A Fourth Order Hermitian Box-Scheme with Fast Solver for the Poisson Problem in a Square

Ali Abbas
Department of Mathematics, LMAM, UMR 7122
University Paul Verlaine-Metz
57012 Metz, France
email: ali.abbas@umail.univ-metz.fr

Jean-Pierre Croisille
Department of Mathematics, LMAM, UMR 7122
University Paul Verlaine-Metz
57012 Metz, France
email: croisil@poncelet.univ-metz.fr

December 12, 2010

Abstract

A new fourth order box-scheme for the Poisson problem in a square with Dirichlet boundary conditions is introduced, extending the approach in [17]. The design is based on a "hermitian box" approach, combining the approximation of the gradient by the fourth order hermitian derivative, with a conservative discrete formulation on boxes of length $2h$. The goal is twofold: first to show that fourth order accuracy is obtained both for the unknown and its gradient; second, to describe a fast direct algorithm, based on the Sherman-Morrison formula and the Fast Sine Transform. Several numerical results in a square are given, indicating an asymptotic $O(N^2 \log_2(N))$ computing complexity.

MSC Subject Classification: 35J25 - 65M15 - 65N30 - 76M12 - 76M20
Key words: Hermitian Scheme - Box-scheme - Finite Volume Method - High Order Compact Scheme - Fast Solver - FFT - Mehrstellenverfahren - Poisson Problem - Sherman-Morrison formula

1 Introduction

The design of high order compact finite-difference schemes for the Laplace equation in a squared or cubic geometry is a classical topic in Applied Mathematics and Scientific Computing. The books [15, 20, 33, 35, 26, 29, 25] address this question as a fundamental educative piece on numerical methods. We refer to [13, 14] for examples of the numerous articles published in the 1960's on the subject. For works in the last decade, we refer to [16, 36, 37, 40, 11]. Beyond the design of specific numerical schemes, which deals with accuracy and stability, the need of an efficient fast solver is a crucial issue to perform practical computations. On this question we refer to the recent review [5] and the references therein. The use of such solvers in canonical geometries persists to be at the heart of many computing codes in physics. Examples are fluid dymanics (compressible or incompressible Navier-Stokes equations), [34, 21, 3], the Helmholtz equation [9, 12], computations in astrophysics, [32] or in geophysics, [38].

Here, we introduce a new fourth order compact scheme on a cartesian grid for the Poisson problem in a rectangle, whose design is based on the preliminary work [17]. The scheme, referred as *Hermitian Box-Scheme* in the sequel, combines a finite volume "box" approach with an hermitian computation of the derivative. It provides by construction an approximation to u and ∇u. In addition, only the averages of the data $f(x, y)$ are used instead of the pointwise values $f(x_i, y_j)$ as in most finite-difference schemes. This is an interesting feature when approximating Poisson problems in electromagnetism.

The main part of the paper is devoted to a fast resolution procedure which uses the decomposition of the matrix of the scheme is the sum of a diagonal matrix (in a spectral basis) and of a low-rank auxiliary matrix. This runs along the lines of Golub (see Appendix in [19]). The practical resolution is performed by a direct resolution algorithm using the Sherman-Morrison formula and the FFT, (see [7], or [2] in a different context).

The outline of the paper is as follows. In Section 2, we briefly recall the general principle of design of the Hermitian Box-Scheme in dimension one. Two specific examples are given. First, the second order version of the scheme is recalled (called *HB-Scheme 1*). Then a new fourth order scheme is introduced, (*HB-Scheme 2*). In Section 3 we expand on the building block of the fast solver, namely the diagonal form of the HB-Scheme 1. The matrix form of the HB-Scheme 2 appears as a perturbation of the HB-Scheme 1. In Section 4, we describe the design of HB-Schemes in a square as well as the fast solver principle for the two versions of the HB-Scheme. A short comparison with other fourth order schemes and an estimate of the computing cost is given. Numerical results given in Section 5 indicate a fourth order accuracy of the new scheme and an an asymptotic $O(N^2 \log_2(N))$ computing complexity. In section 6, we briefly outline future works, insisting on the general character of this kind of schemes.

2 One-dimensional Hermitian Box-Scheme

In this section, we recall briefly the principle of construction of the Hermitian Box-Scheme (*HB-Scheme*) in dimension one, [17]. Let us consider the one-dimensional Poisson problem on the interval $\Omega = (a, b)$ with length $L = b - a$,

$$(2.1) \qquad \begin{cases} -u''(x) = f(x), \quad a < x < b, \\ u(a) = g_a, \ u(b) = g_b. \end{cases}$$

Equation (2.1) is recast in mixed form:

$$(2.2) \qquad \begin{cases} v'(x) + f(x) = 0, & (a)\ , \\ v(x) - u'(x) = 0, & (b)\ , \\ u(a) = g_a, \ u(b) = g_b, & (c). \end{cases}$$

We lay out on Ω a regular grid $x_j = a + jh$, $0 \le j \le N$ with stepsize $h = L/N$. The unknowns are denoted by $u_j \simeq u(x_j)$ and $u_{x,j} \simeq u'(x_j)$. The vectors $\tilde{U}, \tilde{U}_x \in \mathbb{R}^{N+1}$ (including boundary points) are

$$(2.3) \qquad \widetilde{U} = [u_0, u_1, u_2, ..., u_N]^T; \quad \widetilde{U}_x = [u_{x,0}, u_{x,1}, ..., u_{x,N}]^T.$$

The vectors $U, U_x \in \mathbb{R}^{N-1}$ stand for the unknowns at internal points,

$$(2.4) \qquad U = [u_1, u_2, ..., u_{N-1}]^T; \quad U_x = [u_{x,1}, ..., u_{x,N-1}]^T.$$

As in box-schemes, [30], the HB-Scheme is derived from the integration of the two equations $(2.2)_{a,b}$ on a box $K_j =]x_{j-1}, x_{j+1}[$ of length $2h$.
Suppose given the averaged values of the source term $f(x)$ on the box K_j,

$$(2.5) \qquad \Pi^0 f_j = \frac{1}{2h} \int_{K_j} f(x)dx, \quad 1 \le j \le N-1.$$

The conservation equation $(2.2)_a$ is approximated by

$$(2.6) \qquad -\frac{u_{x,j+1} - u_{x,j-1}}{2h} = \Pi^0 f_j, \quad 1 \le j \le N-1.$$

Second, the equation $(2.2)_b$ is integrated on the box K_j. This yields

$$\frac{1}{2h}\int_{x_{j-1}}^{x_{j+1}} v(x)dx = \frac{u(x_{j+1}) - u(x_{j-1})}{2h}, \quad 1 \le j \le N-1. \tag{2.7}$$

Approximating the integral in the left-hand side of (2.7) by the Simpson formula suggests the fourth-order Hermitian approximation ,

$$\frac{1}{6}u_{x,j-1} + \frac{2}{3}u_{x,j} + \frac{1}{6}u_{x,j+1} = \frac{u_{j+1} - u_{j-1}}{2h}, \ 1 \le j \le N-1. \tag{2.8}$$

To fully define the HB-Scheme, we have further to specify four boundary conditions:

- The Dirichlet boundary conditions $u(a) = g_a$, $u(b) = g_b$ simply translate to

$$u_0 = g_a, \ u_N = g_b. \tag{2.9}$$

- The Hermitian relations (2.8) at points $j = 1$ and $j = N-1$ require to approximate the values $u_{x,0}, u_{x,N}$ in terms of U, U_x, u_0, u_N.

Example 1: A first example of this approximation is obtained by approximating the derivatives $u_{x,0}, u_{x,N}$ in terms of the internal unknowns by the two relations

$$\begin{cases} u_{x,0} = \dfrac{3}{2}\left(\dfrac{u_1 - u_0}{h} - \dfrac{1}{3}u_{x,1}\right), \\ u_{x,N} = \dfrac{3}{2}\left(\dfrac{u_N - u_{N-1}}{h} - \dfrac{1}{3}u_{x,N-1}\right). \end{cases} \tag{2.10}$$

Formula (2.10) is considered in [17]. It coincides with the Simpson formula (2.8) in the case of an odd function $u(x)$ at the boundary.

Equations (2.6), (2.8), (2.9), (2.10) constitute the *HB-Scheme 1*. It is compactly written as: find $\tilde{U}, \tilde{U}_x \in \mathbb{R}^{N+1}$ solution of

$$\begin{cases} -\dfrac{u_{x,j+1} - u_{x,j-1}}{2h} = \Pi^0 f_j, \quad 1 \le j \le N-1, \\ \dfrac{2}{3}u_{x,0} + \dfrac{1}{3}u_{x,1} = \dfrac{u_1 - u_0}{h}, \\ \dfrac{1}{6}u_{x,j-1} + \dfrac{2}{3}u_{x,j} + \dfrac{1}{6}u_{x,j+1} = \dfrac{u_{j+1} - u_{j-1}}{2h}, \ \ 1 \le j \le N-1, \\ \dfrac{2}{3}u_{x,N} + \dfrac{1}{3}u_{x,N-1} = \dfrac{u_N - u_{N-1}}{h}, \\ u_0 = g_a, \ \ u_N = g_b. \end{cases} \tag{2.11}$$

Example 2: A second approximation of $u_{x,0}, u_{x,N}$ in terms of U, U_x, u_0, u_N is given by

$$\begin{cases} \dfrac{1}{3}u_{x,0} + \dfrac{2}{3}u_{x,1} = \dfrac{1}{h}\left(\dfrac{1}{6}u_2 + \dfrac{2}{3}u_1 - \dfrac{5}{6}u_0\right), \quad (a), \\ \dfrac{1}{3}u_{x,N} + \dfrac{2}{3}u_{x,N-1} = \dfrac{1}{h}\left(\dfrac{5}{6}u_N - \dfrac{2}{3}u_{N-1} - \dfrac{1}{6}u_{N-2}\right), \quad (b). \end{cases} \tag{2.12}$$

Relation $(2.12)_a$ is obtained in the following way. Let us consider *a priori* a relation of the form

$$\alpha u_{x,0} + (1-\alpha) u_{x,1} = \beta \frac{u_1 - u_0}{h} + \gamma \frac{u_2 - u_0}{2h}, \tag{2.13}$$

where the parameters α, β, γ satisfy the highest possible consistency order at $x_0 = 0$. It is easily verified by Taylor expansions that this maximum order is three and is achieved by the values

$$\alpha = \frac{1}{3}, \;\; \beta = \frac{2}{3}, \;\; \gamma = \frac{1}{3}. \tag{2.14}$$

An analogous treatment is performed for $(2.12)_b$. Using the boundary conditions $(2.12)_{ab}$ we obtain the *HB-Scheme 2*: find $\tilde{U} = [u_0, u_1, \cdots, u_{N-1}, u_N]^T$, $\tilde{U}_x = [u_{x,0}, \cdots, u_{x,N}]^T \in \mathbb{R}^{N+1}$ such that

$$\begin{cases} -\dfrac{u_{x,j+1} - u_{x,j-1}}{2h} = \Pi^0 f_j, \quad 1 \le j \le N-1, \\ \dfrac{1}{3} u_{x,0} + \dfrac{2}{3} u_{x,1} = \dfrac{1}{h}\left(\dfrac{1}{6} u_2 + \dfrac{2}{3} u_1 - \dfrac{5}{6} u_0\right), \\ \dfrac{1}{6} u_{x,j-1} + \dfrac{2}{3} u_{x,j} + \dfrac{1}{6} u_{x,j+1} = \dfrac{u_{j+1} - u_{j-1}}{2h}, \;\; 1 \le j \le N-1, \\ \dfrac{1}{3} u_{x,N} + \dfrac{2}{3} u_{x,N-1} = \dfrac{1}{h}\left(\dfrac{5}{6} u_N - \dfrac{2}{3} u_{N-1} - \dfrac{1}{6} u_{N-2}\right), \\ u_0 = g_a, \;\; u_N = g_b. \end{cases} \tag{2.15}$$

Note that specifying the approximation of $u_{x,0}, u_{x,N}$ in terms of U, U_x can be interpretated as specifying a discrete *Dirichlet-to-Neumann operator*

$$(f, g) \mapsto (u'(x_0), u'(x_N)). \tag{2.16}$$

HB-Schemes 1 and 2 only differ by the definition of the discrete version of (2.16). The HB-Scheme 1 is observed to be second order accurate in general, whereas the HB-Scheme 2 is expected to be fourth order accurate. We refer to [1] for a study of the truncation error analysis of both schemes.

3 Matrix form of the one-dimensional Hermitian Box-Scheme

3.1 Finite-Difference and Matrix Notation

In this section, we summarize the finite-difference and matrix notation used in the sequel. We call l_h^2 the space of grid functions $(u_i)_{0 \le i \le N}$. The subspace of grid functions $(u_i)_{0 \le i \le N}$ with $u_0 = u_N = 0$ is called $l_{h,0}^2$. We review in the sequel several finite-difference operators acting on grid functions.

