

Some Applications of Adaptive Domain Decomposition Methods in Fluid Dynamics

Claudio Carlenzoli *

Abstract

Advection-diffusion equations are often used as kernels for the simulation of various kinds of problems. When dealing with Navier-Stokes problems for instance the ratio of the advective part to the diffusive one is represented by the flow Reynolds number. Domain decomposition methods based on Dirichlet/Neumann iterations are effective only when the diffusive part is dominant, whereas if the convective part becomes more relevant the natural interface conditions may produce instabilities.

Moving from these considerations we apply the adaptive methods ADN and ARN for the simulation of problems which develop internal or boundary layers holding the smoothness of the numerical solution and keeping a very effective rate of convergence.

Key words: boundary value problems, spectral methods, domain decomposition procedures, convective dominated problems, nonlinear equations, incompressible Navier-Stokes equation.

AMS subject classifications: 65M55, 65M70, 65N35, 65N55.

1 Introduction

In this paper we are interested in some applications of the adaptive schemes introduced in [5]. These schemes were proposed to solve Advection-Diffusion (AD) equations in the framework of nonoverlapping multidomain partitions.

These equations could be the numerical kernel of various kinds of problems. In particular, in this paper we

will investigate the simulation of incompressible Navier-Stokes equations in case of high Reynolds number. Since the diffusive part of the equation depends on this number, whereas the convective part is associated to the velocity of the flow, we may therefore have to treat AD equations that are dominated by advection.

It is well known that in the domain decomposition framework the classical Dirichlet/Neumann method can perform very poorly when equations are dominated by convection. These instabilities are due to treatments of interfaces which although being mathematically correct, are inconsistent with the hyperbolic limit of the advection-diffusion equation.

In the following we give a short outline of the paper.

In Section 2 we illustrate the advection-diffusion equation and its weak formulation with several choices of boundary conditions.

In Section 3 we recall the classical Dirichlet/Neumann method and the adaptive methods proposed in [5] called ADN and ARN methods. Moreover we extend the analysis to the case of subdomain partitions with internal cross points (i.e. a common point of four subdomains)

In Section 4 we illustrate the behavior of the adaptive schemes when dealing with vector fields which form different angles of incidence with the interface. Simulations of physical problems with boundary layer are shown using different subdomain partitions.

In Section 5 we investigate a nonlinear time dependent extension of the AD equation. We use the ARN method at each time level and we show that the numerical solution holds its smoothness although it develops an internal layer during the time evolution.

Finally, in Section 6 we consider the approximation of the incompressible Navier Stokes equation. We make use of a projection method introduced by Chorin in [6] and Temam in [18]. We show, as numerical test, a simulation of the so called *driven cavity*.

All the numerical experiments refer to a discrete approximation by spectral collocation methods.

*Dipartimento di Matematica, Politecnico di Milano.

2 The differential equation

2.1 The advection-diffusion boundary value problem

We consider the boundary-value problem:

$$(1) \begin{cases} L_\varepsilon u \equiv -\varepsilon \Delta u + \mathbf{b} \cdot \nabla u + a u = f & \text{in } \Omega \\ u = g & \text{on } \partial\Omega \end{cases}$$

where $a \geq 0$, Ω is a two dimensional domain with boundary $\partial\Omega$, \mathbf{b} , g and f are given functions.

Let us consider a partition of Ω by M non intersecting open subdomains Ω_i $i=1, \dots, M$ and denote by $\Gamma_{i,j} = \partial\Omega_i \cap \partial\Omega_j$ $i, j=1, \dots, M$ the common boundary between Ω_i and Ω_j (see Fig. 1).

Under these assumptions, problem (1) can be reformulated as follows:

Find $\{u_i \equiv u|_{\Omega_i} \ i=1, \dots, M$ such that, setting $\{f_i \equiv f|_{\Omega_i}\}$:

$$(2) \begin{cases} L_\varepsilon u_i = f_i & \text{in } \Omega_i \\ u_i = g & \text{on } \partial\Omega \cap \partial\Omega_i \\ u_i = u_j & \text{on } \Gamma_{i,j} \\ \Psi_i^{(\delta)}(u_i) + \Psi_j^{(\delta)}(u_j) = 0 & \text{on } \Gamma_{i,j} \end{cases}$$

where

$$\Psi_i^{(\delta)}(u_i) \equiv \varepsilon \frac{\partial u_i}{\partial \mathbf{n}_i} - \delta \mathbf{b} \cdot \mathbf{n}_i u \quad \delta = 0 \text{ or } 1,$$

\mathbf{n}_i is the outward normal unit vector to $\partial\Omega_i$ and $\frac{\partial}{\partial \mathbf{n}_i}$ denotes the normal derivative on $\partial\Omega_i$.

Both choices of $\delta = 0$ and $\delta = 1$ are suitable.

2.2 Variational multidomain formulations and interface conditions

To start with, let us define

$$H^1(\Omega) = \left\{ u | u \in L^2(\Omega) : \nabla u \in (L^2(\Omega))^2 \right\}$$

Moreover, let us set

$$H_0^1(\Omega) = \{ u | u \in H^1(\Omega) : u = 0 \text{ on } \partial\Omega \}$$

and

$$H_g^1(\Omega) = \{ u | u \in H^1(\Omega) : u = g \text{ on } \partial\Omega \}$$

(see [10]).

Let us consider problem (1) in case of $\delta=0$.

The weak form of (1) (which is formally obtained multiplying the first equation by a test function $v \in H^1(\Omega)$ and using the Green formula) is:

Find $u \in H_g^1(\Omega)$:

$$(3) \quad a^0(u, v) \equiv \int_{\Omega} [\varepsilon \nabla u \cdot \nabla v + (\mathbf{b} \cdot \nabla u) v + a uv] d\Omega = \int_{\Omega} f v d\Omega \quad \forall v \in H_0^1(\Omega)$$

Now, if we set:

$$(4) \quad a_i^0(u, v) \equiv \int_{\Omega_i} [\varepsilon \nabla u \cdot \nabla v + (\mathbf{b} \cdot \nabla u) v + a uv] d\Omega_i$$

is readily seen that (3) is equivalent to the following multidomain problem:

Find $\{u_i \equiv u|_{\Omega_i} \ i=1, \dots, M$ such that, setting $\{f_i \equiv f|_{\Omega_i}\}$:

$$(5) \quad \begin{cases} a_i^0(u_i, v) = \int_{\Omega_i} f_i v d\Omega_i & \forall v \in H_0^1(\Omega_i) \\ u_i = g & \text{on } \partial\Omega \cap \partial\Omega_i \\ u_i = u_j & \text{on } \Gamma_{i,j} \end{cases}$$

$$\sum_{l=1}^M a_l^0(u_l, \bar{\varphi}_l) = \sum_{l=1}^M \int_{\Omega_l} f_l \bar{\varphi}_l d\Omega_l \quad \forall \varphi \in \Phi$$

where $i, j = 1, \dots, M$.

Φ is the space of traces on $\Gamma = \bigcup_{i,j=1}^M \Gamma_{i,j}$ of the functions of $H_0^1(\Omega)$ (see [10]), and $\bar{\varphi}_i$ denotes any possible continuous extension of φ to Ω_i (e.g., its harmonic extension).

