L) = u(x)$, $u'(x+L) = u'(x)$ and the coefficients $a(x), \mu(x) > 0$ for $x \in [0, L]$ . We discretize the functions $u(x), a(x), \mu(x)$ and $f(x)$ on a uniform grid of collocation points $x_j = Lj/N$ $(j = 0, 1, ..., N)$ and seek an approximated solution as the finite Fourier series (assuming for simplicity $L = 2\pi$):

$$\gamma(x_j) \equiv \gamma_j = \sum_{|k| \le N/2} \hat{\gamma}_k e^{ikx_j}, \qquad \gamma = u, f, a, \mu. \tag{4.2}$$

The Fourier coefficients $\hat{u}_k$, $\hat{a}_k$, $\hat{\mu}_k$ and $\hat{f}_k$ can be obtained by using the FFT.

The system of linear equations for the unknown values $\hat{u}_k$ becomes:

$$\sum_{\substack{|p| \le N/2 \\ |k-p| \le N/2}} (-p^2 + ipa_{p-k} + b_{p-k})\hat{u}_p = -f_k \tag{4.3}$$

In general, it is a full matrix representation that requires $O(N^2)$ storage locations and $O(N^3)$ operations to invert (using Gauss elimination or other direct methods). Obviously, for multidimensional problems with realistic size (typically $N \sim 10^6$ in each dimension), then, the direct solution of (4.3) is prohibitively expensive.

Instead, a commonly adopted approach to the solution of large linear algebraic systems consists of using iterative methods (like Richardson scheme or, more sophisticated, the CG method). In the simple Richardson scheme the successive approximation to the solution of the linear problem

$$\mathcal{L}u = f \tag{4.4}$$

is obtained via

$$u^{n+1} = u^n + \omega\,(f - \mathcal{L}u^n) \tag{4.5}$$

The optimal value of the relaxation parameter $\omega = \frac{2}{\lambda_{max} + \lambda_{min}}$, where $\lambda_{max}(\mathcal{L})$ and $\lambda_{min}(\mathcal{L})$ are the extreme (maximum and minimum) eigenvalues of the operator $\mathcal{L}$ (e.g. see [11]). At each iteration step the error is reduced by factor $\rho = (\kappa - 1)/(\kappa + 1)$, where

$$\kappa = \lambda_{max}(\mathcal{L})/\lambda_{min}(\mathcal{L}). \tag{4.6}$$

The number of iterations, that is required to reduce the error in the solution by one order of magnitude, is proportional to $\kappa$. Therefore, the iteration method is efficient only if the condition number is bounded while the size of the matrix $\mathcal{L}$ increases.

It can be shown, however, that the condition number grows with the resolution $N$. In particular, for the second order operator (4.1) $\kappa \propto N^2$ regardless which discretization method (spectral or finite differences) is used. Therefore, the straightforward implementation of the iteration procedure (4.5) is inefficient. This can be alleviated by "preconditioning" the problem. The idea is to solve the problem

$$H^{-1}\mathcal{L}u = H^{-1}f \tag{4.7}$$

rather than (4.4). Both operator equations, (4.4) and (4.7), are equivalent but Eq. (4.7) can be iterated efficiently provided that the auxiliary operator $H$ satisfies the following properties:

**(i)** $H$ is a sparse matrix representation in the appropriate basis (so that it can be represented by $O(N)$ storage locations).

**(ii)** $H$ in this basis is inverted.

**(iii)** $H$ approximates $\mathcal{L}$ such that the condition number of the "compound" operator $H^{-1}\mathcal{L}$ be bounded (independent of $N$).

The iteration scheme now looks as follows: Starting with an initial guess $u^0 = 0$, we proceed, then, by solving successively the problems:

$$Hu^{n+1} = f - (\mathcal{L} - H)u^n. \tag{4.8}$$

Several options for choosing the preconditioner $H$ were described in the literature. If a second-order finite-difference approximation to $\mathcal{L}$ is used [10], the condition number becomes *independent* of $N$. The use of a higher order finite-difference preconditioner improves the convergence, but it is partially offset by the additional cost for inverting the finite-difference operator discretized on a larger stencil. Therefore, the high-order finite-difference preconditioners are of limited usage.

## 4.2 Spectral Preconditioner

The idea of the current approach is to approximate the differential operator (Eq. (4.1)) that has *variable* coefficients by the corresponding operator with *constant* coefficients of the form:

$$Hu = -\bar{a}u'' + \bar{\mu}u \tag{4.9}$$

Here $\bar{a}$ and $\bar{\mu}$ are the mean values of $a(x)$ and $\mu(x)$ respectively, obtained by using the proper averaging procedure, such as

$$\bar{a} = \frac{1}{L}\int_0^L a(x)dx. \tag{4.10}$$

The operator $H$ is represented in the Fourier basis by a diagonal matrix, that is easily inverted. At the same time, such *spectral* preconditioner (scheme (4.8)) provides an efficient convergence ($O(1)$ iterations). Therefore, the solution of the problem (4.1) with non-constant coefficients is accomplished with a little more resources (computation and memory) than required to solve the constant-coefficient equation (4.9).

We estimate now the condition number of the operator $H^{-1}\mathcal{L}$ where $\mathcal{L}$ and $H$ are defined in Eq. (4.1) and (4.9). Let $u$ be an eigenfunction of $\mathcal{L}$, normalized by

$$(u,u) \equiv \int_0^L u^* u = 1$$

(here $*$ denotes complex conjugate), and the corresponding eigenvalue is

$$\lambda(\mathcal{L}) = (u, \mathcal{L}u) = \int_0^L u^* \mathcal{L} u dx = \int_0^L u^*[(-au')' + \mu u]dx$$

$$= \int_0^L (a|u'|^2 + \mu |u|^2)dx = (u', au') + (u, \mu u). \tag{4.11}$$

If the coefficients $a(x)$ and $\mu(x)$ vary within the intervals

$$0 < a_1 \le a(x) \le a_2, \; 0 < \mu_1 \le \mu(x) \le \mu_2$$

then (4.11) is bounded by

$$a_1(u',u') + \mu_1(u,u) \le (u', au') + (u, \mu u) \le a_2(u',u') + \mu_2(u,u). \tag{4.12}$$

Let us introduce the operators $H_1$ and $H_2$ such that

$$H_1 u = -a_1 u'' + \mu_1 u$$

$$H_2 u = -a_2 u'' + \mu_2 u \tag{4.13}$$

Then, the following estimates are valid:

$$a_1(u',u') + \mu_1(u,u) = (u, H_1 u) = \lambda(H_1) \ge \lambda_{min}(H_1),$$

$$a_2(u',u') + \mu_2(u,u) = (u, H_2 u) = \lambda(H_2) \le \lambda_{max}(H_2), \tag{4.14}$$

Combining (4.11)-(4.14) we obtain

$$\lambda_{min}(H_1) \le \lambda_{min}(\mathcal{L}), \qquad \lambda_{max}(\mathcal{L}) \le \lambda_{max}(H_2)$$

and thus

$$\kappa(\mathcal{L}) \le \frac{\lambda_{max}(H_2)}{\lambda_{min}(H_1)} \tag{4.15}$$

Similar consideration leads us to the following estimates for the eigenvalues of the preconditioned operator $H^{-1}\mathcal{L}$:

$$\lambda_{min}(H^{-1}H_1) \le \lambda_{min}(H^{-1}\mathcal{L}), \qquad \lambda_{max}(H^{-1}\mathcal{L}) \le \lambda_{max}(H^{-1}H_2),$$

$$\kappa(H^{-1}\mathcal{L}) \le \frac{\lambda_{max}(H^{-1}H_2)}{\lambda_{min}(H^{-1}H_1)} \tag{4.16}$$

The eigenfunctions of (3.13) (on the class of $2\pi$- periodic functions) are

$$w_p(x) = e^{ipx}, \tag{4.17}$$

with the associated eigenvalues

$$\lambda(H_r) = a_r p^2 + \mu_r \qquad r = 1, 2 \tag{4.18}$$

Similar representation in the Fourier basis can be used for the preconditioner $H$:

$$(w_p, Hw_p) = \lambda(H) = \bar{a}p^2 + \bar{\mu}. \tag{4.19}$$

Thus, the eigenvalues of $H^{-1}H_1$ and $H^{-1}H_2$ are given by

$$\lambda(H^{-1}H_r) = \frac{a_r p^2 + \mu_r}{\bar{a}p^2 + \bar{\mu}} \qquad r = 1, 2 \tag{4.20}$$