• *Centered One-dimensional Laplacian*

The one-dimenional three-point Laplacian is

$$\delta_x^2 u_i = \frac{u_{i+1} + u_{i-1} - 2u_i}{h^2}, \;\; 1 \le i \le N-1. \tag{3.1}$$

The $N-1$ eigen grid functions of the operator $-\delta_x^2$ are the $z^k \in l_{h,0}^2$ defined by

$$z_j^k = \left(\frac{2}{L}\right)^{\frac{1}{2}} \sin\left(\frac{k\pi jh}{L}\right), \quad 1 \le j, k \le N-1. \tag{3.2}$$

The grid functions z^k form an orthonormal basis of $l_{h,0}^2$ for the scalar product $(u,v)_h$. The matrix matching δ_x^2 is $-T/h^2$ where the matrix $T \in \mathbb{M}_{N-1}(\mathbb{R})$ is

$$T = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}. \tag{3.3}$$

The dimensionless eigenvectors $Z^k \in \mathbb{R}^{N-1}$ of T are related to z^k by $Z_j^k = \sqrt{h} z_j^k$,

$$Z_j^k = \left(\frac{2}{N}\right)^{1/2} \sin\left(\frac{kj\pi}{N}\right). \tag{3.4}$$

Defining the matrix $Z \in \mathbb{M}_{N-1}$ by$Z = [Z^1, \cdots, Z^{N-1}]$, we have the relations

$$T = Z\Lambda Z, \quad Z^T = Z, \quad I_{N-1} = Z^2 = ZZ^T, \tag{3.5}$$

where $\Lambda = \text{diag}(\lambda_1, ..., \lambda_{N-1})$ is the diagonal matrix of the eigenvalues of T given by

$$\lambda_k = 4\sin^2\left(\frac{k\pi}{2N}\right), \quad 1 \le k \le N-1. \tag{3.6}$$

• *Simpson operator*
The *Simpson finite-difference operator* σ_x is

$$\sigma_x u_j = \frac{1}{6} u_{j-1} + \frac{2}{3} u_j + \frac{1}{6} u_{j+1}, \quad 1 \le j \le N-1. \tag{3.7}$$

Its matching matrix is

$$P_s = I - T/6 = Z(I - \Lambda/6)Z^T. \tag{3.8}$$

• *Centered One-dimensional difference operator*
The centered operator δ_x is

$$\delta_x u_j = \frac{u_{j+1} - u_{j-1}}{2h}, \quad 1 \le i \le N-1. \tag{3.9}$$

The matching matrix is the antisymmetric matrix $K \in \mathbb{M}_{N-1}(\mathbb{R})$ given by

$$K = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & -1 & 0 & 1 \\ 0 & \dots & 0 & -1 & 0 \end{bmatrix}. \tag{3.10}$$

• Denoting $(e_i)_{1\leq i\leq N-1}$ the canonical basis of $\mathbb{R}^{N-1}$, the matrices $F_1, F_2 \in \mathbb{M}_{N-1}(\mathbb{R})$ are defined by

$$(3.11)\qquad \begin{cases} F_1 = e_1e_1^T + e_{N-1}e_{N-1}^T = [e_1, e_{N-1}]\begin{bmatrix} e_1^T \\ e_{N-1}^T \end{bmatrix}, \ (a), \\ F_2 = -e_1e_1^T + e_{N-1}e_{N-1}^T = [-e_1, e_{N-1}]\begin{bmatrix} e_1^T \\ e_{N-1}^T \end{bmatrix}, \ (b). \end{cases}$$

The two following relations hold

$$(3.12)\qquad \begin{cases} F_2.F_1 = F_2, & (a), \\ F_2.F_2 = F_1, & (b). \end{cases}$$

Using the preceding finite-difference operators, the two equations (2.6) and (2.8), common to HB-Schemes 1 and 2, translate to

$$(3.13)\qquad \begin{cases} -\delta_x u_{x,j} = \Pi^0 f_j,\ 1\leq j\leq N-1,\ (a), \\ \sigma_x u_{x,j} = \delta_x u_j,\ 1\leq j\leq N-1,\ (b). \end{cases}$$

Let us turn now to the generic matrix form of any of the boundary conditions (2.10) or (2.12). Using the notation for the boundary values

$$(3.14)\qquad U_L = u_0,\ U_R = u_N,\ U_{x,L} = u_{x,0},\ U_{x,R} = u_{x,N},$$

we claim that (2.10) or (2.12) can be expressed as

$$(3.15)\qquad e_1U_{x,L} + e_{N-1}U_{x,R} = \frac{1}{h}\Big(\mathcal{A}U - h\mathcal{B}U_x + \mathcal{C}\big(e_1U_L + e_{N-1}U_R\big)\Big).$$

Indeed, (2.10) can be written as

(3.16)

$$\begin{cases} u_{x,0} = \frac{1}{h}\big(\frac{3}{2}e_1\big)^TU - \big(\frac{1}{2}e_1\big)^TU_x - \frac{1}{h}\big(\frac{3}{2}e_1\big)^T[u_0,0,..,0,u_N]^T, \\ u_{x,N} = \frac{1}{h}\big(-\frac{3}{2}e_{N-1}\big)^TU - \big(\frac{1}{2}e_{N-1}\big)^TU_x + \frac{1}{h}\big(\frac{3}{2}e_{N-1}\big)^T[u_0,0,..,0,u_N]^T. \end{cases}$$

This translates into (3.15) with the matrices $\mathcal{A} = \mathcal{A}_1$, $\mathcal{B} = \mathcal{B}_1$, $\mathcal{C} = \mathcal{C}_1$ defined by

$$(3.17)\qquad \begin{cases} \mathcal{A}_1 = \frac{3}{2}(e_1e_1^T - e_{N-1}e_{N-1}^T) = -\frac{3}{2}F_2, & (a), \\ \mathcal{B}_1 = \frac{1}{2}(e_1e_1^T + e_{N-1}e_{N-1}^T) = \frac{1}{2}F_1, & (b), \\ \mathcal{C}_1 = \frac{3}{2}(-e_1e_1^T + e_{N-1}e_{N-1}^T) = \frac{3}{2}F_2, & (c). \end{cases}$$

Similarly (2.12) can be written as

(3.18)

$$\begin{cases} u_{x,0} = \frac{1}{h}(2e_1 + \frac{1}{2}e_2)^TU - (2e_1)^TU_x - \frac{1}{h}\left(\frac{5}{2}e_1\right)^T[u_0,0,\cdots,0,u_N]^T, \\ u_{x,N} = -\frac{1}{h}(2e_{N-1} + \frac{1}{2}e_{N-2})^TU - (2e_{N-1})^TU_x + \frac{1}{h}\left(\frac{5}{2}e_{N-1}\right)^T[u_0,0,\cdots,0,u_N]^T. \end{cases}$$

This translates into matrix form as (3.15) with the matrices $\mathcal{A}=\mathcal{A}_2$, $\mathcal{B}=\mathcal{B}_2$, $\mathcal{C}=\mathcal{C}_2$ given by
(3.19)

$$\begin{cases} \mathcal{A}_2 = 2e_1e_1^T + \dfrac{1}{2}e_1e_2^T - 2e_{N-1}e_{N-1}^T - \dfrac{1}{2}e_{N-1}e_{N-2}^T = -2F_2 + \dfrac{1}{2}(e_1e_2^T - e_{N-1}e_{N-2}^T), \quad (a), \\ \mathcal{B}_2 = 2(e_1e_1^T + e_{N-1}e_{N-1}^T) = 2F_1 \quad (b), \\ \mathcal{C}_2 = \dfrac{5}{2}(-e_1e_1^T + e_{N-1}e_{N-1}^T) = \dfrac{5}{2}F_2 \quad (c). \end{cases}$$

Let us consider now the matrix form of the equations in (3.13).
First, $(3.13)_b$ translates to

$$P_sU_x + \frac{1}{6}\big(e_1U_{x,L} + e_{N-1}U_{x,R}\big) = \frac{1}{2h}KU + \frac{1}{2h}F_2(e_1U_L + e_{N-1}U_R). \tag{3.20}$$

It follows from (3.15), (3.20) that

$$U_x = \frac{1}{h}\mathcal{D}U + \frac{1}{h}\mathcal{E}(e_1U_L + e_{N-1}U_R), \tag{3.21}$$

$\mathcal{D}=\mathcal{D}(\mathcal{A},\mathcal{B})$ and $\mathcal{E}=\mathcal{E}(\mathcal{B},\mathcal{C})$ are

$$\begin{cases} \mathcal{D} = \dfrac{1}{2}(P_s - \dfrac{1}{6}\mathcal{B})^{-1}(K - \dfrac{1}{3}\mathcal{A}), \\ \mathcal{E} = \dfrac{1}{2}(P_s - \dfrac{1}{6}\mathcal{B})^{-1}(F_2 - \dfrac{1}{3}\mathcal{C}). \end{cases} \tag{3.22}$$

Second, equation $(3.13)_a$ translates to

$$-\frac{1}{2h}KU_x - \frac{1}{2h}F_2(e_1U_{x,L} + e_{N-1}U_{x,R}) = F. \tag{3.23}$$

where $F = [\Pi^0 f_1, \cdots, \Pi^0 f_{N-1}]^T$. Using (3.15) and (3.21) U_x is eliminated which gives the expression of the HB-Scheme in the sole unknown U as

$$\frac{1}{h^2}\mathcal{H}U = F - \frac{1}{h^2}\mathcal{G}\big(e_1U_L + e_{N-1}U_R\big). \tag{3.24}$$

The matrices $\mathcal{H},\mathcal{G}$ are

$$\begin{cases} \mathcal{H} = -\dfrac{1}{2}\Big(K\mathcal{D} + F_2(\mathcal{A} - \mathcal{B}\mathcal{D})\Big), \\ \mathcal{G} = -\dfrac{1}{2}\Big(K\mathcal{E} + F_2(\mathcal{C} - \mathcal{B}\mathcal{E})\Big), \end{cases} \tag{3.25}$$

or equivalently with (3.22),

$$\begin{cases} \mathcal{H} = -\dfrac{1}{4}(K - F_2\mathcal{B})(P_s - \dfrac{1}{6}\mathcal{B})^{-1}(K - \dfrac{1}{3}\mathcal{A}) - \dfrac{1}{2}F_2\mathcal{A}, \quad (a), \\ \mathcal{G} = -\dfrac{1}{4}(K - F_2\mathcal{B})(P_s - \dfrac{1}{6}\mathcal{B})^{-1}(F_2 - \dfrac{1}{3}\mathcal{C}) - \dfrac{1}{2}F_2\mathcal{C}, \quad (b). \end{cases} \tag{3.26}$$

If needed, the gradient is recovered as a postprocessing by (3.21).

3.2 Diagonal form of the HB-Scheme 1

According to the analysis in the previous section, the matrix form of the HB-Scheme 1 is

$$\frac{1}{h^2}\mathcal{H}_1 U = F - \frac{1}{h^2}\mathcal{G}_1\big(e_1 U_L + e_{N-1}U_R\big). \tag{3.27}$$

The matrices $\mathcal{H}_1, \mathcal{G}_1$ are

$$\begin{cases} \mathcal{H}_1 = -\frac{1}{4}(K - F_2\mathcal{B}_1)(P_s - \frac{1}{6}\mathcal{B}_1)^{-1}(K - \frac{1}{3}\mathcal{A}_1) - \frac{1}{2}F_2\mathcal{A}_1, \quad (a), \\ \mathcal{G}_1 = -\frac{1}{4}(K - F_2\mathcal{B}_1)(P_s - \frac{1}{6}\mathcal{B}_1)^{-1}(F_2 - \frac{1}{3}\mathcal{C}_1) - \frac{1}{2}F_2\mathcal{C}_1, \quad (b). \end{cases} \tag{3.28}$$

We prove one of the essential result of this paper, namely that the resulting matrix $\mathcal{H}_1$ *is diagonal in the spectral basis* Z^k *of the three-point Laplacian.* This is the basis of our fast solver in Section 4.
According to (3.28), and using that $F_2\mathcal{A}_1 = -3\mathcal{B}_1, \;\; F_2\mathcal{B}_1 = -\frac{1}{3}\mathcal{A}_1$, the matrix $\mathcal{H}_1$ can be rewritten as

$$\mathcal{H}_1 = -\frac{1}{4}(K + \frac{1}{3}\mathcal{A}_1)(P_s - \frac{1}{6}\mathcal{B}_1)^{-1}(K - \frac{1}{3}\mathcal{A}_1) + \frac{3}{2}\mathcal{B}_1. \tag{3.29}$$

The matrix $\mathcal{H}_1$ is symmetric, since P_s and $\mathcal{A}_1, \mathcal{B}_1$ are symmetric, (see (3.8), (3.17)) and K is antisymmetric.

Proposition 3.1 *The $(N-1)\times(N-1)$ matrix $\mathcal{H}_1$ in (3.29) is expressed in terms of the matrix T as*

$$\mathcal{H}_1 = P_s^{-1}(T - \frac{1}{4}T^2), \tag{3.30}$$

where $P_s = I - T/6$.

