

Approximate numerical procedures for the Navier-Stokes system through the generalized method of lines

Fabio Silva Botelho

Department of Mathematics

Federal University of Santa Catarina

Florianópolis - SC, Brazil

Abstract

This article develops applications of the generalized method of lines to numerical solutions of the time-independent, incompressible Navier-Stokes system in fluid mechanics. We recall that for such a method, the domain of the partial differential equation in question is discretized in lines (or more generally in curves), and the concerning solutions are written on these lines as functions of the boundary conditions and the domain boundary shape.

Key words: Generalized method of lines, Navier-Stokes system, equivalent elliptic system.

MSC: 65N40, 35Q30

1 Introduction

In this article, we develop approximate solutions for the time independent incompressible Navier-Stokes system, through the generalized method of lines. We recall again, for such a method, the domain of the partial differential equation in question is discretized in lines and the concerning solution is written on these lines as functions of the boundary conditions and boundary shape. We emphasize these last main results are established through applications of the Banach fixed point theorem.

Remark 1.1. *We also highlight the first two paragraphs in this article (a relatively small part) overlaps with the Chapter 27, starting page 512, in the book F.S. Botelho, [2], published in 2020, by CRC Taylor and Francis. However, we emphasize the present article includes substantial new parts, including a concerning software not included in the previous version of 2020. Another novelty in the present version is the establishment of appropriate boundary conditions for an elliptic system equivalent to original Navier-Stokes one. Such new boundary conditions and concerning results are indicated in section 2.*

At this point we describe the system in question.

Consider $\Omega \subset \mathbb{R}^2$ an open, bounded and connected set, whose the regular (Lipschitzian) internal boundary is denoted by Γ_0 and the regular external one is denoted by Γ_1 . For a two-dimensional motion of a fluid on Ω , we denote by $u : \Omega \rightarrow \mathbb{R}$ the velocity field in the direction x of the Cartesian system (x, y) , by $v : \Omega \rightarrow \mathbb{R}$, the velocity field in the direction y and by $p : \Omega \rightarrow \mathbb{R}$, the pressure one. We define $P = p/\rho$, where ρ is the constant fluid density. Finally, ν denotes the viscosity coefficient and g denotes the gravity field. Under such notation and statements, the time-independent incompressible Navier-Stokes system of partial differential equations stands for,

$$\begin{cases} \nu \nabla^2 u - u \partial_x u - v \partial_y u - \partial_x P + g_x = 0, & \text{in } \Omega, \\ \nu \nabla^2 v - u \partial_x v - v \partial_y v - \partial_y P + g_y = 0, & \text{in } \Omega, \\ \partial_x u + \partial_y v = 0, & \text{in } \Omega, \end{cases} \quad (1)$$

$$\begin{cases} u = v = 0, & \text{on } \Gamma_0, \\ u = u_\infty, v = 0, P = P_\infty, & \text{on } \Gamma_1 \end{cases} \quad (2)$$

At first we look for solutions $(u, v, P) \in W^{2,2}(\Omega) \times W^{2,2}(\Omega) \times W^{1,2}(\Omega)$. We emphasize details about such Sobolev spaces may be found in [1]. Finally, standard results on finite differences and existence theory and other results for similar systems may be found in [10, 6, 7, 9, 8] and [11], respectively.

2 Details about an equivalence of a concerning elliptic system

Consider again the Navier-Stokes system

$$\begin{cases} \nu \nabla^2 u - u \partial_x u - v \partial_y u - \partial_x P + g_x = 0, & \text{in } \Omega, \\ \nu \nabla^2 v - u \partial_x v - v \partial_y v - \partial_y P + g_y = 0, & \text{in } \Omega, \\ \partial_x u + \partial_y v = 0, & \text{in } \Omega, \end{cases} \quad (3)$$

$$\begin{cases} u = v = 0, & \text{on } \Gamma_0, \\ u = u_\infty, v = 0, P = P_\infty, & \text{on } \Gamma_1 \end{cases} \quad (4)$$

As previously mentioned, at first we look for solutions $(u, v, P) \in W^{2,2}(\Omega) \times W^{2,2}(\Omega) \times W^{1,2}(\Omega)$.

We are going to obtain an equivalent Elliptic system with appropriate boundary conditions. Our main result is summarized by the following theorem.

Theorem 2.1. *Let $\Omega \subset \mathbb{R}^2$ be an open, bounded, connected set a regular (Lipschitzian) boundary.*

Assume $u, v, P \in W^{2,2}(\Omega)$ are such that

$$\begin{cases} \nu \nabla^2 u - u u_x - v u_y - P_x + g_x = 0, & \text{in } \Omega, \\ \nu \nabla^2 v - u v_x - v v_y - P_y + g_y = 0, & \text{in } \Omega, \\ \nabla^2 P + u_x^2 + v_y^2 + 2u_y v_x - \nabla^2 g = 0, & \text{in } \Omega, \end{cases} \quad (5)$$

$$\begin{cases} u = u_0, v = v_0, & \text{on } \partial\Omega, \\ u_x + v_y = 0, & \text{on } \partial\Omega. \end{cases} \quad (6)$$

Suppose also the unique solution of equation in w

$$\nu \nabla^2 w - u w_x - v w_y = 0, \text{ in } \Omega$$

with the boundary conditions

$$w = 0 \text{ on } \partial\Omega,$$

is

$$w = 0, \text{ in } \Omega.$$

Under such hypotheses, u, v, P solve the following Navier-Stokes system

$$\begin{cases} \nu \nabla^2 u - u u_x - v u_y - P_x + g_x = 0, & \text{in } \Omega, \\ \nu \nabla^2 v - u v_x - v v_y - P_y + g_y = 0, & \text{in } \Omega, \\ u_x + v_y = 0, & \text{in } \Omega, \end{cases} \quad (7)$$

$$\begin{cases} u = u_0, v = v_0, & \text{on } \partial\Omega, \\ u_x + v_y = 0, & \text{on } \partial\Omega. \end{cases} \quad (8)$$

