Parallel Spectral Fourier Algorithm for Fluid Dynamics Problems

L. Vozovoi* M. Israeli* A. Averbuch†

Abstract

We present a high-order parallel algorithm which requires only the minimum inter-processor communication dictated by the physical nature of the problem at hand. This algorithm is applied to the incompressible Navier-Stokes equations. The parallelization is achieved by domain decomposition.

A novel feature of the present approach is that the spatial discretization in subdomains is performed using the Fourier method. To avoid the Gibbs phenomenon, the global functions are decomposed into smooth local pieces. Then the Fourier method is applied on extended local subdomains with spectral accuracy.

The continuity conditions on the interfaces are enforced by adding the homogeneous solutions. Such solutions often have fast decay properties which can be utilized to minimize interprocessor communication. In effect, an overwhelming part of the computation is performed independently in subdomains (processors) or using only local communication.

The present method allows the treatment of problems in various complex geometries by the mapping of curvilinear domains into simpler (rectangular or circular) regions. The operator with non constant coefficients, obtained in the transformed domain, is preconditioned by a (piece-wise) constant coefficient operator, easily inverted in the transformed domain. The problem is then solved with spectral accuracy by (a rapidly convergent) conjugate gradient iteration. The capability of this algorithm is illustrated with results from two problems: a direct numerical simulation of turbulence in a two-dimensional periodic domain and a computation of convective motion in a vertical channel with wavy boundaries.

Key words: Local Fourier Basis, spectral preconditioner, complex geometries, turbulence simulation.

AMS subject classifications: 65P05, 35P05, 35K25.

*Faculty of Computer Science, Technion, Haifa 32000, Israel

†School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel

ICOSAHOM'95: Proceedings of the Third International Conference on Spectral and High Order Methods.

1 Introduction

Direct numerical simulation of turbulent flows at high Reynolds number $Re \sim 10^6 - 10^9$ requires considerable computational resources due to the extreme space and time resolution (the computational requirements scales like Re^3). High order methods, in particular spectral methods, are preferred for turbulence computation since they converge rapidly as the number of degrees of freedom grows. As a consequence, for moderate to high accuracy a smaller number of degrees of freedom may be needed, especially in multi-dimensional problems. It has been also proven that high order methods are beneficial for the long term integration necessary to attain the asymptotic turbulent regime.

Efficient large scale scientific computing can be realized only on massively parallel computers. However, the global nature of spectral methods makes these methods difficult to parallelize. Current parallel spectral algorithms require massive data transfers (for example shuffle based transpose in parallel Fourier algorithms) and global communication with the concomitant communication and synchronization bottlenecks.

In [1, 2, 3] a low communication multi-domain approach is developed which uses the Fourier method for the space discretization in subdomains. Since the truncated Fourier series of a non-periodic function exhibits $O(1)$ spurious oscillations near the boundaries and converges slowly inside the region (the Gibbs phenomenon), the direct application of this method to non-periodic local problems would result in a low order algorithm. To avoid the Gibbs phenomenon, the decomposition of functions into smooth local pieces is performed using a collection of overlapping window functions.

The particular local solutions, being constructed independently (with some "convenient" boundary conditions) have jumps on the interfaces. In the correction step these jumps are removed with the aid of properly weighted homogeneous solutions. The "localization" property of the

elliptic operator dictates that these solutions decay rapidly away from the interfaces. This permits the matching of the local solutions at each interface independently of jumps on the other interfaces. In effect the local communication between the neighboring processors (subdomains) is mostly required.

The above Multidomain Local Fourier (MDLF) method was applied efficiently to problems in periodic rectangular domains decomposed into strips or cells. In these cases the solution of a two-dimensional problem is reduced to the solution a collection of uncoupled one-dimensional problems for the Fourier coefficients with the local matching on the interfaces. The computational algorithm is outlined in section 2. In this paper we present new results obtained by the MDLF method for the computation of two-dimensional turbulence in periodic rectangular domains (section 3).

We also develop the previous MDLF approach for treating irregular geometries (section 4). In the generalized algorithm the irregular region is transformed first into the rectangle. Then the resultant differential equations with non-constant coefficients are solved by a preconditioned iteration method. The piece-wise constant coefficient operator is used in the capacity of the preconditioner. The solution of the corresponding constant coefficient problems in each subdomain is performed using the MDLF method. The method is illustrated by solving the problem of natural convection in a vertical channel with sinusoidal boundaries, heated from the side (section 5).