Remark: In other words, $\mathcal{H}_1$ is diagonal in the spectral basis Z^k. It can be expressed as

$$\mathcal{H}_1 = Z\mathcal{M}Z^T, \tag{3.31}$$

where $\mathcal{M}$ is the diagonal matrix

$$\mathcal{M} = (I - \Lambda/6)^{-1}(\Lambda - \frac{1}{4}\Lambda^2) = \mathrm{diag}(\mu_1, \cdots, \mu_{N-1}). \tag{3.32}$$

The eigenvalues μ_k of $\mathcal{H}_1$ are $\mu_k = (1 - \lambda_k/6)^{-1}(\lambda_k - \lambda_k^2/4)$ or equivalently

$$\mu_k = \frac{\sin^2(\frac{k\pi}{N})}{\frac{2}{3} + \frac{1}{3}\cos(\frac{k\pi}{N})}, \quad 1 \le k \le N-1. \tag{3.33}$$

Proof of Prop.3.1: Consider the grid function z^k in (3.2). It is verified by a direct computation that z^k satisfies the equations

$$
\left\{
\begin{array}{l}
-\dfrac{z^k_{x,j+1} - z^k_{x,j-1}}{2h} = \dfrac{1}{h^2} \dfrac{\sin^2(\frac{k\pi h}{L})}{\frac{2}{3} + \frac{1}{3}\cos\frac{k\pi h}{L}} z^k_j, \quad 1 \le j \le N-1, \\
\dfrac{2}{3} z^k_{x,0} + \dfrac{1}{3} z^k_{x,1} = \dfrac{z^k_1 - z^k_0}{h}, \\
\dfrac{1}{6} z^k_{x,j-1} + \dfrac{2}{3} z^k_{x,j} + \dfrac{1}{6} z^k_{x,j+1} = \dfrac{z^k_{j+1} - z^k_{j-1}}{2h}, \quad 1 \le j \le N-1, \\
\dfrac{2}{3} z^k_{x,N} + \dfrac{1}{3} z^k_{x,N-1} = \dfrac{z^k_N - z^k_{N-1}}{h}, \\
z^k_0 = 0, \quad z^k_N = 0.
\end{array}
\right.
\tag{3.34}
$$

The grid function z^k_x is found to be

$$
z^k_{x,j} = \frac{1}{h}\left(\frac{2}{L}\right)^{\frac{1}{2}} \frac{\sin(\frac{k\pi h}{L})}{\frac{2}{3} + \frac{1}{3}\cos(\frac{k\pi h}{L})} \cos\left(\frac{k\pi j h}{L}\right). \tag{3.35}
$$

Comparing (2.11) and (3.34) and using the definition of the matrix $\mathcal{H}_1$ in (3.24) as the matrix of the HB-Scheme 1, it turns out that (3.34) translates to

$$
\mathcal{H}_1 Z^k = \mu_k Z^k, \tag{3.36}
$$

where μ_k is given by

$$
\mu_k = \frac{\sin^2(\frac{k\pi}{N})}{\frac{2}{3} + \frac{1}{3}\cos(\frac{k\pi}{N})}, \quad 1 \le k \le N-1. \tag{3.37}
$$

Using the value of λ_k in (3.6), it is verified that

$$
\frac{\lambda_k - \frac{1}{4}\lambda_k^2}{1 - \frac{1}{6}\lambda_k} = \frac{\sin^2(\frac{k\pi}{N})}{\frac{2}{3} + \frac{1}{3}\cos(\frac{k\pi}{N})} \triangleq \mu_k. \tag{3.38}
$$

Consequently, the vectors Z^k form a complete set of eigenvectors of $\mathcal{H}_1$. Moreover, the matrix $\mathcal{H}_1$ can be expressed in terms of T as

$$
\mathcal{H}_1 = P_s^{-1}(T - \frac{1}{4}T^2). \tag{3.39}
$$

■

3.3 Matrix form of the HB-Scheme 2

We derive in this section the matrix $\mathcal{H}_2$ of the HB-Scheme 2 in (2.15). According to (3.26), the matrices $\mathcal{H}_2, \mathcal{G}_2$ are (we use $\mathcal{A} = \mathcal{A}_2$, $\mathcal{B} = \mathcal{B}_2$, $\mathcal{C} = \mathcal{C}_2$, with $\mathcal{A}_2$, $\mathcal{B}_2$, $\mathcal{C}_2$ given in (3.19)),

$$
\left\{
\begin{array}{l}
\mathcal{H}_2 = -\dfrac{1}{4}(K - F_2\mathcal{B}_2)(P_s - \dfrac{1}{6}\mathcal{B}_2)^{-1}(K - \dfrac{1}{3}\mathcal{A}_2) - \dfrac{1}{2}F_2\mathcal{A}_2, \quad (a), \\
\mathcal{G}_2 = -\dfrac{1}{4}(K - F_2\mathcal{B}_2)(P_S - \dfrac{1}{6}\mathcal{B}_2)^{-1}(F_2 - \dfrac{1}{3}\mathcal{C}_2) - \dfrac{1}{2}F_2\mathcal{C}_2, \quad (b).
\end{array}
\right.
\tag{3.40}
$$

The matrices P_s, K, F_2 are given in (3.8), (3.10), $(3.11)_b$. The HB-Scheme 2 reads, (see (3.24)),

$$\frac{1}{h^2}\mathcal{H}_2 U = \Pi^0 f - \frac{1}{h^2}\mathcal{G}_2\big(e_1 U_L + e_{N-1}U_R\big). \tag{3.41}$$

In the sequel, we show that $\mathcal{H}_2$ can be expressed as

$$\mathcal{H}_2 = \mathcal{H}_1 + RS^T, \quad R, S \in \mathbb{M}_{N-1,p}(\mathbb{R}),\ p \ll N, \tag{3.42}$$

where $\delta\mathcal{H} = RS^T$ is a low rank perturbation of $\mathcal{H}_1$. The perturbation $\delta\mathcal{H}$ is only due to the fourth order boundary conditions. Later on, we apply to (3.42) the Sherman-Morrison formula, ([22], Chap.2, p. 50) to express $\mathcal{H}_2^{-1}$ in terms of $\mathcal{H}_1^{-1}$ by

$$\mathcal{H}_2^{-1} = \mathcal{H}_1^{-1} - \mathcal{H}_1^{-1}R(I_p + S^T\mathcal{H}_1^{-1}R)^{-1}S^T\mathcal{H}_1^{-1}. \tag{3.43}$$

First, observe that the matrices $\mathcal{A}_2$, $\mathcal{B}_2$ can be expressed in terms of $\mathcal{A}_1$, $\mathcal{B}_1$ as

$$\begin{cases} \mathcal{A}_2 = \mathcal{A}_1 - \frac{1}{2}F_2 + \frac{1}{2}(e_1e_2^T - e_{N-1}e_{N-2}^T), \quad (a), \\ \mathcal{B}_2 = \mathcal{B}_1 + \frac{3}{2}(e_1e_1^T + e_{N-1}e_{N-1}^T) = \mathcal{B}_1 + \frac{3}{2}F_1, \quad (b). \end{cases} \tag{3.44}$$

This yields using (3.12),

$$\begin{cases} F_2\mathcal{A}_2 = F_2\mathcal{A}_1 - \frac{1}{2}F_1 - \frac{1}{2}(e_1e_2^T + e_{N-1}e_{N-2}^T), \\ F_2\mathcal{B}_2 = F_2\mathcal{B}_1 + \frac{3}{2}F_2. \end{cases} \tag{3.45}$$

Therefore the matrix $\mathcal{H}_2$ in $(3.40)_a$ can be expressed as

$$\mathcal{H}_2 = \mathcal{H}_2^{(a)} + \mathcal{H}_2^{(b)}. \tag{3.46}$$

with

(3.47)

$$\begin{cases} \mathcal{H}_2^{(a)} = -\frac{1}{4}(K - F_2\mathcal{B}_1)(P_s - \frac{1}{6}\mathcal{B}_2)^{-1}(K - \frac{1}{3}\mathcal{A}_1) + \frac{3}{8}F_2(P_s - \frac{1}{6}\mathcal{B}_2)^{-1}(K - \frac{1}{3}\mathcal{A}_1) - \frac{1}{2}F_2\mathcal{A}_1, \\ \mathcal{H}_2^{(b)} = \frac{1}{4}F_1 + \frac{1}{4}(e_1e_2^T + e_{N-1}e_{N-2}^T) - \frac{1}{24}(K - F_2\mathcal{B}_2)(P_s - \frac{1}{6}\mathcal{B}_2)^{-1}(F_2 - e_1e_2^T + e_{N-1}e_{N-2}^T). \end{cases}$$

Furthermore

$$\mathcal{B}_2 = \mathcal{B}_1 + \frac{3}{2}[e_1, e_{N-1}]\begin{bmatrix} e_1^T \\ e_{N-1}^T \end{bmatrix}. \tag{3.48}$$

By the Sherman-Morrison formula, $(P_s - \frac{1}{6}\mathcal{B}_2)^{-1}$ is expressed in terms of $(P_s - \frac{1}{6}\mathcal{B}_1)^{-1}$ by

$$\begin{aligned} (P_s - \frac{1}{6}\mathcal{B}_2)^{-1} &= \left(P_s - \frac{1}{6}\mathcal{B}_1 - \frac{1}{4}[e_1 e_{N-1}]\begin{bmatrix} e_1^T \\ e_{N-1}^T \end{bmatrix}\right)^{-1} \\ &= (P_s - \frac{1}{6}\mathcal{B}_1)^{-1} + \begin{bmatrix} v_1, v_2 \end{bmatrix}\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}. \end{aligned} \tag{3.49}$$

The two vectors v_1, v_2 are

$$(3.50)\qquad \begin{cases} v_1 = \dfrac{1}{2(\alpha+\beta)^{\frac{1}{2}}}(P_s - \dfrac{1}{6}\mathcal{B}_1)^{-1}\left(\dfrac{\sqrt{2}}{2}e_1 - \dfrac{\sqrt{2}}{2}e_{N-1}\right), \\ v_2 = \dfrac{1}{2(\alpha-\beta)^{\frac{1}{2}}}(P_s - \dfrac{1}{6}\mathcal{B}_1)^{-1}\left(\dfrac{\sqrt{2}}{2}e_1 + \dfrac{\sqrt{2}}{2}e_{N-1}\right), \end{cases}$$

where the scalars α, β are

$$(3.51)\qquad \begin{cases} \alpha = 1 - \dfrac{1}{4}e_1^T(P_s - \dfrac{1}{6}\mathcal{B}_1)^{-1}e_1, \\ \beta = \dfrac{1}{4}e_{N-1}^T(P_s - \dfrac{1}{6}\mathcal{B}_1)^{-1}e_1. \end{cases}$$

Replacing $(P_s - \frac{1}{6}\mathcal{B}_2)^{-1}$ by its value given in (3.49), we obtain after some algebra,

$$(3.52)\qquad \mathcal{H}_2^{(a)} = \mathcal{H}_1 + w_1 w_1^T + w_2 w_2^T + e_1 s_1^T + e_{N-1}s_2^T,$$

and

$$(3.53)\qquad \mathcal{H}_2^{(b)} = r_1(e_1+e_2)^T + r_2(e_{N-1}+e_{N-2})^T.$$

We deduce the following decomposition

$$(3.54)\qquad \mathcal{H}_2 = \mathcal{H}_1 + RS^T,$$

where RS^T is the perturbation defined by

$$(3.55)\qquad \begin{cases} R = [w_1; w_2; e_1; e_{N-1}; r_1; r_2] \in \mathbb{M}_{N-1,6}(\mathbb{R}), \\ S = [w_1; w_2; s_1; s_2; e_1+e_2; e_{N-1}+e_{N-2}] \in \mathbb{M}_{N-1,6}(\mathbb{R}). \end{cases}$$

The six vectors $w_1, w_2, r_1, r_2, s_1, s_2 \in \mathbb{R}^{(N-1)}$ are

$$(3.56)\qquad \begin{cases} w_1 = -\dfrac{1}{2}(K + \dfrac{1}{3}\mathcal{A}_1)v_1, \\ w_2 = -\dfrac{1}{2}(K + \dfrac{1}{3}\mathcal{A}_1)v_2, \end{cases}$$

$$(3.57)\qquad \begin{cases} r_1 = -\dfrac{1}{24}(-K + \dfrac{1}{4}F_2\mathcal{B}_2)(P_s - \dfrac{1}{6}\mathcal{B}_2)^{-1}e_1 + \dfrac{1}{4}e_1, \\ r_2 = \dfrac{1}{24}(-K + \dfrac{1}{4}F_2\mathcal{B}_2)(P_s - \dfrac{1}{6}\mathcal{B}_2)^{-1}e_{N-1} + \dfrac{1}{4}e_{N-1}, \end{cases}$$

$$(3.58)\qquad \begin{cases} s_1 = -\dfrac{3}{8}(-K - \dfrac{1}{3}\mathcal{A}_2{}^T)(P_s - \dfrac{1}{6}\mathcal{B}_2)^{-1}e_1, \\ s_2 = \dfrac{3}{8}(-K - \dfrac{1}{3}\mathcal{A}_2{}^T)(P_s - \dfrac{1}{6}\mathcal{B}_2)^{-1}e_{N-1}, \end{cases}$$

where the vectors $v_1, v_2 \in \mathbb{R}^{N-1}$ are given (3.50). The matrix $\mathcal{H}_2$ is non symmetric only because of the perturbation $\delta\mathcal{H} = RS^T$.

4 Bidimensional Hermitian Box-Schemes

In this section we extend the design principle of the Hermitian Box-Scheme from dimension one to dimension two. In Section 4.1, the derivation of the bidimensional is carried out without any specific assumption on the discrete form of the derivative on the boundary. In Section 4.2, the general matrix form is obtained using Kronecker algebra in a systematic way. Particular examples are the bidimensional HB-scheme 1 and HB-scheme 2 which are introduced In Section 4.3. They differ only by the approximation of the boundary derivatives in terms of the internal unknowns. A fast solver algorithm is given for each of them.

