Journal of Scientific Computing
https://doi.org/10.1007/s10915-019-01012-2

A Cartesian Compact Scheme for the Navier–Stokes Equations in Streamfunction Formulation in Irregular Domains

Matania Ben-Artzi[1] · Jean-Pierre Croisille[2] · Dalia Fishelov[3]

Received: 6 February 2019 / Revised: 28 June 2019 / Accepted: 11 July 2019

Abstract
In Ben-Artzi et al. (SIAM J Numer Anal 47:3087–3108, 2009) we introduced an embedded Cartesian scheme for the biharmonic problem in two dimensions. Here we extend this methodology to the 2D Navier–Stokes system. Hermite (or Birkhoff) interpolation is invoked in one and two dimensions to obtain finite difference operators. The consistency analysis of the discrete formulas for irregular grids is emphasized. Numerical results demonstrate remarkable accuracy for a series of test cases for flows in elliptical domains.

Keywords Biharmonic problem · Embedded compact scheme · Navier–Stokes equations

1 Introduction

In [3] an embedded Cartesian finite difference scheme for the biharmonic problem in irregular domains was introduced. A regular Cartesian grid is laid out in the embedded domain. In most of the interior physical domain the regular Cartesian grid is kept unchanged. Thus, points which are far from the boundary are the center of regular elements containing the points' immediate neighbors. However, irregular elements are constructed around points which are close to the boundary. As a result, the grid is not necessarily aligned to the boundary, as in the case of body fitted grids.

This paper is dedicated to the memory of Professor Saul Abarbanel.

✉ Dalia Fishelov
daliaf@afeka.ac.il

Matania Ben-Artzi
mbartzi@math.huji.ac.il

Jean-Pierre Croisille
jean-pierre.croisille@univ-lorraine.fr

[1] Institute of Mathematics, The Hebrew University, 91904 Jerusalem, Israel

[2] Department of Mathematics, IECL, UMR 7502, Université de Lorraine, 57045 Metz, France

[3] Afeka - Tel-Aviv Academic College of Engineering, 218 Bnei-Efraim St., 69107 Tel-Aviv, Israel

Published online: 16 July 2019

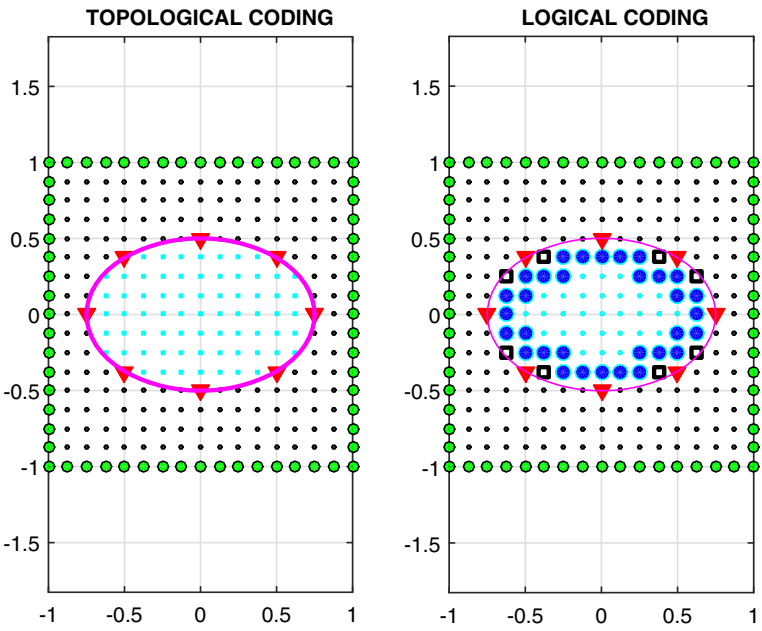

Fig. 1 Ellipse embedded in a square supporting a Cartesian grid. Left panel: Topological coding of the points: the points are exterior (black small circles), boundary (triangles) or internal (clear small circles). Right panel: the logical coding of the different kinds of interior points are subdivided into *Regular calculated* points (small circles), *Irregular calculated* points (large circle), *Edge* points (open squares)

In case the physical domain is not a rectangular, but rather an irregular domain, the scheme results in a Cartesian 9-point compact scheme at points "far" from the boundary. However, the Cartesian stencil is distorted around points which are close to the boundary. A crucial tool in our embedding approach consists of classifying grid points near the boundary. First, the Cartesian points are labeled as interior points if they are inside the physical domain, exterior points if they are outside the domain, and boundary points if they are on the boundary. This is called a *topological coding*. Points which are inside the domain are then divided into three subcategories. This is named a *logical* coding (see Fig. 1). A first category contains *regular calculated* points; their neighboring points reside on the Cartesian grid and are far enough from the boundary. A second category contains *edge* points; these are points which are too close to the boundary and therefore they are not included in the computational stencil, but are replaced by points on the boundary. A third category contains *irregular calculated* points which are close to the boundary and at least one of their neighboring points is too close to the boundary. Irregular calculated points are surrounded by a distorted stencil. Figure 2 shows the stencil of an irregular calculated point M_0, which includes the edge point $\tilde{M}_2$. The topological and logical coding are shown in Fig. 1 for an ellipse. We refer to [3,7] for more details.

Extending finite difference approximations to irregular domains is not a new idea. It originates from early works such as [34]. Important contributions in the 70's with emphasis on the capacitance matrix method include [12,31]. A significant renewal of interest in this approach is recently observed. For example, Poisson solvers are suggested in [19,22]. For time dependent problems we refer to [1,16]. Concerning the Navier–Stokes equations we refer to [21,28]. Approximating the Navier–Stokes equations in pure stream function formulation

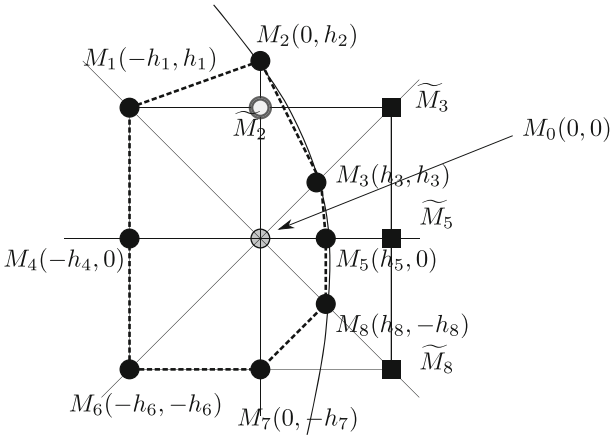

Fig. 2 The figure shows the stencil of the irregular calculated point $\mathbf{M}_0$ as the dash line. It consists of the set $\{\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_5, \mathbf{M}_8, \mathbf{M}_7, \mathbf{M}_6, \mathbf{M}_4\}$. The points $\mathbf{M}_1$, $\mathbf{M}_4$, $\mathbf{M}_6$ and $\mathbf{M}_7$ belong to the Cartesian grid. The points $\mathbf{M}_2$, $\mathbf{M}_3$, $\mathbf{M}_5$ and $\mathbf{M}_8$ belong to the boundary of the domain. The point $\widetilde{\mathbf{M}}_2$ is labeled as an *edge point* since it it too close to the boundary to be a calculated point. The point $\mathbf{M}_2$ is defined as the point on the boundary which lies on the line connecting $\mathbf{M}_0$ and $\widetilde{\mathbf{M}}_2$

using a high order compact scheme was first introduced in [10] in the case of a square domain (see also [20]). Extending this kind of scheme to irregular geometries was considered in [23,30] using a boundary fitted grid. Another approach for handling Navier–Stokes equation in irregular geometries was introduced in [17]. We refer to Sect. 4 for the relation of the scheme [17] to the present work.

Here the compact scheme is based on a particular interpolation polynomial of degree six, combined with a compact interpolation for the gradient. The main features of this approximation have been introduced in [3]. As in the approach to finite differences using an interpolating polynomial presented in classic treatises [14,24,26], the key problem is to find a suitable polynomial which interpolates the "data" on a stencil around any calculated point $\mathbf{x}_{i,j}$. For all calculated points, this is a 9 point compact stencil. At regular points, the stencil is regular; only at irregular points (near the boundary) the stencil is distorted (see Fig. 2). Differentiating this polynomial provides finite difference operators at $\mathbf{x}_{i,j}$ in terms of the data. When collecting the discrete equations for all points $\mathbf{x}_{i,j}$ the data become "unknowns" to be computed as solution of the global linear system. Note that a full mathematical convergence analysis of our scheme is not available yet. However, elements of error bounds and convergence analysis are available in one dimension [6,8,9,11,18]. The interest in this approach is supported by the remarkable accuracy of the numerical results obtained so far [3,7,17].

The outline of the paper is as follows. In Sect. 2 we summarize the one-dimensional design of the compact scheme using a polynomial approach. In Sect. 3 the structure of the two dimensional polynomial, introduced in [3] is outlined. Emphasis is given on the directional splitting between the two axis directions x and y and the two diagonal directions $x + y$ and $x - y$. In Sect. 4, we display several facts concerning discrete operators on compact stencils, regular or not, which are used in our approximation to the Navier–Stokes equation. Finally, Sect. 5 displays numerical results obtained for the Navier–Stokes system in two dimensions for irregular domains.

2 Irregular Finite Difference Operators in One Dimension

In this section, we describe the basics of our finite difference scheme for the one dimensional case.

Let us consider the interval $I = [a, b]$ with the irregular grid

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b. \tag{2.1}$$

We denote by h_j the spacing between neighboring points, thus $h_j = x_j - x_{j-1}$, $j = 1, \ldots, N$ and $\mathbf{h} = [h_1, \ldots, h_N]$.

Furthermore, ϵ and h are defined by (see [3]),

$$\begin{cases} h = \max\limits_j h_j, & (i) \\ \epsilon = \min\limits_j \left(\dfrac{h_{j+1}}{h_j}, \dfrac{h_j}{h_{j+1}} \right). & (ii) \end{cases} \tag{2.2}$$

The design of finite difference operators is based on a particular interpolation procedure of the data. For clarity we consider the following two steps in the analysis.