Note that from (5) we can easily deduce (2). As a matter of fact, counterintegrating by parts the last equation of (5) and using the previous equations, we obtain the flux balance condition:

$$(6) \quad \varepsilon \frac{\partial u}{\partial \mathbf{n}_i} + \varepsilon \frac{\partial u}{\partial \mathbf{n}_j} = 0 \quad \text{on } \Gamma_{i,j}$$

When $\delta=1$, we provide a different weak formulation of (1), which is alternative to (3). The difference stands from the fact that this time we also integrate by parts the convective term, and obtain:

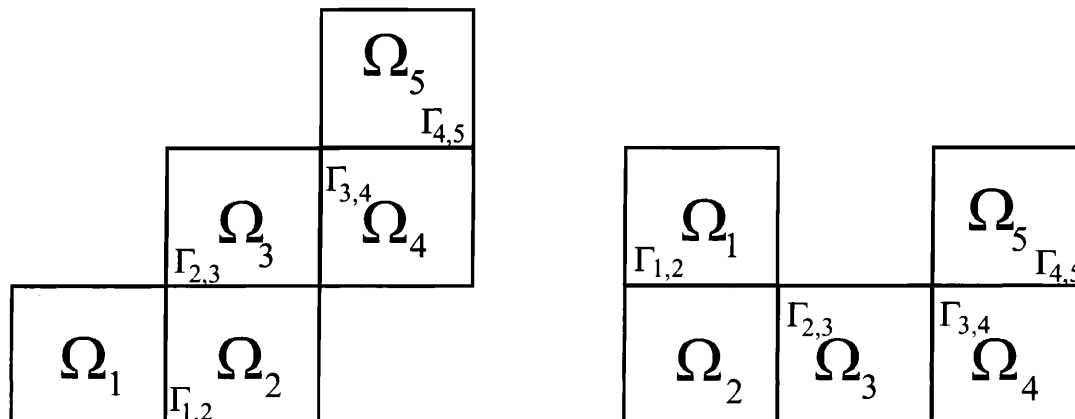


Figure 1: Two examples of computational regions partitioned into non overlapping subdomains (M=5)

Find $u \in H_g^1(\Omega)$:

$$\begin{aligned}
 a^0(u, v) &\equiv \int_{\Omega} [\varepsilon \nabla u \cdot \nabla v - \text{div}(\mathbf{b}v) u + a uv] d\Omega = \\
 (7) \qquad &= \int_{\Omega} f v d\Omega \qquad \forall v \in H_0^1(\Omega)
 \end{aligned}$$

The multidomain formulation of (7) still takes the form (5), provided now $\Psi_i^{(1)}$ is used instead of $\Psi_i^{(0)}$ and $a_i^1(u, v)$ instead of $a_i^0(u, v)$, where:

$$\begin{aligned}
 (8) \qquad a_i^1(u, v) &\equiv \\
 &\int_{\Omega_i} [\varepsilon \nabla u \cdot \nabla v - \text{div}(\mathbf{b}v) u + a uv] d\Omega_i
 \end{aligned}$$

In turns, this is now the weak formulation of (2) and the transmission condition in the current case reads:

$$\begin{aligned}
 (9) \qquad \left(\varepsilon \frac{\partial u}{\partial \mathbf{n}_i} - \mathbf{b} \cdot \mathbf{n}_i u_i \right) + \left(\varepsilon \frac{\partial u}{\partial \mathbf{n}_j} - \mathbf{b} \cdot \mathbf{n}_j u_j \right) &= 0 \\
 &\text{on } \Gamma_{i,j}
 \end{aligned}$$

3 Subdomain iterative methods

3.1 Multidomain methods based on Dirichlet/Neumann iterations

For the sake of simplicity we split the domain Ω into two non overlapping subdomains Ω_1 and Ω_2 and denote by $\Gamma_{1,2}$

their interface. Then the Dirichlet/Neumann subdomain iteration procedure for problem (1) reads:

for a given value λ^1 , we look for a sequence u_i^k with $k \geq 1$ and $i=1,2$ such that:

$$\begin{aligned}
 (10) \qquad &\begin{cases} L_{\varepsilon} u_1^k = f_1 & \text{in } \Omega_1 \\ u_1^k = g & \text{on } \partial\Omega \cap \partial\Omega_1 \\ u_1^k = \lambda^k & \text{on } \Gamma_{1,2} \end{cases} \\
 &\begin{cases} L_{\varepsilon} u_2^k = f_2 & \text{in } \Omega_2 \\ u_2^k = g & \text{on } \partial\Omega \cap \partial\Omega_2 \\ \Psi_2^{(0)}(u_2^k) + \Psi_1^{(0)}(u_1^k) = 0 & \text{on } \Gamma_{1,2} \end{cases}
 \end{aligned}$$

where

$$\lambda^{k+1} = \theta u_2^k + (1 - \theta) \lambda^k \qquad \text{on } \Gamma_{1,2}$$

Choosing a suitable parameter $\theta > 0$ is necessary to achieve convergence.

For the convergence analysis in case of a self adjoint operator see e.g. [1, 8, 11].

When ε tends to zero the effectiveness of the Dirichlet/Neumann method can deteriorate (see [5]). This is due to the fact that the interface conditions can be inconsistent with the hyperbolic limit of the Advection-Diffusion equation. In [5] we have proposed two different approaches to avoid these instabilities.

The first approach (Adaptive Dirichlet Neumann method, ADN for short) is simply based on an adaptive choice of the Dirichlet and Neumann interface conditions according to the direction of the vector field \mathbf{b} .

Indeed, when $\mathbf{b} \cdot \mathbf{n}_{\Gamma}$ is positive on a side of the interface Neumann conditions are there imposed, whereas when

$\mathbf{b} \cdot \mathbf{n}_r$ is negative Dirichlet conditions are in order (here \mathbf{n}_r represents the outward normal unit vector to the interface $\Gamma_{i,j}$).

The second approach (ARN for short) comes directly from the variational formulation (7) of the differential problem (1).

This new approach has been named Adaptive Robin Neumann method as it replaces Dirichlet interface conditions with the Robin ones.

Under the same assumptions used for the Dirichlet/Neumann scheme we can formulate the ARN method as follows:

for a given value λ^1 , μ^1 and u_2^0 we look for a sequence u_i^k , with $k \geq 1$ and $i=1,2$ such that:

$$(11) \quad \begin{cases} L_\varepsilon u_1^k = f_1 & \text{in } \Omega_1 \\ u_1^k = g & \text{on } \partial\Omega \cap \partial\Omega_1 \\ \Psi_1^{(1)}(u_1^k) = -\Psi_2^{(1)}(\lambda^k) & \text{on } \Gamma_1^{in} \\ \Psi_1^{(0)}(u_1^k) = -\Psi_2^{(0)}(u_2^{k-1}) & \text{on } \Gamma_1^{out} \end{cases}$$

$$\begin{cases} L_\varepsilon u_2^k = f_2 & \text{in } \Omega_2 \\ u_2^k = g & \text{on } \partial\Omega \cap \partial\Omega_2 \\ \Psi_2^{(1)}(u_2^k) = -\Psi_1^{(1)}(\mu^k) & \text{on } \Gamma_2^{out} \\ \Psi_2^{(0)}(u_2^k) = -\Psi_1^{(0)}(u_1^k) & \text{on } \Gamma_2^{in} \end{cases}$$

where

$$\lambda^{k+1} = \theta u_2^k + (1 - \theta) \lambda^k \quad \text{on } \Gamma_1^{in}$$

and

$$\mu^{k+1} = \theta u_1^k + (1 - \theta) \mu^k \quad \text{on } \Gamma_2^{out}$$

We have set $\Gamma_1^{in} \equiv \{x \in \Gamma_{1,2} : \mathbf{b} \cdot \mathbf{n}_1 < 0\}$ and $\Gamma_1^{out} \equiv \{x \in \Gamma_{1,2} : \mathbf{b} \cdot \mathbf{n}_1 \geq 0\}$

We remind that the interface conditions required in this approach still impose the continuity of the solution along the interface provided $\mathbf{b} \cdot \mathbf{n}_1 \neq 0$ (and then we can still look for a numerical solution in $H_q^1(\Omega)$).