2 Multidomain local Fourier algorithm

- **Formulation of the problem**

An important application of the present Multidomain Local Fourier (MDLF) method is the solution of the incompressible Navier-Stokes equations.

Governing Equations:

$$(1)\quad \frac{\partial \mathbf{v}}{\partial t} = Re^{-1}\nabla^2\mathbf{v} + \mathbf{N}(\mathbf{v}) - \nabla\Pi + \mathbf{F} \quad \text{in } \Omega.$$

Here $\Omega \subset R^d, d = 2, 3$, $\mathbf{v}(\mathbf{x}, t) = (u, v, w)$ is the velocity, subject to the incompressibility constraint

$$(2)\qquad \nabla \cdot \mathbf{v} = 0 \qquad \text{in } \Omega,$$

Π is the total pressure, $\mathbf{F}$ is the forcing term, and Re is the Reynolds number. The nonlinear term is written in the rotational form

$$(3)\qquad \mathbf{N}(\mathbf{v}) = \mathbf{v} \times (\nabla \times \mathbf{v}).$$

Computational Domain and Boundary Conditions We consider first the rectangular domain $\Omega = \{0 \le x < 2\pi, 0 \le y < 2\pi\}$ with periodic boundary conditions in both directions. The case of irregular domains with Dirichlet boundary conditions is addressed below.

The numerical solution of the problem (1)-(3) with specified boundary conditions requires discretization in both time and space.

- **Discretization in time**

The discretization in time is performed via the 3d-order splitting algorithm of [4]:

$$(4)\quad \frac{\hat{\mathbf{v}} - \sum_{q=0}^{2} \alpha_q \mathbf{v}^{n-q}}{\Delta t} = \sum_{q=0}^{2} \beta_q \mathbf{N}(\mathbf{v}^{n-q}),$$

$$(5)\quad \frac{\hat{\hat{\mathbf{v}}} - \hat{\mathbf{v}}}{\Delta t} = -\nabla\Pi^{n+1}, \quad \nabla^2\Pi^{n+1} = \frac{1}{\Delta t}\nabla \cdot \hat{\mathbf{v}},$$

$$(6)\qquad \frac{\gamma_0 \mathbf{v}^{n+1} - \hat{\hat{\mathbf{v}}}}{\Delta t} = Re^{-1}\nabla^2\mathbf{v}^{n+1}.$$

It consists of an explicit advection step (4), a global pressure adjustment for incompressibility (5) and an implicit viscous step (6).

Semi-implicit schemes of this type are associated with much less severe restriction on the time step, $O(N^{-1})$, than fully explicit schemes, $O(N^{-2})$, where $2\pi/N$ is the mesh size.

As a result of the splitting procedure, a time-dependent problem is reduced to the solution (for each time step) of a sequence of two types canonical elliptic equations:
the modified Helmholtz equation

$$(7)\qquad \nabla^2 u - \lambda^2 u = f(x, y),$$

and of the Poisson equation

$$(8)\qquad \nabla^2 u = g(x, y),$$

The parameter λ in (7) is related to the time-stepping increment, $\lambda \propto 1/\sqrt{Re^{-1}\,\Delta t}$.

- **Domain decomposition. Discretization in space.**

First we describe briefly the algorithm as applied to Eq.(7) in 1-D, then we show the extension onto 2-D case. The detailed description can be found in [1, 2, 3].

Algorithm in 1-D

1. Divide the computational interval $[0, L]$ into P subdomains, $l = L/P$, and discretize the local problems on uniform grids of collocation points (with different resolution in each subdomain, if desired).
2. Decompose the source function $f(x)$ into smooth local pieces $f^{(n)}(x)$ using a collection of overlapping window functions. Each of these functions is equal to one inside a corresponding subdomain and smoothly decay outwards over the distance 2ϵ from both sides.
3. Integrate the local problems on extended intervals $l + 2\epsilon$ with smooth source functions in the right hand side. The values of the computed particular solutions $u_p^{(n)}$ will have jumps through the interfaces.
4. Correct solutions in each subdomain in order to enforce the continuity of $u(x)$ and $u'(x)$ at the interfaces (then the higher derivatives will be matched automatically due to Eq. (7) It can be done by adding the properly weighted homogeneous solutions

$$(9) \qquad u = \bigcup_{n=1}^{P} u^{(n)},$$
$$u^{(n)} = u_p^{(n)} + A_n h_+^{(n)} + B_n h_-^{(n)}.$$