4.1 Design of the Hermitian Box-Scheme in a rectangle

We consider the extension of the HB-Scheme to the bidimensional Poisson Problem in the square $\Omega = (a,b)^2$,

$$\begin{cases} -\Delta u = f, & (x,y) \in \Omega \quad (a) \\ u = g, & \text{on } \partial\Omega \quad (b). \end{cases} \tag{4.1}$$

The square cell $K_{i,j}$ centered at point (x_i, y_j) with length $2h$ is, (see Fig.1):

$$K_{i,j} = [x_i - h, x_i + h] \times [y_j - h, y_j + h], \quad \forall 1 \leq i,j \leq N-1. \tag{4.2}$$

The three discrete unknowns at internal points are

$$U = (u_{i,j})_{1\leq i,j\leq N-1}, \;\; U_x = (u_{x,i,j})_{1\leq i,j\leq N-1}, \;\; U_y = (u_{y,i,j})_{1\leq i,j\leq N-1}, \tag{4.3}$$

with $u_{i,j} \simeq u(x_i, y_j)$, $u_{x,i,j} \simeq \frac{\partial u}{\partial x}(x_i, y_j)$, $u_{y,i,j} \simeq \frac{\partial u}{\partial y}(x_i, y_j)$. The design of the scheme proceeds as in dimension one. First, using the Green theorem, the average of $(4.1)_a$ on $K_{i,j}$ yields (the outer normal vector to $\partial K_{i,j}$ is $\overrightarrow{n}(n_1, n_2)$),

$$-\frac{1}{h^2}\int_{\partial K_{i,j}} (\frac{\partial u}{\partial x} n_1 + \frac{\partial u}{\partial y} n_2)\, d\sigma = \Pi^0 f_{i,j}, \;\; 1 \leq i,j \leq N-1. \tag{4.4}$$

where

$$\Pi^0 f_{i,j} = \frac{1}{4h^2}\int_{K_{i,j}} f(x,y)dxdy, \quad 1 \leq i,j \leq N-1. \tag{4.5}$$

The left-hand side in (4.4) consists of four one-dimensional integrals corresponding to the four edges of $\partial K_{i,j}$. Each of them is approximated in turn by the Simpson formula. This gives the following approximation of (4.4), for

$1 \leq i, j \leq N-1$,

(4.6)

$$\begin{cases} -\frac{1}{(2h)^2}\Big\{ & \left[\frac{1}{6}u_{x,i+1,j-1} + \frac{2}{3}u_{x,i+1,j} + \frac{1}{6}u_{x,i+1,j+1}\right] \\ & -\left[\frac{1}{6}u_{x,i-1,j-1} + \frac{2}{3}u_{x,i-1,j} + \frac{1}{6}u_{x,i-1,j+1}\right] \\ & +\left[\frac{1}{6}u_{y,i-1,j+1} + \frac{2}{3}u_{y,i,j+1} + \frac{1}{6}u_{y,i+1,j+1}\right] \\ & -\left[\frac{1}{6}u_{y,i-1,j-1} + \frac{2}{3}u_{y,i,j-1} + \frac{1}{6}u_{y,i+1,j-1}\right]\Big\} = \Pi^0 f_{i,j}. \end{cases}$$

Second, the connection of u to u_x is obtained as counterparts of (2.8), by

(4.7)

$$\frac{1}{6}u_{x,i-1,j} + \frac{2}{3}u_{x,i,j} + \frac{1}{6}u_{x,i+1,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \;\; 1 \leq i \leq N-1, \;\; 0 \leq j \leq N$$

Similarly in the y-direction, the connection of u_y to u is given by

(4.8)

$$\frac{1}{6}u_{y,i,j-1} + \frac{2}{3}u_{y,i,j} + \frac{1}{6}u_{y,i,j+1} = \frac{u_{i,j+1} - u_{i,j-1}}{2h}, \;\; 1 \leq j \leq N-1, \;\; 0 \leq i \leq N.$$

Equations (4.6), (4.7), (4.8) constitute the core of the HB-Scheme in two dimensions. To fully define the HB-Scheme, we have to add the boundary conditions.

- First, the Dirichlet boundary conditions translate to the four series of relations, along the four sides of the square Ω,

$$\begin{cases} u_{i,0} = g(x_i, y_0), \;\; 0 \leq i \leq N, \\ u_{i,N} = g(x_i, y_N), \;\; 0 \leq i \leq N, \\ u_{0,j} = g(x_0, y_j), \;\; 1 \leq j \leq N-1, \\ u_{N,j} = g(x_N, y_j), \;\; 1 \leq j \leq N-1. \end{cases} \tag{4.9}$$

- Second, the approximation of the normal derivative $u_{x,0,j}, u_{x,N,j}$, $1 \leq j \leq N-1$, and $u_{y,i,0}, u_{y,i,N}$, $1 \leq i \leq N-1$ along each side has to be specified in terms of the internal unknowns (discrete Dirichlet-to-Neumann operator).

  As a first example, the bidimensional form of the *HB-Scheme 1* consists in writing along the four sides of the square the following equations, (see (2.10)),

$$\begin{cases} \frac{2}{3}u_{x,0,j} + \frac{1}{3}u_{x,1,j} = \frac{u_{1,j} - u_{0,j}}{h}, \;\; 0 \leq j \leq N, \\ \frac{2}{3}u_{x,N,j} + \frac{1}{3}u_{x,N-1,j} = \frac{u_{N,j} - u_{N-1,j}}{h}, \;\; 0 \leq j \leq N, \\ \frac{2}{3}u_{y,i,0} + \frac{1}{3}u_{y,i,1} = \frac{u_{i,1} - u_{i,0}}{h}, \;\; 0 \leq i \leq N, \\ \frac{2}{3}u_{y,i,N} + \frac{1}{3}u_{y,i,N-1} = \frac{u_{i,N} - u_{i,N-1}}{h}, \;\; 0 \leq i \leq N. \end{cases} \tag{4.10}$$

Observe that the total number of equations actually coincides with the number of unknowns. The unknowns are $(u_{i,j}, u_{x,i,j}, u_{y,i,j},\ 0 \leq i, j \leq N)$, which gives $3(N+1)^2$ unknowns. On the other hand, we have

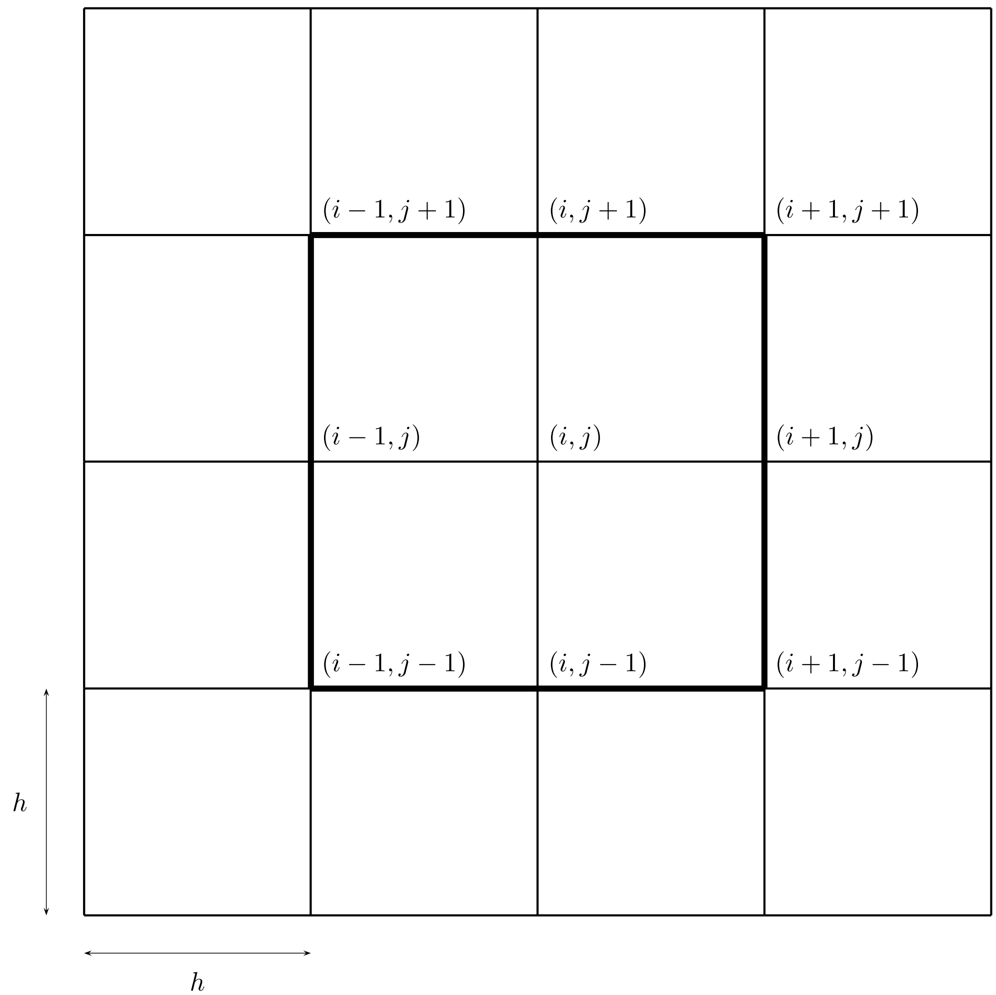

Figure 1: *HB-Scheme in two dimensions. The flux is integrated along the thick contour of the box* $K_{i,j} = [x_i - h, x_i + h] \times [y_j - h, y_j + h]$.

- $(N-1)^2$ equations in (4.6),
- $(N-1)(N+1)$ equations in (4.7) and in (4.8).
- $4N$ equations in (4.9),
- $4(N+1)$ equations in (4.10).

This gives a total number of equations of $(N-1)^2 + 2(N-1)(N+1) + 4N + 4(N+1) = 3(N+1)^2$, which coincides with the total number of equations.
Let us now turn to the finite-difference form of the three equations (4.6), (4.7), (4.8). This form is used in Section 4.2 to derive the matrix form of the HB-Scheme with Kronecker algebra. In analogy to the one-dimensional case, we let finite-difference operators act on the space L_h^2 of two-dimensional grid functions $u = (u_{i,j})_{0\le i,j\le N}$. The discrete unknowns are the three grid functions $(u, u_x, u_y) \in L_h^2$. The subspace of grid functions with homogeneous boundary conditions along the four sides of the square grid is $L_{h,0}^2$. Finite-difference operators in two dimensions useful in the sequel are

- the centered operators δ_x, δ_y

$$\text{(4.11)} \qquad \begin{cases} \delta_x u_{i,j} = \dfrac{u_{i+1,j} - u_{i-1,j}}{2h}, & 1 \le i,j \le N-1, \quad (a), \\ \delta_y u_{i,j} = \dfrac{u_{i,j+1} - u_{i,j-1}}{2h}, & 1 \le i,j \le N-1, \quad (b). \end{cases}$$

- the Simpson average operators σ_x, σ_y

$$\text{(4.12)} \quad \begin{cases} \sigma_x u_{i,j} = \dfrac{1}{6}u_{i-1,j} + \dfrac{2}{3}u_{i,j} + \dfrac{1}{6}u_{i+1,j}, & 1 \le i,j \le N-1, \quad (a), \\ \sigma_y u_{i,j} = \dfrac{1}{6}u_{i,j-1} + \dfrac{2}{3}u_{i,j} + \dfrac{1}{6}u_{i,j+1}, & 1 \le i,j \le N-1, \quad (b). \end{cases}$$

Using these operators, the three equations of the HB-Scheme (4.6), (4.7), (4.8) translate in terms of the grid functions $u, u_x, u_y \in L_h^2$ as
(4.13)

$$\begin{cases} -\sigma_y(\delta_x u_x)_{i,j} - \sigma_x(\delta_y u_y)_{i,j} = \Pi^0 f_{i,j}, \quad 1 \le i,j \le N-1, \ (a), \\ \sigma_x u_{x,i,j} = \delta_x u_{i,j}, \ 1 \le i \le N-1, \ 0 \le j \le N, \ (b), \\ \sigma_y u_{y,i,j} = \delta_y u_{i,j} \ 0 \le i \le N, \ 1 \le j \le N-1, \ (c). \end{cases}$$

4.2 Matrix form of the bidimensional Hermitian Box-Scheme: general case

In this section, we show that the algebraic structure of the HB-Scheme can be interpretated in a simple way using Kronecker matrix algebra. In many situations one can take advantage of that structure to develop fast resolution procedures. For a summary and basic properties of the Kronecker product of matrices, we refer to [22, 28]. For recent applications in fast computing in high dimensions, [23, 27]. We also make use of the mapping usually denoted by the operator "vec" in Kronecker algebra, which maps the grid function $v = (v_{i,j}) \in L^2_{h,0}$ to the vector $V = \text{vec}(v) \in \mathbb{R}^{(N-1)^2}$. The operator "vec" creates a column vector of size $(N-1)^2$ from a $(N-1)\times(N-1)$ matrix A by stacking the column vectors of $A = [a_1; a_2; ..; a_{N-1}]$ below one another as

$$\text{vec}(A) = \begin{pmatrix} a_1 \\ \vdots \\ a_{N-1} \end{pmatrix}. \tag{4.14}$$