Proof. In (5), taking the derivative in x of the first equation and adding with the derivative in y of the second equation, we obtain

$$\begin{aligned} & \nu \nabla^2 (u_x + v_y) - u(u_x + v_y)_x - v(u_x + v_y)_y \\ & - \nabla^2 P - u_x^2 - v_y^2 - 2u_y v_x + \nabla^2 g = 0, \text{ in } \Omega \end{aligned} \quad (9)$$

Assume now u, v, P are such that

$$\nabla^2 P + u_x^2 + v_y^2 + 2u_y v_x - \nabla^2 g = 0, \text{ in } \Omega,$$

From this and (9), we get

$$\nu \nabla^2 (u_x + v_y) - u(u_x + v_y)_x - v(u_x + v_y)_y = 0, \text{ in } \Omega. \quad (10)$$

Denoting $w = u_x + v_y$, from this last equation we obtain

$$\nu \nabla^2 w - u w_x - v w_y = 0, \text{ in } \Omega.$$

From the hypothesis, the unique solution of this last equation with the boundary conditions $w = 0$, on $\partial\Omega$, is $w = 0$.

From this and (10) we have

$$u_x + v_y = 0,$$

in Ω with the boundary conditions

$$u_x + v_y = 0, \text{ on } \partial\Omega.$$

The proof is complete. $\square$

3 An approximate proximal approach

In this section we develop an approximate proximal numerical procedure for the model in question.

Such results are extensions of previous ones published in F.S. Botelho, [3] now for the Navier-Stokes system context.

More specifically, we solve the equation

$$\begin{cases} \nu \nabla^2 u - u \partial_x u - v \partial_y u - \partial_x P = 0, & \text{in } \Omega, \\ \nu \nabla^2 v - u \partial_x v - v \partial_y v - \partial_y P = 0, & \text{in } \Omega, \\ \nabla^2 P + (\partial_x u)^2 + (\partial_y v)^2 + 2(\partial_y u)(\partial_x v) = 0, & \text{in } \Omega, \end{cases} \quad (11)$$

We present a software similar to those presented in [3], with $\nu = 0.0177$, and with

$$\Omega = [0, 1] \times [0, 1]$$

with the boundary conditions

$$u = u_0 = 0.65y(1 - y), \quad v = v_0 = 0, \quad P = p_0 = 0.15 \text{ on } [0, y], \forall y \in [0, 1],$$

$$u = v = P_y = 0, \text{ on } [x, 0] \text{ and } [x, 1], \quad \forall x \in [0, 1],$$

$$u_x = v_x = 0, \text{ and } P = p_f = 0.12 \text{ on } [1, y], \quad \forall y \in [0, 1].$$

The equation (11), in partial finite differences, stands for

$$\begin{aligned} & \nu \left(\frac{u_{n+1} - 2u_n + u_{n-1}}{d^2} + \frac{\partial^2 u_n}{\partial y^2} \right) - u_n \frac{(u_n - u_{n-1})}{d} - v_n \frac{\partial u_n}{\partial y} \\ & - \frac{P_n - P_{n-1}}{d} = 0, \end{aligned} \quad (12)$$

$$\begin{aligned} & \nu \left(\frac{v_{n+1} - 2v_n + v_{n-1}}{d^2} + \frac{\partial^2 v_n}{\partial y^2} \right) - u_n \frac{(v_n - v_{n-1})}{d} - v_n \frac{\partial v_n}{\partial y} \\ & - \frac{\partial P_n}{\partial y} = 0, \end{aligned} \quad (13)$$

$$\begin{aligned} & \left(\frac{P_{n+1} - 2P_n + P_{n-1}}{d^2} + \frac{\partial^2 P_n}{\partial y^2} \right) + (u_n - u_{n-1}) \frac{(u_n - u_{n-1})}{d^2} + \left(\frac{\partial v_n}{\partial y} \right)^2 \\ & + 2 \frac{\partial u_n}{\partial y} \left(\frac{v_n - v_{n-1}}{d} \right) = 0. \end{aligned} \quad (14)$$

After linearizing such a system about U_0, V_0, P_0 and introducing the proximal formulation, for an appropriate non-negative real constant K , we get

$$\begin{aligned} & \nu \left(\frac{u_{n+1} - 2u_n + u_{n-1}}{d^2} + \frac{\partial^2 u_n}{\partial y^2} \right) - (U_0)_n \frac{(u_n - (U_0)_{n-1})}{d} - (V_0)_n \frac{\partial u_n}{\partial y} \\ & - \frac{(P_0)_n - (P_0)_{n-1}}{d} - K u_n + K (U_0)_n = 0, \end{aligned} \quad (15)$$

$$\begin{aligned} & \nu \left(\frac{v_{n+1} - 2v_n + v_{n-1}}{d^2} + \frac{\partial^2 v_n}{\partial y^2} \right) - (U_0)_n \frac{(v_n - (V_0)_{n-1})}{d} - (V_0)_n \frac{\partial v_n}{\partial y} \\ & - \frac{\partial(P_0)_n}{\partial y} - K v_n + K (V_0)_n = 0, \end{aligned} \quad (16)$$

$$\begin{aligned} & \left(\frac{P_{n+1} - 2P_n + P_{n-1}}{d^2} + \frac{\partial^2 P_n}{\partial y^2} \right) + (u_{n+1} - (U_0)_n) \frac{((u)_{n+1} - (U_0)_n)}{d^2} + \left(\frac{\partial(V_0)_n}{\partial y} \right)^2 \\ & + 2 \frac{\partial(U_0)_n}{\partial y} \left(\frac{v_{n+1} - (V_0)_n}{d} \right) - K P_n + K (P_0)_n = 0. \end{aligned} \quad (17)$$