These solutions are two exponential functions $h_-^{(n)} = e^{-\lambda x}$, $h_+^{(n)} = e^{-\lambda(l_n - x)}$, $0 < x < l_n$, which decay into a subdomain. Since $\lambda \gg 1$ for small enough time step, the influence of remote interfaces on the coefficients A_n, B_n for each particular n is negligible. Therefore, these coefficients can be found in terms of jumps at one corresponding interface. This requires only the local communication between neighboring subdomains.

The error caused by the use of an approximate local matching instead of the exact global matching is $\varepsilon = O(e^{-\lambda l})$, *independent* of the resolution N. For example, for $P = 4$, $l = 2\pi/P \approx 1.6$, $Re = 1$ and $\Delta t = 0.04$ we have $\lambda \approx 10$. The error will be then $\varepsilon \approx 10^{-7}$ which is quite satisfactory in most applications. Note that the above restriction on the size Δt due to the local matching might be less severe than that imposed by the use of an explicit scheme in time. In the last case the stability constraint depends on the grid size like $\Delta t Re^{-1} \leq h^2/2$ where $h = 2\pi/N$. For a realistic resolution, e.g. $N = 128$, and the Reynolds number $Re = 1$ we have $h = 2\pi/N \approx 0.05$, $\Delta t \leq 0.0013$. In such a case the use of a semi-implicit scheme, combined with the local matching procedure, is still advantageous over explicit schemes.

Algorithm in 2-D. Decomposition into Strips

1. Decompose the computational rectangular region into parallel strips: $\Omega = \cup_{n=1}^{P} \omega_n$.
2. Apply the FFT in the periodic direction y (along the strips) to obtain a collection of *uncoupled* 1-D ODEs for the Fourier coefficients $\hat{u}_k(x)$:

$$(10) \quad \frac{d^2 \hat{u}_k^{(n)}}{dx^2} - \lambda_k^2 \hat{u}_k^{(n)} = \hat{f}_k^{(n)}(x) \qquad \text{in } \omega_n,$$

where $k = -\frac{N_y}{2}, ..., \frac{N_y}{2}$, $\lambda_k^2 = \lambda^2 + k^2$ for the modified Helmholtz equation and $\lambda_k^2 = k^2$ for the Poisson equation.

3. Solve the problems (10) by the previous 1-D algorithm. The modified Helmholtz equation is solved using the local matching procedure on the interfaces. In the case of the Poisson equation, $\lambda = 0$, $\lambda_k^2 = k^2$, the global matching is required only for the long waves, $k \leq k_*$, whereas the short waves, $k \geq k_*$, can be treated by using local matching on the interfaces. The cut-off wave number k_* should be chosen in accordance with the prescribed accuracy.

In order to access the potential of this approach we present the following numerical tests. We solve the 2-D Poisson equation with the source function $f(x, y)$ corresponding to the solution $u_{ex} = \cos ky \sum_{m=1}^{M} \cos(mx + \alpha_m)/m$ where the phases α_m are chosen randomly. Table 1 gives the maximum relative error $\varepsilon_k = |u - u_{ex}|_{max}/u_{max}$ for several numbers of globally matched harmonics, k (the error of the "full" global matching in this case is approximately $\varepsilon \sim 10^{-12}$).

k	3	5	7	9
ε_k	5.4 (-3)	5.7 (-4)	6.0 (-5)	6.3 (-6)

Table 1. The maximum numerical error when using the global matching for the first k harmonics; $P = 8, N_x = 64 \times P, N_y = 64$

We can see that using the global matching, for example, only for the first 9 long waves, while

treating locally the rest 21 shorter waves, we guarantee the accuracy $\varepsilon_k \leq 10^{-5}$. The relative amount of globally matched waves, $2k^*/N_y$, becomes smaller as the resolution in the periodic direction increases. Note that the accuracy of the local matching relies on the localization properties of the boundary Green's functions of the Helmholtz operator. Therefore, the results in Table 1 depend only slightly on the particular form of the source function $f(x, y)$.

Decomposition into Cells.

The algorithm in this case is more complicated than the previous one. For details see [3].

- **Parallel test** .