The formulation (11) differs slightly from the one in [5] for what concerns the relaxation procedure. In this case we apply the procedure on the whole Robin interface operator $\Psi_i^{(1)}$ $i=1,2$,

$$\left(\varepsilon \frac{\partial u_1^n}{\partial \mathbf{n}_1} - \mathbf{b} \cdot \mathbf{n}_1 u_1^n \right) = -\theta \left(\varepsilon \frac{\partial u_2^n}{\partial \mathbf{n}_2} - \mathbf{b} \cdot \mathbf{n}_2 u_2^n \right) +$$

$$(1 - \theta) \left(\varepsilon \frac{\partial u_2^{n-1}}{\partial \mathbf{n}_2} - \mathbf{b} \cdot \mathbf{n}_2 u_2^{n-1} \right) \quad \text{on } \Gamma_{1,2}$$

and not only on the second part

$$\left(\varepsilon \frac{\partial u_1^n}{\partial \mathbf{n}_1} - \mathbf{b} \cdot \mathbf{n}_1 u_1^n \right) = -\varepsilon \frac{\partial u_2^n}{\partial \mathbf{n}_2} + \mathbf{b} \cdot \mathbf{n}_2$$

$$(\theta u_2^n + (1 - \theta) u_2^{n-1}) \quad \text{on } \Gamma_{1,2}.$$

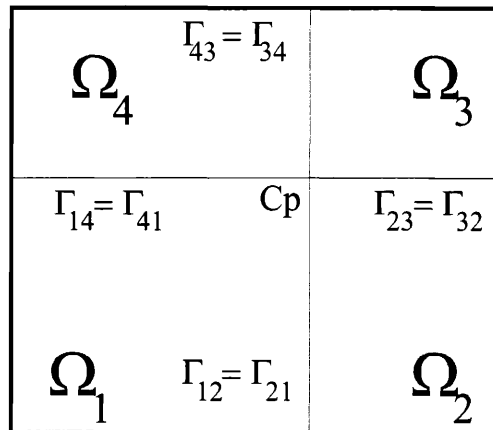


Figure 2: The partition of Ω with an internal cross point

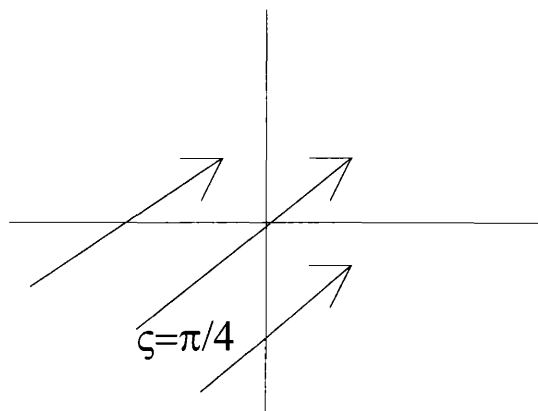


Figure 3: The effect of the vector field $\mathbf{b}^t = (1, 1)$

3.2 Multidomain formulation with “cross points”

In this Section we will illustrate the implementation of the adaptive schemes proposed when the computational domain Ω is partitioned in at least four subregions with a common vertex (the so called *cross point*).

For the sake of simplicity, let us consider a domain Ω partitioned in four subdomains as in Fig. 2.

When using Dirichlet/Neumann schemes the choice of interface conditions for internal cross points is a very crucial problem. In general ones should impose a Dirichlet

condition on each subdomain and then correct this value using a particular equation related to the variational formulation of the differential problem (see [12]).

Using ADN and ARN methods, we would like to avoid the Dirichlet condition on each subdomain and at the same time we would like to use the adaptive conditions even at the internal cross point.

To start with, let us consider the boundary value problem (1) where Ω is a computational domain partitioned as in Fig. 2.

For simplicity let us suppose to deal with $\mathbf{b}^t = (1, 1)$.

Recalling the effects of the vector field \mathbf{b} in the choice of interface conditions in the adaptive schemes ADN and ARN, we have a situation that can be represented as in Fig. 3.

Then the ARN scheme proposed reads as follows:

for given values λ_1^0 , u_2^0 and u_4^0 , we look for a sequence u_i^k , with $k \geq 1$ and $i = 1, 2, 3, 4$ such that:

(12)

$$\begin{cases} L_\varepsilon u_1^k = f_1 & \text{in } \Omega_1 \\ u_1^k = g & \text{on } \partial\Omega \cap \partial\Omega_1 \\ \Psi_{12}^{(0)}(u_1^k) = -\Psi_{21}^{(0)}(u_2^{k-1}) & \text{on } \Gamma_{12} \\ \Psi_{14}^{(0)}(u_1^k) = -\Psi_{41}^{(0)}(u_4^{k-1}) & \text{on } \Gamma_{14} \\ \omega_{12}\Psi_{12}^{(0)}(u_1^k) + \omega_{14}\Psi_{14}^{(0)}(u_1^k) = & \text{on } C_p \\ -\omega_{21}\Psi_{21}^{(0)}(u_2^{k-1}) - \omega_{41}\Psi_{41}^{(0)}(u_4^{k-1}) & \end{cases}$$

$$\begin{cases} L_\varepsilon u_3^k = f_3 & \text{in } \Omega_3 \\ u_3^k = g & \text{on } \partial\Omega \cap \partial\Omega_3 \\ \Psi_{32}^{(1)}(u_3^k) = -\Psi_{23}^{(1)}(u_2^{k-1}) & \text{on } \Gamma_{32} \\ \Psi_{34}^{(1)}(u_3^k) = -\Psi_{43}^{(1)}(u_4^{k-1}) & \text{on } \Gamma_{34} \\ \omega_{32}\Psi_{32}^{(1)}(u_3^k) + \omega_{34}\Psi_{34}^{(1)}(u_3^k) = & \text{on } C_p \\ -\omega_{23}\Psi_{23}^{(1)}(u_2^{k-1}) - \omega_{43}\Psi_{43}^{(1)}(u_4^{k-1}) & \end{cases}$$

$$\begin{cases} L_\varepsilon u_p^k = f_p & \text{in } \Omega_p \\ u_p^k = g & \text{on } \partial\Omega \cap \partial\Omega_p \\ \Psi_{p1}^{(1)}(u_p^k) = -\Psi_{1p}^{(1)}(\lambda_1^k) & \text{on } \Gamma_{p1} \\ \Psi_{p3}^{(0)}(u_p^k) = -\Psi_{3p}^{(0)}(u_3^k) & \text{on } \Gamma_{p3} \\ u_p^k = (\lambda_1^k + u_3^k)/2 & \text{on } C_p \end{cases}$$

where $p = 2, 4$, $\lambda_1^k = \theta u_1^k + (1 - \theta)\lambda_1^{k-1}$ on $\Gamma_{12} \cup \Gamma_{14}$ and $\Psi_{ij}^{(\delta)}(u) = \varepsilon \frac{\partial u}{\partial \mathbf{n}_{ij}} - \delta \mathbf{b} \cdot \mathbf{n}_{ij} u$ with \mathbf{n}_{ij} the unit normal vector from Ω_i to Ω_j .

