Bivariate Splines for Fluid Flows *)

Ming-Jun Lai and Paul Wenston

Abstract. We discuss numerical approximations of the 2D steady state Navier Stokes equations in stream function formulation using bivariate splines of arbitrary degree d and arbitrary smoothness r with $r < d$. We derive the discrete Navier-Stokes equations in terms of B-coefficients of bivariate splines over a triangulation, with curved boundary edges, of any given domain. Smoothness conditions and boundary conditions are enforced through Lagrange multipliers. The pressure is computed by solving a Poisson equation with Neumann boundary conditions. We have implemented this approach in MATLAB and our numerical experiments show that our method is effective. Numerical simulations of several fluid flows will be included to demonstrate the effectiveness of the bivariate spline method.

Table of Contents

§1. Introduction

One of the major tasks for applied mathematicians is to develop efficient methods for numerical solutions of partial differential equations. The steady-state Navier-Stokes equations are one of the most important partial differential equations which have various applications. Although there are many computational methods available in the literature for the numerical solution of the Navier-Stokes equations, new and more efficient methods continue to be developed in order to increase the power of computational flow simulations. For example, in a recent paper [Botella'02], a collocation B-spline method was developed for the solution of the Navier-Stokes equations. In this paper, we use bivariate spline functions over arbitrary triangulations for numerical solution of 2D Navier-Stokes equations. (See [Awanou and Lai'02] for the trivariate spline approximation of 3D Navier-Stokes equations.) Our approach is like the finite element method using triangles to approximate any given 2D polygonal domains and using piecewise polynomials over triangulations to approximate the solution of the Navier-Stokes equations. The main different features are:

*) Department of Mathematics, The University of Georgia, Athens, GA 30602, gawanou@math.uga.edu, mjlai@math.uga.edu, paul@math.uga.edu

(1) no macro-element or locally supported spline functions are constructed;
(2) polynomials of high degrees can be easily used to get a better approximation power;
(3) smoothness can be imposed in a flexible way across the domain at places where the solution is expected to be smooth. For example, the solution of the steady state Navier-Stokes equation is H^2 inside the domain and H^1 near the boundary (cf. [Serrin'62]);
(4) the mass and stiffness matrices can be assembled easily and these processes can be done in parallel;
(5) The stream function formulation will be used and thus the spline approximation of the solution of Navier-Stokes equations satisfies the divergence-free condition exactly;
(6) the matrices that arise are singular which is an important difference from the classical finite element method.
(7) Our spline method leads to a linear system of special structure. We introduce a special numerical method to solve such particularly structured linear systems.

Let us first introduce the stream function formulation. Let $\Omega \subseteq \mathbf{R}^2$ be a simply connected domain and $\mathbf{u} = (u_1, u_2)^T$ be the planar velocity of a fluid flow over Ω. Also, let p be the pressure function, $\mathbf{f} = (f_1, f_2)^T$ be the external body force of the fluid and $\mathbf{g} = (g_1, g_2)^T$ be the velocity of the fluid flow on the boundary $\partial\Omega$. Then the steady state Navier-Stokes equations are

$$\begin{cases} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f}, & (x,y) \in \Omega \\ \text{div}\mathbf{u} = 0, & (x,y) \in \Omega \\ \mathbf{u} = \mathbf{g}, & (x,y) \in \partial\Omega, \end{cases} \tag{1.1}$$

where Δ denotes the usual Laplacian operator and ∇ the gradient operator. After omitting the nonlinear terms, we have the steady state Stokes' equations:

$$\begin{cases} -\nu\Delta\mathbf{u} + \nabla p = \mathbf{f}, & (x,y) \in \Omega \\ \text{div}\mathbf{u} = 0, & (x,y) \in \Omega \\ \mathbf{u} = \mathbf{g} & (x,y) \in \partial\Omega\ . \end{cases} \tag{1.2}$$

Recall the fact that there exists a stream function φ such that $\mathbf{u} = \mathbf{curl}\ \varphi$, i.e., $u_1 = \dfrac{\partial\varphi}{\partial y}$, $u_2 = -\dfrac{\partial\varphi}{\partial x}$. Such φ is unique up to a constant (cf. [Girault and Raviart'86, pp. 37–39].) Thus we may simplify the above Stokes and Navier-Stokes equations by cancelling the pressure term. Consider the Stokes equations first. Replacing $\mathbf{u}$ by $\mathbf{curl}\varphi$ and then differentiating the first equation with respect to y and the second with respect to x, we substract the first equation from the second one to obtain the following fourth order equation

$$\nu\Delta^2\varphi = h$$

with $h = \frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}$. Thus, the Stokes equations become a biharmonic equation:

$$\begin{cases} \nu\Delta^2\varphi = h, & \text{in } \Omega \\ \dfrac{\partial\varphi}{\partial x} = -g_2, & \text{on } \partial\Omega \\ \dfrac{\partial\varphi}{\partial y} = g_1, & \text{on } \partial\Omega \\ \varphi = b_2, & \text{on } \partial\Omega \end{cases} \tag{1.3}$$

where b_2 is an anti-derivative of the tangential derivative of φ along $\partial\Omega$ and will be examined in detail later. By a similar calculation, we easily see that the Navier-Stokes equations become the following fourth order nonlinear equation

$$\begin{cases} \nu\Delta^2\varphi - \dfrac{\partial}{\partial y}\left(\dfrac{\partial\varphi}{\partial y}\dfrac{\partial^2\varphi}{\partial x\partial y} - \dfrac{\partial\varphi}{\partial x}\dfrac{\partial^2\varphi}{\partial y^2}\right) & \\ \qquad -\dfrac{\partial}{\partial x}\left(\dfrac{\partial\varphi}{\partial y}\dfrac{\partial^2\varphi}{\partial x^2} - \dfrac{\partial\varphi}{\partial x}\dfrac{\partial^2\varphi}{\partial x\partial y}\right) = h, & \text{in } \Omega \\ \dfrac{\partial\varphi}{\partial x} = -g_2, & \text{on } \partial\Omega \\ \dfrac{\partial\varphi}{\partial y} = g_1, & \text{on } \partial\Omega \\ \varphi = b_2, & \text{on } \partial\Omega\ . \end{cases} \tag{1.4}$$

Let $H^2(\Omega)$ be the usual Sobolev space and $H_0^2(\Omega)$ be the subspace of $H^2(\Omega)$ of functions whose derivatives of order less than or equal to one all vanish on the boundary $\partial\Omega$. Define the bilinear form $a_2(\varphi,\psi)$ and trilinear form $q(\theta,\varphi,\psi)$ by

$$a_2(\varphi,\psi) = \int_\Omega \Delta\varphi(x,y)\Delta\psi(x,y)dxdy$$

$$q(\theta,\varphi,\psi) = \int_\Omega \Delta\theta(x,y)\left(\frac{\partial\varphi(x,y)}{\partial x}\frac{\partial\psi(x,y)}{\partial y} - \frac{\partial\varphi(x,y)}{\partial y}\frac{\partial\psi(x,y)}{\partial x}\right)dxdy$$

and denote the $L_2(\Omega)$ inner product by

$$\langle h,\psi\rangle = \int_\Omega h(x,y)\psi(x,y)dxdy.$$

We that say $\varphi \in H^2(\Omega)$ is a weak solution of the Stokes equations (1.3) if φ satisfies the following

$$\begin{cases} \nu a_2(\varphi,\psi) = \langle h,\psi\rangle, & \forall\psi \in H_0^2(\Omega) \\ \dfrac{\partial\varphi}{\partial x} = -g_2, & \text{on } \partial\Omega \\ \dfrac{\partial\varphi}{\partial y} = g_1, & \text{on } \partial\Omega \\ \varphi = b_2, & \text{on } \partial\Omega\ . \end{cases}$$

Similarly, a function $\varphi \in H^2(\Omega)$ is a weak solution of the Navier-Stokes equations (1.4) if φ satisfies

$$\begin{cases} \nu a_2(\varphi,\psi) + q(\varphi,\varphi,\psi) = \langle h,\psi\rangle, & \forall \psi \in H_0^2(\Omega) \\ \dfrac{\partial \varphi}{\partial x} = -g_2, & \text{on } \partial\Omega \\ \dfrac{\partial \varphi}{\partial y} = g_1, & \text{on } \partial\Omega \\ \varphi = b_2, & \text{on } \partial\Omega \ . \end{cases}$$

Such weak formulations are referred as the stream function formulation of the Stokes and Navier-Stokes equations, respectively. It is known that the weak solution for Stokes' equations exists and is unique for any $\nu > 0$. For the Navier-Stokes equations, such a weak solution exists for any $\nu > 0$, and is unique when ν is sufficiently large. (See, e.g., [Girault and Raviart'86].)

There are two major advantages using the stream function formulation over the traditional velocity-pressure formulation and vorticity-stream function formulation. Indeed, with the stream function formulation, we need to approximate only one stream function. Otherwise, we need to approximate two components of the velocity and one pressure function if the velocity-pressure formulatio is used or one vorticity and one stream function if the vorticity-stream function formulation is used. In addition, the stream function formulation eliminates the pressure function which does not have appropriate boundary condition. In the vorticity-stream function formulation, the vorticity function does not have appropriate boundary condition. With bivariate spline functions of higher degrees, we are able to approximate stream functions very well. Thus, in this paper, we will use bivariate splines to approximate the stream function of the Stokes and Navier-Stokes equations.

Next we discuss the computation of the pressure functions. By taking divergence, div, of the equations in (1.1) and (1.2), we can easily see that the pressure functions p of the Stokes and Navier-Stokes equations satisfy the following Poisson equations with nonhomogeneous Neumann boundary conditions involving the stream functions.

$$\begin{cases} -\Delta p = -\text{div}(\mathbf{f}), & \text{in } \Omega \\ \dfrac{\partial p}{\partial n} = \nu\Delta(n\cdot \mathbf{curl}\varphi) + n\cdot \mathbf{f} & \text{on } \partial\Omega \end{cases} \tag{1.5}$$

for the Stokes equation and

$$\begin{cases} -\Delta p = -\text{div}(\mathbf{f}) + \text{div}[(\mathbf{curl}\varphi\cdot\nabla)\mathbf{curl}(\varphi)], & \text{in } \ \Omega \\ \dfrac{\partial p}{\partial n} = \nu\Delta(n\cdot\mathbf{curl}\varphi) + n\cdot\mathbf{f} + n\cdot[(\mathbf{curl}\varphi\cdot\nabla)(\mathbf{curl}\varphi)], & \text{on } \partial\Omega \end{cases} \tag{1.6}$$

for the Navier-Stokes equations. We will use these boundary conditions to compute p aftey we find the spline approximations of the stream function φ.