At this point denoting $\nu = e_1$, we define

$$(T_1)_n = \left(-(U_0)_n \frac{(u_n - (U_0)_{n-1})}{d} - (V_0)_n \frac{\partial u_n}{\partial y} - \frac{(P_0)_n - (P_0)_{n-1}}{d} \right) \frac{d^2}{e_1} + \frac{\partial^2 u_n}{\partial y^2} d^2,$$

$$(T_2)_n = \left(-(U_0)_n \frac{(v_n - (V_0)_{n-1})}{d} - (V_0)_n \frac{\partial v_n}{\partial y} - \frac{\partial(P_0)_n}{\partial y} \right) \frac{d^2}{e_1} + \frac{\partial^2 v_n}{\partial y^2} d^2,$$

and

$$\begin{aligned} (T_3)_n &= (u_{n+1} - (U_0)_n) \frac{((u)_{n+1} - (U_0)_n)}{d^2} d^2 + \left(\frac{\partial(V_0)_n}{\partial y} \right)^2 d^2 \\ &+ 2 \frac{\partial(U_0)_n}{\partial y} \left(\frac{v_{n+1} - (V_0)_n}{d} \right) d^2 + \frac{\partial^2 P_n}{\partial y^2} d^2. \end{aligned} \quad (18)$$

Therefore, we may write

$$u_{n+1} - 2u_n + u_{n-1} + K u_n \frac{d^2}{e_1} + (T_1)_n + (f_1)_n = 0,$$

where

$$(f_1)_n = -K(U_0)_n \frac{d^2}{e_1},$$

$\forall n \in \{1, \dots, N-1\}$.

In particular for $n = 1$, we obtain

$$u_2 - 2u_1 + u_0 + (T_1)_1 + (f_1)_1 = 0,$$

so that

$$u_1 = a_1 u_2 + b_1 u_0 + c_1 (T_1)_1 + (h_1)_1 + (E_r)_1,$$

where

$$a_1 = \left(2 + K \frac{d^2}{e_1}\right)^{-1},$$

$$b_1 = a_1$$

$$c_1 = a_1$$

$$(h_1)_1 = a_1 (f_1)_1,$$

$$(E_r)_1 = 0.$$

Similarly, for $n = 2$ we get

$$u_3 - 2u_2 + u_1 + (T_1)_2 + (f_1)_2 = 0,$$

so that

$$u_2 = a_2 u_3 + b_2 u_0 + c_2 (T_1)_2 + (h_1)_2 + (E_r)_2,$$

where

$$a_2 = \left(2 + K \frac{d^2}{e_1} - a_1\right)^{-1},$$

$$b_2 = a_2 b_1$$

$$c_2 = a_2 (c_1 + 1)$$

$$(h_1)_2 = a_2 ((h_1)_1 + (f_1)_2),$$

$$(E_1)_2 = a_2 (c_1 ((T_1)_1 - (T_1)_2)).$$

Reasoning inductively, having

$$u_{n-1} = a_{n-1} u_n + b_{n-1} u_0 + c_{n-1} (T_1)_{n-1} + (h_1)_{n-1} + (E_r)_{n-1},$$

we obtain

$$u_n = a_n u_{n+1} + b_n u_0 + c_n (T_1)_n + (h_1)_n + (E_r)_n, \quad (19)$$

where

$$a_n = \left(2 + K \frac{d^2}{e_1} - a_{n-1}\right)^{-1},$$

$$b_n = a_n b_{n-1}$$

$$c_n = a_n (c_{n-1} + 1)$$

$$(h_1)_n = a_n((h_1)_{n-1} + (f_1)_n),$$

$$(E_r) = a_n((E_r)_{n-1} + c_{n-1}((T_1)_{n-1} - (T_1)_n)),$$

$\forall n \in \{1, \dots, N-1\}$.

Observe now that $n = N-1$ we have $u_{N-1} = u_N$, so that

$$\begin{aligned} u_{N-1} \approx & a_{N-1}u_{N-1} + b_{N-1}u_0 + c_{N-1}(T_1)_{N-1} + (h_1)_{N-1} \\ & a_{N-1}u_{N-1} + b_{N-1}u_0 \\ & + c_{N-1} \frac{\partial^2 u_{N-1}}{\partial y^2} d^2 \\ & + c_{N-1} \left(-(U_0)_n \frac{(u_n - (U_0)_{n-1})}{d} - (V_0)_n \frac{\partial u_n}{\partial y} - \frac{(P_0)_n - (P_0)_{n-1}}{d} \right) \frac{d^2}{e_1} \\ & + (h_1)_{N-1} \end{aligned} \quad (20)$$

This last equation is a second order ODE in u_{N-1} which must be solved with the boundary conditions

$$u_{N-1}(0) = u_{N-1}(1) = 0.$$

Summarizing we have obtained u_{N-1} .

Similarly, we may obtain v_{N-1} and P_{N-1} .

Having u_{N-1} we may obtain u_{N-2} with $n = N-2$ in equation (19) (neglecting $(E_r)_{N-2}$.)

Similarly, we may obtain v_{N-2} and P_{N-2} .

Having u_{N-2} we may obtain u_{N-3} with $n = N-3$ in equation (19) (neglecting $(E_r)_{N-3}$.)

Similarly, we may obtain v_{N-3} and P_{N-3} .

And so on up to obtaining u_1, v_1 and P_1 .