We remind that the weights ω_{ij} are related to the Gauss Lobatto quadrature weights according to the variational formulation of the differential problem.

The ADN scheme can be obtained in a very easy way from the ARN one substituting Robin conditions with

Dirichlet conditions and taking $\omega_{p3} = \omega_{3p} = 1/2$ with $p = 2, 4$ in the second step of the scheme (12).

For more general vector fields \mathbf{b} the generalization of the scheme is straightforward. The only exceptions are those related to a vector field orthogonal to one edge of the subdomains (i.e. in Fig. 3 $\varsigma = 0, \pi/2, \pi, \dots$). In such a case we could impose the interface conditions as if \mathbf{b} was ingoing (or outgoing) to one subdomain. For instance, if $\mathbf{b}^t = (1, 0)$ (i.e. $\varsigma = \pi/2$) we can arbitrary decide to require the interface conditions as if $\mathbf{b}^t = (1 \pm \nu, 0)$ with $\nu > 0$ (and therefore we can use scheme (12)).

4 Numerical results

In this section we want to study the behavior of the adaptive methods proposed when the vector field \mathbf{b} forms different angles of incidence with the interface Γ . It is well known that this angle influences the effectiveness of several subdomain iterative schemes when dealing with convective dominated equations (see [17, 2]).

We consider a computational domain $\Omega = (-1, 1) \times (0, 1)$ partitioned in two subdomains with common interface $\Gamma = \{(x, y) : x = 0, 0 < y < 1\}$.

The domain decomposition methods introduced before can be implemented using any kind of finite dimensional method to approximate the boundary value problems induced on each subdomain.

Here, we use the spectral collocation method with a *weak* treatment of boundary and interface conditions. For details see, e.g., [4, 13, 7].

The boundary value problem considered is

$$(13) \begin{cases} L_\varepsilon u \equiv -\varepsilon \Delta u + \begin{pmatrix} \sin \varsigma \\ \cos \varsigma \end{pmatrix} \cdot \nabla u = f & \text{in } \Omega \\ u = g & \text{on } \partial\Omega \end{cases}$$

Note that the magnitude of the vector field \mathbf{b} is equal to one independently of the angle ς (see Fig. 4 and 5). To start with, we consider as test function $u(x, y) = e^{x+y}$, the right hand side f and the boundary values are computed accordingly. The numerical approximation is based on the spectral collocation method using 10×10 nodes in each subdomain.

The stopping criterion is fulfilled when the difference between two subsequent iterates is less than 10^{-10} (Euclidean norm of the relative error).

We can see in Fig. 6 that the optimal number of ADN iterations grows smoothly as far the angle ς tends to zero, whereas in ARN method if ς is close to zero the convergence rate deteriorates when ε gets large. Such behavior

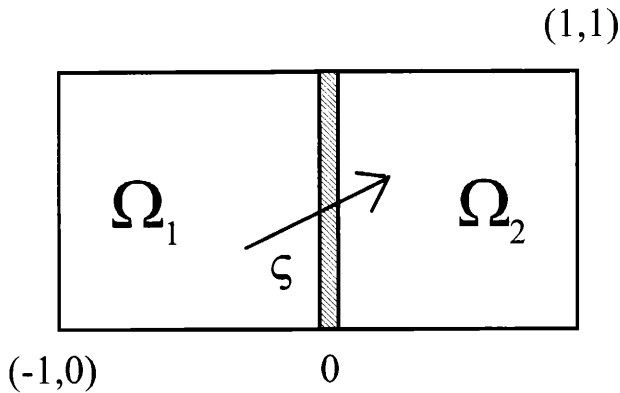


Figure 4: Vector field \mathbf{b} with an angle of $\pi/4$.

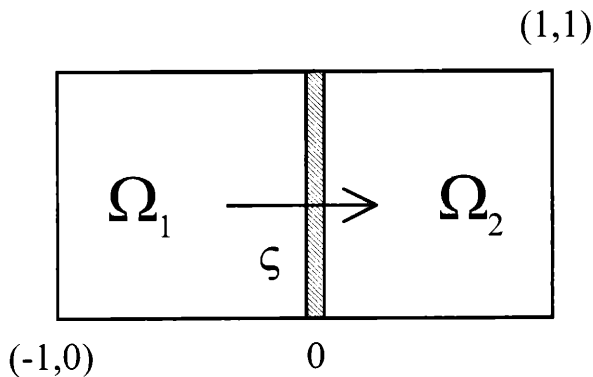


Figure 5: Vector field \mathbf{b} with an angle of $\pi/2$.

is strictly related to the Robin interface conditions $\Psi_i^{(1)}$ imposed. Involving the normal derivative *weighted* with the viscosity ε , the Robin condition deteriorates the convergence as far $\frac{\varepsilon}{|\mathbf{b} \cdot \mathbf{n}_r|}$ grows on the interface. In this case the conditions $\Psi_i^{(1)}$ and $\Psi_i^{(0)}$ tend to be the same (in particular $\Psi_i^{(1)}$ converges to $\Psi_i^{(0)}$) and ARN method becomes unstable.

The convergence of the ARN method, however, looks optimal even in the case of $\mathbf{b} \cdot \mathbf{n}_r = 0$ when ε gets small (see in Fig. 6 the curve relative to $\varepsilon = 10^{-4}$). In this case actually both the interface conditions involved by $\Psi_i^{(0)}$ and $\Psi_i^{(1)}$ can be written as follows (we recall that a *weak* treatment is in order),

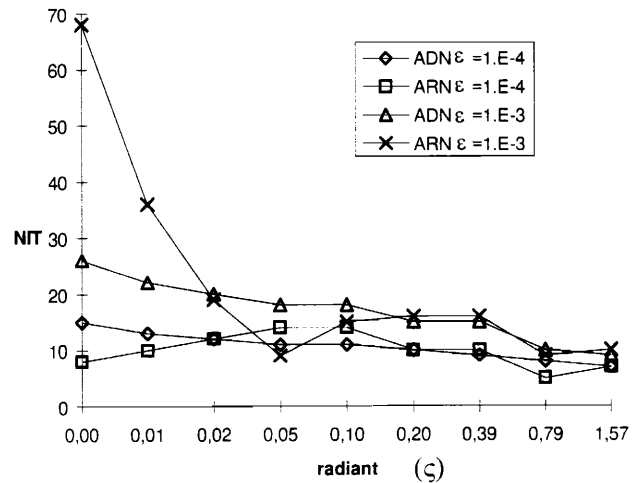


Figure 6: Number of iterations for ADN and ARN methods

$$(14) \quad \omega L_\varepsilon u_1^n + O(\varepsilon) \frac{\partial u_1^n}{\partial \mathbf{n}_1} = \omega f_1 + O(\varepsilon) \frac{\partial u_2^n}{\partial \mathbf{n}_2} + \gamma (f_2 - L_\varepsilon u_2^n)$$

We remind that (14) is the *weak* formulation of the Neumann condition ([4]) and therefore ω and γ are related to the weights of the Gauss Lobatto quadrature formula.