The paper is organized as follows: In §2, we first introduce bivariate spline spaces and in particular, the B-form representation of spline functions. The constrained minimization problems that will be used to generate approximate solutions of the stokes and Navier Stoke equations lead, by way of Lagrange multipliers, to linear systems of the form

$$\begin{bmatrix} L^T & A \\ 0 & L \end{bmatrix} \begin{bmatrix} \lambda \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix},$$

with A singular. We then introduce an iterative method to solve the above system and discuss its convergence. We continue in §3 with a discussion of the spline solution of the 2D Stokes equations, an equivalent biharmonic equation. In §4, we discuss the spline approximations of the 2D Navier-Stokes equations and prove the convergence of two methods for solving the discrete nonlinear equations. In §5, we present numerical results for standard benchmark flows such as the driven cavity flows, backward-facing step flows, and flows around circular obstacles. Also, we also present some other fluid flow experiments.

§2. Preliminaries

2.1. Bivariate Spline Functions

Given a bounded polygonal domain $\Omega \in \mathbb{R}^2$, let $\triangle$ be a triangulation of Ω. Let $d \geq 1$ and $r \geq -1$ be two fixed integers. We introduce the spline spaces

$$S_d^r(\triangle) = \{s \in C^r(\Omega), \; s|_t \in \mathbb{P}_d, \; \forall t \in \triangle\},$$

where $\mathbb{P}_d$ denotes the space of bivariate polynomials of total degree d.

In this paper, the B-form representation of splines on triangulations will be used (cf. [Farin'86] or [de Boor'87]). Let $T = \langle v_1, v_2, v_3 \rangle$ be a non-degenerate triangle with $\mathbf{v}_i = (x_i, y_i)$, $i = 1, 2, 3$. It is well-known that every point $\mathbf{v} = (x, y)$ can be written uniquely in the form

$$\mathbf{v} = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3, \tag{2.1}$$

with

$$\lambda_1 + \lambda_2 + \lambda_3 = 1, \tag{2.2}$$

where λ_1, λ_2, and λ_3 are called the barycentric coordinates of the point $\mathbf{v} = (x, y)$ relative to the triangle T. Moreover each λ_i is a linear polynomial in x, y. Let

$$B_{ijk}^d(v) = \frac{d!}{i!j!k!} \lambda_1^i \lambda_2^j \lambda_3^k, \qquad i + j + k = d.$$

They are called the Bernstein-Bézier polynomials of degree d. In fact, the set

$$\mathcal{B}^d = \{B_{ijk}^d(x, y, z), \quad i + j + k = d\}$$

is a basis for the space of polynomials $\mathbb{P}_d$. As a consequence any polynomial p of degree d can be written uniquely in terms of B^d_{ijk}'s, i.e.,

$$p = \sum_{i+j+k=d} c_{ijk} B^d_{ijk}. \tag{2.3}$$

The representation (2.3) for polynomials is referred to as the B-form with respect to T. Let

$$\mathcal{D}_{d,T} = \{\xi_{ijk} = \frac{i\mathbf{v}_1 + j\mathbf{v}_2 + k\mathbf{v}_3}{d}, i+j+k=d, T \in \triangle\} \tag{2.4}$$

be a set of the domain points of degree d over triangulation $\triangle$. For each spline function $s \in S^r_d(\triangle)$, since s restricted to each triangle $T \in \triangle$ is a polynomial of degree d, we may write

$$s|_T = \sum_{i+j+k=d} c^T_{ijk} B^d_{ijk}, \quad T \in \triangle.$$

Such a representation is called the B-form representation of the spline function s (cf. [de Boor'87]). We denote by $\mathbf{c} := \{c^T_{ijk}, i+j+k=d, T \in \triangle\}$ the B-coefficient vector of s.

To evaluate a polynomial in B-form, there is the so-called de Casteljau algorithm which we now describe. For $p = \sum\limits_{i+j+k=d} c_{ijk} B^d_{ijk}$, let us write $c_{ijk} =: c^{(0)}_{ijk}(\lambda)$ with $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ being the barycentric coordinates of $\mathbf{v} = (x, y)$ with respective to T and define for a positive integer $r \geq 1$

$$\begin{aligned} c^{(r)}_{ijk}(\lambda) &= \lambda_1 c^{(r-1)}_{i+1,j,k}(\lambda) + \lambda_2 c^{(r-1)}_{i,j+1,k}(\lambda) \\ &\quad + \lambda_3 c^{(r-1)}_{i,j,k+l}(\lambda). \end{aligned}$$

We have then

$$p = \sum_{i+j+k=d-r} c^{(r)}_{ijk}(\lambda) B^{d-r}_{ijk}, \quad 0 \leq r \leq d.$$

In particular, for $r = d$, we have

$$p = c^{(d)}_{0,0,0}(\lambda)$$

which is the value of p at $\mathbf{v} = (x, y)$ whose barycentric coordinates are $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ with $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

We next discuss how to take derivatives of polynomials in B-form. We start with formulas for the directional derivatives of p in a direction defined by a vector $\mathbf{u}$. We have:

$$D_{\mathbf{u}} p = d \sum_{i+j+k=d-1} c^{(1)}_{ijk}(\mathbf{a}) B^{d-1}_{ijk},$$

with $\mathbf{a} = (a_1, a_2, a_3)$ the T-coordinates of $\mathbf{u}$; that is,

$$\mathbf{u} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3,$$

with

$$a_1 + a_2 + a_3 = 1.$$

Note that if $\mathbf{u} = \mathbf{v}_1 - \mathbf{v}_2$ is the direction vector, the T-coordinates of $\mathbf{u}$ are $(1, -1, 0)$. In general, we have

$$D^m_{\mathbf{u}} p(\mathbf{v}) = \frac{d!}{(d-m)!} \sum_{i+j+k=d-m} c^{(m)}_{ijk}(\mathbf{a}) B^{d-m}_{ijkl}(\mathbf{v}). \tag{2.5}$$

Note that for arbitrary direction vector $u = (u_1, u_2) \in \mathbb{R}^2$,

$$D_{\mathbf{u}} p = u_1 \frac{\partial}{\partial x} p + u_2 \frac{\partial}{\partial y} p.$$

For a triangle $T = \langle v_1, v_2, v_3 \rangle$, if we let $\mathbf{u} = \mathbf{v}_2 - \mathbf{v}_1 = (x_2 - x_1, y_2 - y_1)$, $\mathbf{v} = \mathbf{v}_3 - \mathbf{v}_1 = (x_3 - x_1, y_3 - y_1)$, it follows that

$$\begin{aligned} D_x p =& \frac{(y_3 - y_1)}{2A_T} D_{\mathbf{u}} p - \frac{(y_2 - y_1)}{2A_T} D_{\mathbf{v}} p \\ D_y p =& \frac{(x_2 - x_1)}{2A_T} D_{\mathbf{v}} p - \frac{(x_3 - x_1)}{2A_T} D_{\mathbf{u}} p, \end{aligned} \tag{2.6}$$

where A_T is the area of T.

There are precise formulas for the integrals and inner products of polynomials in B-form (cf. [Chui and Lai'90]).

Lemma 2.1. *Let p be a polynomial of degree d with B-coefficients c_{ijk}, $i+j+k = d$ on a triangle T. Then*

$$\int_T p(x,y)\,dxdy = \frac{A_T}{\binom{d+2}{2}} \sum_{i+j+k=d} c_{ijk},$$

where $A_T =$ area of T .

Lemma 2.2. *Let q be another polynomial with B-coefficients $d_{ijk}, i+j+k = d$, the inner product of p and q over t is given by*

$$\int_t p(x,y)q(x,y)\,dxdy = \frac{A_T}{\binom{2d}{d}\binom{2d+2}{2}} \sum_{\substack{i+j+k=d \\ r+s+t=d}} c_{ijk} d_{rst} \binom{i+r}{i}\binom{j+s}{j}\binom{k+t}{k}. \tag{2.7}$$

We next discuss the smoothness conditions for a spline function s in $S^r_d(\triangle)$. These are well-known conditions on the coefficients of s that will assure that s has certain global smoothness properties.

Theorem 2.3. *Let $t = \langle v_1, v_2, v_3 \rangle$ and $t' = \langle v_1, v_2, v_4 \rangle$ be two triangles with common edge $\langle v_1, v_2 \rangle$. Then s is of class C^r on $t \cup t'$ if and only if*

$$c^{t'}_{ijm} = \sum_{\mu+\nu+\kappa=m} c^t_{i+\mu,j+\nu,\kappa} B^m_{\mu,\nu,\kappa}(v_4), \ m = 0, \ldots, r, \ i + j = d - m.$$

For a proof, see [Farin'86] or [de Boor'87]. This theorem guarantees the existence of a matrix H such that s is in $C^r(\Omega)$ if and only if

$$H\mathbf{c} = 0,$$

where $\mathbf{c}$ encodes the B-coefficients of s.

2.2. A Matrix Iterative Method

The constrained minimization problems that will be used to generate approximate solutions of the stokes and Navier Stoke equations will lead, by way of Lagrange multipliers, to linear systems of the form

$$\begin{bmatrix} L^T & A \\ 0 & L \end{bmatrix} \begin{bmatrix} \lambda \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix}, \tag{2.8}$$

with A singular and appropriate matrices L and vectors F and G. Computing a least squares solution of the above system would allow us to compute $\mathbf{c}$ when $\mathbf{c}$ is unique. However, when the size of a system is very large, any least squares method will be too costly. We present here an iterative method for solving the above system which involves matrices of smaller size. Consider the following sequence of problems:

$$\begin{bmatrix} L^T & A \\ -\epsilon I & L \end{bmatrix} \begin{bmatrix} \lambda^{(k+1)} \\ \mathbf{c}^{(k+1)} \end{bmatrix} = \begin{bmatrix} F \\ G - \epsilon\lambda^{(k)} \end{bmatrix}, \tag{2.9}$$

for $k = 0, 1, \ldots$, with an initial guess $\lambda^{(0)}$, e.g., $\lambda^{(0)} = 0$, and I the identity matrix. Assume that the size of L is $m \times n$ and the size of A is $n \times n$ with $m < n$. Note that (2.9) reads

$$\begin{aligned} A\mathbf{c}^{(k+1)} + L^T\lambda^{(k+1)} &= F \\ L\mathbf{c}^{(k+1)} - \epsilon\lambda^{(k+1)} &= G - \epsilon\lambda^{(k)}. \end{aligned} \tag{2.10}$$

Multiplying on the left the second equation in (2.10) by L^T, we get

$$L^T L\mathbf{c}^{(k+1)} - \epsilon L^T\lambda^{(k+1)} = L^T G - \epsilon L^T \lambda^{(k)}$$

or $L^T\lambda^{(k+1)} = \frac{1}{\epsilon}L^T L\mathbf{c}^{(k+1)} - \frac{1}{\epsilon}L^T G + L^T\lambda^{(k)}$. We substitute this into the first equation in (2.10) to get

$$(A + \frac{1}{\epsilon}L^T L)\mathbf{c}^{(k+1)} = F + \frac{1}{\epsilon}L^T G - L^T\lambda^{(k)}. \tag{2.11}$$

Using again the first equation in (2.10), i.e., $A\mathbf{c}^{(k)} = F - L^T\lambda^{(k)}$ to replace F in (2.11), we have

$$(A + \frac{1}{\epsilon}L^T L)\mathbf{c}^{(k+1)} = A\mathbf{c}^{(k)} + \frac{1}{\epsilon}L^T G,$$

for all $k \geq 1$. This leads to the following

Algorithm 2.4. *Fix $\epsilon > 0$. Given an initial guess $\lambda^{(0)} \in Im(L)$, e.g., $\lambda^{(0)} = 0$, we define $\mathbf{c}^{(1)}$ by*

$$\mathbf{c}^{(1)} = (A + \frac{1}{\epsilon} L^T L)^{-1} (F + \frac{1}{\epsilon} L^T G - L^T \lambda^{(0)})$$

and iteratively define

$$\mathbf{c}^{(k+1)} = (A + \frac{1}{\epsilon} L^T L)^{-1} (A\mathbf{c}^{(k)} + \frac{1}{\epsilon} L^T G), \tag{2.12}$$

for $k = 1, 2, \ldots$, where $Im(L)$ is the range of L.