In the discrete setting, the values of $p$ are localized within the range $-N/2 \leq p \leq N/2$, and they depend on the resolution $N$ in the physical domain. Hence, our task is reduced to obtain the estimates:

$$\lambda_{min}(H^{-1}H_1) = \min_{0 \leq |p| \leq N/2} \frac{a_1 p^2 + \mu_1}{\bar{a}p^2 + \bar{\mu}} \tag{4.21}$$

and

$$\lambda_{max}(H^{-1}H_2) = \max_{0 \leq |p| \leq N/2} \frac{a_2 p^2 + \mu_2}{\bar{a}p^2 + \bar{\mu}}. \tag{4.22}$$

Using the relations

$$a_1 \leq \bar{a} \leq a_2, \qquad \mu_1 \leq \bar{\mu} \leq \mu_2 \tag{4.23}$$

and defining $n = N/2$, one can easily prove that

$$\lambda_{min}(H^{-1}H_1) = \begin{cases} \mu_1/\bar{\mu}, & \frac{\mu_1}{\bar{\mu}} < \frac{a_1}{\bar{a}} \\ \frac{a_1 n^2 + \mu_1}{\bar{a}n^2 + \bar{\mu}}, & \frac{\mu_1}{\bar{\mu}} > \frac{a_1}{\bar{a}} \end{cases} \tag{4.24}$$

$$\lambda_{max}(H^{-1}H_2) = \begin{cases} \mu_2/\bar{\mu}, & \frac{\mu_2}{\bar{\mu}} > \frac{a_2}{\bar{a}} \\ \frac{a_2 n^2 + \mu_2}{\bar{a}n^2 + \bar{\mu}}, & \frac{\mu_2}{\bar{\mu}} < \frac{a_2}{\bar{a}}. \end{cases} \tag{4.25}$$

Finally, we get

$$\kappa(H^{-1}\mathcal{L}) \leq \max\left(\frac{\mu_2}{\mu_1}, \frac{\mu_2}{\bar{\mu}} \cdot \frac{\bar{a}n^2 + \bar{\mu}}{a_1 n^2 + \mu_1}, \frac{a_2 n^2 + \mu_2}{\bar{a}n^2 + \bar{\mu}} \cdot \frac{\bar{\mu}}{\mu_1}, \frac{a_2 n^2 + \mu_2}{a_1 n^2 + \mu_1}\right). \tag{4.26}$$

In specific cases where one of the coefficients $a(x)$ or $\mu(x)$ is constant, we have:

$$\kappa(H^{-1}\mathcal{L}) \leq \frac{\mu_2}{\mu_1}, \qquad a = \text{const}$$

$$\kappa(H^{-1}\mathcal{L}) \leq \frac{a_2 + m}{a_1 + m}, \qquad \mu = \text{const}, \ m \equiv \mu/n^2. \tag{4.27}$$

Note that in these cases the condition number does not depend upon the choice of the preconditioner unless the conditions (4.23) are held.

The important conclusion is that the condition number of the operator $H^{-1}\mathcal{L}$, with the preconditioner (4.9), is independent from the resolution $N$ but depends on the variance of the coefficients $a(x)$ and $\mu(x)$ in Eq. (4.1). Naturally, the condition number increases if the coefficients vary in a wider range, i.e. for larger ratios $\mu_2/\mu_1$ and $a_2/a_1$.

We demonstrate the convergence of the CG iterative method, using the spectral preconditioner, on

$$\mathcal{L}u = u'' - \mu(x)u = f(x), \qquad x \in [-1,1] \tag{4.28}$$

where $\mu(x) = \mu_0(1+\epsilon sin(\pi\omega x))$, $\mu_0 = 10, \omega = 1.4$ and the forcing function $f(x)$ corresponds to the solution reference $u_{ex}(x) = sin(\pi\omega_0 x)$, $\omega_0 = 3$. The corresponding preconditioner operator $H$ is

$$Hu = u'' - \bar{\mu}u \tag{4.29}$$

Figures 1-3 demonstrate the decay of the maximum point-value error for the number of iterations with different parameters $\bar{\mu}$ and $\epsilon$. Figure 1 is related to the case $\bar{\mu} = \mu_0$, $\epsilon = 0.1, 0.3, 0.5$ ($\mu_{max}/\mu_{min} = \frac{11}{9}, \frac{13}{7}, 3$ respectively). For the range of the variance of $\mu$ the minimum error (for a given resolution) is achieved after a few (6-10) iterations, *independent* of the resolution. Figure 2 shows the convergence in $\epsilon = 0.5$ for different $\bar{\mu} = 6, 10, 14$. According to our prediction, in this particular case ($a(x)$ =const), the rate of convergence is almost independent of the choice of $\bar{\mu}$.

In practice, the current algorithm is more robust than we can expect according to estimates (4.26)-(4.27). Indeed, for $\epsilon = 1$ the minimum value $\mu_{min} = \mu_0(1-\epsilon) = 0$ and, thus, the upper-bound of the condition number tends to infinity according to (4.27). The numerical experiment gives, however, fairly rapid convergence at $\epsilon = 1$ and even at larger $\epsilon = 2$ (see Figure 3).

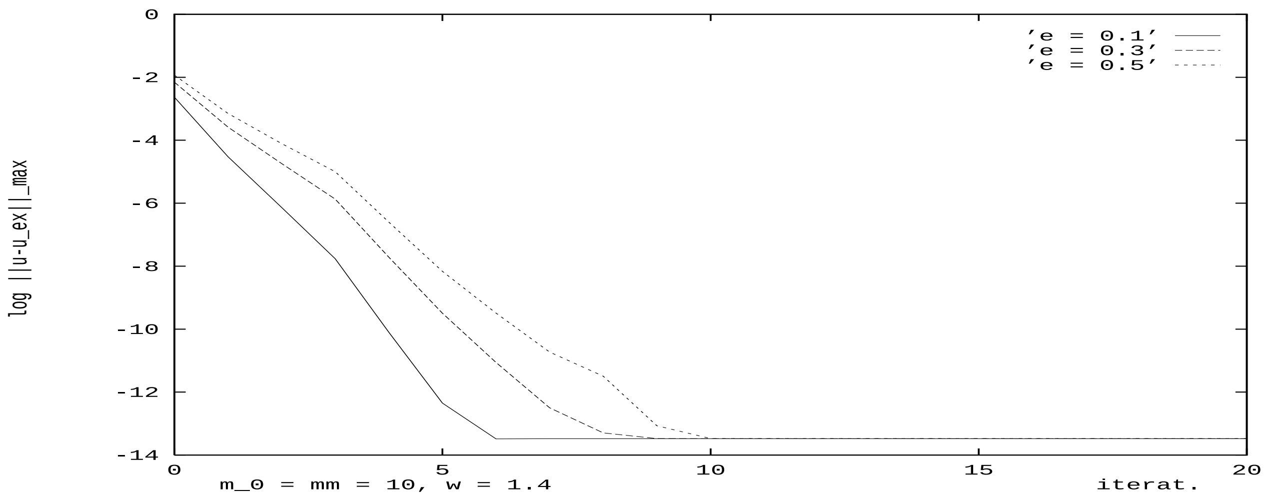


Figure 1: Decay of the maximum point-value error for the number of iterations with different parameters $\bar{\mu}$ and $\epsilon$ where $\bar{\mu} = \mu_0,\ \epsilon = 0.1, 0.3, 0.5$ ($\mu_{max}/\mu_{min} = \frac{11}{9}, \frac{13}{7}, 3$).

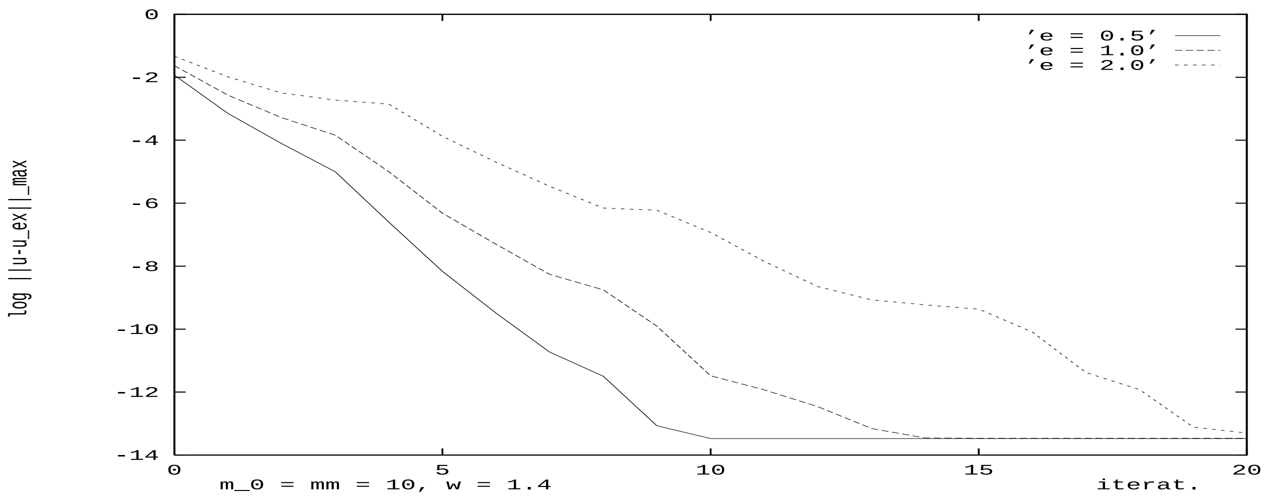


Figure 2: Convergence in $\epsilon = 0.5$ for different $\bar{\mu} = 6, 10, 14$.