In ARN method ω is exactly one of them whereas γ is still a weight of the Gauss Lobatto formula but it is multiplied by a factor dependent on the magnitude of $\frac{\varepsilon}{|\mathbf{b}|}$ in the computational subdomain Ω_i . Thus, when ε is close to zero the condition (14) tends to be the natural outflow condition for the hyperbolic limit of the advection-diffusion equation, which reads,

$$(15) \quad L_0 u_1^n = f_1$$

In the second test problem we consider the same differential operator as in (13). In this case, however, we are interested in a physical problem with boundary layers.

Then we prescribe the numerical solution to be zero on the boundary of the computational domain and the right hand side f equal to one.

When ε tends to zero the solution develops a boundary layer depending on the direction of the vector field \mathbf{b} . For instance if ς is equal to $\frac{\pi}{2}$ the layer will be close to the right edge of the domain, whereas when ς is equal to $\frac{\pi}{4}$ it will develop on the top-right vertex of the domain (see Fig. 7,8,9 and 10).

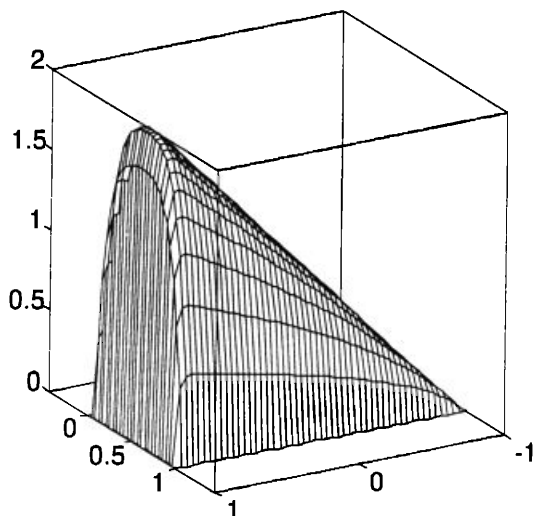


Figure 7: $\varepsilon = 10^{-2}$, $\varsigma = \pi/2$, $\theta = 1$, 6x6 and 9 iterations

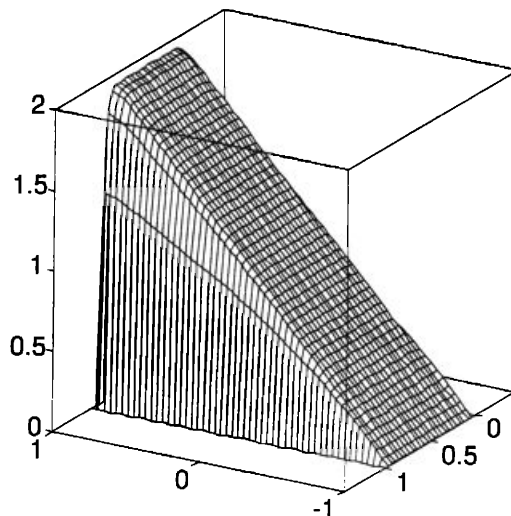


Figure 8: $\varepsilon = 10^{-3}$, $\varsigma = \pi/2$, $\theta = 0.98$, 16x16 and 7 iterations

In each example we have used a different subdomain partition in order to capture the development of the boundary layer. Figures 7 and 8 refer to the following decomposition: $\Omega_1 = (-1, 0.95) \times (0, 1)$ and $\Omega_2 = (0.95, 1) \times (0, 1)$. For the last two figures we have: $\Omega_1 = (-1, 0.95) \times (0, 0.95)$, $\Omega_2 = (0.95, 1) \times (0, 0.95)$, $\Omega_3 = (0.95, 1) \times (0.95, 1)$ and $\Omega_4 = (-1, 0.95) \times (0.95, 1)$. We report in the captions of the figures the degrees of freedom used and the number of iterations obtained to get convergence using the ARN scheme.

Notice that in the last two examples we have used a partition containing a cross point as described in the last section.

To illustrate in particular the behavior of the conditions described in Section 3.2 we show in Fig. 11 the number of iterations with respect to the impinging angle ς . Further we report in the round brackets of the legend the relaxation parameter θ . The numerical approximation is still based on the spectral collocation method using 12x12 nodes in each subdomain.

The stopping criterion is fulfilled when the difference between two subsequent iterates is less than 10^{-10} (Euclidean norm of the relative error).

As we can see, the convergence rate deteriorates as far as the angle tends to zero. This is due to the fact that if $\mathbf{b} \cdot \mathbf{n}_r$ tends to zero the Robin and Neumann conditions are very similar as previous explained.

When ε is not sufficiently small, using the ARN scheme

is equivalent to impose on both sides of the interface Neumann conditions and this produces instabilities which affect the effectiveness of the scheme.

To avoid this lack of consistency we imposed a lower bound on $\mathbf{b} \cdot \mathbf{n}_{ij}$. When $|\mathbf{b} \cdot \mathbf{n}_{ij}|$ is less than $\eta\varepsilon$ (where η is a fixed constant) we add to $\mathbf{b} \cdot \mathbf{n}_{ij}$ the quantity $-\eta\varepsilon + \mathbf{b} \cdot \mathbf{n}_{ij}$ (we recall that $\mathbf{b} \cdot \mathbf{n}_{ij}$ is less than zero because we are considering inflow boundary points). In practice the Robin condition imposed becomes:

$$\Psi_{ij}^{(1)}(u) = \varepsilon \frac{\partial u}{\partial \mathbf{n}_{ij}} + \max(|\mathbf{b} \cdot \mathbf{n}_{ij}|, \eta\varepsilon) u.$$

We can see in Fig. 11 how $\eta = 10$ (see the round brackets in the legend) improves the rate of convergence of the scheme in case of small ς .

Finally in Fig. 12 we illustrate the behavior of the scheme with respect to the degrees of freedom. As we can see, when the impinging angle is small the correction introduced before doesn't seem to be still effective.

Since no theory is available at the moment, we can just give some heuristic explanation to this bad behavior.

When we increase the number of degrees of freedom the spectral approximation of the differential operator becomes in some sense *more elliptic*. The eigenvalues of the pseudospectral operator actually grow like N^4 where N is the polynomial degree used in the approximation.