This algorithm was briefly discussed in [Gunzburger'89]. A convergence result was pointed out. That is, $\|\mathbf{c} - \mathbf{c}^{k+1}\| \leq C\epsilon \|\mathbf{c} - \mathbf{c}^{(k)}\|$ for some constant $C > 0$. According to this result for convergence one clearly needs to choose ϵ small enough to guarantee that $C\epsilon < 1$. Here we present the following convergence result (cf. [Awanou and Lai'02] for a proof) which ensures the convergence for any $\epsilon > 0$.

Theorem 2.5. *Let $A = A_s + A_a$ where $A_s = \frac{1}{2}(A + A^T)$ is the symmetric part of A and $A_a = \frac{1}{2}(A - A^T)$ is the anti-symmetric part of A. Furthermore, suppose that A_s is positive definite with respect to L, that is, $\mathbf{x}^T A_s \mathbf{x} = 0$ and $L\mathbf{x} = 0$ imply that $\mathbf{x} = 0$. Then for any $\epsilon > 0$, the matrix*

$$A + \frac{1}{\epsilon} L^T L$$

is invertible. That is, the Algorithm 2.4 is well-defined. Furthermore, there exists positive constants $C(\epsilon) > 1$ and $\gamma(\epsilon) < 1$ depending on ϵ but independent of k such that

$$\|\mathbf{c} - \mathbf{c}^{(k+1)}\| \leq C(\epsilon) \left(\gamma(\epsilon)\right)^{k+1},$$

for $k \geq 1$.

Our numerical experiments show that it gives the same result as the least squares method. Since any least squares method is computationally expensive, we recommend the iterative method for large problems.

§3. Spline Approximations of the Stokes Equations

In this section, we consider spline approximations of the 2D Stokes equations in the stream function formulation. That is, we consider the following biharmonic equation with nonhomogeneous boundary conditions g_1 and g_2:

$$\begin{cases} \nu a_2(\varphi, \psi) = \langle h, \psi \rangle, & \forall \psi \in H_0^2(\Omega) \\ \dfrac{\partial \varphi}{\partial x} = -g_2, & \text{on } \partial\Omega \\ \dfrac{\partial \varphi}{\partial y} = g_1, & \text{on } \partial\Omega \\ \varphi = b_2, & \text{on } \partial\Omega, \end{cases} \tag{3.1}$$

where b_2 is an anti-derivative of the tangential derivative of φ along $\partial\Omega$ which can be computed based on g_1 and g_2. For simplicity, let b_1 be the normal derivative of φ along $\partial\Omega$. Note that b_1 can be easily computed using g_1 and g_2.

It is well-known that the weak solution of (3.1) exists and is unique (cf. [Girault and Raviart'86]). We are going to approximate the weak solution of (3.1) by using bivariate spline functions. Let $\triangle$ be a triangulation of Ω if Ω is a polygonal domain. Otherwise, we choose vertices $v_1, \cdots, v_n$ on $\partial\Omega$ which include all the corner points of $\partial\Omega$. Instead of the piecewise linear interpolation of the vertices, we use a C^0 piecewise quadratic spline curve $s_{\partial\Omega}$, which interpolates the vertices, to approximate the boundary $\partial\Omega$. Clearly, we may assume that $s_{\partial\Omega}$ is a much better approximation of $\partial\Omega$ than the piecewise linear interpolant based on an appropriate choice of $v_1, \cdots, v_n$. Adding interior vertices $v_{n+1}, \cdots, v_N$ inside Ω, we triangulate Ω using the vertices $v_1, \cdots, v_N$. For convenience, we let $\triangle$ be the triangulation consisting of all interior triangles and boundary triangles with piecewise quadratic edges.

Let d and r be two positive integer with $d > r$ and let $\mathcal{S} \subset S_d^0(\triangle)$ be the space of spline functions which are C^r inside of Ω. Note that $\mathcal{S} \subset H^2(\Omega)$. Recall from §2.1 that $s \in S_d^0(\triangle)$ can be given by

$$s(x,y) = \sum_{i+j+k=d} c^t_{i,j,k} B^t_{i,j,k}(x,y), \forall (x,y) \in t \in \triangle.$$

Let $\mathbf{c} = (c^t_{i,j,k}, i+j+k=d, t \in \triangle)$ be the B-coefficient vector of s. Recall from §2.1 that there is a matrix H such that if $s \in \mathcal{S}$, then

$$H\mathbf{c} = 0.$$

Although in general we may not be able to find a spline function $s \in \mathcal{S}$ satisfying the boundary conditions exactly, we can find a spline $S_b \in \mathcal{S}$ approximating the boundary conditions. If the the boundary conditions are continuous, the interpolation is the obvious choice. Assuming that $v_1, \cdots, v_b$ are arranged in the counterclockwise direction, S_b interpolates b_2 at $d+1$ distinct points on any straight boundary edge $\langle v_i, v_{i+1}\rangle$ and at $2d+1$ distinct points on any quadratic boundary curved edge $v_i \smile v_{i+1}$. Also $\frac{\partial S_b}{\partial n_i}$ interpolates b_1 at d distinct points on any straight boundary edge $\langle v_i, v_{i+1}\rangle$ and at $2d-1$ disctinct points on any quadratic boundary curved edge $v_i \smile v_{i+1}$, where n_i denotes the outer normal direction of boundary edge $\langle v_i, v_{i+1}\rangle$ or $v_i \smile v_{i+1}$. If the boundary conditions are not continuous, we can still use interpolation except that the interpolation values now come from a conveniently choosen smooth approximation of the bounday conditions. Since the values of any spline and its normal derivative at any of the interpolation points are linear combinations of the B-coefficients of the spline, we may use

$$B\mathbf{c} = \mathbf{b} \tag{3.2}$$

to express the approximate boundary conditions. Let $\mathcal{S}_0 = \mathcal{S} \cap H_0^2(\Omega)$. Our spline approximation of the biharmonic equation (3.1) is to find $s_{h,\mathbf{b}} \in \mathcal{S}$ satisfying the approximate boundary condition (3.2) such that

$$\nu a_2(s_{h,\mathbf{b}}, s) = \langle h, s\rangle, \forall s \in \mathcal{S}_0. \tag{3.3}$$

Using the Lax-Milgram Theorem, problem (3.3) has an unique solution $s_{h,\mathbf{b}}$ in $\mathcal{S}$. Based on the theory of elliptic equations (cf. [Evans'98]), solving problem (3.3) is equivalent to finding $\mathbf{c}$ which minimizes the following energy functional

$$E(s) = \frac{\nu}{2} a_2(s,s) - \langle h, s\rangle$$

subject to $s \in \mathcal{S}$ satisfying the approximation conditions. In terms of B-coefficients, the above energy functional can be approximated by

$$\tilde{E}(\mathbf{c}) := \frac{\nu}{2}\mathbf{c}^T A\mathbf{c} - \mathbf{c}^T M\mathbf{h}$$

subject to $H\mathbf{c} = 0$ and $B\mathbf{c} = \mathbf{b}$, where $A =$ diag $(A_t, t \in \triangle)$ is a bending matrix with blocks

$$A_t = \left[\int_t \Delta B^t_{i,j,k} \Delta B^t_{l,m,n} dxdy\right]_{\substack{i+j+k=d\\ l+m+n=d}}, t \in \triangle,$$

$M =$ diag $(M_t, t \in \triangle)$ is a mass matrix with blocks

$$M_t = \left[\int_t B^t_{i,j,k} B^t_{l,m,n} dxdy\right]_{\substack{i+j+k=d\\ l+m+n=d}}, t \in \triangle$$

and $\mathbf{h} = (h^t_{l,m,n}, l+m+n=d, t \in \triangle)$ is a vector with

$$s_h = \sum_{l+m+n=d} h_{l,m,n} B^t_{l,m,n} dxdy \in S^0_d(\triangle)$$

the spline interpolation of h over the domain points $\{\xi^t_{l,m,n}, l+m+n=d, t \in \triangle\}$. We note that the mass and bending matrices, M and A, are diagonal block matrices whose blocks can be assembled trivially and in parallel. The matrix H is sparse with each row involving at most $(r+1)(r+2)/2+1$ nonzero elements. Also, the matrix B is very sparse with each row containing at most $d+1$ nonzero elements.

Following the Lagrange multiplier method, we let

$$L(\mathbf{c}, \alpha, \beta) = \frac{\nu}{2}\mathbf{c}^T A\mathbf{c} + \alpha^T H\mathbf{c} + \beta^T B\mathbf{c} - \mathbf{c}^T M\mathbf{h}$$

and compute local minimizers. It follows that we need to solve the following

$$\begin{bmatrix} H^T & B^T & \nu A \\ 0 & 0 & H \\ 0 & 0 & B \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} M\mathbf{h} \\ 0 \\ \mathbf{b} \end{bmatrix}.$$

We note that the existence and uniqueness of $s_{h,\mathbf{b}}$ implies that the there eixsts a unique solution $\mathbf{c}$ satisfying the above linear system. Clearly, A is positive definite with respect to $[H; B]$. Thus, our iterative method introduced in §2.2 can be applied to this system and the iterative solutions converge.

Next we use the approach presented above to compute the pressure function assuming that we have found the approximation of the stream function φ. That is, we need to solve the Poisson problem with Neumann boundary conditions (1.5). Note that we are seeking the pressure in

$$L_0^2(\Omega) = \{p \in L^2(\Omega), \int_\Omega p = 0\}.$$

Again we can use the spline space $\mathcal{S}$ as an approximating space. For a spline approximation s_p of p with B-coefficient vector c_p, its integral over a triangle is simply a weighted sum of its B-coefficients (cf. Lemma 2.1), hence the spline approximation of the zero mean value condition, $\int_\Omega p = 0$, can be written

$$Uc = 0,$$

with U a vector of size N. The pressure is also the solution of a minimization problem, namely that of minimizing the functional

$$P(v) = \frac{1}{2}\int_\Omega |\Delta v|^2 - \int_\Omega (-hv - \int_{\partial\Omega} (\mathbf{f} \cdot \mathbf{n}v + \nu\Delta(\mathbf{curl}\varphi \cdot \mathbf{n})v$$

over $L_0^2(\Omega)$.