  The parallel test is performed on a collection of Digital Alpha workstations (Alpha-farm), 128 MB/unit, with Giga-switch and PVM software package. The computational domain is a 3-D periodic box, decomposed into "slabs" ($P = 2-10$). For the resolution 128^3 the effect of using the local-global matching, when solving the Poisson equation, results in 2-3 times reduction of the executable time as compared to the "fully" global matching.

  When solving the complete Navier-Stokes Eqs. (1)-(3) the implicit viscous step is always computed using the local matching. The use of the global-local matching in the pressure step gives about $15 - 20\%$ saving in the CPU time as compared to the global matching. This effect is expected to be even more pronounced for a larger number of processors. Detailed discussion of the parallel performance is given in [9].

- **Test of accuracy** .

  The following example testifies the accuracy of the MDLF algorithm. We consider the exact solution of Eqs. (1)-(3)

$$v_x = \frac{\partial \psi}{\partial y}, v_y = -\frac{\partial \psi}{\partial x},$$

$$\psi(x, y) = g(t) e^{(\cos \omega x)(\cos \omega y)}$$

$$g(t) = \frac{1}{40}\left(\frac{1}{10} e^{(\sin\ 2t - 3 \sin\ 2\pi t)} + \cos(2\sqrt{2}t)\right) + c$$

  with $Re = 10^2, \omega = 5,\ c = 0.075$ and the forcing function $f(x, y, t)$ computed accordingly (this solution was suggested in [5] to demonstrate the capability of the multilevel Nonlinear Galerkin method for the long-time numerical integration).

The evolution of the maximum relative error in time is shown in Fig. 1; $dt = 5 \cdot 10^{-3}$ and $\cdot 10^{-4}$, the strip decomposition with $P = 4$ is applied, the space resolution is 64×64 . The error remains small on a large time scale and it decreases as the size of the time step Δt becomes smaller (the spatial error for this example is $O(10^{-11})$ can be found with $g(t) = \text{const}\ (t)$).

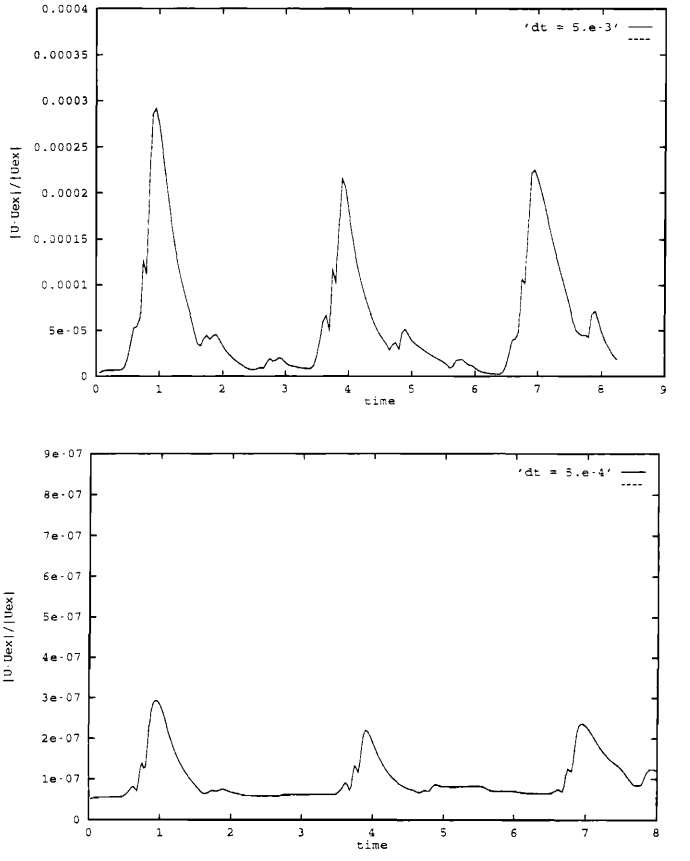

Figure 1: Evolution of the relative maximum error in time for $\Delta t = 5 \cdot 10^{-3}$ (upper plot) and $\Delta t = 5 \cdot 10^{-4}$ (lower plot).