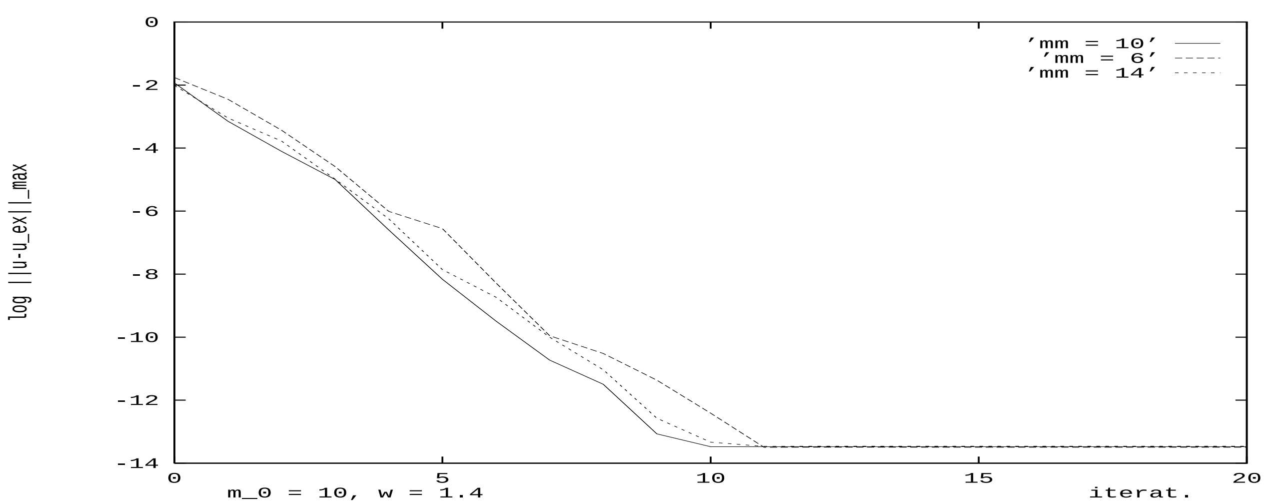


Figure 3:

The following remark is important; Although we are after a periodic solution for (3.28), the accuracy of the iterated algorithm is *not affected* by the fact whether $\mu(x)$ is periodic or not (in our example $\mu(x)$ is not periodic on $x \in [-1,1]$ for $\omega = 1.4$). Moreover, the accuracy is independent from the smoothness of $\mu(x)$ inside the computational interval since the residual $(\mathcal{L} - H)u^n = (\mu(x) - \bar{\mu})u^n$ contains only multiplication by this coefficient, but not the derivative of $\mu(x)$ itself. The last conclusion can be illustrated by figure 4, which corresponds to $\mu(x) = |x|, \bar{\mu} = 0.5, u_{ex} = \sin \pi x$. The same is true for a more general case such as Eq. (4.1).

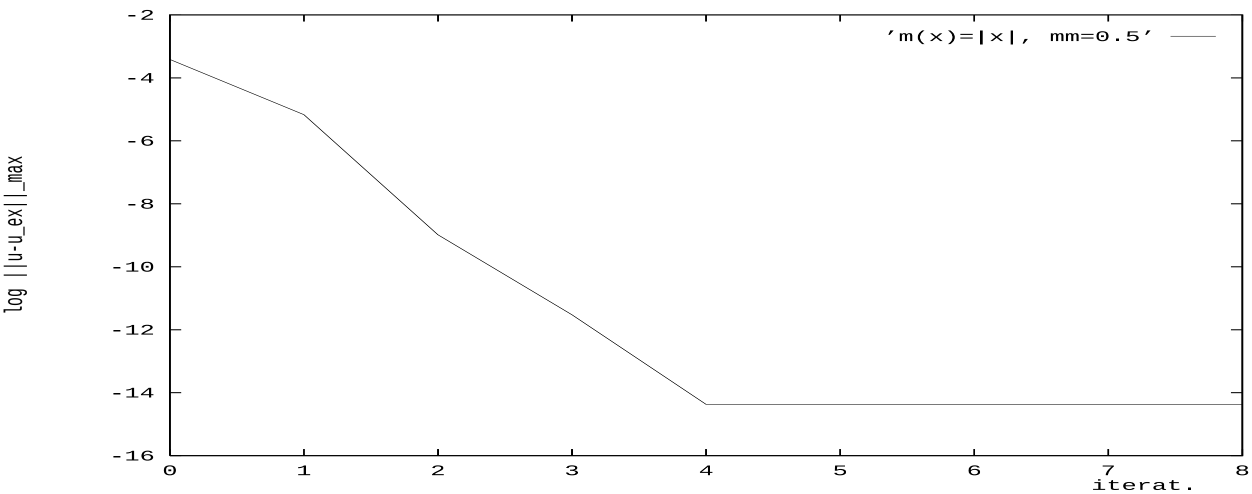


Figure 4: The accuracy is independent from the smoothness of $\mu(x)$ inside the computational interval. Here we have $\mu(x) = |x|, \bar{\mu} = 0.5, u_{ex} = \sin \pi x$.

## 4.3 Comparison With the Finite-Difference Preconditioners

The 2-d order finite-difference approximation of (4.9) is

$$\mathcal{H}_{II} = -\bar{a}\frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x)^2} + \bar{\mu}u_j = f_j \tag{4.30}$$

and it has the same eigenfunctions (4.17) as the differential equation (4.9). The corresponding eigenvalues are:

$$(w_p, \mathcal{H}_{II}w_p) = \bar{a}\frac{4\sin^2(p\Delta x/2)}{(\Delta x)^2} + \bar{\mu} \qquad p = -N/2, -N/2+1, ..., N/2, \tag{4.31}$$

where $\Delta x = 2\pi/N$.

The same arguments apply to the case of the spectral preconditioner which leads us to the estimates of the condition number of the operator $\mathcal{H}_{II}^{-1}\mathcal{L}$:

$$\kappa(\mathcal{H}_{II}^{-1}\mathcal{L}) \leq \frac{\lambda_{max}(\mathcal{H}_{II}^{-1}H_2)}{\lambda_{min}(\mathcal{H}_{II}^{-1}H_1)} \tag{4.32}$$

where

$$\lambda_{max}(\mathcal{H}_{II}^{-1}H_2) = \max_{0\leq|p|\leq N/2} \frac{a_2p^2+\mu_2}{\bar{a}\frac{4\sin^2(p\Delta x/2)}{(\Delta x)^2}+\bar{\mu}}, \tag{4.33}$$

$$\lambda_{min}(\mathcal{H}_{II}^{-1}H_1) = \min_{0\leq|p|\leq N/2} \frac{a_1p^2+\mu_1}{\bar{a}\frac{4\sin^2(p\Delta x/2)}{(\Delta x)^2}+\bar{\mu}} \tag{4.34}$$

Introducing $\xi = p\Delta x/2,\ 0 \leq \xi \leq \pi/2$, we can rewrite (4.30) as follows:

$$\lambda_{max}(\mathcal{H}_{II}^{-1}H_2) = \max_{0\leq|p|\leq N/2} \frac{a_2\xi^2+\mu_2(\pi/N)^2}{\bar{a}\sin^2\xi+\bar{\mu}(\pi/N)^2} \tag{4.35}$$

For the 4-d order finite-difference approximation of (3.9) $\mathcal{H}_{IV}$:

$$(w_p, \mathcal{H}_{IV}w_p) = \bar{a}\frac{\cos(2p\Delta x)-16\cos(p\Delta x)+15}{6(\Delta x)^2}+\bar{\mu}u_j \qquad p=-N/2, -N/2+1, ..., N/2$$

# 5 Preconditioned Conjugate Gradient Method in Multi-Domains Decompositions

If the coefficients $a(x)$ and $\mu(x)$ vary significantly in the computational interval, the preconditioner $H(\bar{a},\bar{\mu})$, obtained by the averaging of $a(x)$ and $\mu(x)$ over the whole interval, is a poor approximation to $\mathcal{L}(x) = \mathcal{L}(a(x),\mu(x))$. Consequently, a large number of iterations are required to achieve a predetermined accuracy. This can be alleviated by combining the previous iterative algorithm with domain decomposition technique, described in section 2. The idea is to approximate the operator $\mathcal{L}(x)$ by a *sequence* of constant coefficient operators which are best fitted to $\mathcal{L}(x)$ on smaller sub-intervals.