We now write P in terms of $\mathbf{c}_p$. Let G_p interpolate $\mathbf{f} \cdot \mathbf{n} + \nu(\Delta\mathbf{curl}\varphi) \cdot \mathbf{n}$ on the boundary. The later is computed by using the spline approximation $s_{h,\mathbf{b}}$ of the solution φ of the Stokes equations. Let R be the matrix which restricts the B-coefficients c of a spline over Ω to the B-coefficients of the spline restricted to $\partial\Omega$. We have

$$P(c_p) = \frac{1}{2}(c_p)^T K c_p - \mathbf{h}^T M c_p - G_p^T M_b R c_p,$$

where K is the stiffness matrix which is a diagonal block matrix similar to A and M_b is the mass matrix for the bivariate splines of degree d over the edges of $\partial\Omega$. Following the Lagrange multiplier method, we need to solve the following linear system:

$$\begin{bmatrix} U & H^T & K \\ 0 & 0 & H \\ 0 & 0 & U \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ c_p \end{bmatrix} = \begin{bmatrix} M\mathbf{h} + R^T M_b R G_p \\ 0 \\ 0 \end{bmatrix}.$$

Again, it is easy to see that K is positive definite with respect to $[H; U]$. The iterative method described in §2.2 converges.

§4. Spline Approximations of the Navier-Stokes Equations

In this section, we consider the bivariate spline approximation of the 2D Navier-Stokes equations in the stream function formulation. Let $\varphi \in H^2(\Omega)$ be the weak solution of the Navier-Stokes equations

$$
\begin{cases}
\nu a_2(\varphi,\psi) + q(\varphi,\varphi,\psi) = \langle h, \psi\rangle, & \forall \psi \in H_0^2(\Omega) \\
\dfrac{\partial \varphi}{\partial x} = -g_2, & \text{on } \partial\Omega \\
\dfrac{\partial \varphi}{\partial y} = g_1, & \text{on } \partial\Omega \\
\varphi = b_2, & \text{on } \partial\Omega\ .
\end{cases}
\tag{4.1}
$$

It is known that the weak solution φ exists and is unique if ν is sufficiently large or h is sufficiently small (cf. [Girault and Raviart'86]). We shall use the approach discussed in the previous section to derive spline approximations of the stream function φ.

With the same spline space $\mathcal{S}$ and the same notation as in the previous section, we have

$$a_2(\varphi,\psi) = \mathbf{c}^T A \mathbf{d},$$

where $\mathbf{c}$ and $\mathbf{d}$ stand for the B-coefficient vectors of spline functions $\varphi, \psi \in \mathcal{S}$. The difference with the previous section is the presence of the nonlinear term, the trilinear form q defined by

$$q(\theta,\varphi,\psi) = \int_\Omega \Delta\theta(x,y)\left(\frac{\partial\varphi(x,y)}{\partial x}\frac{\partial\psi(x,y)}{\partial y} - \frac{\partial\varphi(x,y)}{\partial y}\frac{\partial\psi(x,y)}{\partial x}\right) dxdy. \tag{4.2}$$

Let us first get acquainted with the trilinear form (4.2). Let $\mathbf{e}$ encode the B-coefficient vector of the spline function θ in $\mathcal{S}$. We can, by tedious computations, show that there is a matrix $Q(\mathbf{e})$, which is a linear function of $\mathbf{e}$, such that

$$q(\theta,\varphi,\psi) = \mathbf{d}^T Q(\mathbf{e})\mathbf{c}. \tag{4.3}$$

There are several useful properties of $Q(\mathbf{e})$ given below.

Lemma 4.1. *$Q(\mathbf{e})$ is anti-symmetric, i.e., $Q(\mathbf{e})^T = -Q(\mathbf{e})$.*

Proof: We first note that $q(\theta,\varphi,\psi) = -q(\theta,\psi,\varphi)$. In terms of B-coefficient vectors, we have

$$\mathbf{d}^T Q(\mathbf{e})\mathbf{c} = -\mathbf{c}^T Q(\mathbf{e})\mathbf{d}.$$

Thus, it follows from $\mathbf{d}^T Q(\mathbf{e})^T\mathbf{c} = \mathbf{c}^T Q(\mathbf{e})\mathbf{d}$ that $\mathbf{d}^T Q(\mathbf{e})^T\mathbf{c} = -\mathbf{d}^T Q(\mathbf{e})\mathbf{c}$ for all $\mathbf{c}$ and $\mathbf{d}$. Hence, $Q(\mathbf{e})^T = -Q(\mathbf{e})$. □

Consequently, we have

Lemma 4.2. *Fix $\mathbf{e}$. $\mathbf{d}^T Q(\mathbf{e})\mathbf{d} = 0$ for any $\mathbf{d}$ satisfying $H\mathbf{d} = 0$.*

Next we will need the following

Lemma 4.3. *There exists a constant* C_1 *such that*

$$|\mathbf{d}^T Q(\mathbf{e})\mathbf{c}| \le C_1 \|\mathbf{c}\| \, \|\mathbf{d}\| \, \|\mathbf{e}\|.$$

Proof: The Markov inequality and a tedious computation yields the above estimate. □

The coefficient vector $\mathbf{c}$ of th spline approximation of the weak solution satisfies $H\mathbf{c} = 0$ and $B\mathbf{c} = \mathbf{b}$ and

$$\nu \mathbf{c}^T A\mathbf{d} + \mathbf{c}^T Q(\mathbf{c})\mathbf{d} = \mathbf{h}^T M\mathbf{d}, \tag{4.4}$$

for all coefficient vectors $\mathbf{d}$ of splines in $\mathbf{S}_0$. The existence of $\mathbf{c}$ can be shown by using the same argument for the existence of the weak solution ϕ satisfying (4.1) (cf. [Girault and Raviart'86] and [Lai and Wenston'00]). We are mainly interested in computing $\mathbf{c}$.

We now observe that if $\mathbf{c}$ and λ are choosen so that

$$\begin{cases} \nu A\mathbf{c} - Q(\mathbf{c})\mathbf{c} + [H^T \; B^T]\lambda = M\mathbf{h} \\ \qquad\qquad\qquad\qquad\quad H\mathbf{c} = 0 \\ \qquad\qquad\qquad\qquad\quad B\mathbf{c} = \mathbf{b} \end{cases} \tag{4.5}$$

then

$$\nu \mathbf{c}^T A\mathbf{d} + \mathbf{c}^T Q(\mathbf{c})\mathbf{d} + \lambda^T \begin{bmatrix} H \\ B \end{bmatrix} \mathbf{d} - \mathbf{h}^T M\mathbf{d} = 0$$

for all vectors $\mathbf{d}$. Since both $H\mathbf{d} = 0$ and $H\mathbf{d} = 0$ for all coefficient vectors $\mathbf{d}$ of splines in $\mathbf{S}_0$ (4.4) follows.

We next derive two methods to linearize the nonlinear equations (4.5), using a simple iteration algorithm and Newton's method.

Algorithm 4.4 (A simple iteration algorithm). *Let* $(\mathbf{c}^{(0)}$ *be the solution of the linear problem (i.e. the associated Stokes equations) and for* $n = 0, 1, \ldots,$ *define* $(\mathbf{c}^{(n+1)}, \lambda^{(n+1)})$ *as the solution of*

$$\begin{aligned} \nu A\mathbf{c}^{(n+1)} - Q(\mathbf{c}^{(n)})\mathbf{c}^{(n+1)} + [H^T \; B^T]\lambda^{(n+1)} &= M\mathbf{h} \\ H\mathbf{c}^{(n+1)} &= 0 \\ B\mathbf{c}^{(n+1)} &= \mathbf{b}. \end{aligned} \tag{4.6}$$

We use Algorithm 2.4 to find the new iterates $\mathbf{c}^{(n+1)}$ of the above Algorithm 4.4. Note that A is symmetric and positive definite with respect to H and B and that $Q(\mathbf{c}^{(n)})$ is anti-symmetric by Lemma 4.1. Theorem 2.5 ensures that the matrix iterative method converges for the linear system in (4.6).

To apply Newton's method to solve (4.5), we consider the mapping Γ defined by

$$\Gamma : (\mathbf{c}, \lambda) \mapsto (\nu A\mathbf{c} - Q(\mathbf{c})\mathbf{c} + [H^T \; B^T]\lambda - M\mathbf{h}, H\mathbf{c}, B\mathbf{c} - \mathbf{b})$$

and seek to solve $\Gamma(\mathbf{c},\lambda) = 0$. We write $\mathbf{X}^{(n)} = (\mathbf{c}^{(n)}, \lambda^{(n)})$ and let $\mathbf{X}^{(0)}$ be the solution of the linear Stokes problem. Then we define $\mathbf{X}^{(n+1)}$ by the equation

$$\Gamma'(\mathbf{X}^{(n)})(\mathbf{X}^{(n+1)} - \mathbf{X}^{(n)}) = -\Gamma(\mathbf{X}^{(n)}).$$

Thanks to the bilinearity of the mapping $(\mathbf{c},\mathbf{d}) \to Q(\mathbf{c})\mathbf{d}$, the above equation leads to

$$\begin{aligned} \nu A\mathbf{c}^{(n+1)} - Q(\mathbf{c}^{(n)})\mathbf{c}^{(n+1)} - Q(\mathbf{c}^{(n+1)} - \mathbf{c}^{(n)})\mathbf{c}^{(n)} + [H^T\ B^T]\lambda^{(n+1)} &= M\mathbf{h} \\ H\mathbf{c}^{(n+1)} &= 0 \\ B\mathbf{c}^{(n+1)} &= \mathbf{b}. \end{aligned} \tag{4.7}$$

It can be shown that $Q(\mathbf{c}^{(n+1)} - \mathbf{c}^{(n)})\mathbf{c}^{(n)}$ can be written $\widetilde{Q}(\mathbf{c}^{(n)})(\mathbf{c}^{(n+1)} - \mathbf{c}^{(n)})$ with $\widetilde{Q}(\mathbf{c}^{(n)})\mathbf{c}^{(n)} = Q(\mathbf{c}^{(n)})\mathbf{c}^{(n)}$ for some matrix $\widetilde{Q}(\mathbf{c}^{(n)})$. We have our second algorithm

Algorithm 4.5 (Newton's method). *Let $(\mathbf{c}^{(0)}$ be the solution of the Stokes equations and for $n = 0, 1, 2, \ldots$ define $(\mathbf{c}^{(n+1)}, \lambda^{(n+1)})$ to be the solution of*

$$\begin{aligned} \nu A\mathbf{c}^{(n+1)} - Q(\mathbf{c}^{(n)})\mathbf{c}^{(n+1)} - \widetilde{Q}(\mathbf{c}^{(n)})\mathbf{c}^{(n+1)} + [H^T\ B^T]\lambda^{(n+1)} &= M\mathbf{h} - \widetilde{Q}(\mathbf{c}^{(n)})\mathbf{c}^{(n)} \\ H\mathbf{c}^{(n+1)} &= 0 \\ B\mathbf{c}^{(n+1)} &= \mathbf{b}. \end{aligned} \tag{4.8}$$

For the linear system (4.8), Algorithm 2.4 converges by Theorem 2.5 because $Q(\mathbf{c}^{(n)})$ is anti-symmetric and $\nu A + \widetilde{Q}(\mathbf{c}^{(n)})$ is positive definite with respect to $[H;\ B]$ for ν sufficiently large. Indeed, since A is positive definite with respect to H, we have

$$\mathbf{x}^T A\mathbf{x} \geq \alpha_0 \|\mathbf{x}\|^2$$

for any $\mathbf{x}$ satisfying $H\mathbf{x} = 0$. It follows from Lemma 4.3,

$$\nu\mathbf{x}^T A\mathbf{x} + \mathbf{x}^T\widetilde{Q}(\mathbf{c}^{(n)})\mathbf{x} \geq \nu\alpha_0||\mathbf{x}||^2 - C_1||\mathbf{c}^{(n)}||\,||\mathbf{x}||^2.$$

By Lemma 4.6 below, the sequence $\|\mathbf{c}^{(n)}\|$ is bounded and hence,

$$\nu\mathbf{x}^T A\mathbf{x} + \mathbf{x}^T\widetilde{Q}(\mathbf{c}^{(n)})\mathbf{x} \geq (\nu\alpha_0 - C_1C_2)||\mathbf{x}||^2.$$

Therefore, for ν sufficiently large, $\nu A + \widetilde{Q}(\mathbf{c}^{(n)})$ is positive definite with respect to $[H; B]$.