The vectors $U, U_x, U_y \in \mathbb{R}^{(N-1)^2}$ are defined by

$$U = \text{vec}(u), \quad U_x = \text{vec}(u_x), \quad U_y = \text{vec}(u_y). \tag{4.15}$$

We need also the mapping of two vectors $v, w \in \mathbb{R}^{N-1}$ onto a grid function with v and w in left and right positions,

$$[v_1, \ldots, v_{N-1}, 0_{N-1}, \ldots, 0_{N-1}, w_1, \ldots, w_{N-1}]^T = e_1 \otimes v + e_{N-1} \otimes w. \tag{4.16}$$

The transpose maps two vectors $v, w \in \mathbb{R}^{N-1}$ onto a grid function with v and w respectively in bottom and top positions,

(4.17)
$$[v_1, 0, \ldots, 0, w_1, v_2, 0, \ldots, 0, w_{2,}, 0, \ldots, 0, v_{N-1}, 0, .., 0, w_{N-1}]^T = v \otimes e_1 + w \otimes e_{N-1}.$$

Here we denote $U_L, U_R \in \mathbb{R}^{N-1}$ the left and right Dirichlet boundary vector data at $x = a$ and $x = b$. Similarly, U_B, U_T are the Bottom and Top Dirichlet data at $y = a$ and $y = b$,

$$\begin{cases} U_L = [u_{0,1}, .., u_{0,N-1}]^T, \\ U_R = [u_{N,1}, .., u_{N,N-1}]^T, \\ U_B = [u_{1,0}, .., u_{N-1,0}]^T, \\ U_T = [u_{1,N}, .., u_{N-1,N}]^T. \end{cases} \tag{4.18}$$

The boundary gradient vectors in $\mathbb{R}^{N-1}$ are denoted by $(U_{x,L}, U_{y,L})$, $(U_{x,R}, U_{y,R})$, and $(U_{x,T}, U_{y,T})$, $(U_{x,B}, U_{y,B})$. For example, we have

$$U_{x,L} = [u_{x,0,1}, u_{x,0,2}, \cdots, u_{x,0,N-2}, u_{x,0,N-1}]^T. \tag{4.19}$$

We denote the four corner values $u_{0,0}, u_{0,N}, u_{N,0}, u_{N,N}$ by

$$U_{LB} = u_{0,0}, \; U_{LT} = u_{0,N}, \; U_{RB} = u_{N,0}, \; U_{RT} = u_{N,N}. \tag{4.20}$$

Similarly, the derivatives at the four corner points are

$$\begin{cases} (U_{x,LB}, U_{y,LB}) = (u_{x,0,0}, u_{y,0,0}), \\ (U_{x,RB}, U_{y,RB}) = (u_{x,N,0}, u_{y,N,0}), \\ (U_{x,LT}, U_{y,LT}) = (u_{x,0,N}, u_{y,0,N}), \\ (U_{x,RT}, U_{y,RT}) = (u_{x,N,N}, u_{y,N,N}). \end{cases} \tag{4.21}$$

We consider now all the equations needed to eliminate the internal and boundary derivatives in terms of $U = \mathrm{vec}(u)$, U_L, U_R, U_B and U_T.

• First, (4.7, 4.8) translate in matrix form as the extension of (3.21) to the bidimensional case. This gives the derivatives $U_x = \mathrm{vec}(u_x), U_y = \mathrm{vec}(u_y)$ in terms of $U = \mathrm{vec}(u)$, U_L, U_R, U_B, U_T as

$$\left\{ \begin{array}{l} U_x = \dfrac{1}{h}\Big((\mathcal{D}\otimes I)U + (\mathcal{E}\otimes I)(e_1\otimes U_L + e_{N-1}\otimes U_R)\Big), \quad (a), \\ U_y = \dfrac{1}{h}\Big((I\otimes\mathcal{D})U + (I\otimes\mathcal{E})(U_B\otimes e_1 + U_T\otimes e_{N-1})\Big), \quad (b). \end{array} \right. \tag{4.22}$$

• Second, the normal derivatives along the four sides of the square are deduced from (3.15). The bidimensional vector form turns out to be

$$\left\{ \begin{array}{l} e_1\otimes U_{x,L} + e_{N-1}\otimes U_{x,R} = \dfrac{1}{h}\Big((\mathcal{A}\otimes I)U - h(\mathcal{B}\otimes I)U_x + (\mathcal{C}\otimes I)\big(e_1\otimes U_L + e_{N-1}\otimes U_R\big)\Big), \\ U_{y,B}\otimes e_1 + U_{y,T}\otimes e_{N-1} = \dfrac{1}{h}\Big((I\otimes\mathcal{A})U - h(I\otimes\mathcal{B})U_y + (I\otimes\mathcal{C})\big(U_B\otimes e_1 + U_T\otimes e_{N-1}\big)\Big). \end{array} \right. \tag{4.23}$$

As an example, the case of the boundary conditions for the HB-scheme 1 in (4.10) translates to (4.23) with $\mathcal{A} = \mathcal{A}_1$, $\mathcal{B} = \mathcal{B}_1$, $\mathcal{C} = \mathcal{C}_1$ (see (3.17)).

• Third, we need the tangential derivatives along the four edges of the square. Consider for example the bottom side, where the tangential derivative is the derivative in the $x-$ direction. Using (4.7) with $j = 0$ and (3.21), we observe that $U_{x,B}$ can be expressed in terms of U_B as

$$U_{x,B} = \frac{1}{h}\mathcal{D}U_B + \frac{1}{h}\mathcal{E}(e_1 U_{LB} + e_{N-1}U_{RB}). \tag{4.24}$$

Using also the Dirichlet-to-Neumann relation along the bottom side considered as a segment, (see (3.15)), the $x-$ derivatives $U_{x,LB}$, $U_{x,RB}$ at the two bottom corner points LB and RB are expressed in terms of the bottom values $U_B \in \mathbb{R}^{N-1}$, $U_{LB}, U_{RB} \in \mathbb{R}$, by

$$e_1 U_{x,LB} + e_{N-1}U_{x,RB} = \frac{1}{h}\Big(\mathcal{A}U_B - h\mathcal{B}U_{x,B} + \mathcal{C}\big(e_1 U_{LB} + e_{N-1}U_{RB}\big)\Big). \tag{4.25}$$

The tangential derivatives along the three other sides are derived in a similar way.

By the principle already used in the one-dimensional case, we derive the matrix form of the finite difference equation of (4.6), which is $(4.13)_a$,

$$-\sigma_y(\delta_x u_x)_{i,j} - \sigma_x(\delta_y u_y)_{i,j} = \Pi^0 f_{i,j}, \quad 1 \le i,j \le N-1. \tag{4.26}$$

Consider the first term $\sigma_y(\delta_x u_x)$. For all $v \in L_h^2$, the grid function $\sigma_y v \in L_h^2$ is defined by (see $(4.12)_b$),

$$\sigma_y v_{i,j} = \frac{1}{6}v_{i,j-1} + \frac{2}{3}v_{i,j} + \frac{1}{6}v_{i,j+1}, \quad 1 \le i,j \le N-1. \tag{4.27}$$

Therefore, with $V = \mathrm{vec}(v) \in \mathbb{R}^{(N-1)^2}$, the vec operator applied to $\sigma_y v$ can be expressed as

$$\mathrm{vec}(\sigma_y v) = (I \otimes P_s)V + \frac{1}{6}(I \otimes e_1)V_B + \frac{1}{6}(I \otimes e_{N-1})V_T. \tag{4.28}$$

Using $v = \delta_x u_x$, in (4.28) gives

$$\mathrm{vec}(\sigma_y(\delta_x u_x)) = (I\otimes P_s)\,\mathrm{vec}(\delta_x u_x) + \frac{1}{6}(I\otimes e_1)(\mathrm{vec}(\delta_x u_x))_B + \frac{1}{6}(I\otimes e_{N-1})(\mathrm{vec}(\delta_x u_x))_T. \tag{4.29}$$

Furthermore, $\mathrm{vec}(\delta_x u_x) \in \mathbb{R}^{(N-1)^2}$ is expressed as

$$\begin{aligned} \mathrm{vec}(\delta_x u_x) &= \frac{1}{2h}(K \otimes I)U_x - \frac{1}{2h}(e_1 \otimes I)U_{x,L} + \frac{1}{2h}(e_{N-1} \otimes I)U_{x,R} \\ &= \frac{1}{2h}(K \otimes I)U_x + \frac{1}{2h}(F_2 \otimes I)(e_1 \otimes U_{x,L} + e_{N-1} \otimes U_{x,R}). \end{aligned} \tag{4.30}$$

Substituting (4.30) in (4.29) yields

$$\begin{aligned} \mathrm{vec}(\sigma_y(\delta_x u_x)) &= \frac{1}{2h}(I \otimes P_s)\Big((K \otimes I)U_x + (F_2 \otimes I)(e_1 \otimes U_{x,L} + e_{N-1} \otimes U_{x,R})\Big) \\ &+ \frac{1}{12h}(I \otimes e_1)\Big(KU_{x,B} + F_2(e_1 U_{x,LB} + e_{N-1}U_{x,RB})\Big) \\ &+ \frac{1}{12h}(I \otimes e_{N-1})\Big(KU_{x,T} + F_2(e_1 U_{x,LT} + e_{N-1}U_{x,RT})\Big). \end{aligned}$$

This provides the decomposition of the grid function $\sigma_y(\delta_x u_x)$ in four terms $(a),(b),(c),(d)$ as

$$\left\{ \begin{aligned} \mathrm{vec}(\sigma_y(\delta_x u_x)) &= \frac{1}{2h}(K \otimes P_s)U_x \quad (a) \\ &+ \frac{1}{2h}(F_2 \otimes P_s)\big(e_1 \otimes U_{x,L} + e_{N-1} \otimes U_{x,R}\big) \quad (b) \\ &+ \frac{1}{12h}(I \otimes e_1)KU_{x,B} + \tfrac{1}{12h}(I \otimes e_1)F_2(e_1 U_{x,LB} + e_{N-1}U_{x,RB}) \quad (c) \\ &+ \frac{1}{12h}(I \otimes e_{N-1})KU_{x,T} + \tfrac{1}{12h}(I \otimes e_{N-1})F_2(e_1 U_{x,LT} + e_{N-1}U_{x,RT}) \quad (d). \end{aligned} \right. \tag{4.31}$$

Each term $(a),(b),(c),(d)$ in the preceding identity is expanded, by expressing the derivatives in terms of the unknown U and of the boundary data.

• *Term* (a)

Using $(4.22)_a$, the term (a) can be expressed as

$$\begin{aligned} (a) &= \frac{1}{2h}(K \otimes P_s)U_x \\ &= \frac{1}{2h^2}(K \otimes P_s)\Big((\mathcal{D} \otimes I)U + (\mathcal{E} \otimes I)(e_1 \otimes U_L + e_{N-1} \otimes U_R)\Big). \end{aligned}$$

• *Term (b)*
The term (b) corresponds to the normal derivatives along the boundary. Using (4.23), it can be written as

$$\begin{aligned}(b) &= \frac{1}{2h}(F_2 \otimes P_s)(e_1 \otimes U_{x,L} + e_{N-1} \otimes U_{x,R}) \\ &= \frac{1}{2h^2}(F_2 \otimes P_s)\Big\{(\mathcal{A} \otimes I)U - h(\mathcal{B} \otimes I)U_x + (C \otimes I)(e_1 \otimes U_L + e_{N-1} \otimes U_R)\Big\} \\ &= \frac{1}{2h^2}(F_2 \otimes P_s)\Big\{(\mathcal{A} \otimes I)U - (\mathcal{B} \otimes I)\Big((\mathcal{D} \otimes I)U + (\mathcal{E} \otimes I)(e_1 \otimes U_L + e_{N-1} \otimes U_R)\Big) \\ &+ (C \otimes I)(e_1 \otimes U_L + e_{N-1} \otimes U_R)\Big\}.\end{aligned}$$

This is rewritten as
(4.32)
$$(b) = \frac{1}{2h^2}\Big\{\Big(F_2(\mathcal{A} - \mathcal{B}\mathcal{D}) \otimes P_s\Big)U + \Big(F_2(\mathcal{C} - \mathcal{B}\mathcal{E}) \otimes P_s\Big)(e_1 \otimes U_L + e_{N-1} \otimes U_R)\Big\}.$$

• *Term (c)*
The term (c) corresponds to the tangential derivatives along the bottom side. Using (4.24, 4.25), it is expressed as

$$(c) = \frac{1}{12h}(I \otimes e_1)KU_{x,B} + \frac{1}{12h}(I \otimes e_1)F_2(e_1U_{x,LB} + e_{N-1}U_{x,RB}), \tag{4.33}$$

or equivalently

$$\begin{aligned}(c) &= \frac{1}{12h^2}(I \otimes e_1)K\Big(\mathcal{D}U_B + \mathcal{E}(e_1U_{LB} + e_{N-1}U_{RB})\Big) \\ &+ \frac{1}{12h^2}(I \otimes e_1)F_2\Big(\mathcal{A}U_B - h\mathcal{B}U_{x,B} + \mathcal{C}(e_1U_{LB} + e_{N-1}U_{RB})\Big).\end{aligned}$$