3 Parallel simulation of 2-D turbulence

In this section we present new results on numerical simulation of decaying turbulence (without forcing term) in the periodic box using the above algorithm. The initial velocity field in the Fourier domain is chosen as in [6]:

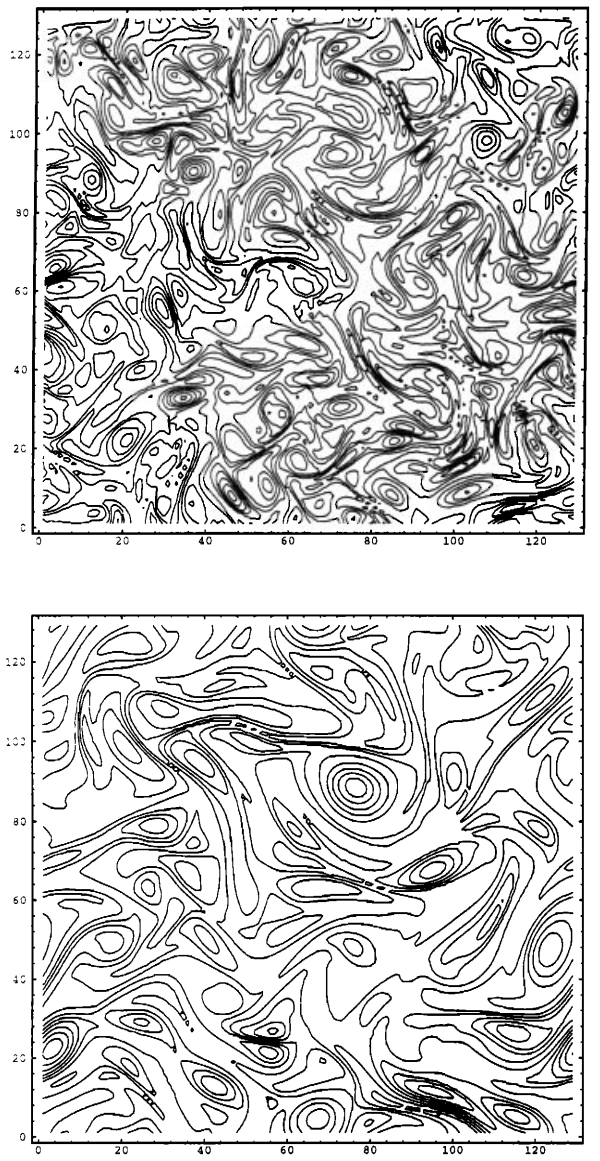

Figure 2. Contour plots of vorticity at $t = 20$ (upper plot) and $t = 100$ (lower plot).

$$\hat{\psi}(k) = |\hat{\psi}(k)|e^{i\theta(k)}$$

$$|\hat{\psi}(k)| = \begin{cases} \frac{C}{k\sqrt{1+(k/6)^4}}, & |k| < 20.5 \\ 0, & |k| \geq 20.5 \end{cases}$$

where ψ is the stream function defined by

$$\mathbf{v} = \{u, v, 0\}, \quad u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x},$$

The phase $\theta(k)$ is chosen randomly on the interval $[0, 2\pi]$.

The enstrophy spectra are plotted in Fig.3 for $t = 20$ and 100. Naturally, the small scales dissipate faster than the large scales.

The computed solution (the contour plot of vorticity $\zeta = -\Delta\psi$) is shown in Fig. 2 at $t = 20$ and $t = 100$; $\Delta t = 0.01$, $Re = 10^4$, the resolution 128×128, the number of processors $P = 4$. Practically the same pictures are obtained using the resolution 256×256 and the time steps in the range $\Delta t = 10^{-4} - 10^{-1}$.

These plots are very similar (regarding the flow structure, the sizes of the eddies) to those obtained in [6] using a pseudo-spectral Fourier method in a single domain (they are not expected to be identical since the phase $\theta(k)$ is random field). Unfortunately, the above cited paper does not contain the energy spectrum plots to compare with. At least one conclusion can be made at this point: the MDLF algorithm allows a stable time integration at high Reynolds numbers with adequate resolution of small features in the flow. The accuracy of this algorithm was illustrated independently by the example in the previous section.