- *Interpolation of the data* u_j, u'_j:

We first consider the case where each interpolating point x_j carries the two interpolating data u_j and u'_j. The values u_j and u'_j are supposed to be *known data*. The *primary data* u_j stands for some approximation of $u(x_j)$ and the value u'_j approximates the derivative $u'(x_j)$, $j = 1, \ldots, N-1$. In approximation theory [32], these data are considered to be given, either as exact data or as measurements. To each point x_j, $j = 1, \ldots, N-1$, we associate a polynomial $p(x)$

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5. \tag{2.3}$$

Assuming the knowledge of the six data $u_j, u'_j, u_{j+1}, u'_{j+1}, u_{j-1}, u'_{j-1}$ (see Fig. 3), $p(x)$ is the solution of

$$\begin{cases} p(x_j) = u_j, & p(x_{j-1}) = u_{j-1}, & p(x_{j+1}) = u_{j+1}, \\ p'(x_j) = u'_j, & p'(x_{j-1}) = u'_{j-1}, & p'(x_{j+1}) = u'_{j+1}. \end{cases} \tag{2.4}$$

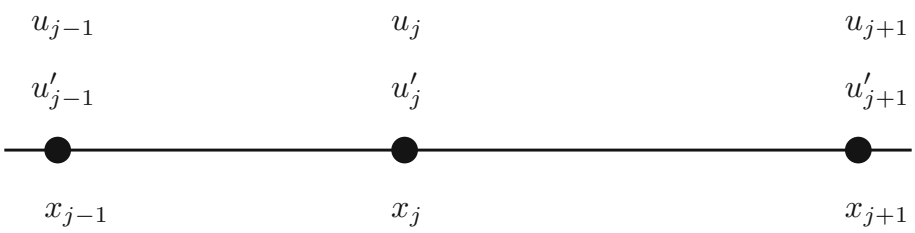

Fig. 3 One dimensional Hermite interpolation (2.4). The unique polynomial $p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5$ with the six data $u_{j-1}, u'_{j-1}, u_j, u'_j, u_{j+1}, u'_{j+1}$ is given in (2.5)

The coefficients a_α in (2.3) are expressed in terms of $u_{j-1}, u_j, u_{j+1}, u'_{j-1}, u'_j, u'_{j+1}$ by

$$\begin{aligned} a_0 &= u_j, \quad a_1 = u'_j \\ (a_\alpha)_j &= A_{1,\alpha}(h_j, h_{j+1})u_{j-1} + (-1)^\alpha A_{1,\alpha}(h_{j+1}, h_j)u_{j+1} + A_{2,\alpha}(h_j, h_{j+1})u_j \\ &\quad + B_{1,\alpha}(h_j, h_{j+1})u'_{j-1} - (-1)^\alpha B_{1,\alpha}(h_{j+1}, h_j)u'_{j+1} + B_{2,\alpha}(h_j, h_{j+1})u'_j, \end{aligned} \tag{2.5}$$

The rational fractions $A_{1,\alpha}(h_1, h_2)$, $A_{2,\alpha}(h_1, h_2)$, $B_{1,\alpha}(h_1, h_2)$ and $B_{2,\alpha}(h_1, h_2)$, $2\alpha \le 5$ are

$$\begin{cases} A_{1,2}(h_1, h_2) = \dfrac{h_2^2\,(5\,h_1 + 3\,h_2)}{h_1^2\,(h_1 + h_2)^3}, \quad A_{2,2}(h_1, h_2) = -\dfrac{3\,h_1^2 - 4\,h_2\,h_1 + 3\,h_2^2}{h_1^2\,h_2^2}, \\ B_{1,2}(h_1, h_2) = \dfrac{h_2^2}{h_1\,(h_1 + h_2)^2}, \quad B_{2,2}(h_1, h_2) = -2\,\dfrac{h_1 - h_2}{h_2\,h_1}, \end{cases} \tag{2.6}$$

$$\begin{cases} A_{1,3}(h_1, h_2) = -2\,\dfrac{h_2\,\left(h_2\,h_1 - {h_2}^2 + 5\,{h_1}^2\right)}{{h_1}^3\,(h_1 + h_2)^3}, \quad A_{2,3}(h_1, h_2) = 2\,\dfrac{(h_1 - h_2)\left({h_1}^2 - 3\,h_2\,h_1 + {h_2}^2\right)}{{h_1}^3{h_2}^3}, \\ B_{1,3}(h_1, h_2) = -\dfrac{h_2\,(2\,h_1 - h_2)}{{h_1}^2\,(h_1 + h_2)^2}, \quad B_{2,3}(h_1, h_2) = \dfrac{{h_1}^2 - 4\,h_2\,h_1 + {h_2}^2}{h_1}, \end{cases} \tag{2.7}$$

$$\begin{cases} A_{1,4}(h_1, h_2) = \dfrac{-5\,h_2\,h_1 - 4\,{h_2}^2 + 5\,{h_1}^2}{{h_1}^3\,(h_1 + h_2)^3}, \quad A_{2,4}(h_1, h_2) = \dfrac{4\,{h_1}^2 - 7\,h_2\,h_1 + 4\,{h_2}^2}{{h_1}^3{h_2}^3}, \\ B_{1,4}(h_1, h_2) = \dfrac{-2\,h_2 + h_1}{{h_1}^2\,(h_1 + h_2)^2}, \quad B_{2,4}(h_1, h_2) = 2\,\dfrac{h_1 - h_2}{{h_1}^2{h_2}^2}. \end{cases} \tag{2.8}$$

$$\begin{cases} A_{1,5}(h_1, h_2) = 2\,\dfrac{2\,h_1 + h_2}{{h_1}^3\,(h_1 + h_2)^3}, \quad A_{2,5}(h_1, h_2) = 2\,\dfrac{h_1 - h_2}{{h_1}^3{h_2}^3}, \\ B_{1,5}(h_1, h_2) = \dfrac{1}{{h_1}^2\,(h_1 + h_2)^2}, \quad B_{2,5}(h_1, h_2) = \dfrac{1}{{h_1}^2{h_2}^2}. \end{cases} \tag{2.9}$$

Denoting

$$\bar{\mathrm{u}} = \left[[u_1, \ldots, u_{N-1}], [u'_1, \ldots, u'_{N-1}]\right], \tag{2.10}$$

the discrete operator $(D_x^\alpha \bar{\mathrm{u}})_j \simeq \frac{d^\alpha}{dx^\alpha} u(x_j)$ (also denoted $\partial_x^\alpha u(x_j)$) at x_j is related to a_α by

$$(D_x^\alpha \bar{\mathrm{u}})_j = \alpha!\,(a_\alpha)_j \quad \alpha = 2, 3, 4, 5. \tag{2.11}$$

We should denote the coefficients a_α by $a_\alpha(\bar{\mathrm{u}}, \mathbf{h})$ and the discrete derivative $(D_x^\alpha \bar{\mathrm{u}})_j$ by $D_x^\alpha(\bar{\mathrm{u}}, \mathbf{h})_j$, but instead we adopt the notation (2.11). The operator $(D_x^\alpha \bar{\mathrm{u}})_j$ approximates $\partial_x^\alpha u(x_j)$ in terms of the six data $(u_k, u'_k)_{j-1 \le k \le j+1}$ and of the two mesh spacing h_j, h_{j+1}. The truncation error analysis for $D_x^\alpha \bar{\mathrm{u}}$ is given in Sect. 7.

- *Approximating the derivative u'_j by Hermitian derivatives $u_{x,j}$*

The discrete operators above depend on the gridfunction $\bar{\mathrm{u}}$ in (2.10). In particular, the vector of the derivatives $[u'_1, \ldots, u'_{N-1}]$ is assumed to be known. However, in our context below, the data u'_j is not available.[1] Thus, u'_j must be interpolated from the primal data u_j. Here, this interpolation is obtained by mean of an interpolation polynomial called $q(x)$. This polynomial is a priori different from $p(x)$ in (2.3). We define the polynomial $q(x)$ associated with x_j as

$$q(x) = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4. \tag{2.12}$$

[1] In some numerical methods, as the the finite element Discontinuous Galerkin approach, u' is kept as a primal unknown.

The gridfunction $\mathfrak{u}_x = [u_{x,1}, \ldots, u_{x,N-1}]$ is calculated implicitly as follows. Suppose that $u_{x,j-1}$ and $u_{x,j+1}$ are given. Then the vector $\mathbf{b} = [b_0, b_1, b_2, b_3, b_4]^T \in \mathbb{R}^5$ is the solution of the Hermite interpolation problem

$$\begin{cases} q(x_{j-1}) = u_{j-1}, \quad q(x_j) = u_j, \quad q(x_{j+1}) = u_{j+1}, \\ q'(x_{j-1}) = u_{x,j-1}, \quad q'(x_{j+1}) = u_{x,j+1}. \end{cases} \tag{2.13}$$

The vector $\mathbf{u}_x = [u_{x,j}]_{1\le j\le N-1}$ is then *defined* from the five data u_{j-1}, u_j, u_{j+1}, $u_{x,j-1}, u_{x,j+1}$ by

$$u_{x,j} = q'(x_j). \tag{2.14}$$

Relations (2.14) for $j = 1, \ldots, N-1$ form a linear system with unknown $\mathbf{u}_x$, which is given for a general grid by

$$\alpha_{1,j} u_{x,j} + u_{x,j} + \alpha_{2,j} u_{x,j} = \beta_{1,j} u_{j-1} + \beta_{2,j} u_j + \beta_{3,j} u_{j+1}. \tag{2.15}$$

The five coefficients $\alpha_{1,j}, \alpha_{2,j}, \beta_{1,j}, \beta_{2,j}$ and $\beta_{3,j}$ are

$$\begin{cases} \alpha_{1,j} = \dfrac{h_{j+1}^2}{(h_{j+1}+h_j)^2}, \quad \alpha_{2,j} = \dfrac{h_j^2}{(h_{j+1}+h_j)^2}, \quad \beta_{2,j} = \dfrac{2(h_{j+1}-h_j)}{h_j h_{j+1}}, \\ \beta_{1,j} = -\dfrac{2h_{j+1}^2(2h_j + h_{j+1})}{h_j(h_j+h_{j+1})^3}, \quad \beta_{3,j} = \dfrac{2h_j^2(2h_{j+1}+h_j)}{h_{j+1}(h_j+h_{j+1})^3}. \end{cases} \tag{2.16}$$

Relation (2.15) is called the *Hermitian closure for the derivative*. In [3, Lemma 4.2], it is shown that

$$\max_{1\le j\le N-1} |u'(x_j) - u_{x,j}| \le C \frac{h^4}{\epsilon} \max_{x\in[a,b]} |\partial_x^5 u(x)|, \tag{2.17}$$

where ϵ is given in $(2.2)_{ii}$. In the case of a regular grid $h = h_j = h_{j+1}$, $j = 1, \ldots, N-1$, (2.15) becomes the standard Hermitian relation given by

$$\begin{cases} \dfrac{1}{6}u_{x,j-1} + \dfrac{2}{3}u_{x,j} + \dfrac{1}{6}u_{x,j+1} = \dfrac{u_{j+1}-u_{j-1}}{2h}, \quad 1 \le j \le N-1, \\ u_{x,0} = u'(a), \quad u_{x,N} = u'(b). \end{cases} \tag{2.18}$$

Assuming periodicity, the truncation error on a uniform grid is given by.[2]

$$(u^*)_{x,j} - (\partial_x u)_j^* = -\frac{h^4}{180}(\partial_x^5 u)_j^* + O(h^6). \tag{2.19}$$

Replacing the exact derivatives $u'_j (= \partial_x u_j)$ in (2.11) by the approximate values $u_{x,j}$ gives the finite difference operator denoted by $\widetilde{D}_x^\alpha \mathfrak{u}$:

$$(\widetilde{D}_x^\alpha \mathfrak{u})_j = \alpha! \, (a_\alpha)_j([\mathfrak{u}], [\mathfrak{u}_x]) \quad \alpha = 2, 3, 4, 5. \tag{2.20}$$

The gridfunction $\mathfrak{u}_x = [u_{x,1}, \ldots, u_{x,N-1}]$ is not $[u'_1, \ldots, u'_{N-1}]$ but it is rather its approximation by (2.15). In addition this approximation is non-local (see Sect. 7). This implies that the truncation error at x_j depends on the full vector $\mathbf{h}$ and not only on h_j, h_{j+1} as in (7.5). Consider the particular case of the two operators $\widetilde{D}_x^2$ and $\widetilde{D}_x^4$, which satisfy

$$(\widetilde{D}_x^2 \mathfrak{u})_j \simeq \partial_x^2 u(x_j), \quad (\widetilde{D}_x^4 \mathfrak{u})_j \simeq \partial_x^4 u(x_j). \tag{2.21}$$

We have

[2] Refer to Sect. 7 for more comments on truncation error analysis.