Using (4.24) again, the term (c) can be further expanded as

$$\begin{aligned}(c) &= \frac{1}{12h^2}(I \otimes e_1)(K\mathcal{D} + F_2\mathcal{A})U_B - \frac{1}{12h^2}(I \otimes e_1)F_2\mathcal{B}\Big(\mathcal{D}U_B + \mathcal{E}(e_1U_{LB} + e_{N-1}U_{RB})\Big) \\ &+ \frac{1}{12h^2}(I \otimes e_1)\Big(K\mathcal{E} + F_2\mathcal{C}\Big)\Big(e_1U_{LB} + e_{N-1}U_{RB}\Big) \\ &= \frac{1}{12h^2}(I \otimes e_1)\Big(K\mathcal{D} + F_2(\mathcal{A} - \mathcal{B}\mathcal{D})\Big)U_B \\ &+ \frac{1}{12h^2}(I \otimes e_1)\Big(K\mathcal{E} + F_2(\mathcal{C} - \mathcal{B}\mathcal{E})\Big)\Big(e_1U_{LB} + e_{N-1}U_{RB}\Big).\end{aligned}$$

Using the expression of the matrices $\mathcal{H}$ and $\mathcal{G}$ given in (3.25), we obtain finally

$$(c) = -\frac{1}{6h^2}\mathcal{H}U_B \otimes e_1 - \frac{1}{6h^2}\mathcal{G}\Big(e_1U_{LB} + e_{N-1}U_{RB}\Big) \otimes e_1. \tag{4.34}$$

• *Term (d)*
The term (d) is

$$(d) = -\frac{1}{6h^2}\mathcal{H}U_T \otimes e_{N-1} - \frac{1}{6h^2}\mathcal{G}\Big(e_1 U_{LT} + e_{N-1}U_{RT}\Big) \otimes e_{N-1}. \tag{4.35}$$

Rearranging in (4.31) the term $(a)+(b)$ and adding (c) and (d), we find that

$$\begin{aligned} \mathrm{vec}(\sigma_y(\delta_x u_x)) &= -\frac{1}{h^2}(\mathcal{H}\otimes P_s)U - \frac{1}{h^2}(\mathcal{G}\otimes P_s)(e_1\otimes U_L + e_{N-1}\otimes U_R) \\ &- \frac{1}{6h^2}\mathcal{H}U_B \otimes e_1 - \frac{1}{6h^2}\mathcal{G}(e_1 U_{LB} + e_{N-1}U_{RB})\otimes e_1 \\ &- \frac{1}{6h^2}\mathcal{H}U_T \otimes e_{N-1} - \frac{1}{6h^2}\mathcal{G}(e_1 U_{LT} + e_{N-1}U_{RT})\otimes e_{N-1}. \end{aligned} \tag{4.36}$$

Symmetrically, the matrix form of the term $\sigma_x(\delta_y u_y)$ in (4.26) is,

$$\begin{aligned} \mathrm{vec}(\sigma_x(\delta_y u_y)) &= -\frac{1}{h^2}(P_s\otimes\mathcal{H})U - \frac{1}{h^2}(P_s\otimes\mathcal{G})(U_B\otimes e_1 + U_T\otimes e_{N-1}) \\ &- \frac{1}{6h^2}e_1\otimes\mathcal{H}U_L - \frac{1}{6h^2}e_1\otimes\mathcal{G}(e_1U_{LB} + e_{N-1}U_{LT}) \\ &- \frac{1}{6h^2}e_{N-1}\otimes\mathcal{H}U_R - \frac{1}{6h^2}e_{N-1}\otimes\mathcal{G}(e_1U_{RB} + e_{N-1}U_{RT}). \end{aligned} \tag{4.37}$$

Collecting the terms in (4.36, 4.37), we find out that the matrix form of (4.26) is

$$\frac{1}{h^2}\big(\mathcal{H}\otimes P_s + P_s\otimes\mathcal{H}\big)U = F - G_x - G_y. \tag{4.38}$$

The vectors F corresponds to the averaged source term in (4.5),

$$F = \mathrm{vec}(\Pi^0 f). \tag{4.39}$$

The vectors $G_x, G_y \in \mathbb{R}^{(N-1)^2}$ correspond to the non homogeneous boundary conditions,

$$\begin{cases} G_x = & \frac{1}{h^2}(\mathcal{G}\otimes P_s)(e_1\otimes U_L + e_{N-1}\otimes U_R) \\ \quad + & \frac{1}{6h^2}\Big(\mathcal{H}U_B + \mathcal{G}(e_1U_{LB} + e_{N-1}U_{RB})\Big)\otimes e_1 \\ \quad + & \frac{1}{6h^2}\Big(\mathcal{H}U_T + \mathcal{G}(e_1U_{LT} + e_{N-1}U_{RT})\Big)\otimes e_{N-1}, \end{cases} \tag{4.40}$$

and

$$\begin{cases} G_y = & \frac{1}{h^2}(P_s\otimes\mathcal{G})(U_B\otimes e_1 + U_T\otimes e_{N-1}) \\ \quad + & \frac{1}{6h^2}e_1\otimes\Big(\mathcal{H}U_L + \mathcal{G}(e_1U_{LB} + e_{N-1}U_{LT})\Big) \\ \quad + & \frac{1}{6h^2}e_{N-1}\otimes\Big(\mathcal{H}U_R + \mathcal{G}(e_1U_{RB} + e_{N-1}U_{RT})\Big). \end{cases} \tag{4.41}$$

The only parameter to specify in (4.38) consists in the choice of the discrete Dirichlet-to-Neumann approximation in (4.23). This choice results in specific matrices $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, which in turn determine the matrices $\mathcal{D}$, $\mathcal{E}$ in (3.22) and then $\mathcal{H}$, $\mathcal{G}$ in (3.26).

4.3 Fast solver for the HB-Scheme in two dimensions

4.3.1 Fast Solver for the HB-Scheme 1

It results from (4.38) that the matrix form of the *HB-Scheme 1* is

$$\frac{1}{h^2}\left(\mathcal{H}_1 \otimes P_s + P_s \otimes \mathcal{H}_1\right) U = F - G_x - G_y. \tag{4.42}$$

The matrices $\mathcal{H}_1, \mathcal{G}_1$ in (3.28) are substituted to the matrices $\mathcal{H}, \mathcal{G}$ in (4.40, 4.41). The bidimensional HB-Scheme 1 reads: find $U \in \mathbb{R}^{(N-1)^2}$ solution of

$$H_1 U = f \tag{4.43}$$

where the matrix H_1 is defined by

$$H_1 = \mathcal{H}_1 \otimes P_s + P_s \otimes \mathcal{H}_1, \tag{4.44}$$

and

$$f = h^2(F - G_x - G_y). \tag{4.45}$$

The vectors F, G_x, $G_y \in \mathbb{R}^{(N-1)^2}$ are given in (4.39, 4.40, 4.41). Since P_s, $\mathcal{H}_1$ are both diagonal in the basis Z^k, with eigenvalues $\alpha_k = 1 - \lambda_k/6$, μ_k, the matrix H_1 is diagonal in the basis $Z^k \otimes Z^l$ with spectral decomposition, (see (3.5), (3.32)),

$$H_1 = \big(Z \otimes Z\big)\Big(\mathcal{M} \otimes (I - \Lambda/6) + (I - \Lambda/6) \otimes \mathcal{M}\Big)\big(Z^T \otimes Z^T\big). \tag{4.46}$$

The eigenvalues are $\beta_{k,l}$ given by

$$\beta_{k,l} = \mu_k \alpha_l + \alpha_k \mu_l, \quad 1 \le k, l \le N-1. \tag{4.47}$$

Solving (4.43) is performed by FFT using the following algorithm. It extends the classical FFT algorithm used to solve the five-point Laplacian in a rectangle.

Algorithm 4.1 *Algorithm 1: (Fast FFT algorithm for the HB-Scheme 1)*

- **Step 1:** *Decompose the source term $f = h^2(F - G_x - G_y)$, (see (4.39),(4.40), (4.41)) with $\mathcal{H} = \mathcal{H}_1$, $\mathcal{G} = \mathcal{G}_1$), on the orthonormal basis $Z^k \otimes Z^l$ of $\mathbb{R}^{(N-1)^2}$. Recall that μ_k and $\alpha_k = 1 - \frac{1}{6}\lambda_k$ are the eigenvalues of matrix $\mathcal{H}_1$ and P_s respectively. This step consists of computing the coefficients $f^Z_{k,l} = (f, Z^k \otimes Z^l)$, $1 \le k, l \le N-1$ and is performed by Fast Sine Transform, [31].*

- **Step 2:** *Compute the components* $U^Z_{k,l}$ *of the solution in the Fourier space by*

$$U^Z_{k,l} = \frac{f^Z_{k,l}}{\beta_{k,l}}, \quad 1 \le k,l \le N-1. \tag{4.48}$$

- **Step 3:** *Assemble componentwise the solution using the decomposition of the grid function* $U \in \mathbb{R}^{(N-1)^2}$ *in* $Z^k \otimes Z^l$ *by*

$$U_{i,j} = \sum_{k,l=1}^{N-1} U^Z_{k,l} Z^k_i Z^l_j. \tag{4.49}$$

The grid function $u \in L^2_{h,0}$ *is such that* $U = \mathrm{vec}(u)$, *therefore*

$$u_{i,j} = U_{i,j}, \quad 1 \le i,j \le N-1. \tag{4.50}$$

Steps 1 and 3 are $O(N^2 \log_2(N))$, and Step 2 is $O(N^2)$, which gives a $O(2N^2 \log_2(N)) + O(N^2)$ algorithm.

4.3.2 Fast Solver for the HB-Scheme 2

In this section, we derive a fast direct resolution procedure for the fourth-order *HB-Scheme 2*. Here (4.38) is expressed as

$$\frac{1}{h^2}\Big(\mathcal{H}_2 \otimes P_s + P_s \otimes \mathcal{H}_2\Big)U = F - G_x - G_y. \tag{4.51}$$

In (4.51), the matrices $\mathcal{H}_2$ and P_s are given in $(3.40)_a$, (3.8). The three terms of the right-hand side are given in (4.39), (4.40), (4.41) whith $\mathcal{G} = \mathcal{G}_2$, (see$(3.40)_b$). Our first result states the algebraic structure of (4.51). Observe that the matrix structure of the HB-Scheme 2 keeps exactly the same shape as the HB-Scheme 1. Using the structure of the matrix $\mathcal{H}_2$ in (3.54), we obtain that

$$\underbrace{\mathcal{H}_2 \otimes P_s + P_s \otimes \mathcal{H}_2}_{H_2} = \underbrace{(\mathcal{H}_1 \otimes P_s + P_s \otimes \mathcal{H}_1)}_{H_1} + \underbrace{\big(RS^T \otimes P_s + P_s \otimes RS^T\big)}_{\delta H}. \tag{4.52}$$

The linear system to solve is therefore

$$H_2 U = f \tag{4.53}$$

where $f = h^2(F - G_x - G_y)$. Furthermore, rank $(\delta H) = 12(N-1)$ compared to rank $(H_1) = (N-1)^2$. Let us give the detailed structure of δH. The matrix P_s can be expressed as

$$P_s = Z\,\mathrm{diag}(\alpha_1, \cdots, \alpha_{N_1})Z^T. \tag{4.54}$$

Defining the vector $Z'^{,k} = \alpha_k^{1/2} Z^k$, the matrices $Z', P_s \in \mathbb{M}_{N-1}(\mathbb{R})$ are

$$Z' = [Z'^{,1}, \cdots, Z'^{,N-1}], \quad P_s = Z' Z'^{,T}. \tag{4.55}$$

Therefore δH can be written as

$$
\begin{aligned}
\delta H &= RS^T \otimes P_s + P_s \otimes RS^T \\
&= \Big[R_1 \otimes Z', \cdots, R_6 \otimes Z', Z' \otimes R_1, \cdots, Z' \otimes R_6\Big] \begin{bmatrix} S_1^T \otimes Z'^{,T} \\ \vdots \\ S_6^T \otimes Z'^{,T} \\ Z'^{,T} \otimes S_1^T \\ \vdots \\ Z'^{,T} \otimes S_6^T \end{bmatrix}.
\end{aligned}
$$

This result in $\delta H = LM^T$ where the rectangular matrices $L, M \in \mathbb{M}_{(N-1)^2, 12(N-1)}(\mathbb{R})$ are

$$L = [L_1, \cdots, L_{12}], \quad M = [M_1, \cdots, M_{12}]. \tag{4.56}$$

The matrices $L_i, M_i \in \mathbb{M}_{(N-1)^2,(N-1)}(\mathbb{R})$ are for $1 \le i \le 6$,

$$\begin{cases} L_i = R_i \otimes Z', & L_{i+6} = Z' \otimes R_i, \\ M_i = S_i \otimes Z', & M_{i+6} = Z' \otimes S_i. \end{cases} \tag{4.57}$$

The basis of our fast algorithm is the Sherman-Morrison formula (see ([22], Chap. 2, pp. 50) applied to the matrix (4.52). It consists in expressing the inverse of the matrix $H_2 = H_1 + LM^T$ as

$$H_2^{-1} = H_1^{-1} - H_1^{-1} L \Big(I_{12(N-1)} + M^T H_1^{-1} L \Big)^{-1} M^T H_1^{-1}. \tag{4.58}$$

According to (4.58), the following algorithm summarizes the solution procedure. Indications of the computing complexity are given at each step of the algorithm. Despite the apparent length of the algorithm, which is given in full details, the implementation keeps to be simple, using standard vector operations in *matlab* or FORTRAN 90.