Lemma 4.6. *Suppose that $\mathbf{c}^{(n+1)}$ is the solution of (4.8). Then*

$$\|\mathbf{c}^{(n+1)}\| \leq C$$

for a positive constant C independent of n.

Proof: Let $\mathbf{c}$ be the solution of (4.5). Let $\mathbf{e}^{(n+1)} = \mathbf{c} - \mathbf{c}^{(n+1)}$ and $\tau^{(n+1)} = \lambda - \lambda^{(n+1)}$. Then we have

$$\begin{aligned}&\nu A\mathbf{e}^{(n+1)} + Q(\mathbf{c}^{(n)})\mathbf{c}^{(n+1)} + Q(\mathbf{c}^{(n+1)})\mathbf{c}^{(n)} - Q(\mathbf{c})\mathbf{c} + [H;B]^T\tau^{(n+1)}\\ &= Q(\mathbf{c}^{(n)})\mathbf{c}^{(n)}\end{aligned}$$

which gives

$$\nu A\mathbf{e}^{(n+1)} - Q(\mathbf{c}^{(n)})\mathbf{e}^{(n+1)} - Q(\mathbf{e}^{(n)})\mathbf{c} + Q(\mathbf{c}^{(n+1)})\mathbf{c}^{(n)} + [H;B]^T\tau^{(n+1)} = Q(\mathbf{c}^{(n)})\mathbf{c}^{(n)},$$

or

$$\begin{aligned}&\nu A\mathbf{e}^{(n+1)} - Q(\mathbf{c}^{(n)})\mathbf{e}^{(n+1)} - Q(\mathbf{e}^{(n)})\mathbf{c} - Q(\mathbf{e}^{(n+1)})\mathbf{c}^{(n)}\\ &+ Q(\mathbf{e}^{(n)})\mathbf{c}^{(n)} + [H;B]^T\tau^{(n+1)} = 0.\end{aligned}$$

We multiply this equation on the left by $(\mathbf{e}^{(n+1)})^T$ and get

$$\nu(\mathbf{e}^{(n+1)})^T A\mathbf{e}^{(n+1)} = -(\mathbf{e}^{(n+1)})^T Q(\mathbf{e}^{(n)})\mathbf{e}^{(n)} + (\mathbf{e}^{(n+1)})^T Q(\mathbf{e}^{(n+1)})\mathbf{c}^{(n)}.$$

It follows that

$$\begin{aligned}\nu\alpha_0||\mathbf{e}^{(n+1)}|| &\le C_1\big(||\mathbf{e}^{(n)}||^2 + ||\mathbf{e}^{(n+1)}||\,||\mathbf{c}^{(n)}||\big)\\ &\le C_1\Bigg(||\mathbf{e}^{(n)}||^2 + ||\mathbf{e}^{(n+1)}||(||\mathbf{e}^{(n)}|| + ||\mathbf{c}||)\Bigg).\end{aligned}$$

We now use an induction to prove the desired result. Assume that $\|\mathbf{e}^{(n)}\| \le K$ with $K \le \frac{1}{2}\frac{\nu\alpha_0 - C_1\|\mathbf{c}\|}{C_1}$. This is alway possible if ν is large enough. Then we have

$$\big(\nu\alpha_0 - C_1(K + \|\mathbf{c})\|\big)||\mathbf{e}^{(n+1)}|| \le C_1||\mathbf{e}^{(n)}||^2$$

or

$$||\mathbf{e}^{(n+1)}|| \le \frac{C_1}{\nu\alpha_0 - C_1(K + \|\mathbf{c}\|}K^2 \le K.$$

This completes the proof with $C_2 = K + \|\mathbf{c}\|$. □

5.1. Computational Experiments on Driven Cavity Flows

Example 4.7. *Let us consider the standard cavity flow over unit square domain $[0,1] \times [0,1]$. The boundary conditions are $\mathbf{u} = (u_1, u_2) = (0,0)$ for all four line boundary pieces except for $u_1(x,1) = 1$ when $0 \le x \le 1$. With Reynolds numbers are 10, 100, 1000, 5000, and 10000, the stream lines of the cavity flow are shown in Figs 1–5. We use a triangulation with only 256 triangles and bivariate splines of degree 8 and smoothness 2. All the computation is done using a PC with MS Window 2000. We can see that the contours of the stream functions are very closed*

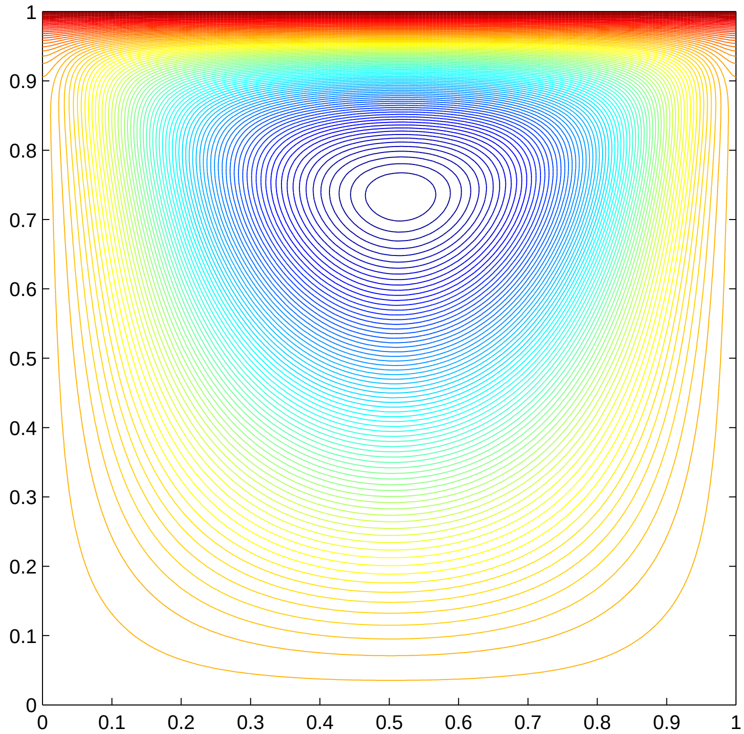

Fig. 1 The stream lines of driven cavity flow (Renolds Number=10)

to those computed by using finite difference method as shown in [Griebel, and et.'98]. Certainly, our method can compute the stream functions for much higher Reynolds numbers.

Also, the vorticity of the flows,

$$\psi(x,y) = \frac{\partial^2}{\partial x^2}\phi(x,y) + \frac{\partial^2}{\partial y^2}\phi(x,y),$$

are shown as in Figs. 6–10. They are very similar to those computed by using finite difference method given in [Griebel and et.'98].

5.2. Computational Experiments on Backward Facing Step Flows

Next we consider the well-known backward facing step flows. The inflow condition at the left inlet is the parabolic flow with maximum value 1. The outflow condition at the right outlet is
Outflow condition: Both velocity components remain the same in the direction normal to the boundary. That is,

$$\frac{\partial^2}{\partial n^2}\varphi = 0 \text{ and } \frac{\partial}{\partial n}\frac{\partial}{\partial \tau}\varphi = 0$$

at the outlet, where τ, n denote the tangential and normal direction of the outlet. The following are some numerical simulations of the backward-facing step flows For

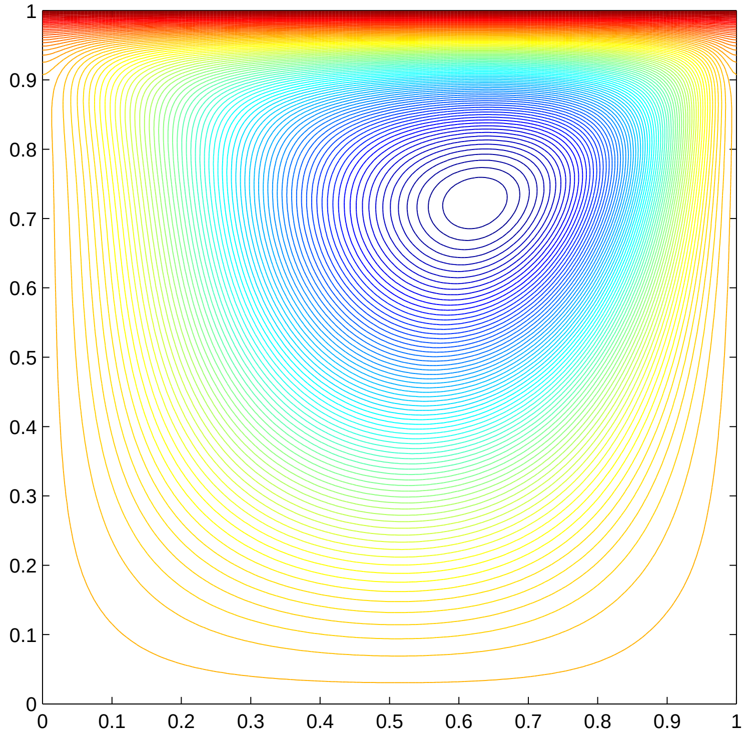

Fig. 2 The stream lines of driven cavity flow (Renolds Number=100)

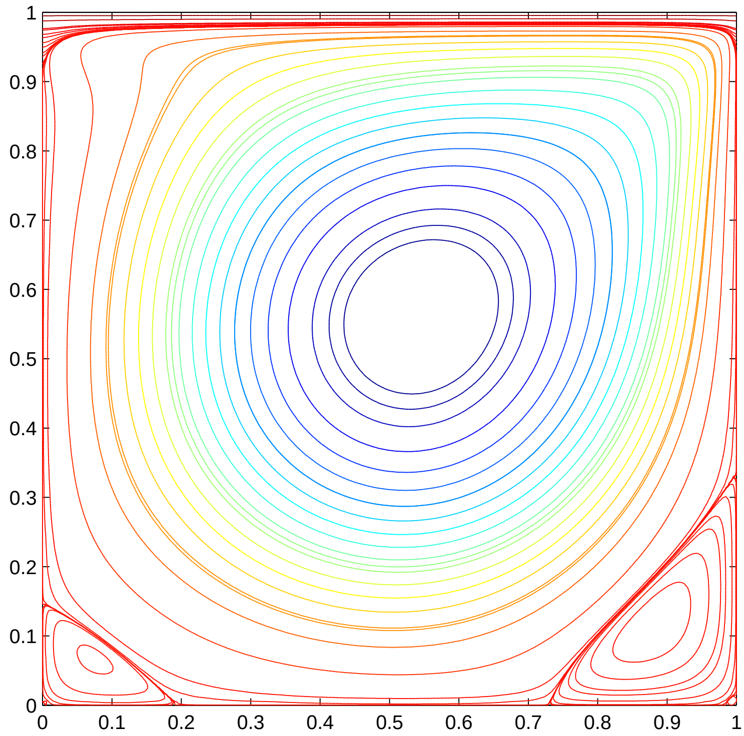

Fig. 3 The stream lines of driven cavity flow (Renolds Number=1,000)

detailed explanation of these flows, see, e.g. [Griebel and al.'98] and the reference therein.