The next step is to replace $\{(U_0)_n, (V_0)_n, (P_0)_n\}$ by $\{u_n, v_n, P_n\}$ and repeat the process until an appropriate convergence criterion is satisfied.

Here we present a concerning software in MATLAB based in this last algorithm (with small changes and differences where we have set $K = 0$)

1. *clear all*
 $m_8 = 3000;$
 $d = 1/m_8;$
 $m_9 = 120;$
 $d_1 = 1/m_9;$
 $e_1 = 0.1;$
 $K = 0.0;$
 $m_2 = \text{zeros}(m_9 - 1, m_9 - 1);$
2. *for* $i = 2 : m_9 - 2$
 $m_2(i, i) = -2.0;$
 $m_2(i, i + 1) = 1.0;$
 $m_2(i, i - 1) = 1.0;$

```

end;
m2(1,1) = -1.0;
m2(1,2) = 1.0;
m2(m9-1,m9-1) = -1.0;
m2(m9-1,m9-2) = 1.0;
m22 = zeros(m9-1,m9-1);
3. for i = 2 : m9-2
    m22(i,i) = -2.0;
    m22(i,i+1) = 1.0;
    m22(i,i-1) = 1.0;
end;
m22(1,1) = -2.0;
m22(1,2) = 1.0;
m22(m9-1,m9-1) = -2.0;
m22(m9-1,m9-2) = 1.0;
m1a = zeros(m9-1,m9-1);
m1b = zeros(m9-1,m9-1);
4. for i = 1 : m9-2
    m1a(i,i) = -1.0;
    m1a(i,i+1) = 1.0;
end;
m1a(m9-1,m9-1) = -1.0;
5. for i = 2 : m9-1
    m1b(i,i) = 1.0;
    m1b(i,i-1) = -1.0;
end;
m1b(1,1) = 1.0;
m1 = (m1a + m1b)/2;
Id = eye(m9-1);
a(1) = 1/(2 + K * d^2/e1);
b(1) = 1/(2 + K * d^2/e1);
c(1) = 1/(2 + K * d^2/e1);
6. for i = 2 : m8-1
    a(i) = 1/(2 - a(i-1) + K * d^2/e1);
    b(i) = a(i) * b(i-1);
    c(i) = (c(i-1) + 1) * a(i);
end;

```

```

7. for i = 1 : m9 - 1
    u5(i, 1) = 0.65 * i * d1 * (1 - i * d1);
    end;
    Pf = 0.12 * ones(m9 - 1, 1);
    uo = u5;
    vo = zeros(m9 - 1, 1);
    po = 0.15 * ones(m9 - 1, 1);
    Uo = 0.25 * ones(m9 - 1, 1);
    Vo = 0.05 * ones(m9 - 1, 1);
    Po = 0.05 * ones(m9 - 1, 1);

8. for i = 1 : m8 - 1
    U1(:, i) = Uo;
    V1(:, i) = Vo;
    P1(:, i) = Po;
    end;

9. for k7 = 1 : 28
    e1 = e1 * .94;
    b14 = 1.0;
    k1 = 1;
    k1max = 500;
    while (b14 > 10-4) and (k1 < 1000)
        k1 = k1 + 1;
        b12 = 1.0;
        kmax = 100;
        k = 1; i = 1;
        while (b12 > 10-4) and (k < kmax)
            k = k + 1;
            M50 = (Id - a(m8 - 1) * Id - c(m8 - 1) * m22/d12 * d2)
            M50 = M50 + c(m8 - 1) * diag(Vo) * m1/d1 * d2/e1;
            z1 = b(m8 - 1) * uo + K * U1(:, m8 - 1) * d2 * c(m8 - i)/e1;
            M60 = (Id - a(m8 - 1) * Id - c(m8 - 1) * m22/d12 * d2)
            M60 = M60 + c(m8 - 1) * diag(Vo) * m1/d1 * d2/e1;
            z2 = b(m8 - 1) * vo + K * V1(:, m8 - 1) * d2 * c(m8 - i)/e1;
            M70 = (Id - c(m8 - 1) * m2/d12 * d2);
            z3 = a(m8 - 1) * Pf + b(m8 - 1) * po + c(m8 - 1) * ((m1/d1 * Vo). * (m1/d1 * Vo) * d2)
            z3 = z3 + K * P1(:, m8 - 1) * d2 * c(m8 - i)/e1;
            U(:, m8 - 1) = inv(M50) * z1;
            V(:, m8 - 1) = inv(M60) * z2;
            P(:, m8 - 1) = inv(M70) * z3;
        end
    end
end