Algorithm 4.2 *Algorithm 2 (Fast FFT algorithm for the HB-Scheme 2)*

- **Step 1:** *Let* $f = h^2(F - G_x - G_y) \in \mathbb{R}^{(N-1)^2}$ *be the source-term vector. Solve the linear system*

  $$H_1 g = f. \tag{4.59}$$

  It is solved using the Algorithm 4.1 at a cost $O(2N^2 \log_2(N)) + O(N^2)$. *The vector* $g \in \mathbb{R}^{(N-1)^2}$ *is stored for the Step 7.*

- **Step 2:** *Given* $g \in \mathbb{R}^{(N-1)^2}$ *in Step 1, compute the vector* $M^T g \in \mathbb{R}^{12(N-1)}$ *defined by*

  $$M^T g = \Big[M_1^T g, M_2^T g, .., M_{12}^T g\Big]^T. \tag{4.60}$$

The first component $M_1^T g \in \mathbb{R}^{(N-1)}$ *is*

$$M_1^T g = \Big[(S_1 \otimes Z'^{,1})^T g, \cdots, (S_1 \otimes Z'^{,N-1})^T g\Big]^T. \tag{4.61}$$

For $1 \leq l \leq N-1$, *the* l*-component of the vector* $M_1^T g$ *in (4.61) is*

$$\begin{aligned}(S_1 \otimes Z'^{,l})^T g &= \sum_{i=1}^{N-1} (S_1)_i \sum_{j=1}^{N-1} g_{i,j} Z_j'^{,l} \\ &= \alpha_l^{1/2} \sum_{i=1}^{N-1} (S_1)_i \sum_{j=1}^{N-1} g_{i,j} Z_j^l.\end{aligned} \tag{4.62}$$

The terms

$$\sum_{j=1}^{N-1} g_{i,j} Z_j^l, \;\; 1 \leq i \leq N-1, \tag{4.63}$$

are computed by FFT using

$$\sum_{j=1}^{N-1} g_{i,j} Z_j^l = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{j=1}^{N-1} g_{i,j} \sin\left(\frac{jl\pi}{N}\right). \tag{4.64}$$

The same method is used to compute $M_2^T g, \cdots, M_6^T g$. *Similarly, the* $k-th$ *component in* $M_7^T g$ *is*

$$\begin{aligned}(Z'^{,k} \otimes S_1)^T g &= \sum_{j=1}^{N-1} (S_1)_j \sum_{i=1}^{N-1} g_{i,j} Z_i'^{,k} \\ &= \alpha_k^{1/2} \sum_{j=1}^{N-1} (S_1)_j \sum_{i=1}^{N-1} g_{i,j} Z_i^k.\end{aligned} \tag{4.65}$$

Again the FFT is used to compute

$$\sum_{i=1}^{N-1} g_{i,j} Z_i^k = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=1}^{N-1} g_{i,j} \sin\left(\frac{ik\pi}{N}\right). \tag{4.66}$$

The vectors $M_8^T g, \cdots, M_{12}^T g$ *are computed by the same method. The* $N-1$ *FFT computations in (4.64), (4.66) are performed at a cost* $O(2N^2 \log_2(N))$. *For each* $1 \leq l \leq N-1$, *the scalar product in (4.62) or (4.65) gives an additional* $2N$ *cost. The cost of the scalar products is therefore* $24N^2$. *In summary the cost of Step 2 is* $O(2N^2 \log_2(N)) + 24N^2$.

- **Step 3:** *Solve the* $12(N-1) \times 12(N-1)$ *auxiliary linear system,*

$$(I_{12(N-1)} + M^T H_1^{-1} L) w = M^T g. \tag{4.67}$$

The computing complexity of the whole algorithm relies on the efficiency of this solving. We use the GMRES method. No preconditionning is used for the time now. A computing cost analysis is given in Section 4.3.3.

- **Step 4:** *The solution $w \in \mathbb{R}^{12(N-1)}$ in Step 3 is decomposed as*

$$w = [w_1, w_2, ..., w_{12}]^T, w_1, w_2, .., w_{12} \in \mathbb{R}^{N-1}. \tag{4.68}$$

  Compute the vector $t = Lw,\ w \in \mathbb{R}^{12(N-1)},\ t \in \mathbb{R}^{(N-1)^2}$ by

$$t = t_1 + t_2 + t_3 + ... + t_{12}, \quad t_l = L_l w_l,\ 1 \le l \le 12 \tag{4.69}$$

  with

(4.70)
$$\begin{cases} t_1 = (R_1 \otimes [Z'^{,1}, \cdots, Z'^{,N-1}])w_1\ , \quad t_2 = (R_2 \otimes [Z'^{,1}, \cdots, Z'^{,N-1}])w_2, \\ ... \\ t_5 = (R_5 \otimes [Z'^{,1}, \cdots, Z'^{,N-1}])w_5\ , \quad t_6 = (R_6 \otimes [Z'^{,1}, ..., Z'^{,N-1}])w_6, \\ t_7 = ([Z'^{,1}, \cdots, Z'^{,N-1}] \otimes R_1)w_7\ , \quad t_8 = ([Z'^{,1}, \cdots, Z'^{,N-1}] \otimes R_2)w_8, \\ ... \\ t_{11} = ([Z'^{,1}, \cdots, Z'^{,N-1}] \otimes R_5)w_{11}, \quad t_{12} = ([Z'^{,1}, \cdots, Z'^{,N-1}] \otimes R_6)w_{12}. \end{cases}$$

  The components $(t_m)_{i,j},\ 1 \le i,j \le N-1,\ 1 \le m \le 12$ are

(4.71)
$$\begin{cases} (t_1)_{i,j} = (R_1)_i \sum_{l=1}^{N-1} \alpha_l^{1/2}(w_1)_l Z_j^l\ , \quad (t_2)_{i,j} = (R_2)_i \sum_{l=1}^{N-1} \alpha_l^{1/2}(w_2)_l Z_j^l, \\ ... \\ (t_5)_{i,j} = (R_5)_i \sum_{l=1}^{N-1} \alpha_l^{1/2}(w_5)_l Z_j^l\ , \quad (t_6)_{i,j} = (R_6)_i \sum_{l=1}^{N-1} \alpha_l^{1/2}(w_6)_l Z_j^l, \\ (t_7)_{i,j} = (R_1)_j \sum_{k=1}^{N-1} \alpha_k^{1/2}(w_7)_k Z_i^k\ , \quad (t_8)_{i,j} = (R_2)_j \sum_{k=1}^{N-1} \alpha_k^{1/2}(w_8)_k Z_i^k, \\ ... \\ (t_{11})_{i,j} = (R_5)_j \sum_{k=1}^{N-1} \alpha_k^{1/2}(w_{11})_k Z_i^k\ , \quad (t_{12})_{i,j} = (R_6)_j \sum_{k=1}^{N-1} \alpha_k^{1/2}(w_{12})_k Z_i^k. \end{cases}$$

  Each sum in each right hand side in (4.71) is computed by FFT, which gives a cost of $O(12N\log_2(N))$. Each of the 12 vectors $t_1, \cdots, t_{12}$ requires also N^2 multiplications. The sum in (4.69) requires further $12N^2$ additions. This gives a global cost of $O(12N\log_2(N)) + 24N^2$.

- **Step 5:** *Solve the linear system in $\mathbb{R}^{(N-1)^2}$*

$$w = H_1^{-1}t, \tag{4.72}$$

  via the fast FFT solver as in Step 1. The cost is $O(2N^2\log_2(N)) + O(N^2)$.

- **Step 6:** *Assemble the solution $U \in R^{(N-1)^2}$ of the linear system (4.51) by*

$$U = g - w, \tag{4.73}$$

  where $g, w \in R^{(N-1)^2}$ are given in (4.59, 4.72). The cost is $O(N^2)$.

- **Step 7:** *Compute (if needed) the hermitian gradient $U_x, U_y \in R^{(N-1)^2}$ given in (4.22) as a post-processing of the grid values of U.*

Remark:
Not that the spectral radius of $H_1^{-1}\delta H$ is close to 1.48. This discards an iterations using $H_1^{-1}\delta H$ as iterative matrix.

4.3.3 Further comments on the computational cost of Algorithm 4.2

1- Resolution of (4.67)
The complexity of the previous algorithm relies on the efficiency of solving the auxiliary linear system (4.67). The matrix of the linear system is $I_{12(N-1)} + M^T H_1^{-1} L$ where

$$M^T H_1^{-1} L = \begin{bmatrix} M_1^T H_1^{-1} L_1 & M_1^T H_1^{-1} L_2 & .. & M_1^T H_1^{-1} L_{12} \\ M_2^T H_1^{-1} L_1 & M_2^T H_1^{-1} L_2 & .. & M_2^T H_1^{-1} L_{12} \\ .. & .. & .. & .. \\ M_{12}^T H_1^{-1} L_1 & M_{12}^T H_1^{-1} L_2 & .. & M_{12}^T H_1^{-1} L_{12} \end{bmatrix} \quad (4.74)$$

Suppose that k_{it} matrix-vector products are needed to reach a prescribed accuracy (for a specified iterative solver). Due to the structure of the matrix (4.74), it turns out that each matrix vector product $M^T H_1^{-1} Lw$ requires:

- One matrix vector product $w \in \mathbb{R}^{12(N-1)} \mapsto Lw := t \in \mathbb{R}^{(N-1)^2}$. This product coincides with Step 4.
- One resolution $H_1 s = t$, where $s, t \in \mathbb{R}^{(N-1)^2}$ which coincides with Step 1.
- One matrix-vector product $s \in \mathbb{R}^{(N-1)^2} \mapsto M^T s \in \mathbb{R}^{12(N-1)}$ which coincides with Step 2.

Summing up the costs in Step 4, Step 2, Step 1 gives a cost of $O(2N^2 \log_2(N))$ $+O(49N^2)$ for each matrix vector product in (4.67). This has to be multiplied by the number of iterations k_{it}. The total cost for Algorithm 2 4.2 is found to be $O(6N^2 \log_2(N) + 51N^2) + k_{it} O(2N^2 \log_2(N) + 49N^2)$. In Table 4.3.3 the condition number of the matrix (4.67) is displayed. It appears to be independent of the grid size.
2- Comparison with other fourth order schemes
Let us briefly comment the differences between the scheme (4.53) with other fourth order schemes on regular cartesian grids. The classical fourth order scheme of Collatz [15] is usually solved using multigrid solvers, [41, 24]. This makes difficult an accurate cost analysis comparison with any direct solver. In [8, 35] a family of high order compact finite finite difference schemes (HODIE methods) is systematically studied in two and three dimensions for elliptic or Helmholtz problems. A broad series of test cases is solved using direct and iterative solvers. A specific solver is the Fourier-tridiagonal method, [10]. The Orthogonal Spline Collocation (OSC) method ([6], [4]) is fourth order accurate for the unknown and the gradient. It uses Hermite cubic splines. For many HODIE schemes as well as for the OSC method, it is possible to compute in a preliminary step the discrete spectral basis. This allows a direct resolution at a cost $O(C_1 N^2 \log_2(N) + O(C_2 N^2)$ with optimal constants C_1, C_2. This is also the case for the HB1 scheme (see Algorithm 4.1). We refer to [5] for additional properties of OSC schemes. Finally, the algorithm in [11] based on a subtractional solver uses a decomposition of the solution in an analytical part

Mesh size	8	16	32	64	128	256	512
Cond. number	11.3376	11.2784	11.2299	11.2015	11.1851	11.1757	11.1703

Table 1: Condition number of the matrix $I_{12(N-1)} + M^T H_1^{-1} L$

and a spectral part. This algorithm performs optimally in particular cases.
The design of the HB-scheme is quite different from the preceding schemes. It relies on the mixed form of the equation. The fourth order accuracy for u and ∇u results from the one of quadrature formulas and not from pointwise Taylor expansions. As in finite volume methods, only averaged values of the source term are used. at the points (ih, jh). The FFT/Sherman-Morrison solver proposed in Algorithm 4.2 is an example of a fast solver for (4.53).

5 Numerical Results

In this section we display some numerical results which prove the efficiency of the fourth-order version of HB-Scheme 2 in different cases of computational interest. Moreover, in some cases, the HB-Scheme 1 is also observed to be fourth order accurate. The F90 computing code is sequential and on a desktop computer with a processor Intel i7, 3.20 Ghz, 6GB memory. The compiler is *g95* with *-O3* optimization level. We use the package FFTPACK5 for the Fast Fourier Transform, [39]. The reported CPU time is obtained using the `cputime` function. We use the discrete L_h^2 norm to measure the errors, defined by

$$\|u - u_h\|_h = \left(h^2 \sum_{i,j=1}^{N-1} (u(x_i, y_j) - u_{i,j})^2 \right)^{\frac{1}{2}}. \tag{5.1}$$

Recall that taking into account non homogeneous boundary conditions gives an additional contribution to the right-hand side for near boundary points which is given in the G_x and G_y vectors, see (4.40, 4.41). The average operator Π^0 in (4.39) is approximated by the (tensorial) Simpson formula, which is fourth order.
The number of GMRES iterations for the resolution of (4.67) is observed to be independent of N in practice. For a relative residual condition set to 10^{-13}, 15 iterations are typically needed.
Case 1: We consider the Gaussian function $u(x, y) = \exp(-((x - 0.5)^2 + (y - 0.5)^2))$. This case is considered in [11]. The results are reported on Table 2 up to a grid of size $N = 1024$ in the $x-$ and $y-$ directions. Observe the fourth-order accuracy for the three unknowns (up to the computer accuracy).