We illustrate this approach by the following example:

$$\mathcal{L}(r)u = u'' + \frac{1}{r}u' - \frac{m^2}{r^2}u - \lambda^2u = f(r), \qquad r\in[r_0, R]$$

$$u'(r_0) = u(R) = 0, \tag{5.1}$$

Eq.(5.1) is of special interest since it represents the radial part of the modified Helmholtz equation in the annulus (after the application of the Fourier transform on the azimuthal direction).

Assume that the exact solution is $u_{ex}(x) = (r-r_0)^2\,(r-R))$ on the interval $r_0 = 0.1,\ R = 1$ such that the coefficients $1/r,\ 1/r^2$ are modified by one and two orders of magnitude in the range of $[r_0, R]$ respectively.

We divide the computational interval $[r_0, R]$ into $P$ sub-intervals $[l_j, l_{j+1}]$, where $l_j = r + (j-1)l$; $l = (R - r_0)/P$, $j = 1, 2, ..., P$, and approximate the coefficients $b(x)1/r$ and $c(r) = 1/r^2$ in a step-wise manner

$$\bar{b}_j = \frac{1}{l}\int_{l_j}^{l_{j+1}} b(r)dr, \quad \bar{c}_j = \frac{1}{l}\int_{l_j}^{l_{j+1}} c(r)dr \tag{5.2}$$

like in Figure 5.

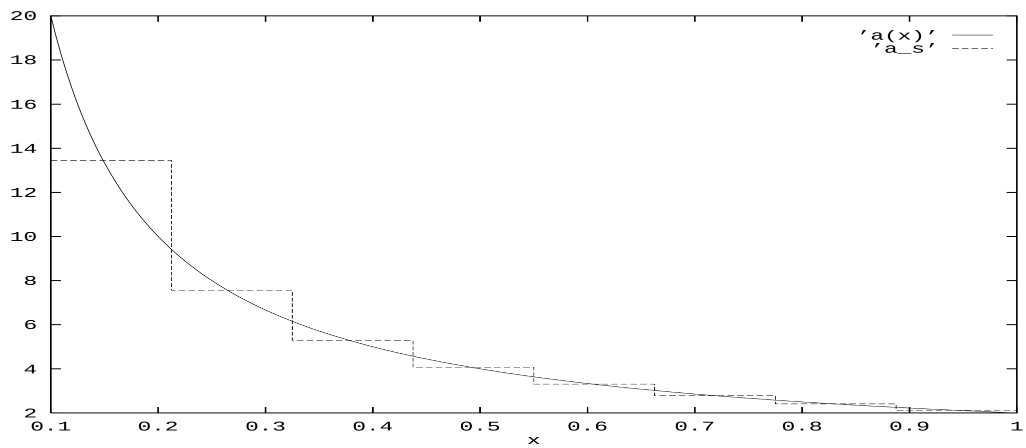


Figure 5: Step-wise approximation of the variable coefficient $\frac{1}{r}$ in Eq. (5.1)

Now, we define a sequence of operators $H_j$

$$H_j u = u'' + \bar{b}_j u' + \bar{c}_j u, \qquad r \in [l_j, l_{j+1}], \tag{5.3}$$

where each one approximates the operator $\mathcal{L}(r)$ in Eq. (5.1) on the corresponding sub-interval $[l_j, l_{j+1}]$. The spectrally accurate solution of Eq. (5.1) is obtained using the appropriate iteration scheme. The constant coefficient operators Eq. (5.3) serve as preconditioners to $\mathcal{L}(r)$ on the corresponding sub-intervals. In each iterative step the matching of the local solutions is necessary to impose the continuity conditions on the interfaces $r = l_j$:

$$r = l_j, \qquad u_j^n = u_{j+1}^n, \qquad \frac{du_j^n}{dr} = \frac{du_{j+1}^n}{dr},$$

The convergence history of the preconditioned conjugate gradient (PCG) iterations is shown in Figure 6 for different numbers of sub-divisions $P = 2, 4, 8, 12$ (the local resolution is fine enough to ensure the maximum accuracy for a given $P$, $m = 1$ in Eq. (5.1)).

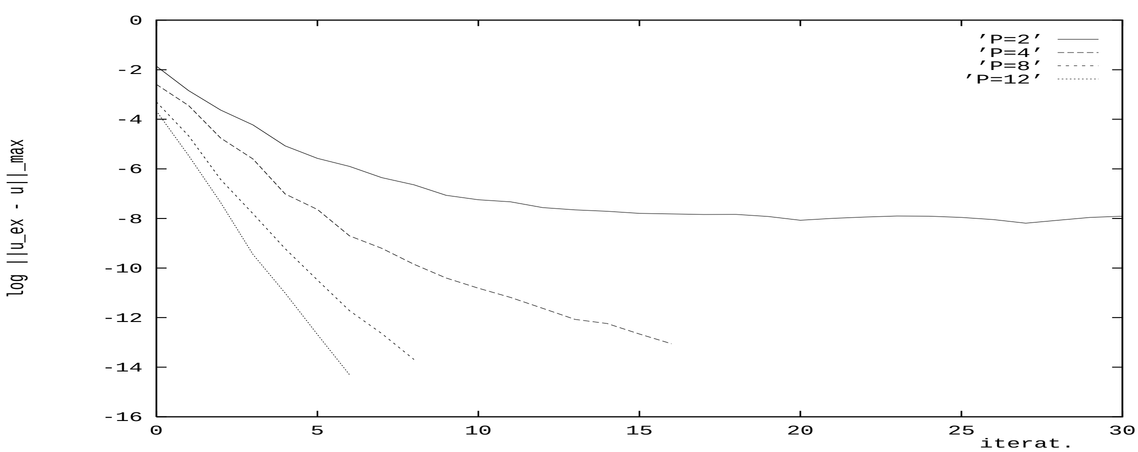


Figure 6: Convergence history of the PCG iterations for different numbers of sub-divisions $P = 2, 4, 8, 12$.

We can see that the convergence is improved drastically when the PCG method is combined with the domain-decomposition strategy that allows a better approximation of the original problem, with non-constant coefficients, by a sequence of constant-coefficient problems.

If the inner radius $r = r_0$ of the annulus shrinks to zero, the accuracy of the solution Eq. (5.1) deteriorates rapidly since the variance of the coefficients for the terms $u'$ and $u$ on the interval $x \in [r_0, R]$ grows as $1/r_0$ and $1/r_0^2$ respectively. The main factor that causes the deterioration of the convergence in the PCG method is the term $(1/r)u'$ which violates the self-adjoint form of the operator. The eigenvalues of Eq. (5.1) are not positive-definite, as opposed to Eq.(3.11) for a self-adjoint operator Eq. (4.1), and thus the estimates of section (2.1 ????) can not be used in the current case.

The maximum point-wise error after 30 PCG iterations is shown in Figure 7 for $P = 16$, $N/P = 32$, $m = 1$, and $r_0$ varies within $10^{-2} < r_0 < 10^{-3}$.