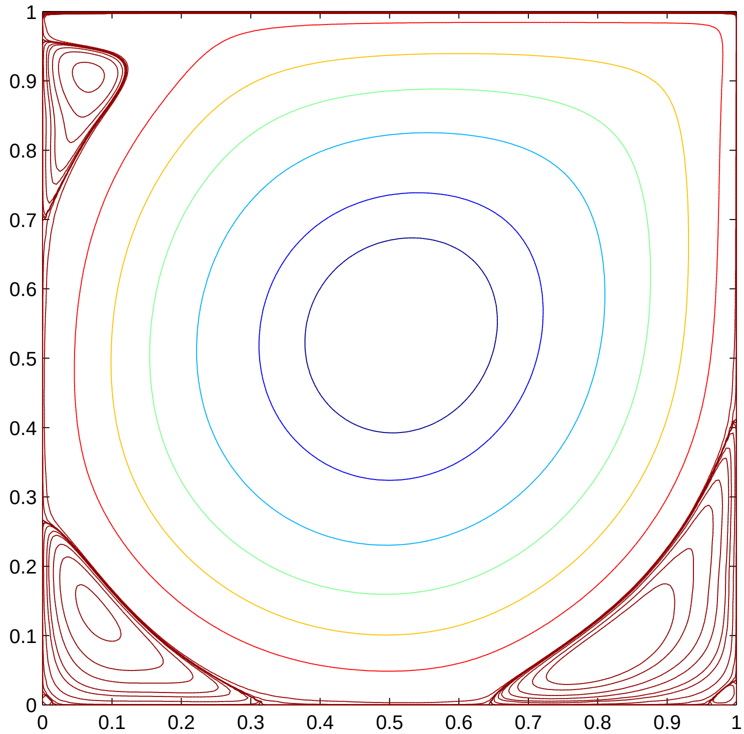

Fig. 4　The stream lines of driven cavity flow (Renolds Number=5,000)

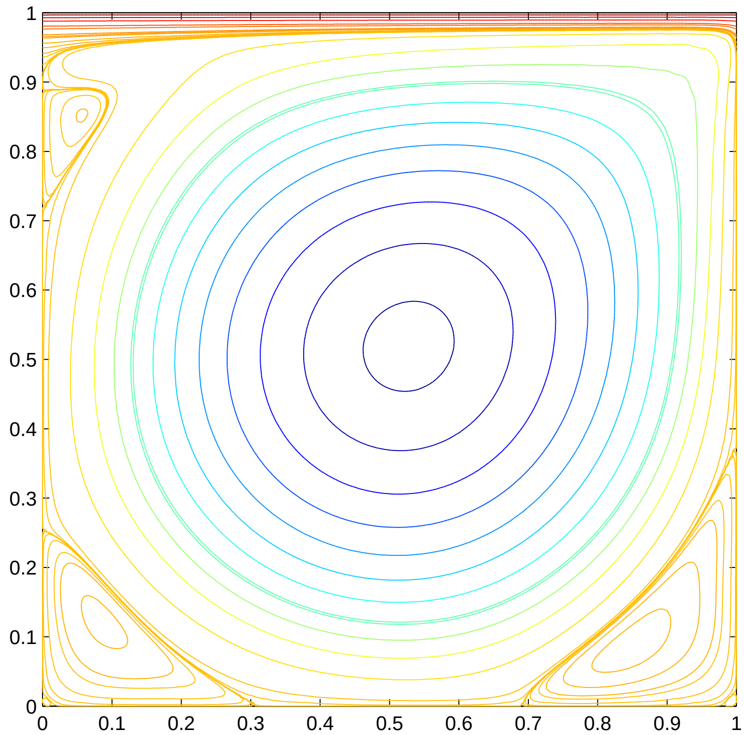

Fig. 5　The stream lines of driven cavity flow (Renolds Number=10,000)

5.3. Computational Experiments of Flows around circular obstacle

In addition to the no-slip and outflow boundary conditions in the previous

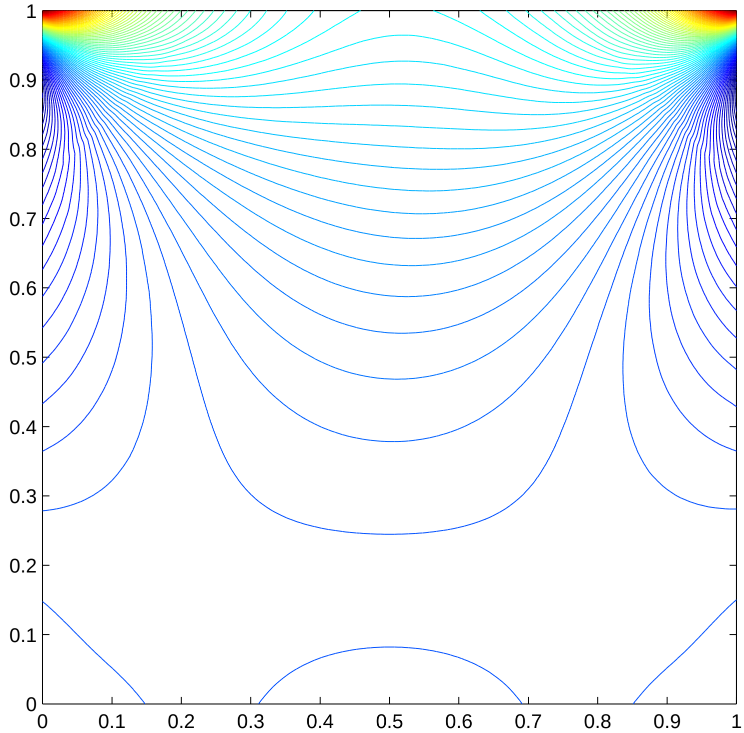

Fig. 6 The contour of the vorticity of the driven cavity flow (Renolds Number=10)

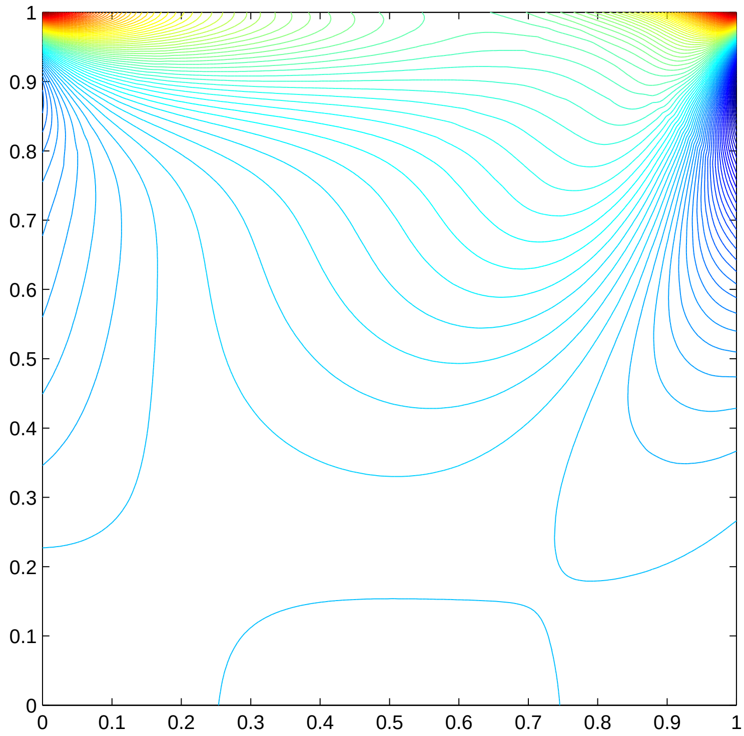

Fig. 7 The contour of the vorticity of driven cavity flow (Renolds Number=100)

subsections, we need the following freeslip conditions for simulating the flows around some obstacles.

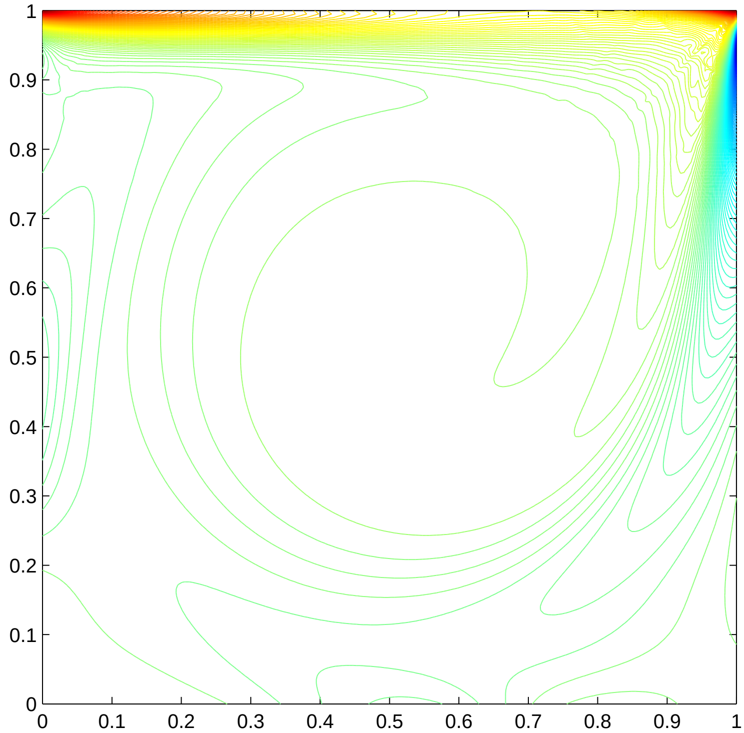

Fig. 8 The contour of the vorticity of driven cavity flow (Renolds Number=1,000)