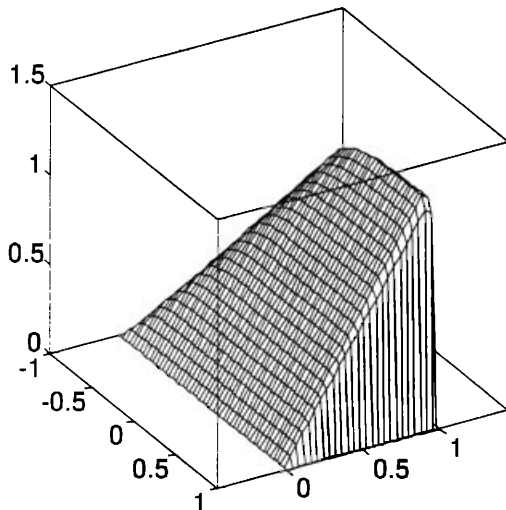


Figure 9: $\varepsilon = 10^{-2}$, $\zeta = \pi/4$, $\theta = 0.96$, 10x10 and 12 iterations

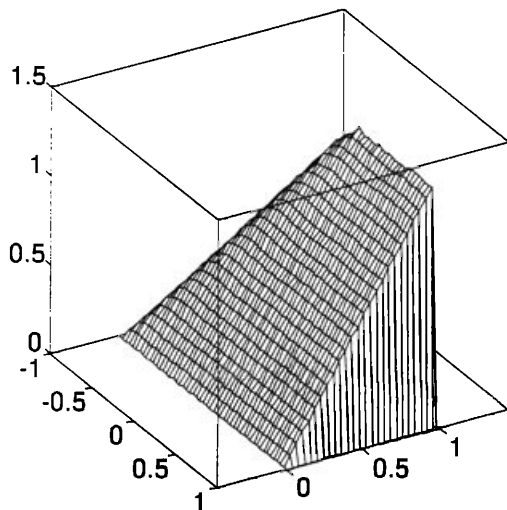


Figure 10: $\varepsilon = 2 \cdot 10^{-3}$, $\zeta = \pi/4$, $\theta = 1.1$, 12x12 and 17 iterations

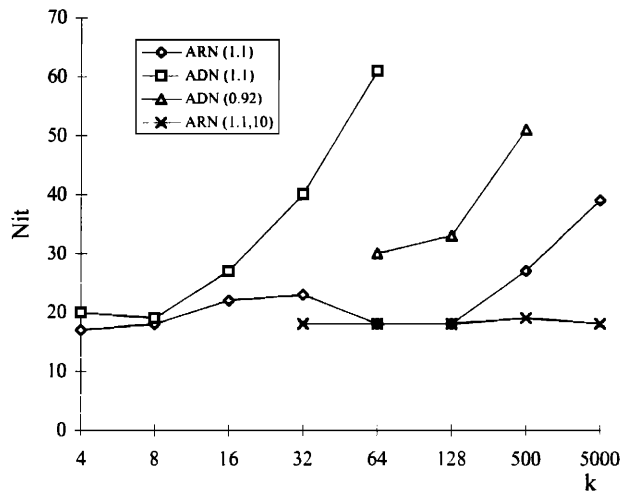


Figure 11: Number of iterations for ADN and ARN methods with an impinging angle $\zeta = \pi/k$

However the second order operator is multiplied by ε and this allows us to treat the whole operator like an hyperbolic one when N is quite small.

On the contrary when N grows the pseudospectral operator is more elliptic and then the Robin condition involved in the ARN scheme is no more sufficient to ensure a fast convergence.

Since the normal derivative is a first order operator and since for first order operators the eigenvalues of the spectral approximation grow like N^2 , we have chosen η as a linear function of N^2 (i.e. $\eta = \frac{\sqrt{2}N^2}{14}$) obtaining a sensible improving in the convergence rate as shown in Fig. 12 (the case labeled in the legend with a star).

In the third test problem we consider a vector field \mathbf{b} that changes direction along the interface Γ . In particular we have chosen

$$(16) \quad \mathbf{b} = \begin{pmatrix} \sin\left(\frac{\pi}{2}(2y-1)\right) \\ \cos\left(\frac{\pi}{2}(2y-1)\right) \end{pmatrix}$$

Note that the magnitude of \mathbf{b} is still equal to one.

The computational domain and the test function are still those considered in the first example. Fig. 13 shows the number of iterations using ARN and ADN methods for different choices of the relaxation parameter θ . ARN looks

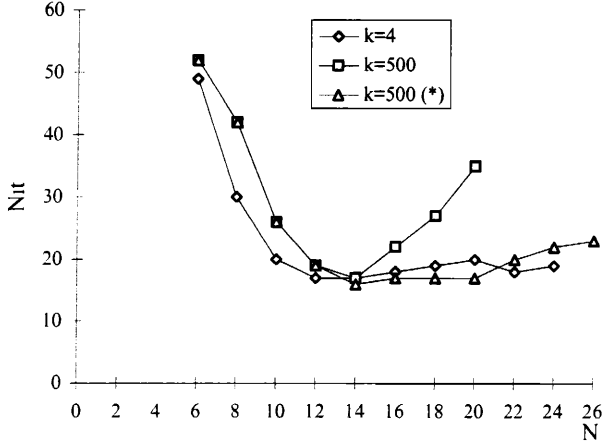


Figure 12: Number of iterations for the ARN scheme with an impinging angle $\zeta = \pi/k$, $k=4$ and 500

more stable than ADN. Moreover in ARN method the optimal relaxation parameter is bigger than the one of ADN and consequently less iterations are occurred to get convergence.

The numerical approximation is still based on the spectral collocation method using 10×10 nodes in each subdomain. Again the stopping criterion is fulfilled when the difference between two subsequent iterates is less than 10^{-10} (Euclidean norm of the relative error).

5 Extension to the nonlinear case

In this section we are interested in the extension of the previous approach to a time dependent nonlinear advection-diffusion equation. We consider the model problem,

$$(17) \quad \frac{\partial u}{\partial t} - \varepsilon \Delta u + \mathbf{b}(u) \cdot \nabla u + a u = f \quad \text{in } \Omega \times (0, T)$$

where $\mathbf{b}(u)$ is a vector function depending on u .

To advance in time, we introduce a finite difference temporal approximation. We note by u^k the value of u at the time level $(k \Delta t)$, where Δt is the temporal discretization parameter.

We consider a semi-implicit scheme as follows:

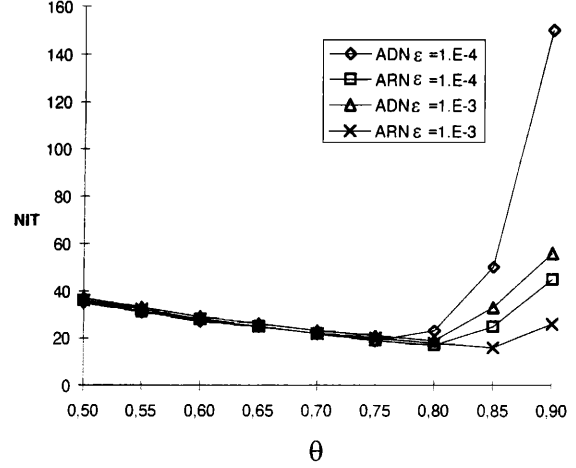


Figure 13: Number of iterations for ADN and ARN methods for the vector field $\mathbf{b} = (\sin(\frac{\pi}{2}(2y - 1)), \cos(\frac{\pi}{2}(2y - 1)))^t$

Given u^0 , solve

$$(18) \quad \left(\frac{1}{\Delta t} + a \right) u^{k+1} - \varepsilon \Delta u^{k+1} + \mathbf{b}(u^k) \cdot \nabla u^{k+1} = f^{k+1} + \frac{u^k}{\Delta t} \quad k = 0, 1, \dots$$

In this way at each time level we have to solve an advection-diffusion problem like the one faced in Sections 3.1 and 3.2.