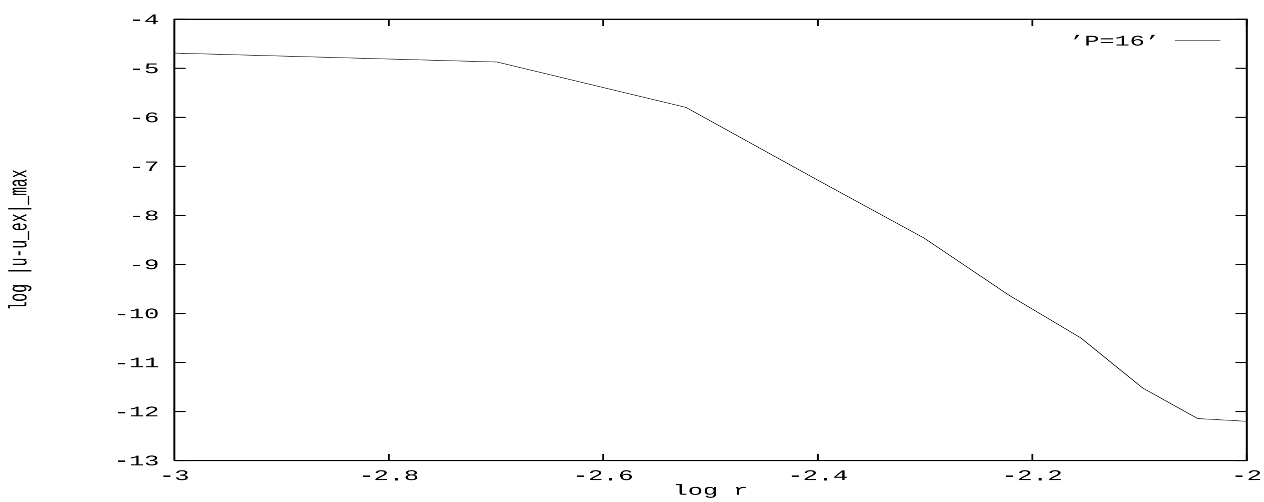


Figure 7: The maximum point-wise error after 30 PCG iterations for $P = 16$, $N/P = 32$, $m = 1$, and $r_0$ varies within $10^{-2} < r_0 < 10^{-3}$.

Although we can preserve the high accuracy by splitting the interval $[r_0, R]$ into a progressively larger number of sub-intervals $P$ while $r$ decreases, it is going to be too expensive.

## 5.1 Transformation to Self-Adjoint Form

In the case of Eq. (5.1) there exists a transformation which permits us to get rid of the term that has first derivative, and thus to bring the operator to a self-adjoint form. We define a new variable as follows:

$$v = u\sqrt{r} \tag{5.4}$$

Substituting 5.4 into Eq. (5.1) we get the following equation for $v$:

$$v'' - \left(\frac{m^2 - 1/4}{r^2} + \lambda^2\right) v = f(r)\sqrt{r}, \qquad r \in [r_0, R]$$

$$v'(r_0) = v(R) = 0, \tag{5.5}$$

The convergence using PCG is shown in Fig. 8 in the cases of Eqs. (5.1) and (5.5) for $r_0 = 0.005,\ m = 1,\ R = 1,\ P = 8\ N/P = 32$.

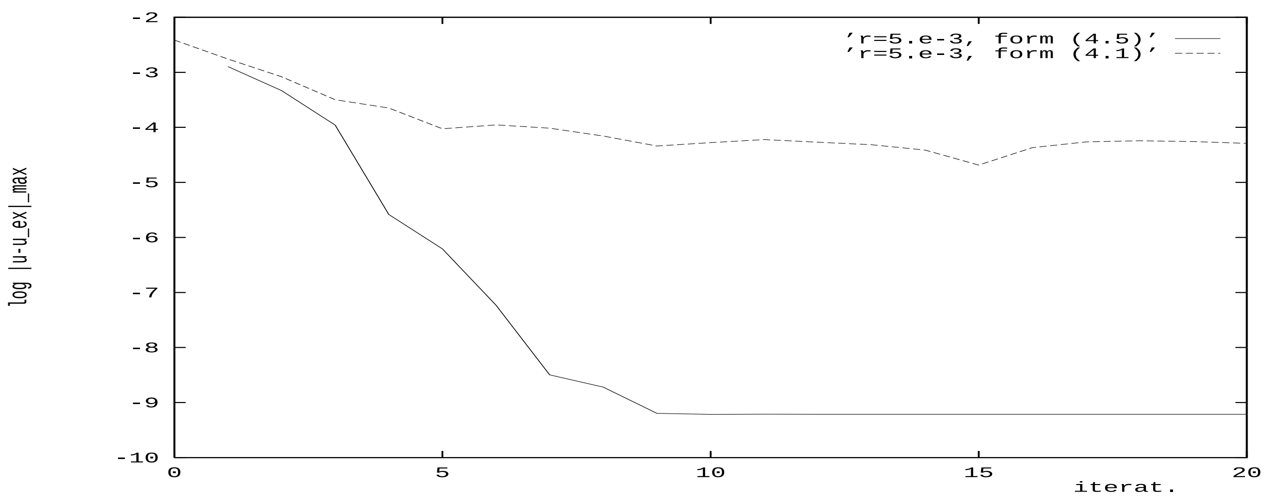


Figure 8: The convergence using for $r_0 = 0.005,\ m = 1,\ R = 1,\ P = 8\ N/P = 32$.

The superiority of the self-adjoint form (5.5) is vividly seen (the difference in the accuracy after 10 iterations makes up 5 orders of magnitude).

Moreover, using the form (5.5) we can treat annulus with an arbitrary small inner radius, including the limit $r_0 \to 0$. In this case a regular solution of Eq. (5.1) must behave near the pole $r = r_0$ like $u(r) \propto r^2$. Therefore, $v'(r) = u'(r)\sqrt{r} + u(r)/(2\sqrt{r}) \to 0$ as $r \to 0$. We can also estimate the second term in Eq. (5.5) near the pole as $v/r^2 \propto \sqrt{r}$ to compute the residual $\left(\overline{1/x^2} - 1/x^2\right) v \to 0$ in the limit $r \to 0$. The convergence of this method is shown in Fig. 9 for $r = 0$, $m = 1$, $N/P = 32$, and $P = 4, 16, 32$. The maximum accuracy is achieved after 13 PCG iterations independent of the number of sub-intervals $P$. The accuracy is improved by increasing the number $P$.

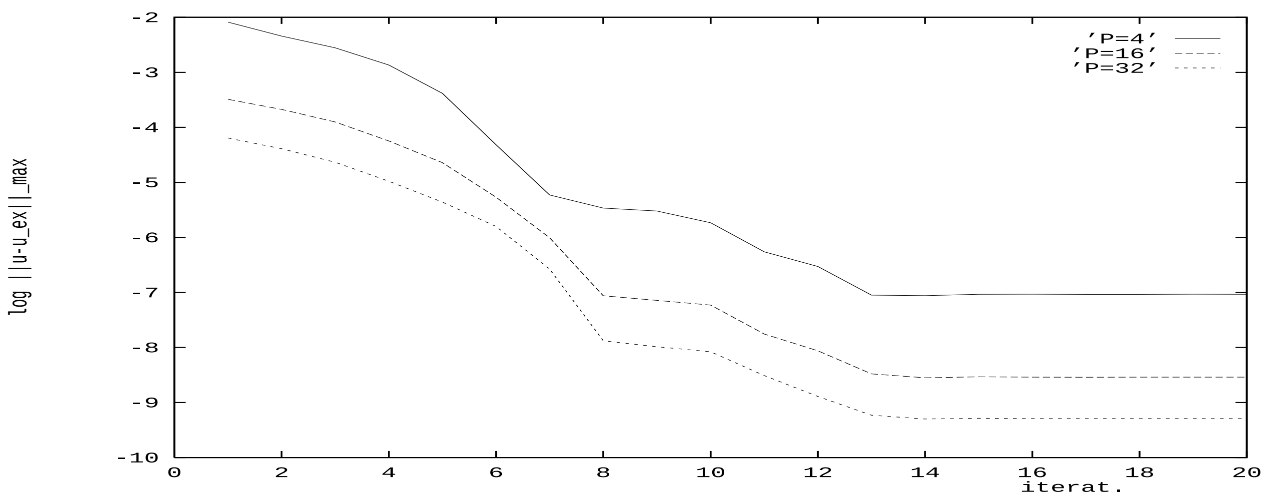


Figure 9: The convergence of this method for $r = 0$, $m = 1$, $N/P = 32$, and $P = 4, 16, 32$. The maximum accuracy is achieved after 13 PCG iterations independent of the number of sub-intervals $P$. The accuracy is improved by increasing the number $P$.

# 6 Problems in Complex Geometries

In this section we apply the multi-domain Fourier method to the solution of problems in complicated geometries. Consider the second-order elliptic equation

$$\nabla^2 u - \lambda^2 u = f(x, y), \qquad x \text{ in } \Omega, \tag{6.1}$$

in the region $\Omega = \{0 \le x \le L,\ f_1(x) \le y \le f_2(x)\}$. Here $f_1(x)$ and $f_2(x)$ are smooth $L$-periodic functions of $x$, having the mean values $\bar{f_1} = 0$ and $\bar{f_2} = 1$ when averaged over the interval $L$. The following boundary conditions on $u(x, y)$ are given:

$$y = f_1(x), \qquad u = \phi_1(x),$$
$$y = f_2(x), \qquad u = \phi_2(x),$$

$$u(0, y) = u(L, y). \tag{6.2}$$

where $\phi_1(x)$ and $\phi_2(x)$ are smooth $L$- periodic functions of $x$.