```

```

    b12 = max(abs(U(:, m8 - 1) - Uo));
    Uo = U(:, m8 - 1);
    Vo = V(:, m8 - 1);
    Po = P(:, m8 - 1);
    end;
10. for i = 2 : m8 - 1
    b12 = 1.0;
    kmax2 = 100; k = 1;
    while (b12 > 10-4) and (k < kmax2)
    k = k + 1;
    M50 = (Id - c(m8 - i) * m22/d12 * d2)
    M50 = M50 + c(m8 - i) * (-diag(Uo) * d + diag(Vo) * m1/d1 * d2)/e1;
    z1 = b(m8 - i) * uo + a(m8 - i) * U(:, m8 - i + 1)
    z1 = z1 + c(m8 - i) * (-Uo * U(:, m8 - i + 1) * d + (P(:, m8 - i + 1) - Po) * d)/e1;
    z1 = z1 + K * U1(:, m8 - i) * d2 * c(m8 - i)/e1;
    M60 = (Id - c(m8 - i) * m22/d12 * d2)
    M60 = M60 + c(m8 - i) * (-diag(Uo) * d + diag(Vo) * m1/d1 * d2)/e1;
    z2 = b(m8 - i) * vo + a(m8 - i) * V(:, m8 - i + 1)
    z2 = z2 + c(m8 - i) * (-Uo * V(:, m8 - i + 1) * d + m1 * Po/d1 * d2)/e1;
    z2 = z2 + K * V1(:, m8 - i) * d2 * c(m8 - i)/e1;
    M70 = (Id - c(m8 - i) * m2/d12 * d2);
    z3 = b(m8 - i) * po + a(m8 - i) * P(:, m8 - i + 1)
    z3 = z3 + c(m8 - i) * ((U(:, m8 - i + 1) - Uo) * (U(:, m8 - i + 1) - Uo)
    z3 = z3 + (m1/d1 * Vo) * (m1/d1 * Vo) * d2);
    z3 = z3 + 2 * (m1/d1 * Uo) * (V(:, m8 - i + 1) - Vo) * d
    z3 = z3 + K * P1(:, m8 - i) * d2 * c(m8 - i)/e1;
    U(:, m8 - i) = inv(M50) * z1;
    V(:, m8 - i) = inv(M60) * z2;
    P(:, m8 - i) = inv(M70) * z3;
    b12 = max(abs(U(:, m8 - i) - Uo));
    Uo = U(:, m8 - i);
    Vo = V(:, m8 - i);
    Po = P(:, m8 - i);
    end; end;
    b14 = max(max(abs(U - U1)));
    U1 = U;
    V1 = V;
    P1 = P;
    k1

```

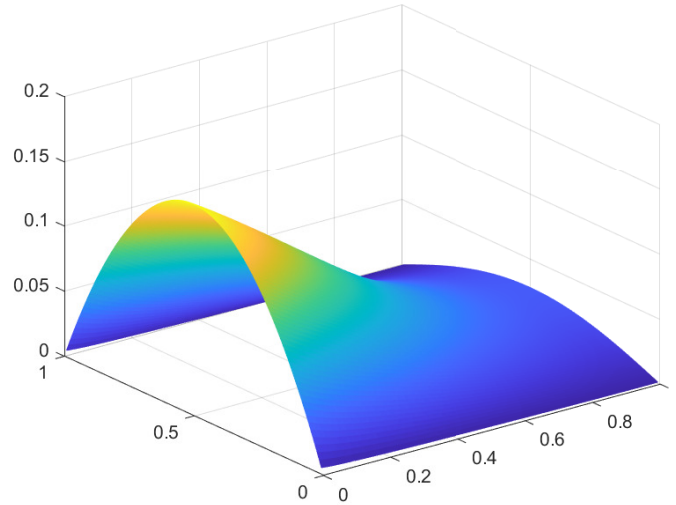


Figure 1: solution $U(x, y)$ for the case $\nu = 0.0177$.

```

U(m9/2,10)
end;
k7
end;
for i = 1 : m9 - 1
y(i) = i * d1;
end;
for i = 1 : m8 - 1
x(i) = i * d;
end;
mesh(x,y,U)

```

For the field of velocities U , V and the pressure field P , please see figures 1, 2 and 3, respectively.

4 The generalized method of lines for the Navier-Stokes system

In this section we develop the solution for the Navier-Stokes system through the generalized method of lines, which was originally introduced in [4], with further developments in [5].

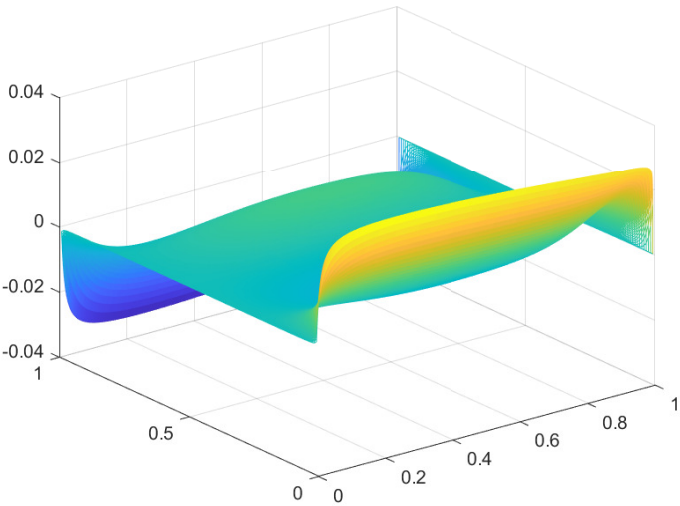


Figure 2: solution $V(x, y)$ for the case $\nu = 0.0177$.

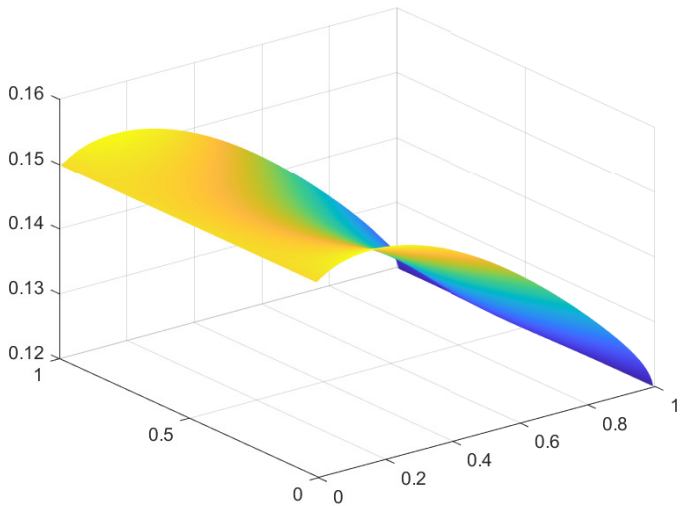


Figure 3: solution $P(x, y)$ for the case $\nu = 0.0177$.