Proposition 2.1 *Let $u(x)$ be a smooth function. Assuming that the grid (2.1) is such that*

$$c\, h \le h_j \le h, \quad c > 0, \tag{2.22}$$

The truncation errors for $\widetilde{D}_x^2$ and $\widetilde{D}_x^4$

$$\tilde{\mathfrak{t}}_2(u) = (\widetilde{D}_x^2 u^*) - (\partial_x^2 u)^*, \quad \tilde{\mathfrak{t}}_4(u) \quad = (\widetilde{D}_x^4 u^*) - (\partial_x^4 u)^* \tag{2.23}$$

satisfy the estimates

$$\begin{cases} |\tilde{\mathfrak{t}}_2(u)| \le Ch^3, & (i) \\ |\tilde{\mathfrak{t}}_4(u)| \le Ch, & (ii) \end{cases} \tag{2.24}$$

where C is a constant depending only on $u(x)$.

Proof The truncation error $\tilde{\mathfrak{t}}_2(u)$ is decomposed as

$$\begin{aligned} \tilde{\mathfrak{t}}_2(u) &= \widetilde{D}_x^2 u^* - (\partial_x^2 u)^* \\ &= \underbrace{\widetilde{D}_x^2 u^* - D_x^2 u^*}_{(I)} + \underbrace{D_x^2 u^* - (\partial_x^2 u)^*}_{(II)}. \end{aligned} \tag{2.25}$$

The term (I) is the truncation error $\mathfrak{t}_2(u)$ in $(7.5)_a$. It satisfies the estimate

$$|(II)| \le C(u)h^4. \tag{2.26}$$

Invoking (2.5), (2.6), (2.17), (2.20) and assuming that $c \le \epsilon$, it is easy to see that (II) satisfies

$$\begin{aligned} |(II)| &\le 2B_{1,2}(h_j, h_{j+1})|u_{x,j-1} - \partial_x u_{j-1}| \\ &\quad + 2B_{1,2}(h_{j+1}, h_j)|u_{x,j+1} - \partial_x u_{j+1}| \\ &\quad + 2B_{2,2}(h_j, h_{j+1})|u_{x,j} - \partial_x u_j| \\ &\le Ch^3, \end{aligned} \tag{2.27}$$

where C is a constant depending only on u. A similar argument is used for $(2.24)_{ii}$. □

In the case of a regular grid, (i.e with $h = h_j$, $1 \le j \le N-1$), the operators $\widetilde{D}_x^2 u$ and $\widetilde{D}_x^4 u$ are

$$\begin{cases} \widetilde{D}_x^2 \mathrm{u}_j = \tilde{\delta}_x^2 u_j = 2\delta_x^2 u_j - \delta_x u_{x,j}, \\ \widetilde{D}_x^4 \mathrm{u}_j = \delta_x^4 u_j = \dfrac{12}{h^2}\left(\delta_x u_{x,j} - \delta_x^2 u_j\right). \end{cases} \tag{2.28}$$

The nominal truncation error (i.e. assuming periodicity) of $\widetilde{D}_x^2$ and of of $\widetilde{D}_x^4$ are

$$\begin{cases} (\widetilde{D}_x^2 u^*)_j - (\partial_x^2 u)_j^* = \dfrac{h^4}{360}(\partial_x^6 u)_j^* + O(h^6), \\ (\widetilde{D}_x^4 u^*)_j - (\partial_x^4 u)_j^* = -\dfrac{h^4}{720}(\partial_x^8 u)_j^* + O(h^6). \end{cases} \tag{2.29}$$

In [4,6], the operators $\tilde{\delta}_x^2$ and δ_x^4 were introduced using the polynomial $q(x)$ only, and not the polynomial $p(x)$. The reason is that when using $u_{x,j}$ instead of $\partial_x u(x_j)$ in $p(x)$, then $p(x)$ and $q(x)$ are identical. Here for clarity of the presentation we choose to introduce separately the two polynomials $p(x)$ and $q(x)$. Another Hermitian closure for the derivative, different from (2.14), could be selected to define the approximate values $u_{x,j}$ (see for example Equation (3.37) in [5]). In fact, in the two dimensional setting introduced in [3], the polynomials p and q are distinct, as will be seen below in Sect. 3.

Remark 2.1 The truncation results in Proposition 2.1 are in agreement with Equations (4.27) and (4.35) of [17].

Remark 2.2 We refer to [25,32,33] for a general presentation of Hermite interpolation.

Remark 2.3 The linear system (2.15) is distinct from the one employed to define the cubic splines derivatives, [2, (2.1.14) p.12]. However, the two systems are close. Both coincide with (2.18) in the case of a regular grid. See also [11].

3 Finite Difference Scheme on Two Dimensional Irregular Grids by Interpolation

In two dimensions we consider a Cartesian grid in a rectangular domain embedding an irregular domain Ω. The definition of the finite difference operators on Cartesian grids can be obtained in two ways. The first is by expanding finite difference formulas in Taylor series (see [27]). In the second approach a local interpolating polynomial is defined around each computational point which is the center point of an element (regular or irregular) inside the physical domain. Successive differentiation of this polynomial provides discrete differential operators. This is done in [35] for the Biharmonic operator. The latter has been extended in [3] to define a high order approximation to the biharmonic operator on the stencil shown in Fig. 4. The polynomial $P(x, y)$, defined in [3], is the solution of a specific Birkhoff interpolation problem.[3] The polynomial $P(x, y) \in \mathcal{P}_{19} = \text{Span}\{l_k, k = 1, \ldots 19\}$ where the homogeneous polynomials $l_k(x)$ are

$$\begin{cases} l_1(x,y) = 1, \quad l_2(x,y) = x, \quad l_3(x,y) = x^2, \quad l_4(x,y) = x^3, \\ l_5(x,y) = x^4, \quad l_6(x,y) = x^5, \quad l_7(x,y) = y, \quad l_8(x,y) = y^2, \quad l_9(x,y) = y^3, \\ l_{10}(x,y) = y^4, \quad l_{11}(x,y) = y^5, \quad l_{12}(x,y) = xy, \quad l_{13}(x,y) = xy(x+y), \\ l_{14}(x,y) = xy(x-y), \quad l_{15}(x,y) = xy(x+y)^2, \quad l_{16}(x,y) = xy(x-y)^2, \quad l_{17}(x,y) = xy(x+y)^3, \\ l_{18}(x,y) = xy(x-y)^3, \quad l_{19}(x,y) = x^2y^2(x^2+y^2). \end{cases} \tag{3.1}$$

The polynomial $P(x, y)$ is equivalently expressed as

$$P(x,y) = P(0,0) + P_1(x) + P_2(y) + xyQ(x,y), \tag{3.2}$$

where

$$\begin{cases} \text{(a)} & P_1(sx) = a_2x + a_3x^2 + a_4x^3 + a_5x^4 + a_6x^5, \\ \text{(b)} & P_2(y) = a_7y + a_8y^2 + a_9y^3 + a_{10}y^4 + a_{11}y^5, \\ \text{(c)} & Q(x,y) = a_{12} + a_{13}(x+y) + a_{14}(x-y) + a_{15}(x+y)^2 + a_{16}(x-y)^2 \\ & \qquad + a_{17}(x+y)^3 + a_{18}(x-y)^3 + a_{19}xy(x^2+y^2). \end{cases} \tag{3.3}$$

Consider $\psi(x, y)$ a given function and $\psi^*_{i,j}$ its restriction to the grid. The quintic polynomials $P(0, 0) + P_1(x)$ and $P(0, 0) + P_2(y)$ solve interpolation problems of the form (2.4). The

[3] In a Birkhoff problem, the set of data carried by each point in the stencil is dependent on the point. We refer to [29] for a theory of multivariate Birkhoff interpolation.

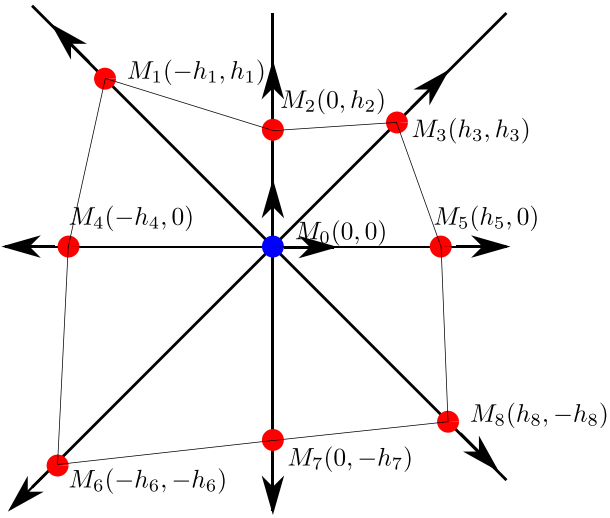

Fig. 4 Case of an irregular stencil: the 8 neighbor nodes of M_0 are the points M_1, M_2, M_3, M_4, M_5, M_6, M_7 and M_8 with $\mathbf{h} = [h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8]$. The 19 data determining uniquely the polynomial $P(x, y)$ in (3.2) are of two kinds. First, the 9 data for u are marked with circles. Second, the 10 data corresponding to directional derivatives are marked with arrows

polynomial $P(0,0) + P_1(x) \in \text{Span}\{1, x, x^2, x^3, x^4, x^5\}$ is given by the 6 horizontally aligned data (see Fig. 4):

$$\begin{cases} \Gamma_4(\psi) = \psi^*(\mathbf{M}_4), \quad \Gamma_5(\psi) = \psi^*(\mathbf{M}_0), \quad \Gamma_6(\psi) = \psi^*(\mathbf{M}_5), \\ \Gamma_{13}(\psi) = -\partial_x\psi^*(\mathbf{M}_4), \quad \Gamma_{14}(\psi) = \partial_x\psi^*(\mathbf{M}_0), \quad \Gamma_{16}(\psi) = \partial_x\psi^*(\mathbf{M}_5). \end{cases} \tag{3.4}$$

Similarly, the polynomial $P(0,0) + P_2(y) \in \text{Span}\{1, y, y^2, y^3, y^4, y^5\}$ is determined by the 6 vertically aligned data

$$\begin{cases} \Gamma_8(\psi) = \psi^*(\mathbf{M}_7), \quad \Gamma_5(\psi) = \psi^*(\mathbf{M}_0), \quad \Gamma_2(\psi) = \psi^*(\mathbf{M}_2), \\ \Gamma_{18}(\psi) = -\partial_y\psi^*(\mathbf{M}_7), \quad \Gamma_{15}(\psi) = \partial_y\psi^*(\mathbf{M}_0), \quad \Gamma_{11}(\psi) = \partial_y\psi^*(\mathbf{M}_2). \end{cases} \tag{3.5}$$

Consider now the 8 diagonally aligned data. They are denoted as:

$$\begin{cases} \Gamma_3(\psi) = \psi^*(\mathbf{M}_3), \quad \Gamma_7(\psi) = \psi^*(\mathbf{M}_6), \quad \Gamma_9(\psi) = \psi^*(\mathbf{M}_8), \quad \Gamma_1(\psi) = \psi^*(\mathbf{M}_1), \\ \Gamma_{12}(\psi) = (\partial_x + \partial_y)\psi^*(\mathbf{M}_3), \quad \Gamma_{17}(\psi) = (-\partial_x - \partial_y)\psi^*(\mathbf{M}_6), \\ \Gamma_{19}(\psi) = (\partial_x - \partial_y)\psi^*(\mathbf{M}_8), \quad \Gamma_{10}(\psi) = (-\partial_x + \partial_y)\psi^*(\mathbf{M}_1). \end{cases} \tag{3.6}$$