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

$$\eta = \frac{y - f_1(x)}{f_2(x) - f_1(x)} \qquad \xi = x \tag{6.3}$$

Define

$$\alpha = f_1, \qquad \beta = f_2 - f_1,$$

$$\eta_y = \frac{1}{\beta}, \qquad \eta_x = -\ \frac{\alpha' + \eta\,\beta'}{\beta},$$

$$\eta_{xx} = -\ \frac{\alpha'' + \beta'' + 2\eta_x\,\beta'}{\beta}$$

$$J(\xi,\eta) = \eta_x^2 + \eta_y^2 \tag{6.4}$$

then, we can rewrite Eq. (6.1) in the transformed domain $\omega$, as:

$$\mathcal{L}u = u_{\eta\eta} + (2\eta_x/J)u_{\xi\eta} + (1/J)u_{\xi\xi} + (\eta_{xx}/J)u_\eta - (\lambda^2/J)u = f(\xi,\eta)/J, \qquad \text{in } \omega \tag{6.5}$$

(here subscripts $x$ and $y$ denote derivatives with respect to the corresponding coordinates, while $'$ denotes the derivative of a function in one variable). The derivatives of $f_1(x)$ and $f_2(x)$ in (6.4) can be computed either analytically or numerically (using the discrete Fourier transform of the node-values and spectral differentiation in the Fourier domain).

In our method it is convenient to deal with uniform boundary conditions. This can be achieved by changing the variables:

$$u = v + \eta\phi_2(\xi) + (1-\eta)\phi_1(\xi) \tag{6.6}$$

The new variable $v(\xi,\eta)$ vanishes on the stretched boundaries

$$\eta = 0,1 \qquad\qquad v = 0,$$

$$v(0,\eta) = v(L,\eta). \tag{6.7}$$

and it satisfies the same differential Eq. (6.5) but with the forcing term in the RHS that is modified to:

$$\mathcal{L}v = F(\xi,\eta) = \frac{1}{J}(f + \tilde{f}),$$

where

$$\tilde{f} = 2\eta_x(\phi_1' - \phi_2') - \eta_{xx}(\phi_1 - \phi_2) + \lambda^2(\eta\phi_2 + (1-\eta)\phi_1) - (\eta\phi_2'' + (1-\eta)\phi_1'') \tag{6.8}$$

Eq. (6.8), with variable coefficients, generated by the mapping (6.3), can be solved by using the preconditioned CG method as described in sections 4 and 5. To accomplish this, we divide the computational domain $\omega = \bigcup_{j=1}^{S}\omega_j$ into strips, orientated along the axis $x$, and in each sub-domain we construct the appropriate spectral preconditioner as follows:

$$H_j u = u_{\eta\eta} + 2(\overline{\eta_x/J})_j\ u_{\xi\eta} + (\overline{1/J})_j\ u_{\xi\xi} + (\overline{\eta_{xx}/J})_j\ u_\eta - (\overline{\lambda/J})_j\ u, \qquad \text{in } \omega_j \tag{6.9}$$

The coefficients $(\overline{\eta_x/J})_j$, $(\overline{1/J})_j$ and $(\overline{\eta_{xx}/J})_j$ are obtained by the averaging of the corresponding non-constant coefficients in Eq. (6.5) over a sub-domain $\omega_j$ like

$$\bar{C}_j = \frac{1}{L/S}\int_{\omega_j} C(\xi,\eta)\,d\xi\,d\eta \tag{6.10}$$

Assume that there is a symmetry across the middle point $x = L/2$, $f_1(x) = f_1(L - x)$, $f_2(x) = f_2(L - x)$. We can omit the term with mixed derivatives in Eq. (6.9) since in this case $\overline{(\eta_x/J)}_j = 0$. If the boundaries do not have such a parity symmetry, we can either ignore this term in the preconditioner (since the averaged values $\overline{(\eta_x/J)}_j$ still remains small for a wide range of geometric parameters), or take it into account with extra minor complication of the algorithm (instead of complete separation of harmonics after the Fourier transform in the $\xi$ direction, we get pairs of coupled harmonics).

Let us summarize all the computational steps of the algorithm. Given the problem, that is described by Eqs. (6.1) and (6.2) in the region $\Omega = \{0 \leq x \leq L,\ f_1(x) \leq y \leq f_2(x)\}$:

1. Compute the first and the second derivatives of $f_1(x)$ and $f_2(x)$ (using the FFT if these functions are given in discrete form).

2. Compute the non-constant coefficients (Eq. (6.4)) of the operator $\mathcal{L}$ (Eq. (6.5)) in the mapped domain $\omega = \{0 \leq \xi \leq L,\ 0 \leq \eta \leq 1\}$.

3. Compute the new source function $F(\xi, \eta)$ (Eq. (6.8)), generated by transforming Eq. (6.6) to the zero boundary condition (Eq. (6.7)).

4. Decompose $\omega$ into strips, stretched in the periodic direction $\xi$: $\omega = \bigcup_{j=1}^{S} \omega_j$, where $\omega_j = \{0 \leq \xi \leq L,\ (j-1)/S \leq \eta \leq j/S\}$.

5. Descritize the sub-problems
$$\mathcal{L}v_j = F_j, \qquad \text{in } \omega_j \tag{6.11}$$
on a uniform mesh of collocation points (the mesh-sizes $\Delta\xi_j, \Delta\eta_j$ may depend, in general, on $j$ to allow a different resolution in each sub-domain).

6. Compute the averages $\overline{(1/J)}_j$ $\overline{(\eta_{xx}/J)}_j$ over the sub-domains $\omega_j$ as in Eq. (6.10) to construct local preconditioners (Eq. (6.9)).

7. Iterate, using the PCG scheme with the operators $H_j$ as preconditioners, until the prescribed accuracy is achieved (details are given in Appendix).

8. Transform back to the variable $u$ using Eq. (6.6), and into the original domain $\Omega$, using Eq. (6.3).

# 7 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.

## 7.1 Stationary Problems

Consider the solution of the Poisson equation

$$\Delta u = 1, \qquad \text{in } \Omega \tag{7.1}$$

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:

$$y_1(y) = -(1 + \delta cos\ kx), \qquad y_2(y) = 1 + \delta cos\ kx. \tag{7.2}$$

The following boundary conditions are used:

$$y = y_{1,2}, \qquad u = 0, \quad u(0, y) = u(L, y). \tag{7.3}$$

This problem is solved by the generalized MDLF method. The stretching transformation Eq. (6.3) is used to transform $\Omega$ into the rectangular domain $\omega$ which is decomposed afterwards into parallel strips:

$$\omega = \bigcup_{n=1}^{P} \omega_n, \qquad \omega_n = \{0 \le \xi \le L,\ \bar{\eta}_{n-1} \le \eta \le \bar{\eta}_n\}. \tag{7.4}$$

In each subdomain $\omega_n$ a spectral preconditioner $H_n$ is constructed in the form of Eq. (4.9) with constant coefficients, computed by the averaging of the corresponding variable coefficients in Eq. (4.4) within the strip $\omega_n$.