We present a software in MATHEMATICA for $N = 10$ lines for the case in which

$$\begin{cases} \nu \nabla^2 u - u \partial_x u - v \partial_y u - \partial_x P = 0, & \text{in } \Omega, \\ \nu \nabla^2 v - u \partial_x v - v \partial_y v - \partial_y P = 0, & \text{in } \Omega, \\ \nabla^2 P + (\partial_x u)^2 + (\partial_y v)^2 + 2(\partial_y u)(\partial_x v) = 0, & \text{in } \Omega, \end{cases} \quad (21)$$

Such a software refers to the algorithm presented in Chapter 27, in [2], in polar coordinates, with $\nu = 1.0$, and with

$$\Omega = \{(r, \theta) \in \mathbb{R}^2 : 1 \leq r \leq 2, 0 \leq \theta \leq 2\pi\},$$

$$\partial\Omega_1 = \{(1, \theta) \in \mathbb{R}^2 : 0 \leq \theta \leq 2\pi\},$$

and

$$\partial\Omega_2 = \{(2, \theta) \in \mathbb{R}^2 : 0 \leq \theta \leq 2\pi\}.$$

The boundary conditions are

$$u = v = 0, \quad P = 0.15 \text{ on } \partial\Omega_1,$$

$$u = -1.0 \sin(\theta), \quad v = 1.0 \cos(\theta), \quad P = 0.10 \text{ on } \partial\Omega_2.$$

We remark some changes have been made, concerning the original conception, in order to make it suitable through the software MATHEMATICA for such a Navier-Stokes system.

We highlight the nature of this approximation is qualitative.

1. $m_8 = 10$;
2. $\text{Clear}[z_1, z_2, z_3, u_1, u_2, P, b_1, b_2, b_3, a_1, a_2, a_3]$;
3. $\text{Clear}[P_f, t, a_{11}, a_{12}, a_{13}, b_{11}, b_{12}, b_{13}, t_3]$;
4. $d = 1.0/m_8$;
5. $e_1 = 1.0$;
6. $a_1 = 0.0$;
7. $a_2 = 0.0$;
8. $a_3 = 0.15$;
9. $\text{For}[i = 1, i < m_8, i ++,$
 $\quad \text{Print}[i]$;
 $\quad \text{Clear}[b_1, b_2, b_3, u_1, u_2, P]$;
 $\quad b_1[x_-] = u_1[i + 1][x]$;
 $\quad b_2[x_-] = u_2[i + 1][x]$;
 $\quad b_3[x_-] = P[i + 1][x]$;
 $\quad t[i] = 1 + i * d$;
 $\quad du1x = \text{Cos}[x] * (b_1[x] - a_1)/d * t_3 - 1/t[i] * \text{Sin}[x] * D[b_1[x], x] * t_3$;
 $\quad du1y = \text{Sin}[x] * (b_1[x] - a_1)/d * t_3 + 1/t[i] * \text{Cos}[x] * D[b_1[x], x] * t_3$;

- $$du2x = \text{Cos}[x] * (b_2[x] - a_2)/d * t_3 - 1/t[i] * \text{Sin}[x] * D[b_2[x], x] * t_3;$$
- $$du2y = \text{Sin}[x] * (b_2[x] - a_2)/d * t_3 + 1/t[i] * \text{Cos}[x] * D[b_2[x], x] * t_3;$$
- $$dPx = \text{Cos}[x] * (b_3[x] - a_3)/d * t_3 - 1/t[i] * \text{Sin}[x] * D[b_3[x], x] * t_3;$$
- $$dPy = \text{Sin}[x] * (b_3[x] - a_3)/d * t_3 + 1/t[i] * \text{Cos}[x] * D[b_3[x], x] * t_3;$$
10. *For*[$k = 1, k < 6, k++$, (in this example, we have fixed a relatively small number of iterations)
Print[k];
 $z_1 = (u_1[i+1][x] + b_1[x] + a_1 + 1/t[i] * (b_1[x] - a_1) * d + 1/t[i]^2 * D[b_1[x], x, 2] * d^2 - (b_1[x] * du1x + b_2[x] * du1y) * d^2/e_1 - dPx * d^2/e_1)/3.0;$
 $z_2 = (u_2[i+1][x] + b_2[x] + a_2 + 1/t[i] * (b_2[x] - a_2) * d + 1/t[i]^2 * D[b_2[x], x, 2] * d^2 - (b_1[x] * du2x + b_2[x] * du2y) * d^2/e_1 - dPy * d^2/e_1)/3.0;$
 $z_3 = (P[i+1][x] + b_3[x] + a_3 + 1/t[i] * (b_3[x] - a_3) * d + 1/t[i]^2 * D[b_3[x], x, 2] * d^2 + (du1x * du1x + du2y * du2y + 2.0 * du1y * du2x) * d^2)/3.0;$
11. $z_1 = \text{Series}[z_1, \{u_1[i+1][x], 0, 2\}, \{u_1[i+1]'[x], 0, 1\}, \{u_1[i+1]''[x], 0, 1\}, \{u_1[i+1]'''[x], 0, 0\}, \{u_1[i+1]''''[x], 0, 0\}, \{u_2[i+1][x], 0, 1\}, \{u_2[i+1]'[x], 0, 0\}, \{u_2[i+1]''[x], 0, 0\}, \{u_2[i+1]'''[x], 0, 0\}, \{u_2[i+1]''''[x], 0, 0\}, \{P[i+1][x], 0, 1\}, \{P[i+1]'[x], 0, 0\}, \{P[i+1]''[x], 0, 0\}, \{P[i+1]'''[x], 0, 0\}, \{P[i+1]''''[x], 0, 0\}, \{\text{Sin}[x], 0, 1\}, \{\text{Cos}[x], 0, 1\}];$
 $z_1 = \text{Normal}[z_1];$
 $z_1 = \text{Expand}[z_1];$
Print[z_1];
12. $z_2 = \text{Series}[z_2, \{u_1[i+1][x], 0, 1\}, \{u_1[i+1]'[x], 0, 1\}, \{u_1[i+1]''[x], 0, 1\}, \{u_1[i+1]'''[x], 0, 0\}, \{u_1[i+1]''''[x], 0, 0\}, \{u_2[i+1][x], 0, 2\}, \{u_2[i+1]'[x], 0, 0\}, \{u_2[i+1]''[x], 0, 0\}, \{u_2[i+1]'''[x], 0, 0\}, \{u_2[i+1]''''[x], 0, 0\}, \{P[i+1][x], 0, 1\}, \{P[i+1]'[x], 0, 0\}, \{P[i+1]''[x], 0, 0\}, \{P[i+1]'''[x], 0, 0\}, \{P[i+1]''''[x], 0, 0\}, \{\text{Sin}[x], 0, 1\}, \{\text{Cos}[x], 0, 1\}];$
 $z_2 = \text{Normal}[z_2];$
 $z_2 = \text{Expand}[z_2];$
Print[z_2];
13. $z_3 = \text{Series}[z_3, \{u_1[i+1][x], 0, 2\}, \{u_1[i+1]'[x], 0, 1\}, \{u_1[i+1]''[x], 0, 1\}, \{u_1[i+1]'''[x], 0, 0\}, \{u_1[i+1]''''[x], 0, 0\}, \{u_2[i+1][x], 0, 2\}, \{u_2[i+1]'[x], 0, 1\}, \{u_2[i+1]''[x], 0, 0\}, \{u_2[i+1]'''[x], 0, 0\}, \{u_2[i+1]''''[x], 0, 0\}, \{P[i+1][x], 0, 1\}, \{P[i+1]'[x], 0, 1\},$