Proposition 3.1 *For any values $\alpha_1, \ldots, \alpha_8$, there exists a unique polynomial $Q(x, y)$ of the form $(3.3)_c$ satisfying the 8 relations*

$$\Gamma_k(Q) = \alpha_k, \tag{3.7}$$

where $k \in \mathcal{I}_Q = \{3, 7, 9, 1, 12, 17, 19, 10\}$

Proof Since the problem is linear, it is sufficient to prove that $\alpha_k = 0$, $k \in \mathcal{I}_Q$, implies that $Q(x, y) \equiv 0$. This is easily obtained by analyzing $Q(x, y)$ along the two diagonal directions $x = y$, $x = -y$: it is found that Q is a quartic polynomial along these two directions with 5 zero data. This implies $Q(x, y) \equiv 0$ by considering the coefficient of $xy(x^2 + y^2)$. □

As a corollary, there exists a unique solution $P(x, y)$ which is a solution of the 19×19 linear system

$$\Gamma_k(P) = \Gamma_k(\psi^*), \quad 1 \le k \le 19. \tag{3.8}$$

where the 19 data $\Gamma_k(\psi^*)$ are defined in (3.4), (3.5) and (3.6). Denoting

$$\mathbf{\Gamma}(\psi^*) = [\Gamma_k(\psi^*)]_{1 \le k \le 19} : \tag{3.9}$$

the polynomial $P(x, y)$ is expressed as,

$$P(x, y) = \sum_{k=1}^{19} a_k\big(\mathbf{\Gamma}(\psi^*)\big) l_k(x, y). \tag{3.10}$$

The expression of $P(x, y)$ in terms of $\mathbf{\Gamma}(\psi^*) \in \mathbb{R}^{19}$ and the lengths $\mathbf{h} \in \mathbb{R}^8$ is obtained by symbolic calculation.[4] The decomposition (3.2) is applied as follows. First, the polynomials $P_1(x)$ and $P_2(y)$) are calculated. They are of the form (2.3), in the directions x and y respectively. The coefficients of $P(x, 0)$ (resp. $P(0, y)$) are obtained by (2.5) in terms of h_4, h_5 (resp. h_2, h_7). Second, the polynomial $Q(x, y)$ is calculated by solving a 8×8 system deduced from Proposition 3.1. Third, the 8 data $\Gamma_l(Q)$ in (3.6) are expressed in terms of the 19 data $\Gamma_k(P)$, $k = 1, \ldots, 19$. This is easily obtained using (3.2).

4 Discrete Operators on Compact Stencils for the Navier–Stokes Equations

Let $\Omega \subset \mathbb{R}^2$ be a regular bounded domain. The Navier–Stokes system in pure stream function form and with viscosity $\nu > 0$ is

$$\begin{cases} \partial_t \Delta\psi(t, \mathbf{x}) + \Big(\nabla^\perp\psi \cdot \nabla(\Delta)\Big)\psi(t, \mathbf{x}) - \nu\Delta^2\psi(t, \mathbf{x}) = f(t, \mathbf{x}), \\ \psi(t, \mathbf{x}) = g_1(t, \mathbf{x}), \\ \dfrac{\partial\psi}{\partial n}(t, \mathbf{x}) = g_2(t, \mathbf{x}), \quad \mathbf{x} = (x_1, x_2) \in \partial\Omega. \end{cases} \tag{4.1}$$

with $\mathbf{x} = (x_1, x_2) \in \Omega$, $t > 0$. The velocity $\mathbf{v}(t, \mathbf{x})$ is related to ψ by

$$\mathbf{v}(t, \mathbf{x}) = \nabla^\perp\psi(t, \mathbf{x}) = (-\partial_2\psi, \partial_1\psi). \tag{4.2}$$

Three spatial operators must be approximated: the Biharmonic operator $\Delta^2\psi$, the Laplacian operator $\Delta\psi$ and the convective term $C(\psi) = \big(\nabla^\perp\psi \cdot \nabla(\Delta)\big)\psi$. The approximations, described hereafter, are all based on the polynomial P. They are called $\Delta^2_{\mathbf{h}}$, $\Delta_{\mathbf{h}}$ and $C_{\mathbf{h}}$ respectively. The numerical scheme used hereafter is second order accurate. It is expressed as

$$\begin{cases} \dfrac{\Delta_{\mathbf{h}}\psi^{n+1/2} - \Delta_{\mathbf{h}}\psi^n}{\Delta t/2} = -C_{\mathbf{h}}(\psi^n) + \dfrac{\nu}{2}\left(\Delta^2_{\mathbf{h}}\psi^n + \Delta^2_{\mathbf{h}}\psi^{n+1/2}\right) + (f(t^{n+1/4}, \cdot))^*, \\ \dfrac{\Delta_{\mathbf{h}}\psi^{n+1} - \Delta_{\mathbf{h}}\psi^n}{\Delta t} = -C_{\mathbf{h}}(\psi^{n+1/2}) + \dfrac{\nu}{2}\left(\Delta^2_{\mathbf{h}}\psi^n + \Delta^2_{\mathbf{h}}\psi^{n+1}\right) + (f(t^{n+1/2}, \cdot))^*. \end{cases} \tag{4.3}$$

Refer to [6,10] for the derivation of (4.3).

[4] Performed in MAPLE.

4.1 The Biharmonic Operator

In this section, we briefly review various finite difference Biharmonic operators used in regular stencils [6,10] or irregular stencils [3,7,17]. The 9 point stencil in regular form in shown in Fig. 5 and in irregular form in Fig. 4. The Biharmonic operator is

$$\Delta^2\psi(x,y) = \partial_x^4\psi(x,y) + \partial_y^4\psi(x,y) + 2\partial_x^2\partial_y^2\psi(x,y). \tag{4.4}$$

The basic discrete Biharmonic is the one of Stephenson [35]. It operates on the 9 point regular stencil. It is expressed as

$$\Delta_h^2\psi_{i,j} = \delta_x^4\psi_{i,j} + \delta_y^4\psi_{i,j} + 2\delta_x^2\delta_y^2\psi_{i,j}. \tag{4.5}$$

The Hermitian derivative $\psi_{x,i,j}$ (resp. $\psi_{y,i,j}$) is involved in δ_x^4, (resp. δ_y^4), (see Sect. 7). Both δ_x^4 and δ_y^4 are fourth order. The operator Δ_h^2 is second order with truncation error

$$\Delta_h^2(\psi)_{i,j}^* - (\Delta^2\psi)_{i,j}^* = \frac{h^2}{6}\left(\partial_x^4\partial_y^2 + \partial_x^2\partial_y^4)\psi\right)^* + O(h^4). \tag{4.6}$$

The second order accuracy in (4.6) is due to the approximation of the mixed term $(\partial_x^2\partial_y^2\psi)_{i,j}^*$ by $\delta_x^2\delta_y^2\psi_{i,j}^*$. A modified version $\tilde{\Delta}_h^2$ of Δ_h^2 was introduced in [7,17]:

$$\tilde{\Delta}_h^2\psi_{i,j} = \Delta_h^2\psi_{i,j} - \frac{h^2}{6}(\delta_x^2\delta_y^4 + \delta_y^2\delta_x^4)\psi_{i,j}. \tag{4.7}$$

The perturbation is $O(h^2)$. It is designed to cancel the leading truncation term in (4.6). The operator $\tilde{\Delta}_h^2\psi_{i,j}$ is fourth order. The truncation error is

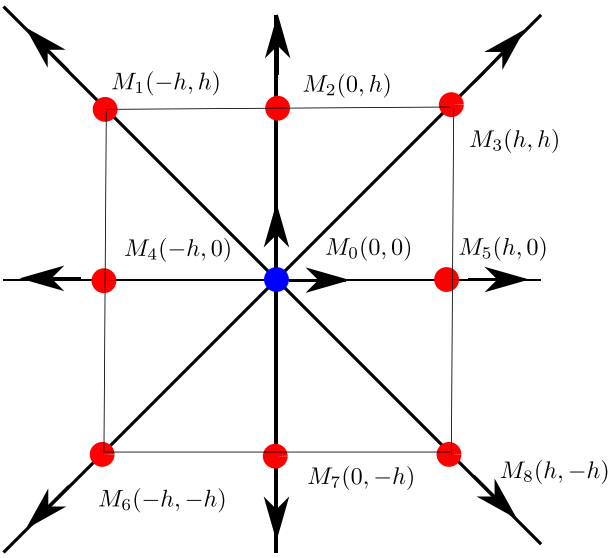

Fig. 5 Same as for Fig. 4, but in the case of a regular stencil with step size h

$$\tilde{\Delta}_h^2(\psi)_{i,j}^* - (\Delta^2\psi)_{i,j}^* = -h^4\left(\frac{1}{720}\left((\partial_x^8\psi)^* + (\partial_y^8\psi)^*\right) + \frac{1}{72}\left(\partial_x^4\partial_y^4\psi\right)^* - \frac{1}{180}\left((\partial_x^2\partial_y^6\psi)^* + (\partial_x^6\partial_y^2\psi)^*\right)\right) + O(h^6). \tag{4.8}$$

Consider now extending (4.7) to the irregular stencil shown in Fig. 4. The polynomial $P(x, y) = \sum_{k=1}^{19} a_k l_k(x, y)$ is associated to the point $\mathbf{M_0}$ with global coordinates (x_i, y_j) and local coordinates (0, 0). The approximate Biharmonic operator at $\mathbf{M_0}$ is defined in terms of the data $\mathbf{\Gamma}(\psi^*) = [\Gamma_l(\psi^*)]_{1\le l\le 19}$ via

$$\Delta_{\mathbf{h}}^2\psi_{i,j} = \sum_{k=1}^{19} a_k(\mathbf{\Gamma}(\psi^*))\Delta^2 l_k(0,0) = 24\Big(a_5(\mathbf{\Gamma}(\psi^*)) + a_{10}(\mathbf{\Gamma}(\psi^*))\Big) + 16\Big(a_{15}(\mathbf{\Gamma}(\psi^*)) - a_{16}(\mathbf{\Gamma}(\psi^*))\Big). \tag{4.9}$$

It has been proved in Theorem 3.11 of [3] that if $(u, \partial_x u, \partial_y u)$ are considered to be known exactly, then in analogy with $(7.5)_c$, $\Delta_{\mathbf{h}}^2\psi_{i,j}$ in (4.9) satisfies

$$|\Delta_{\mathbf{h}}^2(\psi^*)_{i,j} - (\Delta^2\psi)_{i,j}^*| \le Ch^2. \tag{4.10}$$

In fact, this estimate is optimal, since on a regular one dimensional grid, the operator $\Delta_{\mathbf{h}}^2$ coincides with $D_x^4\mathbf{u}_j$ in (2.11), which is second order accurate (see (7.5)(c)).

In Sect. 5 we approximate the first order derivatives $(\partial_x u, \partial_y u)$ via the Hermitian derivative (2.15). The latter are invoked in the approximation (4.9) of the Biharmonic operator. At regular points, the compact stencil is regular (see Fig. 5), i.e. $h_k = h$, $1 \le k \le 8$. It has been proved in [7, pp. 227-228] that the operator (4.9) coincides with (4.7), and therefore is fourth order accurate. At irregular points, which are near boundary, the accuracy drops to a lower value.