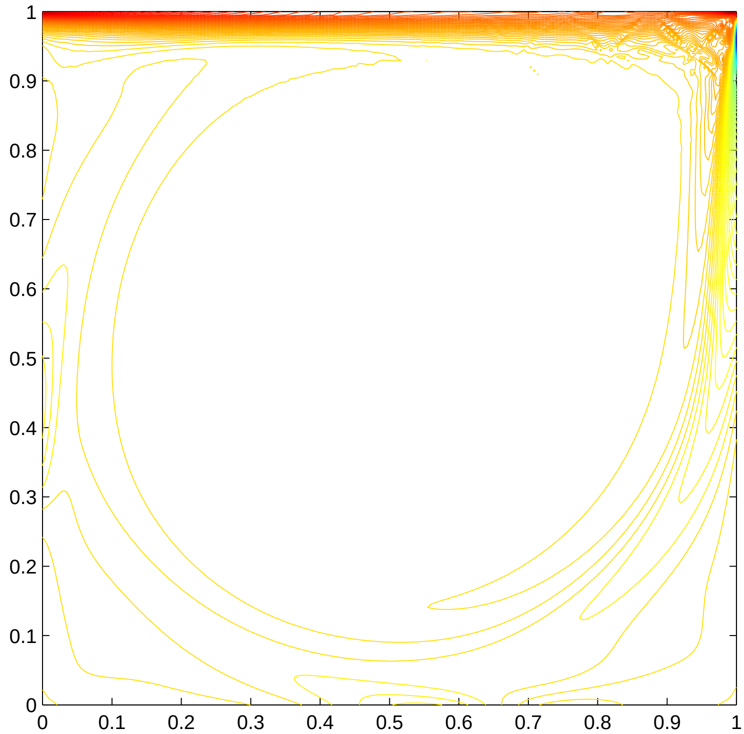

Fig. 9 The contour of the vorticity of driven cavity flow (Renolds Number=5,000)

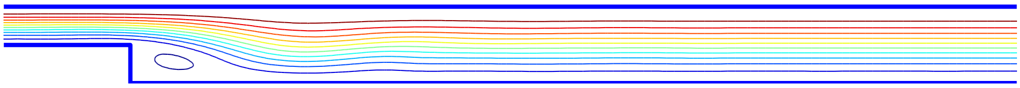

Fig. 11 Flows over a backward facing step
Reynolds number=1

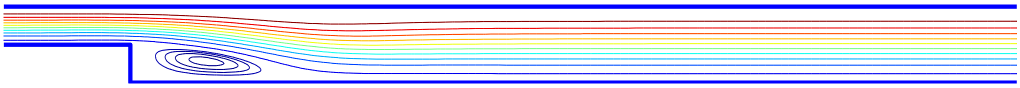

Fig. 12 Flows over a backward facing step
Reynolds number=100

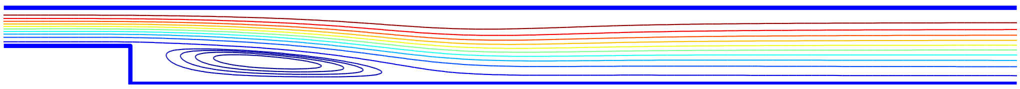

Fig. 13 Flows over a backward facing step
Reynolds numbers=250

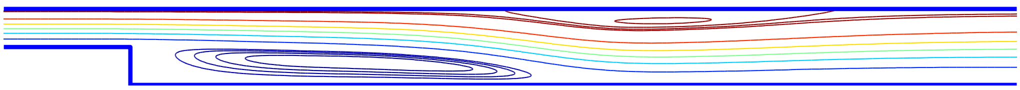

Fig. 14 Flows over a backward facing step
Reynolds number=500

Freeslip conditions: No fluid penetrates the boundary. I.e.,

$$\frac{\partial}{\partial n}\varphi = 0 \text{ and } \frac{\partial}{\partial n}\frac{\partial}{\partial \tau}\varphi = 0$$

along a part of the boundary $\partial\Omega$. Here are some numerical simulations of flows around a circular disk for various Reynolds numbers. The in-flows are at a constant velocity at the left vertical boundary, i.e., $u = 1$ and $v = 0$. These flows can be compared with the ones in [Griebel and et.'98] and [Van Dyke'82].

5.4. Computational Experiments on Other Interested Flows

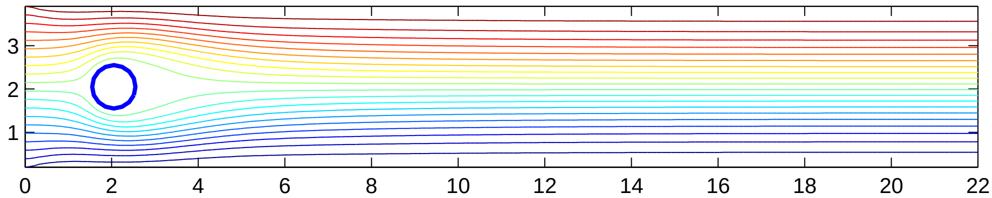

Fig. 15 Flow around a disk with Reynolds number=10

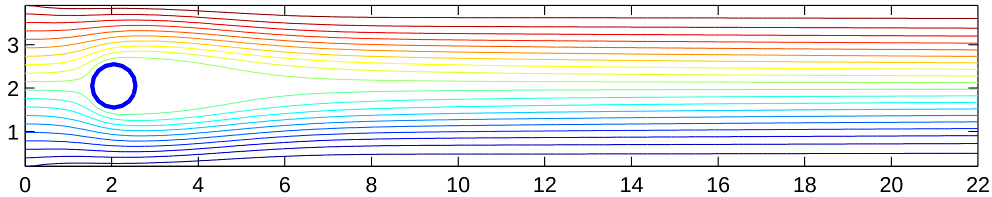

Fig. 16 Flow around a disk with Reynolds number=50

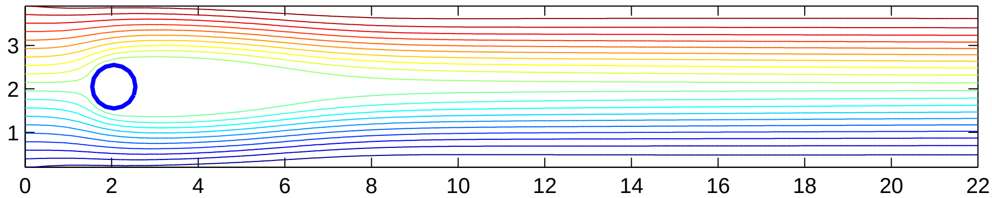

Fig. 17 Flow around a disk with Reynolds number=100

We first show cavity flows over a triangular domain.

Example 4.8. *Let us consider the cavity flow over triangular domain. We uniformly refine the triangle into 256 subtriangles and use bivariate splines of degree 8 and smoothness 2. The boundary conditions are* $\mathbf{u} = (u_1, u_2) = (0, 0)$ *for all three line boundary pieces except for* $u_1(x, 1) = 1$ *when* $0 \leq x \leq 1$*. With Reynolds numbers are* 100, 1000, *and* 5000, *the stream lines of the cavity flow are shown in Figs 18, 20, 22 while the contour of vorticity are shown in Figs. 19, 21 and 23.*

Next we present some additional examples which need inflow and outflow conditions.

Example 4.9. *In this example, we compare with the flows around a bluff car model and streamlined car model as in Fig. 24. The flows are similar to the real simulations as compiled in [Nakayama'78].*

Example 4.10. *In this example, we consider the flow through sudden contraction and enlargement. We compare with the flows with two levels of contraction and enlargement. The flows are constant at the left vertical boundary and are same for the two flows in Fig. 25, but result two different flow pattens.*

§7. References

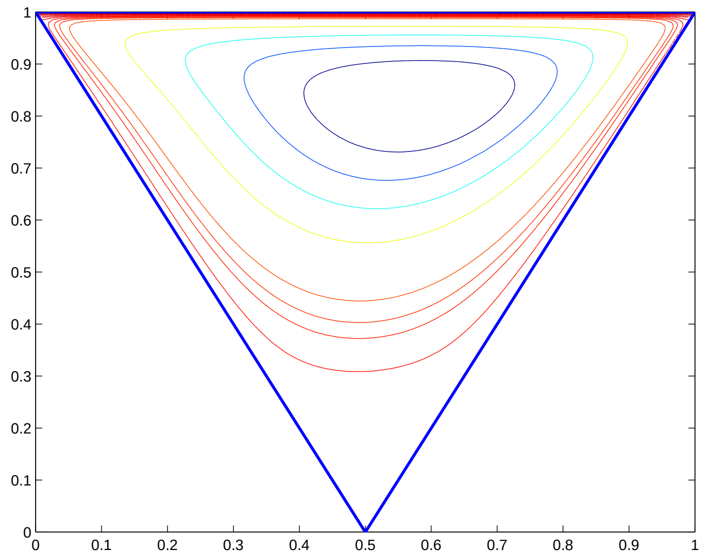

Fig. 18   Driven Cavity Flow (Renolds Number=100)

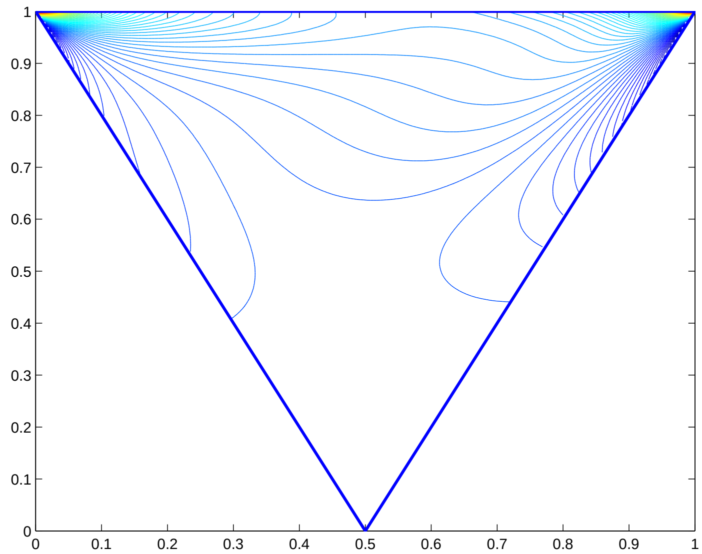

Fig. 19   Contour of the Vorticity of the Driven Cavity Flow (Renolds Number=100)

1. G. Awanou and M. J. Lai, 2002, Trivaraite spline approximations of the 3D Navier-Stokes Equations, submitted, 2002.

2. G. Awanou, M. J. Lai and P. Wenston, 2002, The Multivariate Spline Method for Numerical Solutions of Partial Differential Equations and Scattered Data Fitting, submitted.

3. O. Botella, 2002, On a collocation B-spline method for the solution of the Navier-Stokes equations, Computer & Fluids, **13**, 397–420.

4. C. de Boor, *B*-form Basics, in *Geometric Modeling*, edited by G. Farin, SIAM Publication, Philadelphia, pp 131-148, 1987.