- $\{P[i+1]''[x], 0, 0\}, \{P[i+1]'''[x], 0, 0\}, \{P[i+1]''''[x], 0, 0\},$
 $\{Sin[x], 0, 1\}, \{Cos[x], 0, 1\}\};$
- $z_3 = Normal[z_3];$
 $z_3 = Expand[z_3];$
 $Print[z_3];$
14. $b_1[x] = z_1;$
 $b_2[x] = z_2;$
 $b_3[x] = z_3;$
 $b_{11} = z_1;$
 $b_{12} = z_2;$
 $b_{13} = z_3;];$
15. $a_{11}[i] = b_{11};$
 $a_{12}[i] = b_{12};$
 $a_{13}[i] = b_{13};$
 $Print[a_{11}[i]];$
 $Clear[b_1, b_2, b_3];$
 $u_1[i+1][x] = b_1[x];$
 $u_2[i+1][x] = b_2[x];$
 $P[i+1][x] = b_3[x];$
 $a_1 = Series[b_{11}, \{t_3, 0, 1\}; \{b_1[x], 0, 1\}, \{b_2[x], 0, 1\}, \{b_3[x], 0, 1\},$
 $\{b'_1[x], 0, 0\}, \{b'_2[x], 0, 0\}, \{b'_3[x], 0, 0\}, \{b''_1[x], 0, 0\}, \{b''_2[x], 0, 0\}, \{b''_3[x], 0, 0\}\};$
 $a_1 = Normal[a_1];$
 $a_1 = Expand[a_1];$
 $a_2 = Series[b_{12}, \{t_3, 0, 1\}; \{b_1[x], 0, 1\}, \{b_2[x], 0, 1\}, \{b_3[x], 0, 1\},$
 $\{b'_1[x], 0, 0\}, \{b'_2[x], 0, 0\}, \{b'_3[x], 0, 0\}, \{b''_1[x], 0, 0\}, \{b''_2[x], 0, 0\}, \{b''_3[x], 0, 0\}\};$
 $a_2 = Normal[a_2];$
 $a_2 = Expand[a_2];$
 $a_3 = Series[b_{13}, \{t_3, 0, 1\}; \{b_1[x], 0, 1\}, \{b_2[x], 0, 1\}, \{b_3[x], 0, 1\},$
 $\{b'_1[x], 0, 0\}, \{b'_2[x], 0, 0\}, \{b'_3[x], 0, 0\}, \{b''_1[x], 0, 0\}, \{b''_2[x], 0, 0\}, \{b''_3[x], 0, 0\}\};$
 $a_3 = Normal[a_3];$
 $a_3 = Expand[a_3];$
16. $b_1[x] = -1.0 * Sin[x];$
 $b_2[x] = 1.0 * Cos[x];$
 $b_3[x] = 0.10;$
17. $For[i = 1, i < m8, i ++,$
 $A_{11} = a_{11}[m8 - i];$
 $A_{11} = Series[A_{11}, \{Sin[x], 0, 2\}, \{Cos[x], 0, 2\}\};$
 $A_{11} = Normal[A_{11}];$

```

A11 = Expand[A11];
A12 = a12[m8 - i];
A12 = Series[A12, {Sin[x], 0, 2}, {Cos[x], 0, 2}];
A12 = Normal[A12];
A12 = Expand[A12];
A13 = a13[m8 - i];
A13 = Series[A13, {Sin[x], 0, 2}, {Cos[x], 0, 2}];
A13 = Normal[A13];
A13 = Expand[A13];
18. u1[m8 - i][x_] = A11;
    u2[m8 - i][x_] = A12;
    P[m8 - i][x_] = Expand[A13];
19. t3 = 1.0;
20. Print["u1[" , m8 - i, "] = ", A11];
    Clear[t3];
    b1[x] = A11;
    b2[x] = A12;
    b3[x] = A13;];
21. t3 = 1.0;
22. For[i = 1, i < m8, i++,
    Print["u1[" , i, "] = ", u1[i][x]]]