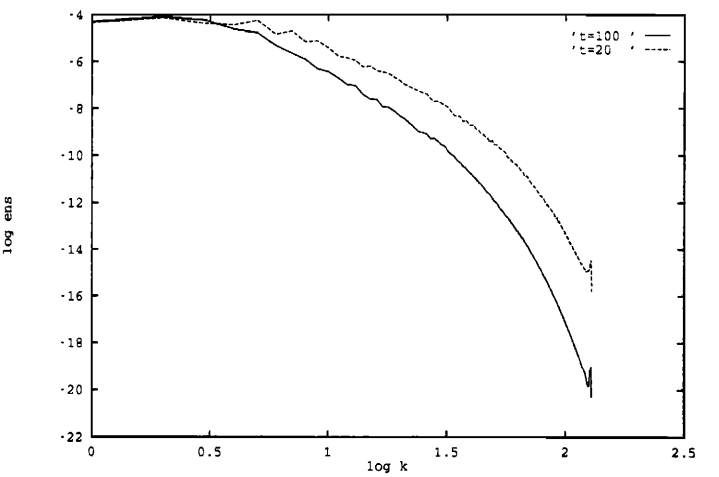

Figure 3. The enstrophy spectra at $t = 20$ and $t = 100$; the resolution is 256×256, $P = 4$.

4 Problems in irregular geometries

Our approach for treating irregular domains has much in common with that of [7]. The idea is to map a complex geometry onto a simpler one and then solve the resultant non-constant coefficient equations by a preconditioned iteration method. Usually a low order (finite-difference or finite-element) approximation is chosen as a preconditioner. We propose to use in the capacity of the preconditioner a constant coefficient operator which has the same structure as the original one. These constant coefficients can be computed as the mean values of the corresponding variable coefficients. Then the above spectral MDLF method is applied on each iteration step to solve constant coefficient problems in a simple geometry. We shall call this constant coefficient operator a *spectral preconditioner.*

The spectral preconditioner gives a better approximation to a non-constant coefficient operator than any low order approximation provided that the variance is small. If the coefficients vary in a large range, then such an approximation is not satisfactory and thus the iteration method would converge slowly. However, in combination with the domain decomposition strategy the spectral preconditioner becomes very efficient. Since the computation in subdomains is performed independently, we can use a step-wise approximation to variable coefficients by taking different mean values in each subdomain. On each patch of the domain this approximation will be accurate if the number of subdomains is large enough.

We illustrate this approach by the following simplest example:

$$\mathcal{L}(r)u = u'' + a(x)u' - \lambda^2 u = f(x), \qquad x \in [0,1] \tag{11}$$

$$u(0) = u(1) = 0, \tag{12}$$

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 $u_{ex}(x) = x(x-1)$.

Instead of solving the problem (11), (12) directly we rather solve it iteratively using a preconditioned iteration method. The most simple Richardson scheme reads:

$$Hu^{n+1} = f - (\mathcal{L} - H)u^n. \tag{13}$$

(more sophisticated iteration schemes, such as conjugate gradient types method, can be used as well). If the operator $\mathcal{L}$ is approximated in a spectral basis, upon convergence of iterations (13) a spectrally accurate solution is obtained. The rate of convergence depends on the choice of a preconditioner H.

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:

$$H_n = u'' + \bar{a}_n u' - \lambda^2 u, \qquad x \in [x_{n-1}, x_n], \tag{14}$$

where

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

Fig. 4 shows the coefficient $a(x)$ on the interval $x \in [0,1]$ and its step-wise approximation $\bar{a}_n$ for $P = 15$.

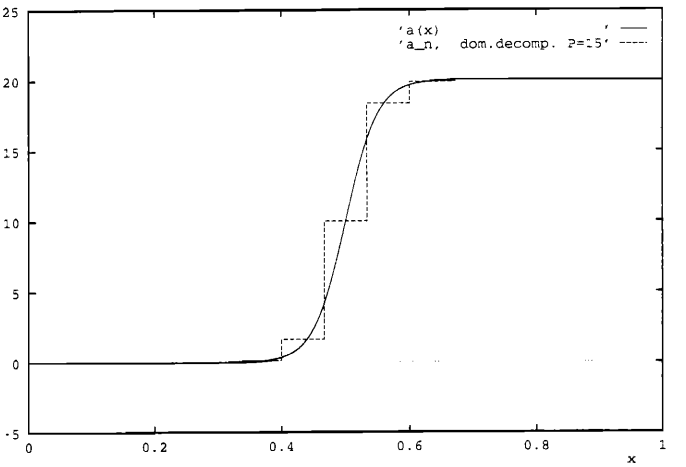

Figure 4. Step-wise approximation of the variable coefficient $a(x) = 10\ (1 + \tan\ 20(x - 0.5))$.