Case 2: We consider the Poisson problem with exact solution $u(x, y) = \ln(x + y^2 + 1)$. The results are reported in Table 5. Note the singularity of the gradient along the boundary.
Case 3: In Table 4 a more difficult case is reported, with isolines along the

Mesh size	$\|u-u_h\|$	$\|u_x-u_{x,h}\|$	T=CPU(in s.)	$T/(N^2\ln_2(N))$	GMRES it.
$N=128$	$2.385(-10)$	$5.793(-10)$	0.12	1.04(-6)	15
conv. rate	3.98	4.01			
$N=256$	$1.504(-11)$	$3.585(-11)$	0.44	8.39(-7)	15
conv. rate	3.99	3.97			
$N=512$	$9,446(-13)$	$2.284(-12)$	1.81	7.67(-7)	15
conv. rate	4.00	2.06			
$N=1024$	$5.884(-14)$	$5.459(-13)$	8.49	8.09(-7)	15

Table 2: Error and convergence rate with the bidimensional HB-Scheme 2 for $u(x,y)=\exp(-(x-0.5)^2-(y-0.5)^2)$ on $[0,1]^2$.

Mesh size	$\|u-u_h\|_h$	$\|u_x-u_{x,h}\|_h$	$\|u_y-u_{y,h}\|_h$	T=CPU(in s.)	$T/(N^2\ln_2(N))$	GMRES it.
$N=128$	$3.478(-10)$	$1.813(-9)$	2.697(-9)	0.11	9.59(-7)	15
conv. rate	4.00	3.83	3.92			
$N=256$	$2.167(-11)$	$1.268(-10)$	1.780(-10)	0.42	8.01(-7)	15
conv. rate	4.00	3.86	3.92			
$N=512$	$1.351(-12)$	$8.721(-12)$	1.171(-11)	1.89	8.01(-7)	15
conv. rate	4.00	3.36	3.62			
$N=1024$	$8.420(-14)$	$8.446(-13)$	9.477(-13)	8.22	7.84(-7)	15

Table 3: Error and convergence rate with the bidimensional HB-Scheme 2 for $u(x,y)=\ln(x+y^2+1)$ on $[0,1]^2$.

Mesh size	$\|u-u_h\|_h$	$\|u_x-u_{x,h}\|_h$	T=CPU(in s.)	$T/(N^2\ln_2(N))$	GMRES it.
$N=512$	$1.163(-7)$	$5.835(-6)$	1.91	8.09(-7)	14
conv. rate	4.00	3.88			
$N=1024$	$7.247(-9)$	$3.967(-7)$	8.65	8.24(-7)	14
conv. rate	4.00	3.90			
$N=2048$	$4.520(-10)$	$2.660(-8)$	37.18	8.06(-7)	14
conv. rate	4.00	3.91			
$N=4096$	$2.821(-11)$	$1.768(-9)$	162.8	8.09(-7)	14

Table 4: Error and convergence rate with the bidimensional HB-Scheme 2 for $u(x,y)=\cos\big(5\pi(x-y)^3\big)$ on $[0,1]^2$.

Mesh size	$\|u-u_h\|_h$	$\|u_x-u_{x,h}\|_h$	T=CPU(in s.)	$T/(N^2\ln_2(N))$	GMRES it.
$N=512$	$9.480(-9)$	$1.283(-7)$	1.66	7.04(-7)	11
conv. rate	4.00	4.00			
$N=1024$	$5.923(-10)$	$8.021(-9)$	7.51	7.16(-7)	11
conv. rate	4.00	4.00			
$N=2048$	$3.702(-11)$	$5.013(-10)$	32.07	6.95(-7)	11
conv. rate	4.00	3.99			
$N=4096$	$2.314(-12)$	$3.156(-11)$	132.7	6.59(-7)	11

Table 5: Error and convergence rate with the bidimensional HB-Scheme 2 for $u(x,y)=\exp\big(-30((x-0.5)^2+(y-0.5)^2)\big)\cos\big(20(x+y-1)\big)$ on $[0,1]^2$.

first diagonal, $u(x,y)=\cos\big(5\pi(x-y)^3\big)$. We display results with grids from $N=512$ to $N=4096$. Again the rate is close to four for the three unknowns $u,\partial_x u,\partial_y u$.

Case 4: We consider in Table 5 a case presented in [11] with the exact function

$$u(x,y)=\exp\big(-30((x-0.5)^2+(y-0.5)^2)\big)\cos\big(20(x+y-1)\big). \tag{5.2}$$

This function is both steep and oscillating and is more difficult than the previous ones to compute. The accuracy obtained on the final grid ($N=4096$) is the same than the one obtained for $N=512$ with the optimal spectral subtractional solver in [11] which uses a decomposition into an explicit part and a computed part. The HB-scheme does not use such a decomposition. In addition, the unknown and the gradient are provided by the scheme.

6 Conclusion

This paper introduces a methodology to design compact finite difference schemes in cartesian geometries. A specific example of interest for the Poisson problem

is worked out. The new scheme appears to be fourth order for the principal unknown and the gradient. The computational cost is found to be $O(C_1 N^2 \log_2(N))$ + $O(C_2 N^2)$. The performances on a desktop are quite good. The scheme does not use any kind of staggered grid. All the unknowns are located at point (x_i, y_j).
The work is going on in several directions including extending the accuracy to order six and eight and the extension to dimension three. In addition, a strategy to generalize this scheme to multiscale grids and to irregular geometries using embedded grids, [18], in underway. We refer to the forthcoming [1] for a detailed numerical analysis.

Acknowledgment: The authors acknowledge the referees for their careful review that helped greatly to enhance this paper. They are also very grateful to Prof. G. Fairweather for his appreciation of this work.

References

[1] A. Abbas. *Schémas compacts hermitiens: algorithmes rapides pour la discrétisation des équations aux dérivées partielles.* PhD thesis, Univ. Paul Verlaine - Metz, to appear.

[2] M. Ben-Artzi, J-P. Croisille, and D. Fishelov. A fast direct solver for the biharmonic problem in a rectangular grid. *SIAM J. Scient. Comp.*, 31(1):303–333, 2008.

[3] M. Ben-Artzi, J-P. Croisille, and D. Fishelov. *Navier-Stokes equations in planar domains.* World Scientific Publishing, to appear, 2011.

[4] B. Bialecki, G. Fairweather, and K.R. Bennett. Fast direct solvers for piecewise Hermite bicubic orthogonal spline collocation equations. *SIAM J. Numer. Anal.*, 29:156–173, 1992.

[5] B. Bialecki, G. Fairweather, and A. Karageorghis. Matrix decomposition algorithms for elliptic boundary value problems: a survey. *Numerical Algorithms*, 2010, to appear.

[6] B. Bialecki, G. Fairweather, and K.A. Remington. Fourier methods for piecewise Hermite bicubic orthogonal spline collocation. *East-West J. Numer. Math*, 2:1–20, 1994.

[7] P. Bjørstad. Fast numerical solution of the biharmonic Dirichlet problem on rectangles. *SIAM J. Numer. Anal.*, 20, No. 1:59–71, 1983.

[8] R.F. Boisvert. Families of high order accurate discretizations of some elliptic problems. *SIAM J. Sci. Stat. Comput.*, 2(3):268–284, 1981.

[9] R.F. Boisvert. A fourth order accurate Fourier method for the Helmholtz equation in three dimensions. *ACM Trans. Math. Soft.*, 13:221–234, 1987.

[10] R.F. Boisvert. Algorithms for special tridiagonal systems. *SIAM J. Sci. Stat. Comput.*, 12(2):423–442, 1991.

[11] E. Braverman, B. Epstein, M. Israeli, and A. Averbuch. A fast spectral subtractional solver for elliptic equations. *J. Sci. Comput.*, 21(1):91–128, 2004.

[12] E. Braverman, M. Israeli, and A. Averbuch. A hierarchical 3-D direct Helmholtz solver by domain decomposition and modified Fourier method. *SIAM J. Sci. Comput.*, 26(5):1504–1524, 2005.

[13] B.L. Buzbee and F.W. Dorr. The direct solution of the biharmonic equation on rectangular regions and the Poisson equation on irregular regions. *SIAM J. Numer. Anal.*, 1974.

[14] B.L. Buzbee, G.H. Golub, and C. W. Nielson. On direct methods for solving Poisson's equations. *SIAM J. Numer. Anal.*, 1970.

[15] L. Collatz. *The Numerical Treatment of Differential Equations*. Springer-Verlag, 3-rd edition, 1960.

[16] G. Coppola and C. Meola. Generalization of the spline interpolation based on the principle of compact schemes. *J. Sci. Comput.*, 17(1-4):695–706, 2002.

[17] J-P. Croisille. A Hermitian Box-Scheme for one-dimensional elliptic equations - Application to problems with high contrasts in the ellipticity. *Computing*, 78:329–353, 2006.

[18] M.D. de Tullio, R. Verzicco, and G. Iaccarino. *Immersed Boundary Techniques for Large-Eddy-Simulation (in Large Eddy Simulation and Related Techniques)*. VKI Lectures Series. Von Karman Institute, 2010 to appear.

[19] L. W. Ehrlich. Solving the biharmonic equations as coupled finite difference equations. *SIAM J. Numer. Anal.*, 8(2):278–287, 1971.

[20] G.E. Forsythe and W.R. Wasow. *Finite Difference Methods for Partial Differential Equations*. Applied Mathematics Series. John Wiley & Sons, 6th edition, 1960.

[21] G.H. Golub, L.C. Huang, H. Simon, and W-P. Tang. A fast Poisson solver for the finite difference solution of the incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.*, 19(5):1606–1624, 1998.

[22] G.H. Golub and C.F. Van Loan. *Matrix computations*. John Hopkins Univ. Press., 1996, 3rd edition.

[23] L. Grasedyck. Existence and computation of low Kronecker-Rank approximations for large linear systems of tensor product structure. *Computing*, 72:247–265, 2004.

[24] M. M. Gupta, J. Kouatchaou, and J. Zhang. Comparison of second order and fourth order scheme discretizations for multigrid Poisson solver. *J. Comput. Phys.*, 132:226–232, 1997.

[25] B. Gustafsson. *High Order Difference Methods for Time Dependent PDE*. Springer-Verlag, 2008.

[26] W. Hackbusch. *Elliptic Differential Equations*, volume 15 of *Springer Series in Comp. Math.* Springer-Verlag, 1992.

[27] W. Hackbusch, B.N. Khoromskij, and E.E. Tyrtyshnikov. Hierarchical Kronecker tensor-product approximations. *Jour. Numer. Math*, 13(2):119–156, 2005.

[28] D.A. Harville. *Matrix algebra from a statistician perspective*. Springer, 2008.

[29] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge Univ. Press, 1996.

[30] H. B. Keller. A new difference scheme for parabolic problems. In *Numerical Solution of Partial Differential Equations, II (SYNSPADE 1970) (Proc. Sympos., Univ. of Maryland, College Park, Md., 1970)*, pages 327–350. Academic Press, New York, 1971.

[31] C. Van Loan. *Computational Frameworks for the Fast Fourier Transform*. SIAM, 1992.

[32] P. Londrillo. Adaptive grid-based gas-dynamics and Poisson solvers for gravitating systems. *Mem. A.A. It. Suppl.*, 4(69):69–74, 2004.

[33] A.R. Mitchell and D.F. Griffiths. *The Finite Difference Method in Partial Differential Equations*. John Wiley & Sons, 1980.

[34] Y. Morinishi, T.S. Lund, O.V. Vasilyev, and P. Moin. Fully conservative higher order finite difference schemes for incompressible flows. *Jour. of Comp. Phys.*, 143:90–124, 1998.

[35] J.R. Rice and R.F. Boisvert. *Solving elliptic problems using ELLPACK*. Springer-Verlag, 1985.

[36] T.K. Sengupta, G. Ganeriwal, and D. De. Analysis of central and upwind compact schemes. *Journal of Computational Physics*, 192:677–694, 2003.

[37] S.E. Sherer and J.N. Scott. High order compact finite difference methods on general overset grids. *Journal of Computational Physics*, 210:459–496, 2005.

[38] K. Shiraishi and T. Matsuoka. Wave propagation simulation using the CIP method of characteristics equations. *Comm. Comput. Physics*, 3(1):121–135, 2008.

[39] P. Swarztrauber. Fast Fourier Transform Algorithms for Vector Computers. *Parallel Computing*, pages 45–63, 1984.

[40] Yin Wang and Jun Zhang. Sixth order compact scheme combined with multigrid method and extrapolation technique for 2d Poisson equation. *Journal of Computational Physics*, 2009.

[41] J. Zhang. An explicit fourth-order compact finite difference scheme for three dimensional convection-diffusion equation. *Comm. Num. Meth*, 14:263–280, 1998.