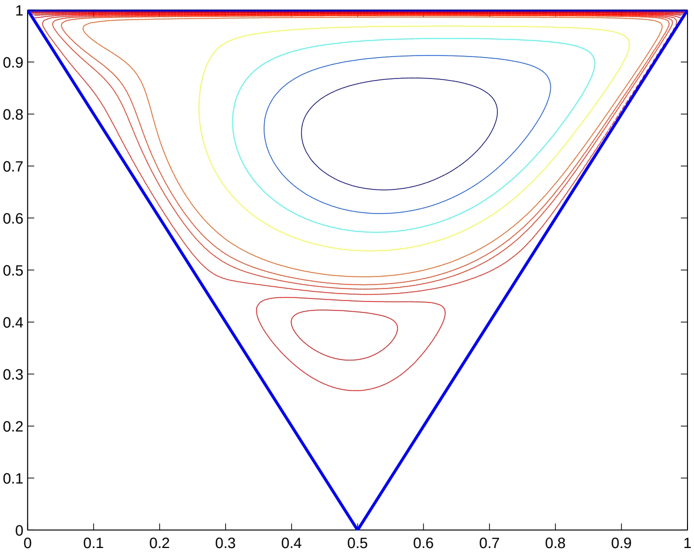

Fig. 20 Driven Cavity Flow (Renolds Number=1000)

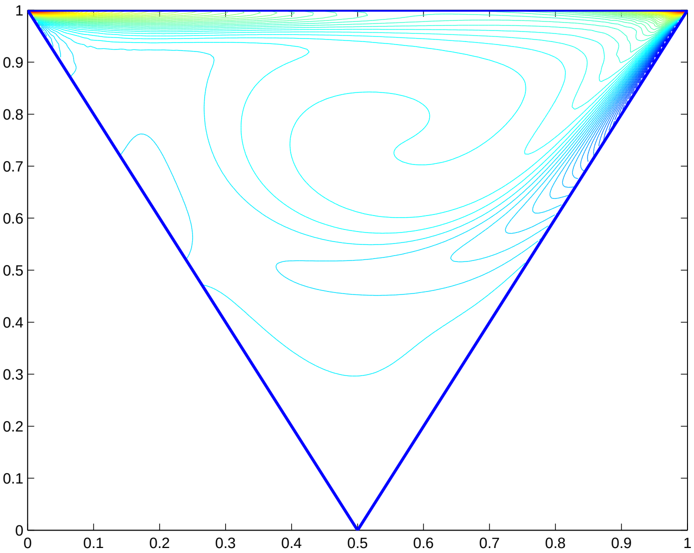

Fig. 21 Contour of the Vorticity of the Driven Cavity Flow (Renolds Number=1000)

5. C. K. Chui and M. J. Lai, 1990, Multivariate vertex splines and finite elements, J. Approx. Theory, **60**, pp. 245–343, 1990.

6. L. Evans, 1998, *Partial Differential Equations*, American Math. Soc., Providence.

7. G. Farin, Triangular Bernstein-Bézier patches, Comput. Aided Geom. Design, **3**, 83–127, 1986.

8. M. Fortin and R. Glowinski, *Augmented Lagrangian Methods*, Elsevier, 1983.

9. M. Griebel, T. Dornseifer, and T. Neunhoeffer, *Numerical Simulation in Fluid Dynamics*, SIAM Publications, philadelphia, 1998.

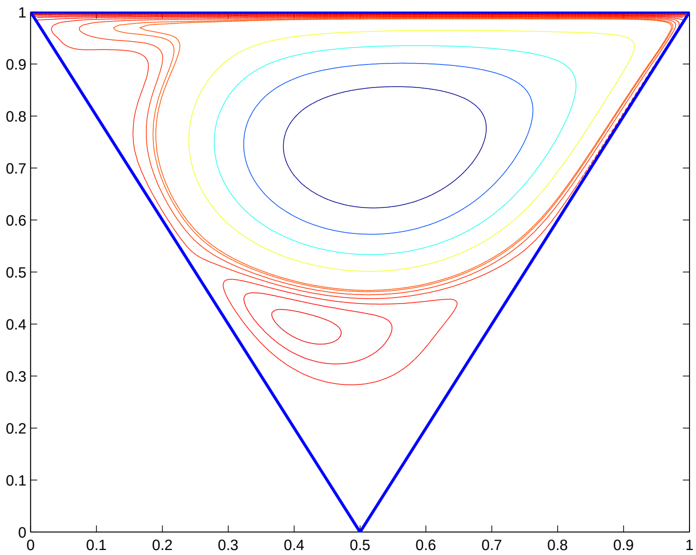

Fig. 22 Driven Cavity Flow (Renolds Number=5000)

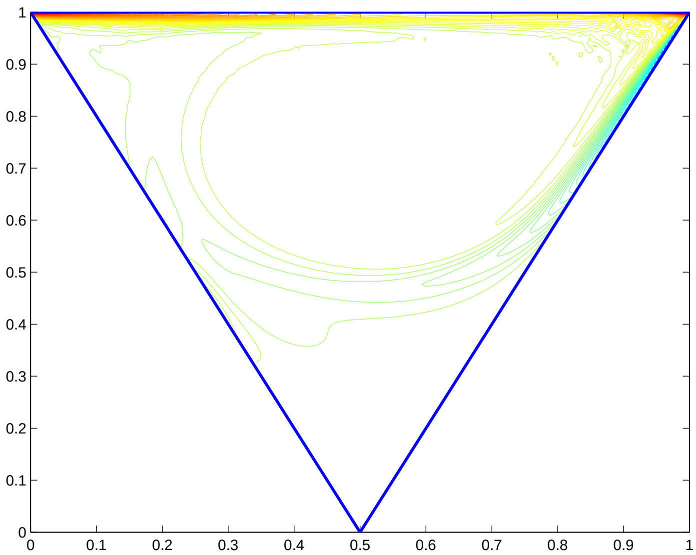

Fig. 23 Contour of the Vorticity of the Driven Cavity Flow (Renolds Number=5000)

10. M. Gunzburger, *Finite Element Methods for Viscous Incompressible Flows*, Academic Press, 1989.

11. V. Girault and P.A. Raviart, *Finite Element Method for Navier-Stokes Equations*, Springer-Verlag, 1986.

12. O. Karakashian, On a Galerkin-Lagrange Multiplier Method for the Stationnary Navier-Stokes Equations, SIAM J. of Numer. Anal., Vol 19, N° 5, (1982) pp 909-923.

13. M. J. Lai and P. Wenston, Bivariate Splines for fluid flows, manuscript, 2002.

14. Y. Nakayama, *Visualized Flow*, Pergamon Press, Oxford, 1978.

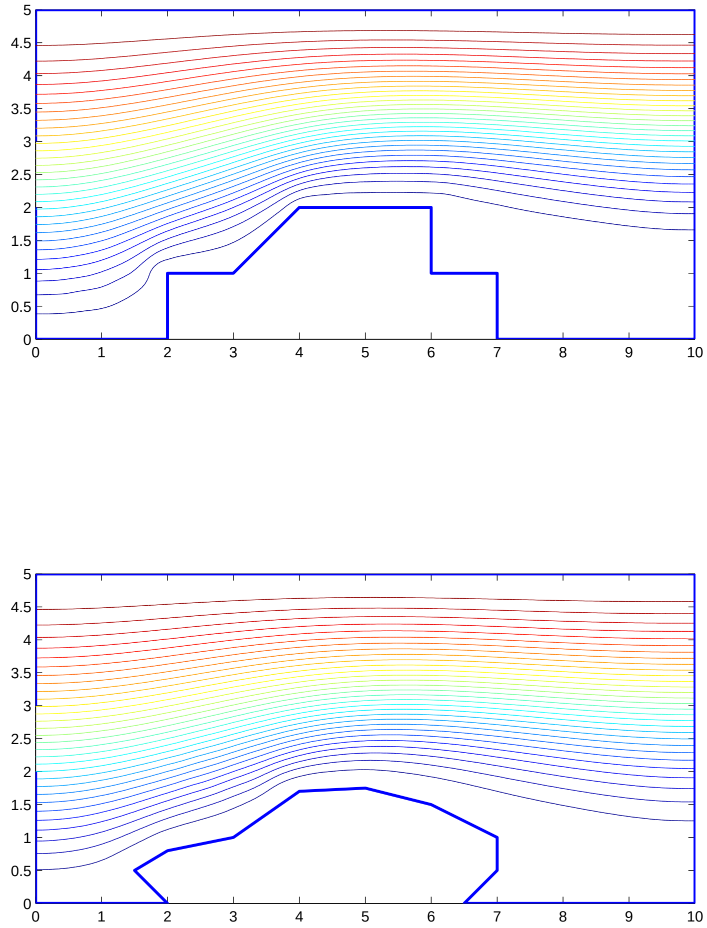

Fig. 24. The flows around a bluff car model and a streamlined car model

15. J. Serrin, On the interior regularity of weak solution of Navier-Stokes equations, Arch. Rath. Mech. Ana. **9** 187-195, 1962.

16. R. Temam, Navier-Stokes Equations. Theory and Numerical Analysis, North-Holland Publishing Co., Amsterdam, 1984.

17. M. Van Dyke, *An Album of Fluid Motion*, Parabolic Press, Stanford, California, 1982.

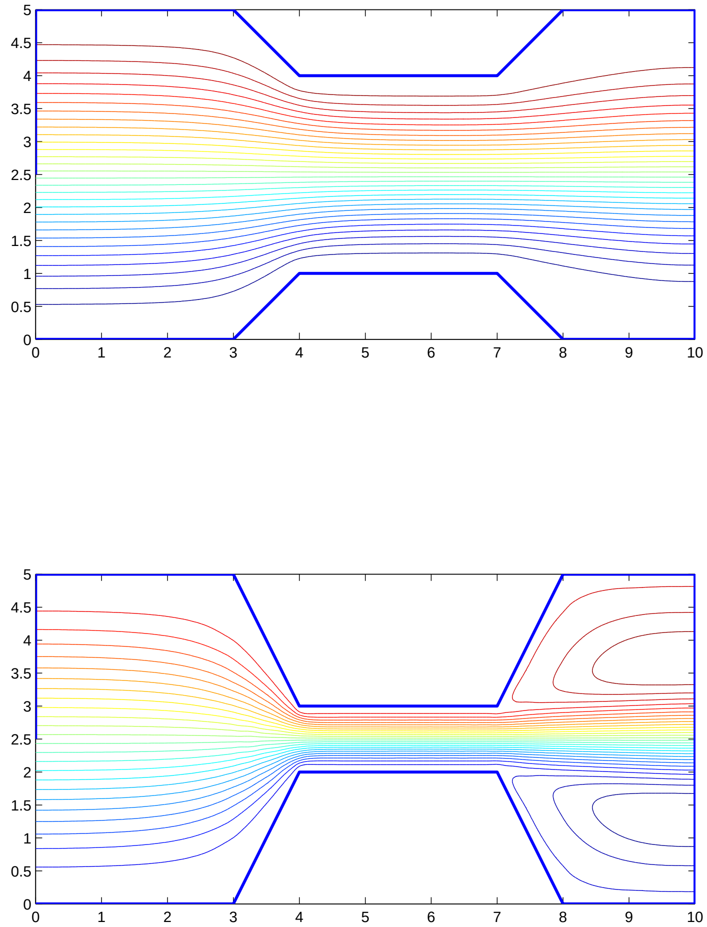

Fig. 25 Flows through sudden contraction and enlargement