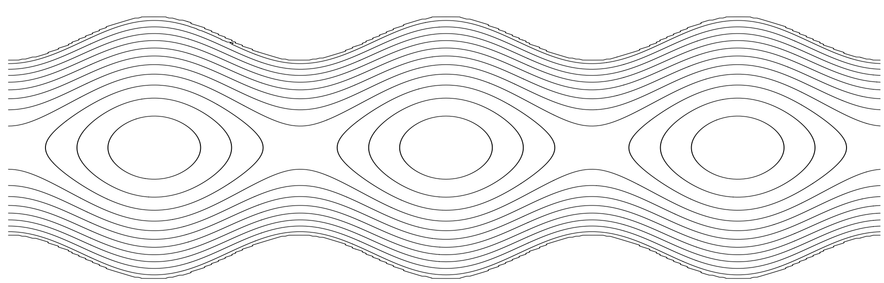

Figure 10: Solution of the Poisson equation (7.1) in a channel with wavy boundaries (lines of constant values $u = const$)

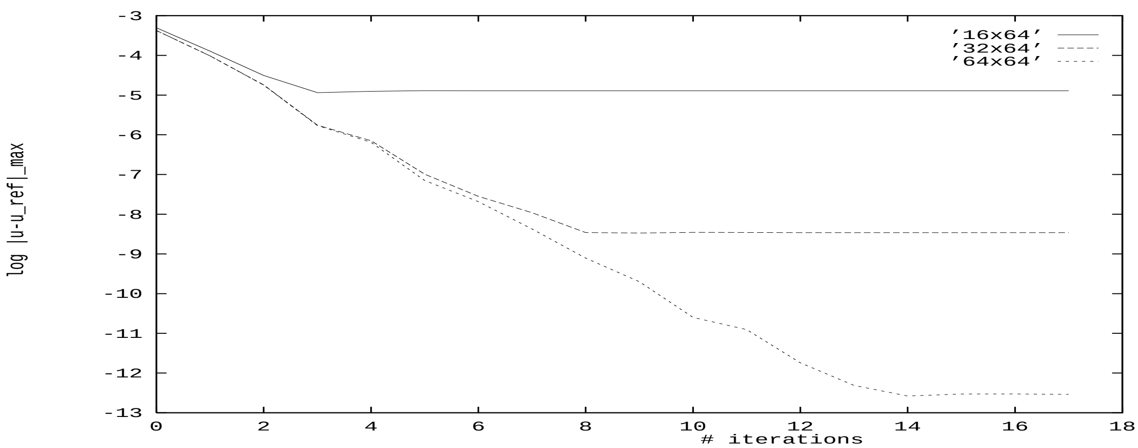


Figure 11: Convergence of the iterative PCG for the geometry in Fig. 10

The computed solution is plotted in Fig. 10 for $\delta = 0.1, k_w = 3, L = 2\pi$. The convergence history of the PCG iterations in this case is shown in Fig. 11 for several resolutions of the lengthwise direction, $N_x = 16, 32, 64$, and a fixed resolution in the other direction, $N_y = 64$. The number of subdomains is $P = 2$. The "exact" solution (reference solution) $u_{ex}$ is computed on the fine grid of size $256 \times 64$. Evidently the rapid convergence of the spectral solution with $N_x$ is preserved in this case.

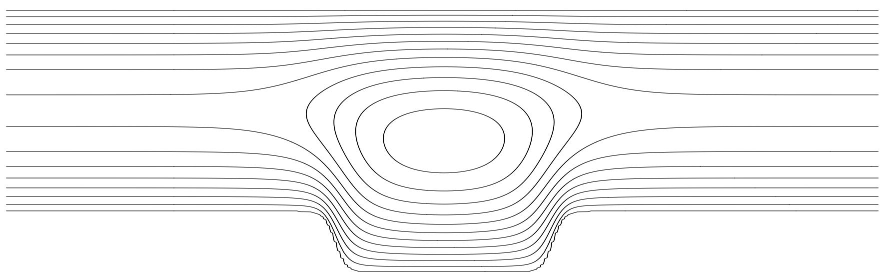

Figure 12: Computed solution of equation (7.1) in a channel with grooves

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. 12 we plot the solution for $x_0 = L/8,\ h = 0.3,\ \alpha = 6,\ L = 2\pi$. In Fig. 13 we plot the convergence history at $N_y = 64$ for different $N_x$. The maximum pointwise error in the asymptotic solution is listed in Table 3; the solution of the reference is computed on the grid $512 \times 64$.

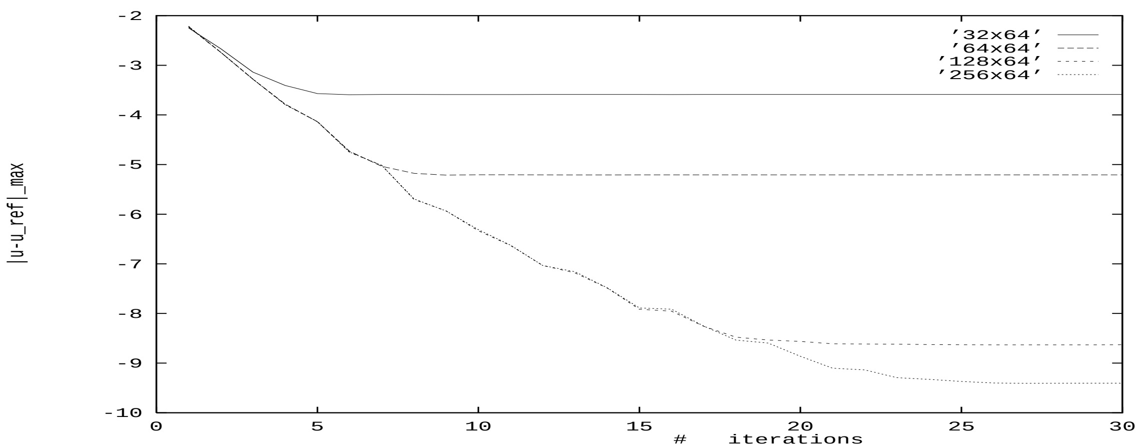


Figure 13: Convergence of the iterative PCG for the geometry in Fig. 12

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

Table 3: Errors in the solution of the Poisson equation (7.1) in a grooved channel, Fig. ($\alpha = 6,\ N_y = 64,\ P = 2$)

The impact of the profile of grooves (parameter $\alpha$) on the accuracy is given in Table 4 (other geometric parameters remain as in Table 3). The accuracy decreases as the profile becomes steeper.

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

Table 4: Dependence of accuracy on the steepness of the groove's profile ($N_x = 128$, $N_y = 64$, $P = 2$)

Unlike the previous case of 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 higher 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. (6.4) in the transformed operator Eq. (6.5). 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. 14 for $n = 9$ (relatively low accuracy sustains after ten 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.

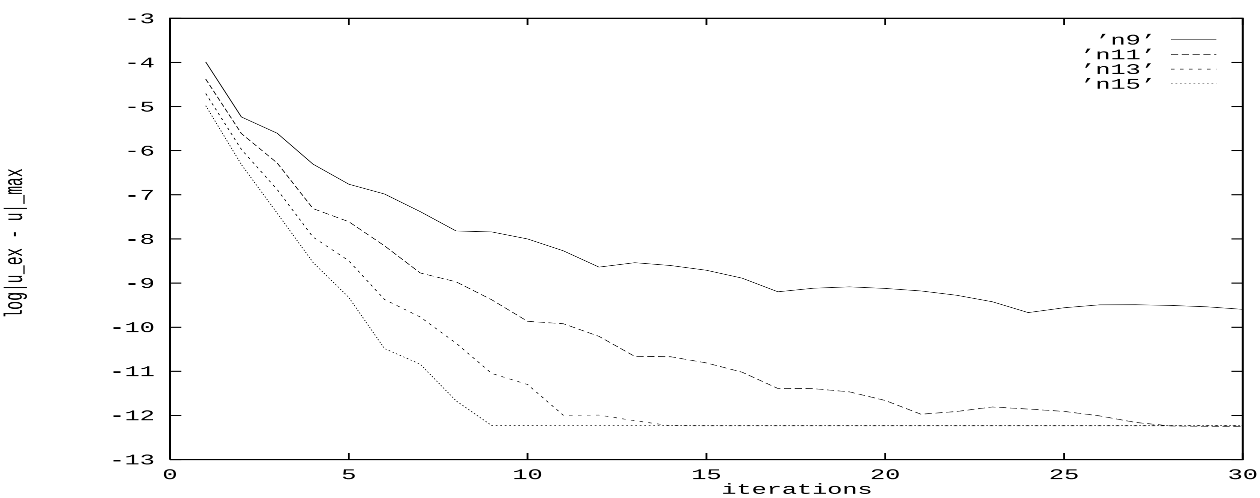


Figure 14: Accelerating the convergence by splitting the computational region and increasing the number $P$ of subdomains.

## 7.2 Time-Dependent Problems

Consider the solution of the heat equation

$$\frac{\partial u}{\partial t} = \nabla^2 u - f(x,y), \qquad \text{in } \Omega \tag{7.5}$$

with Dirichlet boundary conditions

$$u(x,y) = \phi(x,y), \qquad \text{on } \partial\Omega \tag{7.6}$$

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

Time discretization of Eq. (7.5), using an implicit scheme, results in an elliptic equation (6.1) type. For example the simplest two-level Crank-Nicholson scheme reads:

$$\frac{u_{n+1} - u_n}{\tau} = \nabla^2 \left( \frac{u_{n+1} + u_n}{2} \right) - f(x,y) \tag{7.7}$$

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

$$\lambda^2 = 2/\tau, \qquad F_{n+1} = 2f - \nabla^2 u_n - \lambda^2 u_n \tag{7.8}$$

we have

$$\nabla^2_{r,\theta} u_{n+1} - \lambda^2 u_{n+1} = F_{n+1}(r,\theta) \tag{7.9}$$

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. (7.4) of the transformed (rectangular) domain $\omega$ is carried out. For typical $\tau$ used in computations, $\lambda \gg 1$ that justifies the implementation of the local matching procedure for the Helmholtz type equation (7.9) that was discussed in section 2.