Remark 4.1 An alternative approximate Biharmonic operator may be derived from the identity [15]

$$\Delta^2\psi = \frac{2}{3}\left(\frac{\partial^4\psi}{\partial x^4} + \frac{\partial^4\psi}{\partial y^4}\right) + \frac{2}{3}\left(\frac{\partial^4\psi}{\partial \eta^4} + \frac{\partial^4\psi}{\partial \xi^4}\right). \tag{4.11}$$

In (4.11) (η, ξ) are the diagonal variables defined by $\eta = (x + y)/\sqrt{2}$, $\xi = (y - x)/\sqrt{2}$. A discrete counterpart of (4.11) was introduced in [17] as

$$\widehat{\Delta}_{\mathbf{h}}^2 u_{i,j} = \frac{2}{3}(\delta_x^4 u_{i,j} + \delta_y^4 u_{i,j}) + \frac{2}{3}(\delta_\eta^4 u_{i,j} + \delta_\xi^4 u_{i,j}). \tag{4.12}$$

If all the first-order derivatives with respect to x, y, ξ, η are approximated via the Hermitian derivative, then the scheme is fourth-order accurate. If the first-order derivatives with respect to x, y are computed via the Hermitian derivative and the first-order derivatives with respect to ξ, η are computed from the first-order derivatives with respect to x, y via the chain rule, then the scheme is second order accurate. In [17], the approximate Biharmonic operator (4.12) was used only at near boundary points, whereas (4.7) was used at interior points. This resulted in an observed fourth order accuracy of the numerical results.

4.2 Discrete Laplacian and Convective Term

Consider first the Laplacian. The discrete Laplacian used in the sequel is the operator $\Delta_{\mathbf{h}}$ defined by

$$\Delta_{\mathbf{h}}\psi_{i,j} = \sum_{k=1}^{19} a_k(\mathbf{\Gamma}(\psi^*))\Delta l_k(0,0). \tag{4.13}$$

This gives

$$\Delta_{\mathbf{h}}\psi_{i,j} = 2\Big(a_2(\mathbf{\Gamma}(\psi^*)) + a_8(\mathbf{\Gamma}(\psi^*))\Big). \tag{4.14}$$

The discrete Laplacian is easily proved to be fourth order. On a regular grid, $\Delta_{\mathbf{h}}$ satisfies

$$\Delta_{\mathbf{h}} = \tilde{\delta}_x^2 + \tilde{\delta}_y^2. \tag{4.15}$$

In what follows, the convective term $C(\psi)$ is approximated by $C_{\mathbf{h}}(\psi)$ defined by

$$C_{\mathbf{h}}(\psi) = -\psi_y\big[(\Delta \circ \partial_x)P(0,0)\big] + \psi_x\big[(\Delta \circ \partial_y)P(0,0)\big]. \tag{4.16}$$

In terms of the coefficients $a_k(\mathbf{\Gamma}(\psi^*))$ this gives the formula:

$$\begin{aligned} C_{\mathbf{h}}(\psi) = &-\psi_y\Big(6a_4(\mathbf{\Gamma}(\psi^*)) + 2\left\{a_{13}(\mathbf{\Gamma}(\psi^*))) - a_{14}(\mathbf{\Gamma}(\psi^*)\right\}\Big) \\ &+\psi_x\Big(6a_9(\mathbf{\Gamma}(\psi^*)) + 2a_{13}\left\{\mathbf{\Gamma}(\psi^*)) + a_{14}(\mathbf{\Gamma}(\psi^*)\right\}\Big), \end{aligned} \tag{4.17}$$

where a_4, a_9, a_{13}, a_{14} are given in (3.3). In the case of a regular grid, (4.17) is second order accurate. For this reason, the global order in space of the semi discrete scheme (4.3) is second order. This accuracy will be observed numerically below in Sect. 5.

Remark 4.2 We have chosen to approximate the convective term by (4.16) mainly for simplicity. We refer to [5], formula (3.37) for a fourth order accurate convective term in the regular case. Extending the approach in [5] to the present embedding context will be the topic of a forthcoming work. See also [17] for an alternative approach.

5 Numerical Results for Navier–Stokes Equation in Pure Streamfunction

In order to assess the accuracy of our scheme for the system (4.1) three numerical test cases are presented. In Sect. 5.1 we refer to Test cases 1 and 2, where the Navier–Stokes problem (4.1) is approximated in an ellipse. In Test cases 1 and 2 the exact solutions are chosen to be polynomials in space and exponential in time. We inspect the accuracy of the numerical solution by measuring the errors in ψ, $\partial_x\psi$ and $\partial_y\psi$. Test case 2 was also considered in [28], where a Cartesian embedding scheme for the Navier–Stokes equations was applied.

In Sect. 5.2 we approximate the solution of the Navier–Stokes equation in a square. The physical domain (the square) is rotated with some angle θ and then embedded in a bigger square, which is aligned to the x, y grid. We analyze numerically the way the angle of rotation θ of the physical domain affects the accuracy of the computation. For this purpose we consider an analytic solution of the Navier–Stokes equations in the embedded square. We observe the behaviour of the numerical solution as a function of the rotation angle θ. From the numerical results one may observe that the angle of rotation has limited impact on the quality of the numerical results.

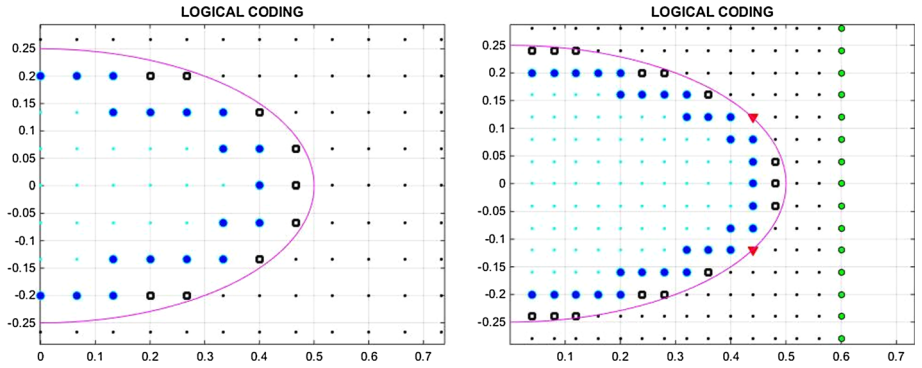

Fig. 6 Cases 1 and 2: The ellipse $x^2/0.5^2 + y^2/0.25^2 = 1$. Left panel: coarser grid with embedding square is $[-1, 1] \times [-1, 1]$. Right panel: finer grid with embedding square is $[-0.6, 0.6] \times [-0.6, 0.6]$. The three types of points inside the domain are: *Regular calculated* points (small circles), *Irregular calculated* points (large circle) and *Edge* points (open squares)

5.1 Navier–Stokes Equation in an Ellipse

5.1.1 Test Case 1

In Test case 1 and Test case 2, we consider the Navier–Stokes equation in an irregular domain Ω, which is defined by the ellipse

$$\frac{x^2}{0.5^2} + \frac{y^2}{0.25^2} = 1. \tag{5.1}$$

This ellipse is equivalently defined as the level set zero of the function

$$\varphi(x, y) = \sqrt{4x^2 + 16y^2} - 1. \tag{5.2}$$

In Test case 1, the forcing function $f(t, x, y)$ is chosen to be the right-hand side of Eq. (4.1), for which the exact solution is

$$\psi_e(t, x, y) = (x^4 + y^4)^2 \exp(t). \tag{5.3}$$

The spatial part $(x^4 + y^4)^2$ is a polynomial of order 8. This polynomial is not included in the space $\mathcal{P}_{19}$ in (3.1). The viscosity is chosen as $\nu = 1/1000$. The following errors are reported in Table 2 at the final time $T_f = 0.5$:

$$\begin{cases} \max\limits_{i,j} |(\psi_e)^*_{i,j}(T_f) - \psi_{i,j}(T_f)|, \\ \max\limits_{i,j} |(\partial_x \psi_e)^*_{i,j}(T_f) - \psi_{x,i,j}(T_f)|, \quad \max\limits_{i,j} |(\partial_y \psi_e)^*_{i,j}(T_f) - \psi_{y,i,j}(T_f)|. \end{cases} \tag{5.4}$$

Figure 7 put later displays the least-square convergence rates corresponding to this test case. Two embedding squares are considered. In the first case the embedding square is $[-1, 1] \times [-1, 1]$ and in the second it is $[-0.6, 0.6] \times [-0.6, 0.6]$. The number of points in each direction x, y are chosen to be the same for both cases. Therefore, the square $[-1, 1] \times [-1, 1]$ corresponds to a coarser grid of the ellipse, and the square $[-0.6, 0.6] \times [-0.6, 0.6]$ corresponds to a finer grid. The detailed embedding grid is shown in Fig. 6 for the two cases. In comparing the two cases, we would like to see whether the accuracy deteriorates in

Table 1 Cases 1 and 2: in the case of a computational grid 61×61, the two embedding squares $[-1, 1] \times [-1, 1]$ and $[-0.6, 0.6] \times [-0.6, 0.6]$ are compared

grid 61×61	Embed. sq. $[-1, 1] \times [-1, 1]$	Embed. sq. $[-0.6, 0.6] \times [-0.6, 0.6]$
Total number of points	$61 \times 61 = 3721$	$61 \times 61 = 3721$
Number of regular calculated points	233	779
Number of irregular calculated points	80	142
Total number of calculated points	313	921
Number of edge points	38	52
Number of boundary points	6	26

The embedding square $[-1, 1] \times [-1, 1]$ corresponds to a coarse grid with 313 vertices inside the ellipse (5.1). The embedding square $[-0.6, 0.6] \times [-0.6, 0.6]$ correspond to a fine grid with 779 vertices inside the ellipse (5.1)

Table 2 Case 1: Time dependent Navier–Stokes equation with exact solution $\psi(t, x, y) = (x^4 + y^4)^2 \exp(t)$

Grid $N \times N$	N_{ite}	e_∞	Rate	$(e_x)_\infty$	Rate	$(e_y)_\infty$	Rate
20×20	20	4.007(−5)		9.3139(−4)		2.3117(−4)	
40×40	80	8.4084(−7)	5.57	4.8964(−5)	4.25	2.3607(−5)	3.29
80×80	320	1.5857(−7)	2.41	3.3584(−6)	3.87	4.4192(−6)	2.42
160×160	1280	3.1146(−8)	2.34	8.6214(−7)	1.96	1.2430(−6)	1.83

The domain is the ellipse $x^2/0.5^2 + y^2/0.25^2 = 1$. The embedding square is $[-1, 1] \times [-1, 1]$, which corresponds to a coarser grid of the ellipse. The final time is $T_f = 0.5$ and the viscosity is $\nu = 0.001$. The asymptotic order of convergence is close to 2 for ψ and grad ψ. This is due to the convective term (4.16) which is approximated to order 2. Note that the errors are very small in magnitude

Table 3 Case 1: Time dependent Navier–Stokes equation with exact solution $\psi(t, x, y) = (x^4 + y^4)^2 \exp(t)$

Grid $N \times N$	N_{ite}	e_∞	Rate	$(e_x)_\infty$	Rate	$(e_y)_\infty$	Rate
20×20	20	5.8548(−5)		1.5547(−4)		6.5855(−5)	
40×40	80	8.7691(−7)	6.06	1.1092(−5)	3.81	1.7557(−5)	1.91
80×80	320	3.2634(−8)	4.75	8.3786(−7)	3.73	1.0619(−6)	4.05
160×160	1280	3.9768(−9)	3.04	1.2299(−7)	2.77	1.6797(−7)	2.66

The domain is the ellipse $x^2/0.5^2 + y^2/0.25^2 = 1$. The embedding square $[-0.6, 0.6] \times [-0.6, 0.6]$, which corresponds to a fine grid of the ellipse. The final time is $T_f = 0.5$ and the viscosity is $\nu = 0.001$

cases where the irregular domain is embedded in a larger computational square. According to Table 1, there are 921 calculated points in the finer grid (which corresponds to embedding the domain in $[-0.6, 0.6] \times [-0.6, 0.6]$) and 313 in the coarser grid (which corresponds to embedding the domain in $[-1, 1] \times [-1, 1]$). As expected, a better accuracy is observed with the finer grid. However, in both cases the errors are small and comparable in magnitude and the convergence rates are satisfactory (around second order and above). See Tables 2, 3 and Fig. 7.