The convergence history of the preconditioned conjugate gradient iterations with the spectral preconditioner (14) is given in Fig.5 for several domain decompositions (parameter $\lambda = 1$). When the number of subdomains is not large enough, $P = 9$, the convergence is rather slow. However, the convergence rate improves substantially for larger P as the local deviations $|a(x) - a_n|$ within the intervals $[x_{n-1}, x_n]$ become smaller.

Note that the computation of inner products, needed when using conjugate-gradient type methods, leads to additional global communications which degrade somewhat the parallel performance.

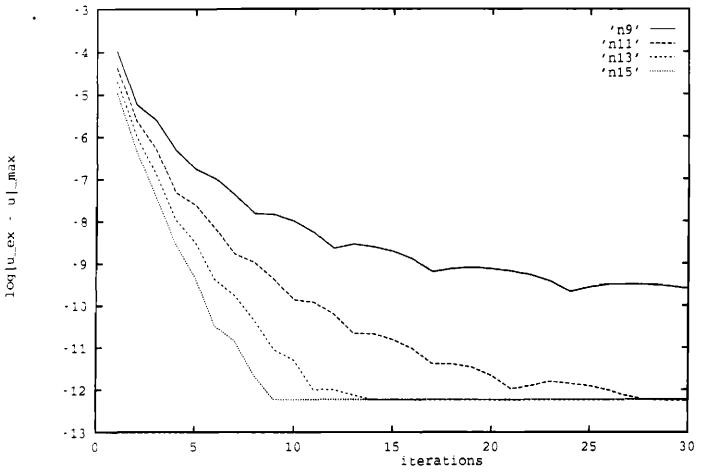

Figure 5. Acceleration of convergence by splitting the computational region into an increasing number of subdomains P.

5 Natural convection in channels with wavy boundaries

As a 2-D example we consider the problem of natural convection in a vertical channel with sinusoidal boundaries:

$$(15)\quad y_1(y) = -(1 + \delta \cos\ k_w x), \quad y_2(y) = 1 + \delta \cos\ k_w x.$$

(this problem was investigated earlier in [8] using a finite-difference method). The computational domain $\Omega = \{0 < x < L, y_1 < y < y_2\}$, where $L = 2\pi/k_*$, $k_* = 1.4$.

The governing equations are written in the $\psi - \phi$ formulation:

$$(16)\quad \frac{\partial \phi}{\partial t} + \frac{\partial \psi}{\partial x}\frac{\partial \phi}{\partial y} - \frac{\partial \phi}{\partial x}\frac{\partial \psi}{\partial y} = \nabla^2 \phi + Gr\frac{\partial T}{\partial y},$$

$$(17)\quad \nabla^2 \psi = -\phi$$

$$(18)\quad \frac{\partial T}{\partial t} + \frac{\partial \psi}{\partial x}\frac{\partial T}{\partial y} - \frac{\partial T}{\partial x}\frac{\partial \psi}{\partial y} = Pr^{-1}\Delta T.$$

Here ψ, ϕ, T are correspondingly stream function, vorticity and temperature; Gr, Pr are the Grashof and Prandtl numbers.

Functions ψ and T are subject to the following boundary conditions:

$$(19)\quad y = y_{1,2}, \quad \psi = \psi_\nu = 0, \quad T = \mp 1,$$

$$(20)\quad \psi(0, y) = \psi(L, y), \qquad T(0, y) = T(L, y).$$

where ψ_ν denotes the normal derivative of ψ on the boundaries.

By using a simple stretching transformation

$$(21)\quad \xi = x, \qquad \eta = \frac{y - y_1(x)}{y_2(x) - y_1(x)}$$

we map a curvilinear region Ω onto the rectangle $\omega = \{0 \le \xi \le L,\ 0 \le \eta \le 1\}$.

The discretization in time is performed via an implicit stiffly stable scheme (4)- (6) A preconditioned conjugate method with the spectral preconditioner is employed for the solution of the system (7), (8) as described in previous sections.

The computed solution (contour plot of the stream function) on the asymptotic stage is shown in Fig.6 for $Gr = 400, Pr = 0, \delta = 0.1$ and $k_w = k_*$. This wave number corresponds to the most unstable periodic motion (critical eigenmode) in the plain vertical layer. Therefore, the perturbation on the boundaries with this wavenumber leads to the resonant amplification of the critical eigenmode.