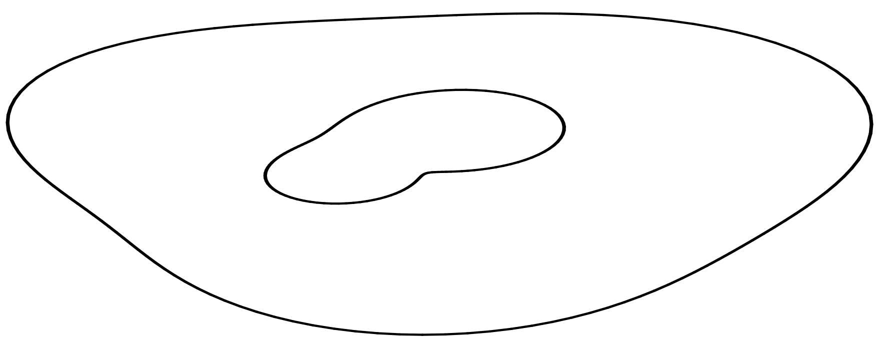

Figure 15: Annular region defined by Eq. (7.7)

As an example we consider the annular region plotted in Fig. 15. The inner and outer boundaries are given, respectively, by

$$\begin{aligned} r_1(\theta) &= 0.3 + 0.1\ \sin\theta + 0.12\ \sin 2\theta, \\ r_2(\theta) &= 1 + 0.1\ \cos\theta + 0.1\ \sin 3\theta. \end{aligned} \tag{7.10}$$

The source function in Eq. (7.5) 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 5. It decays exponentially fast as the resolution in the azimuthal direction increases.

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

Table 5: Errors in steady state solution ( $t \to \infty$) of Eqs. (7.5) and (7.6) in the region Fig. 15; $N_r = 64$, $P = 2$

However, the iterative PCG converges rather slowly in this case (for the resolution $N_\theta = 64$ 45 iterations are needed to achieve convergence). Moreover, for more complicated geometries than the one in Fig. 15 (e.g. annular regions described by Eq. (7.7) 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 (7.1)). More general domain decompositions in the form of rectangular boxes will be able to treat efficiently any complicated geometries.

# A. Appendix

Here we describe in detail the numerical algorithm for the solution of boundary-value problems with non-constant coefficients, based on the preconditioned conjugate gradient method in combination with the domain decomposition technique. The problem is formulated as follows:

$$\mathcal{L}u = u_{yy} + a(x,y)u_{xy} + b(x,y)u_{xx} + c(x,y)u_y + d(x,y)u = f(x,y), \qquad \text{in} \;\; \omega, \qquad (A.1)$$

$$y = 0,1 \qquad\qquad v = 0, \qquad (A.2)$$

where $\omega = (0, L) \times (0, 1)$, and all the functions are periodic in the $x$ direction:

$$\gamma(0,y) = \gamma(L,y), \qquad\qquad \gamma = \{u, f, a, b, c, d\}. \qquad (A.3)$$

We divide the computational domain $\omega$ into parallel strips $\omega_j$, $i = 1,..,S$, stretched in the $x$ direction, in each of which we define the constant coefficient operators $H_j$ as follows:

$$H_j u = u_{yy} + \bar{b}_j \; u_{xx} + \bar{c}_j \; u_y + \bar{d}_j \; u \qquad \text{in} \;\; \omega_j, \qquad (A.4)$$

where

$$\bar{e}_j = \frac{1}{L/S} \int_{\omega_j} e(x,y)dxdy, \qquad\qquad e = \{b, \;\; c, \;\; d\}$$

The second term in Eq.(A.1) with mixed derivatives is not represented in $H_j$. The reasons for that are indicated in section 6 when variable coefficients are generated by the mapping of a complicated domain into a rectangular.

The spectrally accurate solution to Eq.(A.1) is constructed by using the PCG method (e.g. see [11], section 5.3) with the operators $H_j$ serving as preconditioners in the corresponding sub-domains. The algorithm consists of the following steps (the index of a sub-domain in $H_j$ is omitted for the simplicity of the presentation).

1. Initialization:

$$u^0 = 0, \qquad \rightsquigarrow \qquad R^0 = f - \mathcal{L}\ u^0 = f,$$

$$\triangleright H z^0 = R^0,\ z|_{y=0,1} = 0 \quad \rightsquigarrow \qquad z^0,\ z^0_y,\ z^0_{xx},\ z^0_{xy},$$

$$(\mathcal{L} - H) z^0 = (d - \bar{d})\ z^0 + (b - \bar{b})\ z^0_{xx} + (c - \bar{c})\ z^0_y + a\ z^0_{xy}.$$

$$Q^0 = L P^0 = R^0 + (\mathcal{L} - H) z^0,$$

2. Iterations $(n = 1, 2, ...)$:

$$\alpha^n = (R^n, z^n)/(Q^n, P^n),$$

$$u^{n+1} = u^n + \alpha^n P^n,$$

$$R^{n+1} = R^n - \alpha^n Q^n,$$

$$\triangleright H z^{n+1} = R^{n+1},\ z|_{y=0,1} = 0 \qquad \rightsquigarrow \qquad z^{n+1},\ z^n_y\ z^{n+1}_{xx},\ z^{n+1}_{xy},$$

$$(\mathcal{L} - H) z^{n+1} = (d - \bar{d})\ z^{n+1} + (b - \bar{b})\ z^{n+1}_{xx} + (c - \bar{c})\ z^{n+1}_y + a\ z^{n+1}_{xy}$$

$$\beta^n = (R^{n+1}, z^{n+1})/(Q^{n+1}, P^{n+1}),$$

$$P^{n+1} = z^{n+1} + \beta^n P^n,$$

$$\begin{aligned} Q^{n+1} &= \mathcal{L} P^{n+1} \\ &= \beta^n Q^n + \mathcal{L} z^{n+1} \\ &= \beta^n\ Q^n + R^{n+1} + (\mathcal{L} - H) z^{n+1}. \end{aligned}$$

The inner product is defined by

$$(u, v) = \frac{1}{n^i} \sum_{(x_k, y_l) \in\ \omega_j} u(x_k, y_l)\ v(x_k, y_l)$$

where the summation is extended over a $j$-th sub-domain with the total number of nodes $n^j = N_x\ n^j_y$ (here $N_x$ and $N_y = \sum_{j=1}^{S} h^j_y$ are the numbers of collocation points in $x$ and $y$ directions respectively).

The solution of the boundary-value problems, marked by $\triangleright$, is the cardinal stage of this algorithm. Since in each sub-domain $\omega_j$ the coefficients of Eq.(A.4) are constant, the Fourier transform in the periodic direction $x$ results in a collection of uncoupled equations for the Fourier coefficients, to be solved at each iteration step:

$$\hat{H}_j \hat{u}_k = \hat{u}_k'' + \bar{c}_j \hat{u}_k' - (\bar{d}_j + \bar{b}_j k^2)\hat{u}_k = \hat{R}_k(y), \qquad k = -N_x/2, ..., N_x/2 \ j = 1, \ldots, S. \quad (A.5)$$

The solution of 1-D problems (Eq. (A.5)) in the $y$ direction is performed by using the multidomain LFB technique as described in section 2.

## References


[1] D. Gottlieb, Issues in the application of high order schemes, ....

[2] D. Gottlieb, E. Tadmor, The CFL condition for spectral approxinmations to hyperbolic initial-boundary value problems. *ICASE* Report No. 90-42, June 1990.

[3] M. Israeli, L. Vozovoi, A. Averbuch, Spectral multi-domain technique with Local Fourier Basis, *J. Scientific Computing*, **8**, N 2, p. 135-149, (1993).

[4] L. Vozovoi, M. Israeli, A. Averbuch, Spectral multi-domain technique with Local Fourier Basis II: decomposition into cells (to appear in *J. Scientific. Computing.*).

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

[6] M. Israeli, L. Vozovoi, A. Averbuch, Domain decomposition methods for solving parabolic PDE's on multiprocessors, *Applied Numerical. Mathematics*, **12**, N 1-3, (1993).

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

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

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

[10] S. A. Orszag, Spectral methods for problems in Complex Geometries, *Numerical methods for PDE's* .....

[11] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, *Series in Computational Physics*, Springer-Verlag, (1989)