Remark 5.1 The size of the computational domain is usually adjusted to the physical domain as much as possible to reduce the computational effort. However, with our present coding, a direct solver is used to solve the global linear systems. The points exterior to the physical domain as well as the edge points are dummy points associated to the identity matrix in

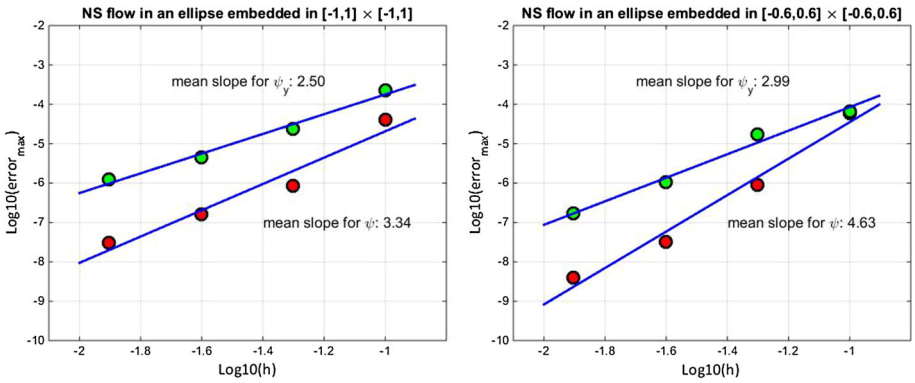

Fig. 7 Case 1: Navier Stokes equations in the ellipse $x^2/0.5^2+y^2/0.25^2 = 1$ with exact solution $\psi(t, x, y) = (x^4 + y^4)^2 \exp(t)$. Regression line for the errors on ψ and $\partial_y\psi$. Left panel: coarse grid with embedding square $[-1, 1] \times [-1, 1]$. The averaged rate of convergence for $|(\partial_x\psi)^*_{i,j}(T_f) - \psi_{x,i,j}(T_f)|)$ is 3.41 (not shown). Right panel: fine grid with embedding square $[-0.6, 0.6] \times [-0.6, 0.6]$. The rate of convergence for $|(\partial_x\psi)^*_{i,j}(T_f) - \psi_{x,i,j}(T_f)|)$ is 3.47 (not shown). The grids are $N = 20, 40, 80, 160$. The accuracy for ψ and grad ψ ranges between 2 and 4. In addition, the error levels are very good, due to the polynomial regularity of the exact solution

global linear systems. Therefore, the computational effort is in fact smaller with the large square $[-1, 1] \times [-1, 1]$ than with the small square $[-0.6, 0.6] \times [-0.6, 0.6]$.

5.1.2 Test Case 2

We consider again the same ellipse (5.1) embedded in the square $[-0.6, 0.6] \times [-0.6, 0.6]$, which corresponds to the fine grid in Sect. 5.1.1. The exact solution is

$$\psi_e(t, x, y) = \begin{cases} K\left(r(x, y)^\alpha - (\frac{1}{4})^\alpha\right)\cos(t), & \text{if } \varphi(x, y) \le 1 \\ 0, & \text{otherwise}, \end{cases} \tag{5.5}$$

where $r(x, y) = x^2 + 4y^2$, $\alpha > 0$ and K are parameters to be picked. The corresponding velocity is

$$\begin{cases} v_1(t, x, y) = -\partial_y\psi_e(t, x, y), \\ v_2(t, x, y) = \partial_x\psi_e(t, x, y). \end{cases} \tag{5.6}$$

The vorticity is

$$\omega_e(t, x, y) = \Delta\psi_e(t, x, y). \tag{5.7}$$

In [28] the particular case $\alpha = 1$, $K = 1$ is considered. The shape of the exact and approximated solution are shown in Fig. 8 (left panel). In Table 4 and in Fig. 9 (left panel) the numerical results are reported using our scheme. Note that in this case the spatial part of the function ψ_e belongs to $\text{Span}(x^4, y^4, x^2y^2) \subset \mathcal{P}_{19}$. So that there is no truncation error in the spatial direction x, y when using the scheme (4.3) and therefore there is no error in space. The errors originate from the temporal discretization. Therefore, fourth order accuracy is expected and observed in Table 4.

A more difficult case is obtained when we chose the exact solution of (5.5) with parameters $\alpha = 2$ and $K = 20$. The shape of the exact and approximated solutions are shown in Fig. 8 (right panel). The error levels for ψ, $\partial_x\psi$ and $\partial_y\psi$ are reported in Table 5. The convergence

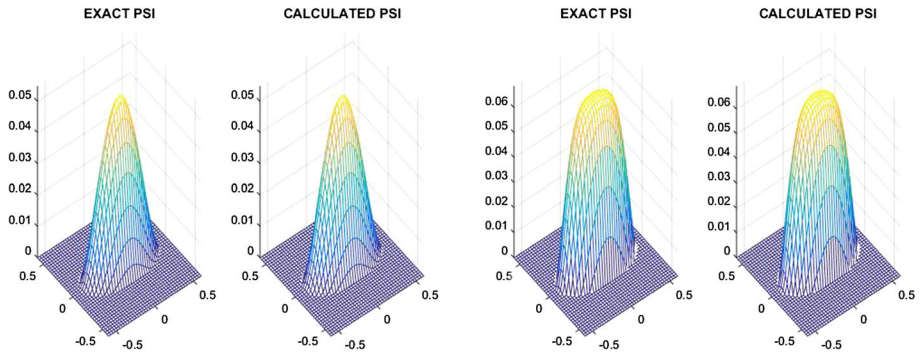

Fig. 8 Case 2: Ellipse (5.1) embedded in the square $[-0.6, 0.6] \times [-0.6, 0.6]$. The Cartesian embedding grid is 40×40 and there are 80 time iterations in (4.3). The exact and approximated solutions $\psi(T_f = 0.5, x, y)$ are shown. Left panel: parameters $\alpha = 1$, $K = 1$ in (5.5). Right panel: parameters $\alpha = 2$, $K = 20$ in (5.5)

Table 4 Case 2: Time dependent Navier–Stokes equation with $\nu = 0.001$ in the ellipse $x^2/0.5^2 + y^2/0.25^2 = 1$ with exact solution (5.5) with the parameter $\alpha = 1$ and $K = 1$

Grid $N \times N$	N_{ite}	e_∞	Rate	$(e_x)_\infty$	Rate	$(e_y)_\infty$	Rate
20×20	20	3.3063(−7)		1.8255(−6)		3.9043(−6)	
40×40	80	1.7633(−8)	4.23	1.2356(−7)	3.88	2.2015(−7)	4.15
80×80	320	1.0386(−9)	4.09	7.8840(−9)	3.97	1.2288(−8)	4.16
160×160	1280	6.8827(−11)	3.91	5.1392(−10)	3.94	7.8928(−10)	3.96

The final time is $T_f = 0.5$. The embedding square is $[-0.6, 0.6] \times [-0.6, 0.6]$. The asymptotic convergence rate is close to 4 for ψ and grad ψ. In this case, the truncation error in space vanishes since the exact solution belongs to the space $\mathcal{P}_{19}$. The error is only due to the time integration scheme

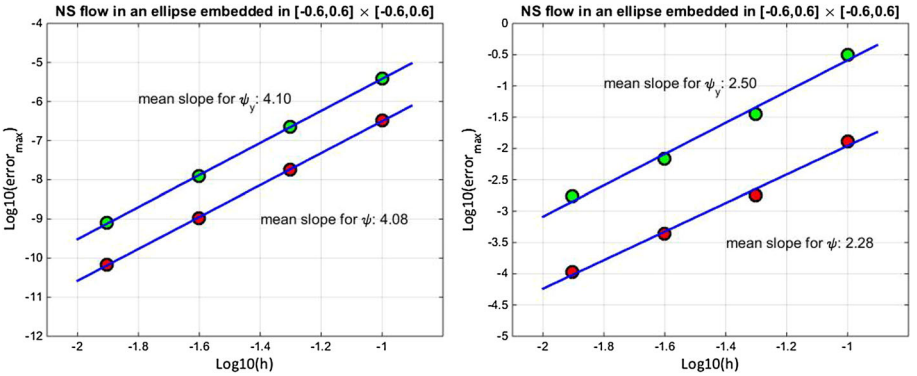

Fig. 9 Case 2: Navier Stokes equations in the ellipse $x^2/0.5^2 + y^2/0.25^2 = 1$ with exact solution (5.5). The embedding square is $[-0.6, 0.6] \times [-0.6, 0.6]$. Left panel: Parameters $\alpha = 1$ and $K = 1$ in (5.5). The rate of convergence for $|(\partial_x \psi)^*_{i,j}(T_f) - \psi_{x,i,j}(T_f)|)$ is 3.94 (not shown). Right panel: Parameters $\alpha = 2$ and $K = 20$ in (5.5). The rate of convergence for $|(\partial_x \psi)^*_{i,j}(T_f) - \psi_{x,i,j}(T_f)|)$ is 2.59 (not shown)

slopes are displayed in Fig. 9. The observed accuracy is around 4 for $\psi(t, x, y)$ grad $\psi(t, x, y)$ for the case $\alpha = 1$, $K = 1$ and around 2 for the case $\alpha = 2$, $K = 20$.

Table 5 Case 2: Time dependent Navier–Stokes equation with $\nu = 0.001$ in the ellipse $x^2/0.5^2 + y^2/0.25^2 = 1$ with exact solution (5.5) with the parameter $\alpha = 2$ and $K = 20$

Grid $N \times N$	N_{ite}	e_∞	Rate	$(e_x)_\infty$	Rate	$(e_y)_\infty$	Rate
20×20	20	1.3111(−2)		9.9770(−2)		3.1718(−1)	
40×40	80	1.7879(−3)	2.87	1.7643(−2)	2.50	3.5885(−2)	3.15
80×80	320	4.3797(−4)	2.03	4.5494(−3)	1.996	6.9431(−3)	2.37
160×160	1280	1.0803(−4)	2.02	1.1213(−3)	2.021	1.7036(−3)	2.03

The final time is $T_f = 0.5$. The embedding square is $[-0.6, 0.6] \times [-0.6, 0.6]$. The asymptotic order of convergence is 2. The error levels for ψ, ψ_x and ψ_y are very good

5.2 Invariance Under Grid Rotation: A Numerical Study

In this test case we consider the domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$ embedded in the computational square $[-1, 1] \times [-1, 1]$. The Navier Stokes equations (4.1) with $\nu = 1/1000$ are solved in Ω with a forcing term $f(t, x, y)$ such that the radial, polynomial in space solution is $\psi_e(t, x, y) = (x^2 + y^2)^4 e^{-t}$. The computation is repeated for a series of rotated positions of $[-0.5, 0.5] \times [-0.5, 0.5]$ in $[-1, 1] \times [-1, 1]$. This permits to evaluate the effect of the position of the calculated points on the global accuracy. In particular, the labelling of near-boundary points changes when the domain changes of position.