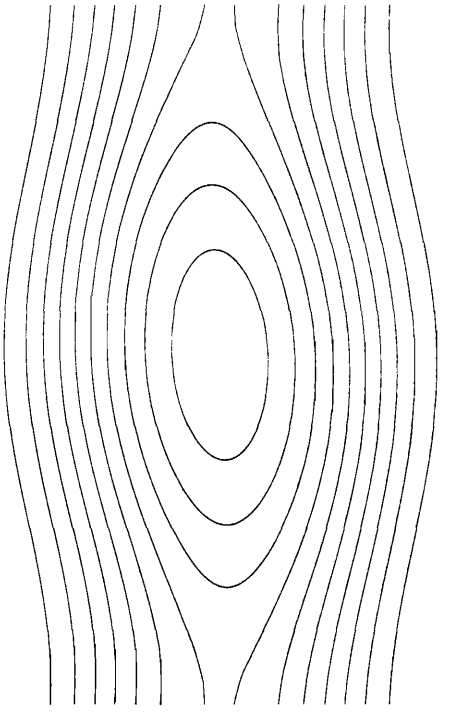

Figure 6. Contour plot of the stream function for $k_w = k_*$; $Gr = 400, Pr = 0$.

The steady solution with $k_w = 2k_*$ ("even" mode) is plotted in Fig.7; $Gr = 300, Pr = 0, \delta = 0.1$. The external forcing has the spatial frequency twice as high as the critical number k_*. In this case some resonance phenomenon of the parametric resonance type can be observed, such as the competition between "even" and "odd" eigenmodes, the reduction of the stability threshold and more.

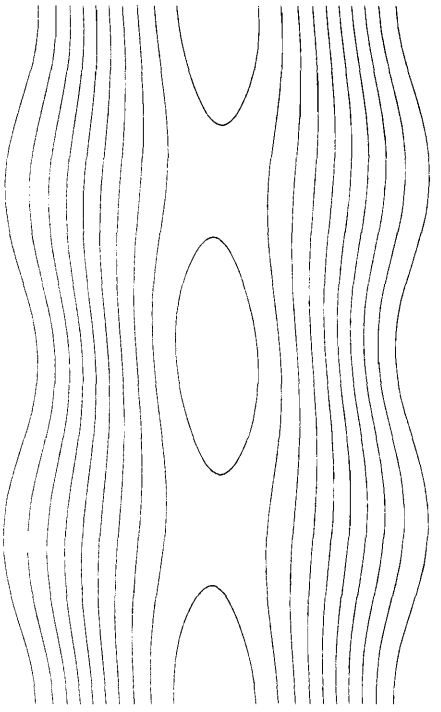

Figure 7. Contour plot of the stream function for $k_w = 2k_*$; $Gr = 300, Pr = 0$.

References

[1] M. Israeli, L. Vozovoi, A. Averbuch, *Spectral multi-domain technique with Local Fourier Basis*, J.Scient.Comput., **8**, (1993), 181–195.

[2] M. Israeli, L. Vozovoi, A. Averbuch, *Parallelizing implicit algorithms for time-dependent problems by parabolic domain decomposition*, J.Scient.Comput., **8**, (1993), 197–212.

[3] L. Vozovoi, M. Israeli, A. Averbuch, *Spectral multi-domain technique with Local Fourier Basis II: decomposition into cells*, SIAM, J.Scient.Comp., **9**, N 3, (1994), pp.311-326.

[4] G. E. Karniadakis, M. Israeli, S. A. Orszag, *High order splitting methods for the incompressible Navier-Stokes Equations*, J.Comput.Phys., **97**, (1991), 414–443.

[5] F.Jaubertau, C.Roisier, R.Temam, *The Nonlinear Galerkin method in computational fluid dynamics*, Applied Numer. Mathem., **6**, (1989/90), p. 361-370.

[6] G.L.Browning and H.-O.Kreiss, *Comparison of Numerical Methods for the calculation of Two-Dimensional Turbulence*, Mathematics of Computation, **52**, No. 186, p.369-388, (1989).

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

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

[9] A. Averbuch, K. Rubinsky, M. Israeli, L. Vozovoi, *Parallel implementation of 2-D and 3-D Navier-Stokes equations on MIMD multiprocessors*, submitted to J. Supercomputing.