Using ARN for instance the interface conditions between Ω_1 and Ω_2 become (for simplicity we report only the conditions relative to u_1^{k+1} , the ones for u_2^{k+1} are imposed in a similar way):

$$(19) \quad \begin{aligned} \varepsilon \frac{\partial u_1^{k+1}}{\partial \mathbf{n}_1} - \mathbf{b}(u_1^k) \cdot \mathbf{n}_1 u_1^{k+1} &= \\ -\varepsilon \frac{\partial u_2^{k+1}}{\partial \mathbf{n}_2} + \mathbf{b}(u_2^k) \cdot \mathbf{n}_2 u_2^{k+1} & \end{aligned}$$

if $\mathbf{b}(u_1^k) \cdot \mathbf{n}_1 < 0$

$$\varepsilon \frac{\partial u_1^{k+1}}{\partial \mathbf{n}_1} = -\varepsilon \frac{\partial u_2^{k+1}}{\partial \mathbf{n}_2}$$

otherwise.

We remind that here u_i^k $i=1,2$ stands for the solution of the i -th subdomain at the time level k . Then we can use the scheme (11) to approximate (18)-(19).

We consider now a test problem in which $\Omega = (-1, -1)^2$, $a = f = 0$, $\mathbf{b}(u) = \begin{pmatrix} u \\ 1 \end{pmatrix}$, $\varepsilon = 10^{-5}$ and the initial condition is

$$u^0(x, y) = \frac{1}{2}(1 + y - xy - 3x)$$

Although initially smooth, the solution of this problem develops an internal layer at some finite time. This fact suggests to use a decomposition of Ω into three subdomains instead of two. This allows to capture the development of the internal layer.

Then we consider $\Omega_1 = (-1, 0) \times (-1, 1)$, $\Omega_2 = (0, 0.4) \times (-1, 1)$ and $\Omega_3 = (0.4, 1) \times (-1, 1)$.

As boundary conditions, we prescribe $u(x, y, t) = u^0(x, y) \forall t > 0$ for all points (x, y) belonging to the lower horizontal as well as the two vertical sides of Ω . Further, we require that the normal derivative must be zero on the upper horizontal edge of the computational domain.

Each time level is solved with ARN method with a relaxation parameter equal to one. We obtained a number of ARN iterations independent of the time level (i.e. 5 iterations). The numerical approximation in space is still based on the spectral collocation method using 16×16 nodes in each subdomain. Again the stopping criterion is fulfilled when the difference between two subsequent iterates is less than 10^{-10} (Euclidean norm of the relative error).

The time step used in this computation is $1/2000$.

In Fig. 14 we have drawn the solution at different time levels T .

We can see that the development of the internal layer doesn't affect the smoothness of the numerical solution. Moreover since the time-dependent solution changes its sign along the interface $\Gamma_{1,2}$ we plot in Fig. 15 the dependence on time of the *zero-point* (i.e. the point where the vector field $\mathbf{b}(u)$ has a zero component with respect to the interface normal direction). We remind that this point indicates the exchange of the interface conditions in the ARN method at each time level k .

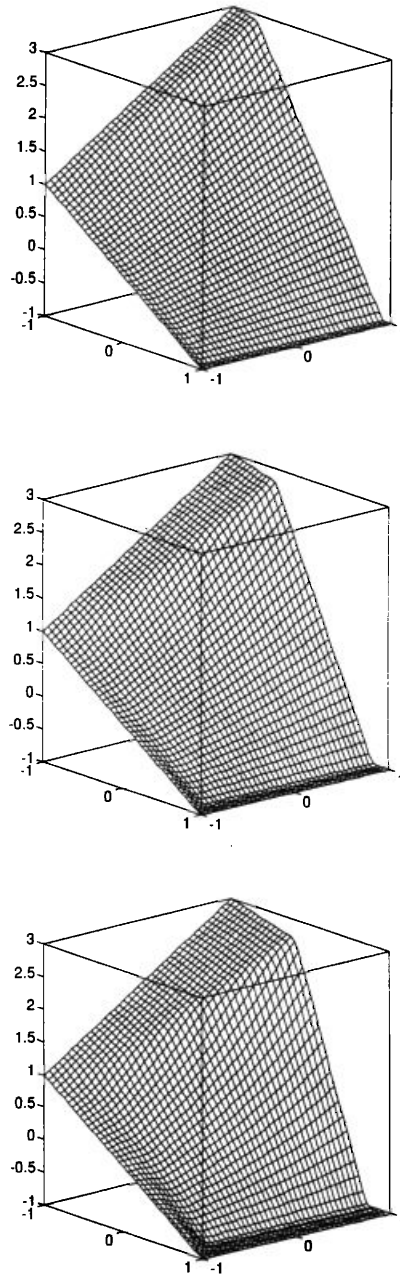


Figure 14: The solution of the nonlinear problem at time levels $T=0.10-0.20-0.30$

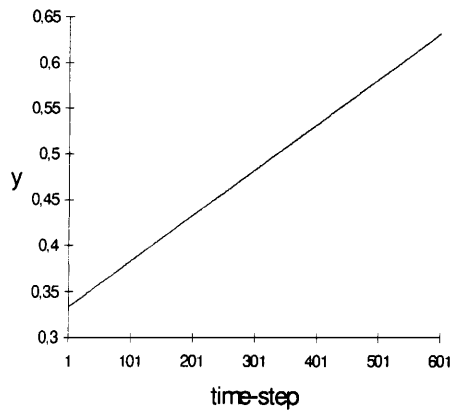


Figure 15: The y-coordinate of the zero-point from T=0 sec to T =0.30 sec

6 The use of adaptive methods in the framework of projection methods for Navier Stokes equations

In this last section we are interested in the approximation of the incompressible Navier Stokes equations

$$(20) \quad \begin{aligned} \frac{\partial \mathbf{u}}{\partial t} - \varepsilon \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p &= \mathbf{f} & \text{in } \Omega \times [0, T] \\ \operatorname{div} \mathbf{u} &= 0 & \text{in } \Omega \times [0, T] \\ \mathbf{u}|_{t=0} &= \mathbf{u}_0 \end{aligned}$$

For the simulation of the equation (20) we will consider the so called *projection methods*.

These schemes have been proposed for solve the unsteady incompressible Navier-Stokes equations (see [6, 18] and some extensions in [9, 20, 19]).

The simplicity and efficiency of projection methods render them particular attractive. The idea behind these kinds of methods is to split the equation into intermediate steps in order to decouple the viscous effects and the incompressibility condition.

A rigorous error analysis for these projection schemes has not been available until recently. In Shen ([14, 15]) the author gave a first error analysis for some frequently used projection schemes. Although many authors have observed second order accuracy for the projection scheme presented in [20] only in the last few months Shen ([16]) in a further paper has established in a rigorous manner this property.