We let rotate Ω from the position $\theta = 0$ to the position $\theta = \pi/4$ with an angle step $\Delta\theta = \pi/360$ (0.5 deg). For each position $\theta_k = k\Delta\theta$, $k = 0, \ldots 90$ the same computation is reproduced. First, a grid 20×20 is considered. Second, a grid 30×30 is used. In each case,

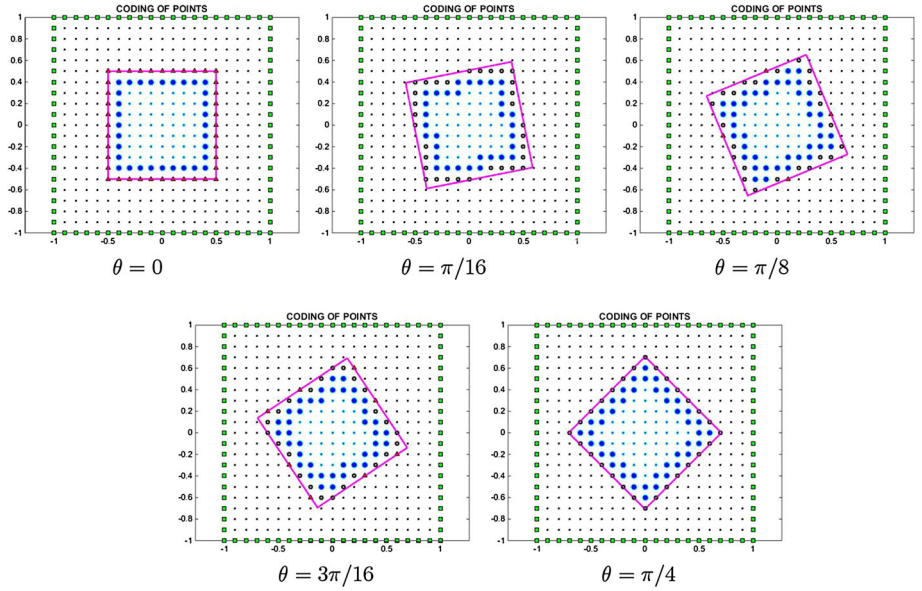

Fig. 10 Effect of the position of the domain in the embedding square on the accuracy. The square $[-0.5, 0.5]$ embedded in the computational square $[-1, 1] \times [-1, 1]$ is shown at positions $\theta = 0$, $\theta = \pi/16$, $\theta = \pi/8$, $\theta = 3\pi/16$ and $\theta = \pi/4$. The type of near boundary points (*Regular calculated*, *Irregular calculated* or *Edge*) varies according to their position with respect to the boundary. See Fig. 1

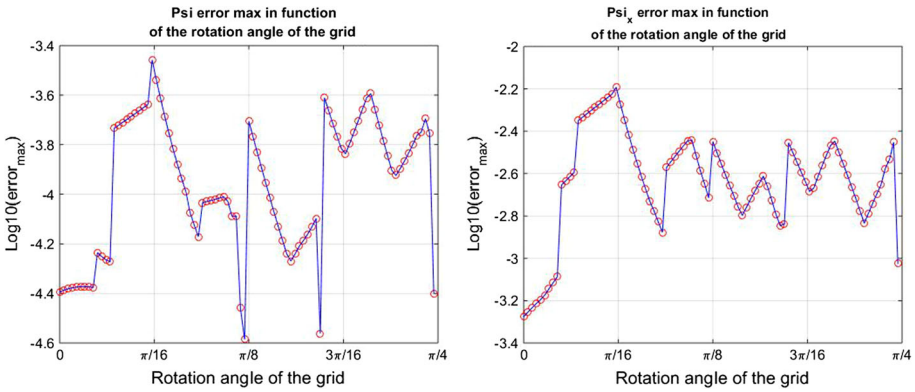

Fig. 11 Case 3: Maximum error for the Navier–Stokes equation in the square with size $[-0.5, 0.5]\times[-0.5, 0.5]$ embedded in $[-1, 1] \times [-1, 1]$. The grid is 20×20. The exact solution is $\psi(t, x, y) = (x^2 + y^2)^4 \exp(-t)$. The final time $T_f = 0.25$ is reached in 20 iterations. The same computation is repeated 91 times on the domain $[-0.5, 0.5] \times [-0.5, 0.5]$ rotated with an angle $\theta = k\frac{\pi}{360}$ for $k = 0, \ldots, 90$. Left panel: maximum error for $\psi(t, x, y)$ at time $T_f = 0.25$ in function of the position of the domain. Right panel: maximum error for $\psi_x(t, x, y)$ at time T_f in function of the position of the domain. The accuracy for ψ_y is similar to the one for ψ_x

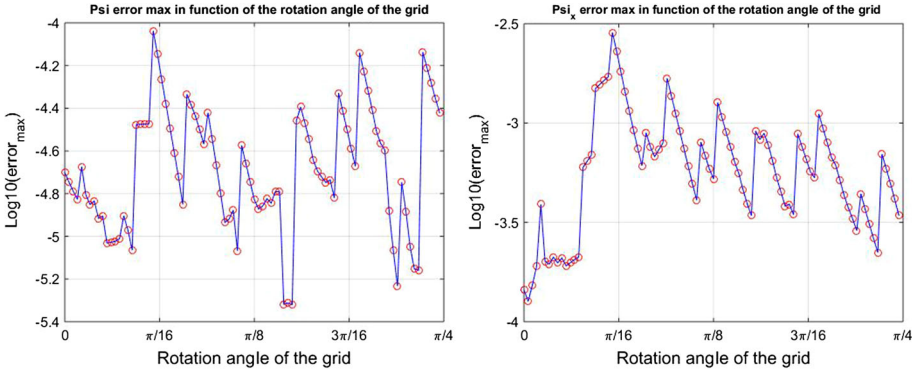

Fig. 12 Case 3: Maximum error for the Navier–Stokes equation in the square $[-0.5, 0.5] \times [-0.5, 0.5]$ embedded in $[-1, 1] \times [-1, 1]$. The grid is 30×30. The exact solution is $\psi(t, x, y) = (x^2 + y^2)^4 \exp(-t)$. The final time $T_f = 0.25$ is reached in 45 iterations. The same computation is repeated 91 times on the domain $[-0.5, 0.5] \times [-0.5, 0.5]$ rotated with an angle $\theta = k\frac{\pi}{360}$ for $k = 0, \ldots, 90$. Left panel: maximum error for $\psi(t, x, y)$ at time $T_f = 0.25$ in function of the position of the domain. Right panel: maximum error for $\psi_x(t, x, y)$ at time T_f in function of the position of the domain. The accuracy for ψ_y is similar to the one for ψ_x

the maximum errors for $\psi(T_f, x, y)$ and $\partial_x\psi(T_f, x, y)$ are evaluated at final time $T_f = 0.25$ using the scheme (4.3)

Several positions of Ω are shown in Fig. 10. The maximum errors for $\psi(T_f, x_i, y_j)$ and $\partial_x\psi(T_f, x_i, y_j)$ with the grid 20×20 in function of the position angle are reported in Fig. 11. The error for $\partial_y\psi(T_f, x_i, y_j)$ is similar to the one obtained for $\partial_x\psi(T_f, x_i, y_j)$. In Fig. 11, one sees that the error level slightly fluctuates in function of the angle within the interval $[10^{-4.6}, 10^{-3.4}]$ for ψ and $[10^{-3.3}, 10^{-2.2}]$ for $\partial_x\psi$. In Fig. 12 the error levels are reported with a grid 30×30. As expected, the error levels are smaller with the grid 30×30 than with the 20×20 grid. The behavior of the error appears to be piecewise linear in all cases. The

jumps in the errors are attributed to the change of labelling of near boundary points when the physical domain rotates. Despite these jumps in the errors, the magnitude of the error varies by a factor of 10 only and remains small. Overall, the errors behave satisfactorily.

6 Conclusion

The numerical approach to the Navier–Stokes equation in pure streamfunction introduced in [10] is extended to irregular domains using embedded finite differences. The methodology presented uses high order interpolating polynomials. The numerical results presented are promising. The accuracy obtained so far is remarkable and is in line with the one observed in the purely Cartesian case.

The approach needs to be extended to more complex problems, with higher Reynolds numbers and more irregular geometries. In addition higher order time schemes with A- and L- stable must be considered. Bifurcation phenomena for flows in elliptical domains will be explored in the near future. Finally performing a stability analysis in the spirit of [13,36,37] is also a perspective.

7 Appendix: Truncation Error Analysis for Finite Differences

In this section, we present a truncation error analysis for finite differences on a bounded interval. Suppose given a regular function $u(x)$ on $[a, b]$. The operator $u(x) \mapsto u^*$ maps the function $u(x)$ on the gridfunction obtained by evaluating $u(x)$ at gridpoints. We write

$$u^* = [u(x_1), \ldots, u(x_{N-1})], \tag{7.1}$$

meaning that the evaluation holds at all scales $h > 0$. The notion of gridfunction is convenient to express the truncation error of a finite difference operator. For example the truncation error $u \mapsto \mathfrak{t}(u)$ of the operator δ_x^2 is the gridfunction obtained by

$$\mathfrak{t}(u) = \delta_x^2(u^*) - (\partial_x^2 u)^*. \tag{7.2}$$

The components of $\mathfrak{t}(u)$ are

$$\begin{aligned} \mathfrak{t}(u)_j &= (\delta_x^2 u^*)_j - (\partial_x^2 u)_j^* \\ &= \frac{h^2}{12}(\partial_x^4 u)_j^* + O(h^4). \end{aligned} \tag{7.3}$$

In (7.3) $u(x)$ is assumed to be periodic on $[a, b]$. This corresponds to a *nominal order* of 2. For any regular function $u(x)$, we denote $\bar{u}^* = [u^*, (\partial_x u)^*]$ where u^*, $(\partial_x u)^*$ are the gridfunctions associated to $u(x)$ and $\partial_x u(x)$ respectively.