```

Here the line expressions for the field of velocity $u = \{u_1[n](x)\}$, where gain we emphasize $N = 10$ lines:

1.

$$u_1[1](x) = 0.0044548 \cos[x] - 0.174091 \sin[x] + 0.00041254 \cos[x]^2 \sin[x] + 0.0260471 \cos[x] \sin[x]^2 - 0.000188598 \cos[x]^2 \sin[x]^2 \quad (22)$$

2.

$$u_1[2](x) = 0.00680614 \cos[x] - 0.331937 \sin[x] + 0.000676383 \cos[x]^2 \sin[x] + 0.0501544 \cos[x] \sin[x]^2 - 0.000176433 \cos[x]^2 \sin[x]^2 \quad (23)$$

3.

$$u_1[3](x) = 0.00775103 \cos[x] - 0.470361 \sin[x] + 0.000863068 \cos[x]^2 \sin[x] + 0.0682792 \cos[x] \sin[x]^2 - 0.000121656 \cos[x]^2 \sin[x]^2 \quad (24)$$

4.

$$u_1[4](x) = 0.00771379 \cos[x] - 0.589227 \sin[x] + 0.000994973 \cos[x]^2 \sin[x] + 0.0781784 \cos[x] \sin[x]^2 - 0.00006958 \cos[x]^2 \sin[x]^2 \quad (25)$$

5.

$$\begin{aligned} u_1[5](x) = & 0.00701567\cos[x] - 0.690152\sin[x] + 0.00106158\cos[x]^2\sin[x] \\ & + 0.0796091\cos[x]\sin[x]^2 - 0.0000330485\cos[x]^2\sin[x]^2 \end{aligned} \quad (26)$$

6.

$$\begin{aligned} u_1[6](x) = & 0.00589597\cos[x] - 0.775316\sin[x] + 0.00104499\cos[x]^2\sin[x] \\ & + 0.0734277\cos[x]\sin[x]^2 - 0.0000121648\cos[x]^2\sin[x]^2 \end{aligned} \quad (27)$$

7.

$$\begin{aligned} u_1[7](x) = & 0.00452865\cos[x] - 0.846947\sin[x] + 0.000931782\cos[x]^2\sin[x] \\ & + 0.0609739\cos[x]\sin[x]^2 - 2.74137 * 10^{-6}\cos[x]^2\sin[x]^2 \end{aligned} \quad (28)$$

8.

$$\begin{aligned} u_1[8](x) = & 0.00303746\cos[x] - 0.907103\sin[x] + 0.000716865\cos[x]^2\sin[x] \\ & + 0.0437018\cos[x]\sin[x]^2 \end{aligned} \quad (29)$$

9.

$$\begin{aligned} u_1[9](x) = & 0.00150848\cos[x] - 0.957599\sin[x] + 0.000403216\cos[x]^2\sin[x] \\ & + 0.0229802\cos[x]\sin[x]^2 \end{aligned} \quad (30)$$

5 Conclusion

In this article, we develop solutions for two-dimensional examples also for the time-independent incompressible Navier-Stokes system, through the generalized method of lines. We also obtain the appropriate boundary conditions for a equivalent elliptic system. Finally, the extension of such results to $\mathbb{R}^3$, compressible and time dependent cases is planned for a future work.

References

- [1] R.A. Adams and J.F. Fournier, Sobolev Spaces, 2nd edn. (Elsevier, New York, 2003).
- [2] F.S. Botelho, Functional Analysis, Calculus of Variations and Numerical Methods in Physics and Engineering, CRC Taylor and Francis, Florida, 2020.
- [3] F.S. Botelho, An Approximate Proximal Numerical Procedure Concerning the Generalized Method of Lines Mathematics 2022, 10(16), 2950; <https://doi.org/10.3390/math10162950>.
- [4] F. Botelho, *Topics on Functional Analysis, Calculus of Variations and Duality*, Academic Publications, Sofia, (2011).
- [5] F. Botelho, *Existence of solution for the Ginzburg-Landau system, a related optimal control problem and its computation by the generalized method of lines*, Applied Mathematics and Computation, 218, 11976-11989, (2012).

- [6] P. Constantin and C. Foias, Navier-Stokes Equation, University of Chicago Press, Chicago, 1989.
- [7] Makram Hamouda, Daozhi Han, Chang-Yeol Jung and Roger Temam, Boundary layers for the 3D primitive equations in a cube: the zero-mode, Journal of Applied Analysis and Computation, 8, No. 3, 2018, 873-889, DOI:10.11948/2018.873.
- [8] Andrea Giorgini, Alain Miranville and Roger Temam, Uniqueness and regularity for the Navier-Stokes-Cahn-Hilliard system, SIAM J. of Mathematical Analysis (SIMA), 51, 3, 2019, 2535-2574, <https://doi.org/10.1137/18M1223459>.
- [9] Ciprian Foias, Ricard M.S. Rosa and Roger M. Temam, Properties of stationary statistical solutions of the three-dimensional Navier-Stokes equations, J. of Dynamics and Differential Equations, Special issue in memory of George Sell, 31, 3, 2019, 1689-1741, <https://doi.org/10.1007/s10884-018-9719-2>.
- [10] J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, SIAM, second edition (Philadelphia, 2004).
- [11] R. Temam, *Navier-Stokes Equations*, AMS Chelsea, reprint (2001).