To start with, we consider the following projection scheme derived from the one proposed in [3]:

Given $\mathbf{u}^0 = \mathbf{u}_0(x, y)$, we solve

$$(21) \quad \begin{cases} \frac{\tilde{\mathbf{u}}^{k+1} - \mathbf{u}^k}{\Delta t} - \varepsilon \Delta \tilde{\mathbf{u}}^{k+1} + B(\tilde{\mathbf{u}}^k, \tilde{\mathbf{u}}^{k+1}) + \gamma \nabla p^k = \mathbf{f}^{k+1} & \text{in } \Omega \\ \tilde{\mathbf{u}}^{k+1} = \mathbf{g} & \text{on } \partial\Omega \end{cases} \quad (22)$$

$$(23) \quad \begin{cases} \operatorname{div} \tilde{\mathbf{u}}^{k+1} - \beta \Delta t \Delta ((1 + \nu \Delta t) p^{k+1} - \gamma p^k) = 0 & \text{in } \Omega \\ \frac{\partial p^{k+1}}{\partial \mathbf{n}} = \gamma \frac{\partial p^k}{\partial \mathbf{n}} & \text{on } \partial\Omega \end{cases}$$

$$(23) \quad \mathbf{u}^{k+1} = \tilde{\mathbf{u}}^{k+1} - \beta \Delta t \nabla ((1 + \nu \Delta t) p^{k+1} - p^k)$$

where $\beta \geq 1$ and $B(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \nabla) \mathbf{v}$.

Since we are interested in simulations of high Reynolds flows the diffusive part of the equation (21) is dominated by the convective one $B(\mathbf{u}, \mathbf{v})$.

Then ARN and ADN methods can be easily adapted to effort the solution of each component of the vector equation (21).

To solve equation (22) we use a classical Dirichlet/Neumann procedure.

As numerical example, we want to show a simulation of a viscous, incompressible flow. We consider the so-called *driven cavity* problem.

The data are as follows: $\Omega = (0, 1)^2$, $\varepsilon = 10^{-3}$, $\mathbf{f} = \mathbf{0}$. The boundary conditions are: $\mathbf{u} = \mathbf{0}$ on $\partial\Omega \setminus \{(x, 1) | 0 \leq x \leq 1\}$ and $\mathbf{u} = (1, 0)^t$ on $\{(x, 1) | 0 \leq x \leq 1\}$. Then the Reynolds number is equal to 1000.

We split the computational domain Ω in two subdomains: $\Omega_1 = (0, 1) \times (0, 0.8)$ and $\Omega_2 = (0, 1) \times (0.8, 1)$. The numerical approximation is still based on the spectral collocation method using 24×18 nodes in each subdomain.

Constants in scheme (21)-(22)-(23) are chosen accordingly to the so called *non incremental* version of the projection scheme (see [9]).

Then we have $\beta = 1$, $\nu = 0$ and $\gamma = 1$. The time step is $\Delta t = 1/20$.

For this simulation we have used the ADN method with a relaxation parameter θ equal to 0.65 obtaining a number

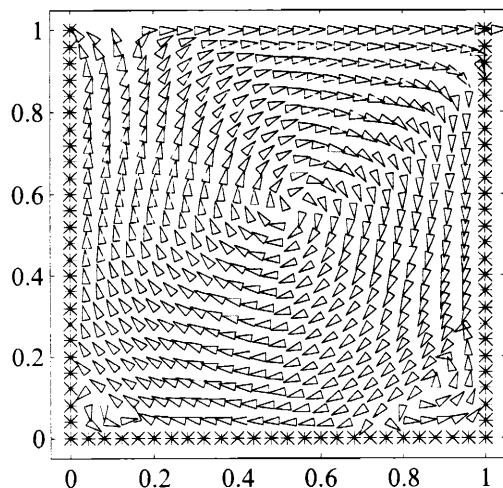


Figure 16: Vector field distribution for the driven cavity, $Re=1000$

of iterations (25/26) independent of the time level (except for the first few iterations when the fluid is not moving yet across the interface).

In Fig. 16 we have shown the vector field distribution of the velocity for the example illustrated.

References

- [1] P.E. Bjorstad and O.B. Widlund (1986), *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM J. Numer. anal. 23 pp.1097-1120;
- [2] E. Brakkee and P. Wilders (1993), *A Domain Decomposition Method for the Advection-Diffusion Equation*, preprint;
- [3] W.Cao and C.Carlenzoli (1994), *A Projection Method for Long-Term Computation of Navier-Stokes Equations*, Quaderno del Seminario Matematico di Brescia, n.17/94;
- [4] C.Carlenzoli and P.Gervasio (1992), *Effective Numerical Algorithms for the Solution of Algebraic Systems Arising in Spectral Methods*, "Applied Numerical Mathematics", 10 pp. 87-113;
- [5] C.Carlenzoli and A.Quarteroni (1993), *Adaptive Domain Decomposition Methods for Advection-Diffusion Problems*, Proceeding of the "Ima-Workshop on Mesh-Adaptivity" July 1993;
- [6] A.Chorin (1968), *Numerical simulation of the Navier-Stokes equations*, Math. Comp,22,pp.745-762;
- [7] A.Frati, F.Pasquarelli and A.Quarteroni (1993), *Spectral approximation to advection diffusion problems by the fictitious interface method*, J. Comput. Phys 107, pp.201-212;
- [8] D.Funaro, A.Quarteroni and P.Zanolli (1988), *An iterative procedure with interface relaxation for domain decomposition methods*, SIAM J. Numer. Anal. 25, pp1231-1236;
- [9] K.Goda (1979), *A multistep technique with implicit difference schemes for calculating two or three dimensional cavity flows*, J. of Comput. Phys., 30, pp.76-95;
- [10] J.-L.Lions and E.Magenes (1968), *Problemes aux Limites non Homogenes et Applications*, Dunod, Paris;
- [11] L.D.Marini and A.Quarteroni (1989), *A relaxation procedure for domain decomposition methods using finite elements*, Numer. Math. 55, pp. 575-598;
- [12] A.Quarteroni and G.Sacchi Landriani (1988), *Doamin Decomposition Preconditioners for the spectral collocation methods*, J. Scientific Comput. 3, pp. 45-75;
- [13] A.Quarteroni and A.Valli (1994), *Numerical approximation of partial differential equations*, Springer Verlag, Heidelberg, in press;
- [14] J. Shen (1992), *On pressure stabilization method and projection method for unsteady Navier-Stokes equations*, in Advances in Computer Methods for Partial Differential Equations, R. Vichnevetsky, D.Knight and G.Richter, eds., IMACS, pp. 658-662;
- [15] J. Shen (1994), *Remarks on the pressure error estimates for the projection methods*, Numer. Math. 67, pp. 513-520;
- [16] J.Shen (1994), *On error estimates of the projection methods for the Navier-Stokes equations: second order schemes*, preprint;
- [17] K.H. Tan and M.J.A. Borsboom (1993), *Problem-Dependent Optimization of Flexible Couplings in Domain Decomposition Methods, with an Application to Advection-Dominated Problems*, preprint;

- [18] R.Temam (1968), *Une methode d'approximation de la solution des equations de Navier-Stokes*, Bull. Soc. Math. France, 98, pp.115-152;
- [19] R.Temam (1991), *Remark on the pressure boundary condition for the projection method*, Theoret. Comput. Fluid Dynamics, 3, pp.181-184;
- [20] J.Van Kan (1991), *A second-order accurate pressure-correction scheme for viscous incompressible flow*, SIAM J. Comput., 7, 3, pp.870-891.