Consider for example the discrete operator D_x^α in (2.11). The truncation error $u \mapsto \mathfrak{t}_\alpha(u)$ is

$$\mathfrak{t}_\alpha(u) = D_x^\alpha \bar{u}^* - (\partial_x^\alpha u)^*. \tag{7.4}$$

The leading term of the truncation error of D_x^α is found[5] in the case of a general irregular grid (2.1) for $\alpha = 1, 2, 3$ and 4 as

[5] With MAPLE.

$$\begin{cases} \mathfrak{t}_2(u)_j = (D_x^2\bar{u}^*)_j - (\partial_x^2 u)_j^* = -\dfrac{1}{360}h_j^2 h_{j+1}^2 \partial_x^6 u_j + O(h^5), \quad (a) \\ \mathfrak{t}_3(u)_j = (D_x^3\bar{u}^*)_j - (\partial_x^3 u)_j^* = \dfrac{1}{60}(h_j^2 h_{j+1} - h_j h_{j+1}^2)\partial_x^6 u_j + O(h^4), \quad (b) \\ \mathfrak{t}_4(u)_j = (D_x^4\bar{u}^*)_j - (\partial_x^4 u)_j^* = -\dfrac{1}{30}(h_j^2 + h_{j+1}^2 - 4h_j h_{j+1})\partial_x^6 u_j + O(h^3), \quad (c) \\ \mathfrak{t}_5(u)_j = (D_x^5\bar{u}^*)_j - (\partial_x^5 u)_j^* = \dfrac{1}{3}(h_{j+1} - h_j)\partial_x^6 u_j + O(h^2) \quad (d). \end{cases} \tag{7.5}$$

The leading terms in the right-hand sides of (7.5) are associated solely with the three grid-points $\{x_{j-1}, x_j, x_{j+1}\}$ and the two mesh spacing h_j, h_{j+1}. In the case where the boundary conditions are taken in account, the truncation error will in general depend on the full set of values $\partial_x^{(5)} u(x_k)$, $1 \le k \le N-1$. In this case, only an estimate of the truncation error is available in general. Even in the case of a regular grid with $h = h_j$, the leading term of the truncation error also depends on the full set of values $\partial_x^{(5)} u(x_k)$, $1 \le k \le N-1$. As an example we have for the Hermitian derivative on a regular grid the following

Proposition 7.1 (Pointwise truncation error of the Hermitian derivative). *Let* $\mathfrak{u} \mapsto \mathfrak{u}_x$ *be the Hermitian gradient defined in* (2.18) *with boundary data* $u_{x,0} = 0$, $u_{x,N} = 0$. *Then the truncation error is expressed as the multi-point Taylor expansion*

$$\mathfrak{t}(u)_i = (u^*)_{x,i} - (\partial_x u)_i^* = -\frac{h^4}{180}\sum_{j=1}^{N-1} \alpha_j^i (\partial_x^{(5)} u)^*(x_j) + O(h^6). \tag{7.6}$$

where

$$\alpha_j^i = \sum_{k=1}^{N-1} \frac{Z_k^i Z_k^j}{1 - \lambda_k/6}, \tag{7.7}$$

and

$$\begin{cases} Z_k^i = \sqrt{\dfrac{2}{N}} \sin\left(\dfrac{ik\pi}{N}\right), \quad 1 \le i, k \le N-1, \\ \lambda_k = 4\sin^2\left(\dfrac{k\pi}{N}\right), \quad 1 \le k \le N-1. \end{cases} \tag{7.8}$$

Proof The relation (2.18) is rewritten

$$\left(I + \frac{h^2}{6}\delta_x^2\right)\mathfrak{u}_x = \delta_x \mathfrak{u}. \tag{7.9}$$

This gives that $\mathfrak{u}_x = \left((I + \frac{h^2}{6}\delta_x^2)^{-1} \circ \delta_x\right)\mathfrak{u}$. Let $x \in [0,1] \mapsto u(x)$ be a regular function. The truncation error $u \mapsto \mathfrak{t}(u)$ is

$$\begin{aligned} \mathfrak{t}(u) &= (u^*)_x - (\partial_x u)^* \\ &\quad \left(I + \frac{h^2}{6}\delta_x^2\right)^{-1}\left(\delta_x u^* - (I + \frac{h^2}{6}\delta_x^2)(\partial_x u)^*\right) \\ &= \left(I + \frac{h^2}{6}\delta_x^2\right)^{-1}\Big(\underbrace{(\delta_x u^* - (\partial_x u)^*)}_{(I)} - \underbrace{\frac{h^2}{6}\delta_x^2(\partial_x u)^*}_{(II)}\Big). \end{aligned} \tag{7.10}$$

The term (I) is expressed as

$$(I) = \frac{h^2}{6}(\partial_x^3 u)^* + \frac{h^4}{120}(\partial_x^5 u)^* + \mathfrak{v}, \tag{7.11}$$

where $\mathfrak{v}$ satisfies

$$\|\mathfrak{v}\|_{L^\infty} \leq Ch^6 \|(\partial_x^7 u)^*)\|_{L^\infty}. \tag{7.12}$$

The term (II) is

$$(II) = \frac{h^2}{6}\delta_x^2(\partial_x u)^* = \frac{h^2}{6}\left((\partial_x^3 u)^* + \frac{h^2}{12}(\partial_x^5 u)^* + \frac{h^4}{360}\mathfrak{w}\right), \tag{7.13}$$

where $\mathfrak{w}$ satisfies

$$\|\mathfrak{w}\|_{L^\infty} \leq C\|((\partial_x^7 u)^*)\|_{L^\infty}. \tag{7.14}$$

Therefore, the truncation error is

$$\mathfrak{t} = \left(I + \frac{h^2}{6}\delta_x^2\right)^{-1}\left(-\frac{h^4}{180}(\partial_x^5 u)^* + O(h^6)\right). \tag{7.15}$$

Using the fact that $-\delta_x^2 Z_k = \lambda_k Z_k$ and that $(Z_k)_{1\leq k\leq N-1}$ is an orthonormal basis of $\mathbb{R}^{N-1}$ gives the gridfunction component $\mathfrak{t}(u)_i$ is

$$\begin{aligned}\mathfrak{t}(u)_i &= -\frac{h^4}{180}\sum_{k=1}^{N-1}(1-\lambda_k/6)^{-1} Z_k^i \sum_{j=1}^{N-1} Z_k^j (\partial_x^5 u)(x_j) + O(h^6)\\ &= -\frac{h^4}{180}\sum_{j=1}^{N-1}\alpha_j^i (\partial_x^5 u)_j^* + O(h^6),\end{aligned}$$

where the coefficients α_j^i are given by (7.7). Therefore the leading term in the truncation error involves all the values $(\partial_x^5 u)_j^*$ as given in (7.6). □

References

1. Abarbanel, S., Ditkowski, A.: Asymptotically stable fourth-order accurate schemes for the diffusion equation on complex shapes. J. Comput. Phys. **133**, 279–288 (1997)
2. Ahlberg, J.H., Nilson, E.N., Walsh, J.L.: The Theory of Splines and Their Application. Academic Press, London (1967)
3. Ben-Artzi, M., Chorev, I., Croisille, J.-P., Fishelov, D.: A compact difference scheme for the biharmonic equation in planar irregular domains. SIAM J. Numer. Anal. **47**, 3087–3108 (2009)
4. Ben-Artzi, M., Croisille, J.-P., Fishelov, D.: Convergence of a compact scheme for the pure streamfunction formulation of the unsteady Navier–Stokes system. SIAM J. Numer. Anal. **44**, 1997–2024 (2006)
5. Ben-Artzi, M., Croisille, J.-P., Fishelov, D.: A high order compact scheme for the pure-streamfunction formulation of the Navier–Stokes Equations. J. Sci. Comput. **42**, 216–250 (2009)
6. Ben-Artzi, M., Croisille, J.-P., Fishelov, D.: Navier–Stokes Equations in Planar Domains. Imperial College Press, London (2013)
7. Ben-Artzi, M., Croisille, J.-P., Fishelov, D.: An embedded compact scheme for biharmonic problems in irregular domains. In: Georgiev, I., Georgiev, K., Todorov, M. (eds.) Advanced Computing in Industrial Mathematics, Series Studies in Computational Intelligence, pp. 11–23. Springer, Berlin (2018)
8. Ben-Artzi, M., Croisille, J.-P., Fishelov, D.: Time evolution of discrete fourth-order elliptic operators. Numer. Methods PDEs **35**, 1429 (2019)
9. Ben-Artzi, M., Croisille, J.-P., Fishelov, D., Katzir, R.: Discrete fourth-order Sturm–Liouville problems. IMA J. Numer. Anal. **38**(3), 1485–1522 (2018)

10. Ben-Artzi, M., Croisille, J.-P., Fishelov, D., Trachtenberg, S.: A pure-compact scheme for the stream-function formulation of Navier–Stokes equations. J. Comput. Phys. **205**, 640–664 (2005)
11. Ben-Artzi, M., Katriel, G.: Spline functions, the biharmonic operator and approximate eigenvalues. Numer. Math. **141**(4), 839–879 (2019)
12. Buzbee, B.L., Dorr, F.W.: The direct solution of the biharmonic equation on rectangular regions and the Poisson equation on irregular regions. SIAM J. Numer. Anal. **11**, 753–763 (1974)
13. Carpenter, M.H., Gottlieb, D., Abarbanel, S.: The stability of numerical boundary treatments for compact high-order finite difference schemes. J. Comput. Phys. **108**, 272–295 (1993)
14. Collatz, L.: The Numerical Treatment of Differential Equations, 3rd edn. Springer, Berlin (1960)
15. Ditkowski, A.: private communication
16. Ditkowski, A., Harness, Y.: High-order embedded finite difference schemes for initial boundary value problems on time dependent irregular domains. J. Sci. Comput. **39**, 394–440 (2009)
17. Fishelov, D.: A new fourth-order compact sheme for the Navier–Stokes equations in irregular geometries. Comput. Math. Appl. **74**, 6–25 (2017)
18. Fishelov, D., Ben-Artzi, M., Croisille, J.-P.: Recent advances in the study of a fourth-order compact scheme for the one-dimensional biharmonic equation. J. Sci. Comput. **53**(1), 55–79 (2012)
19. Gibou, F., Fedkiw, R.P., Cheng, L.-T., Kang, M.: A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. J. Comput. Phys. **176**, 205–227 (2002)
20. Gupta, M.M., Kalita, J.C.: A new paradigm for solving Navier–Stokes equations: streamfunction-velocity formulation. J. Comput. Phys. **207**, 52–68 (2005)
21. Hou, T.Y., Wetton, B.T.R.: Convergence of a finite difference scheme for the Navier–Stokes equations using vorticity boundary conditions stream function-only formulation. SIAM J. Numer. Anal. **29**, 615–639 (1992)
22. Jomaa, Z., Macaskill, C.: The Shortley–Weller embedded finite-difference method for the 3D Poisson equation with mixed boundary conditions. J. Comput. Phys. **229**, 3675–3690 (2010)
23. Kalita, J.C., Dass, A.K., Nidhi, N.: An efficient transient Navier–Stokes solver on compact nonuniform space grids. J. Comput. App. Math. **214**, 148–162 (2008)
24. Kopal, Z.: Numerical Analysis, 2nd edn. Wiley, New York (1961)
25. Kuntzmann, J.: Méthodes Numériques. Interpolation - Dérivées. Dunod (1959)
26. Kunz, K.: Numerical Analysis. McGrawHill, New York (1958)
27. Lele, S.K.: Compact finite-difference schemes with spectral-like resolution. J. Comput. Phys. **103**, 16–42 (1992)
28. Li, Z., Wang, C.: A fast finite difference method for solving Navier–Stokes equations in irregular domains. Commun. Math. Sci. **1**, 180–196 (2003)
29. Lorentz, R.A.: Multivariate Birkhoff interpolation. In: Lecture Notes in Mathematics, vol 1516. Springer, Berlin (1992)
30. Pandit, S.K.: On the use of compact streamfunction-velocity formulation of steady Navier–Stokes equations on geometries beyond rectangular. J. Sci. Comput. **36**, 219–242 (2008)
31. Pereyra, V., Proskurowski, W., Widlund, O.: High order fast Laplace solvers for the Dirichlet problem on general regions. Math. Comput. **31**(137), 1–16 (1977)
32. Powell, M.J.D.: Approximation Theory and Methods. Cambridge University Press, Cambridge (1981)
33. Prenter, P.M.: Splines and Variational Methods. Academic Press, London (1975)
34. Shortley, G.H., Weller, R.: The numerical solution of Laplace's equation. J. Appl. Phys. **9**, 334–348 (1938)
35. Stephenson, J.W.: Single cell discretizations of order two and four for biharmonic problems. J. Comput. Phys. **55**, 65–80 (1984)
36. Svärd, M., Nordström, J.: Review of summation-by-parts schemes for initial-boundary-value problems. J. Comput. Phys. **268**, 17–38 (2014)
37. Wang, S., Kreiss, G.: Convergence of summation-by-parts finite difference methods for the wave equation. J. Sci. Comput. **71**(1), 219–245 (2017)