A FAST DIRECT SOLVER FOR THE BIHARMONIC PROBLEM IN A RECTANGULAR GRID *

MATANIA BEN-ARTZI †, JEAN-PIERRE CROISILLE‡, AND DALIA FISHELOV §

Abstract. We present a fast direct solver methodology for the Dirichlet biharmonic problem in a rectangle. The solver is applicable in the case of the second order Stephenson scheme [34] as well as in the case of a new fourth order scheme, which is discussed in this paper. It is based on the capacitance matrix method ([10], [8]). The discrete biharmonic operator is decomposed into two components. The first is a diagonal operator in the eigenfunction basis of the Laplacian, to which the FFT algorithm is applied. The second is a low rank perturbation operator (given by the capacitance matrix), which is due to the deviation of the discrete operators from diagonal form. The Sherman-Morrison formula [18] is applied to obtain a fast solution of the resulting linear system of equations.

Key words. Fast solver, FFT, biharmonic problem, capacitance matrix method, Sherman-Morrison formula, Navier-Stokes equations, streamfunction formulation, vorticity, compact scheme, driven cavity, Stephenson scheme.

AMS subject classifications. 35J40 - 65F05 - 65F15 - 65M06 - 76D05

1. Introduction.

The accurate resolution of linear fourth order elliptic problems, such as the biharmonic equation, is of great importance in many fields of applied mathematics. Specific examples are the computation of the streamfunction in incompressible fluid dynamics or in problems in elasticity theory. From the numerical point of view, two essential issues are the accuracy of the scheme and the availability of a fast solver.

Finite-difference schemes for the biharmonic problem on rectangles fall broadly into two categories: Coupled and non-coupled schemes. In coupled schemes, the problem is decomposed into two Poisson problems, combined with boundary conditions. Classical works include [32], [33], [14], [28]. When one applies such a scheme to the Dirichlet biharmonic problem, one encounters difficulties in the design of artificial boundary conditions for the first Poisson problem, whereas the second one is overdetermined. In the fluid dynamics context, this question is related to the design of an artificial boundary condition for the vorticity of the flow.

Here we focus on a non-coupled scheme, namely the nine point (second order) Stephenson scheme introduced in [34]. This scheme has been successfully applied to the numerical simulation of the time-dependent Navier-Stokes system (see [17], [4], [3]). Its main advantage, in contrast with the usual thirteen point scheme (see e.g. [24, 12, 10, 8]), is that it allows the construction of a time-dependent solver, which is closely related to the partial differential problem. A crucial feature of the scheme is the absence of artificial boundary conditions on the vorticity. This permits the construction of a compact scheme, which handles near boundary points in the same manner as internal points (see also [20], [19]). The solution of the linear system of discretized equations requires a fast solver.

*This work is partially supported by the High Council for Scientific and Technological Cooperation between France and Israel.

† Institute of Mathematics, The Hebrew University, Jerusalem 91904, Israel, mbartzi@math.huji.ac.il

‡ Department of Mathematics, LMAM, UMR 7122, University Paul Verlaine-Metz, BP 80794, F-57012 Metz, France, jean-pierre.croisille@univ-metz.fr

§ Afeka-Tel-Aviv Academic College of Engineering, 218 Bnei-Efraim St., Tel-Aviv 69107, Israel, daliaf@post.tau.ac.il.

The main goal of this paper is to present a fast solver for compact discretizations of the biharmonic problem. This is done by the *capacitance matrix* formulation. In the first part of the paper we construct the capacitance matrix formulation for the Stephenson scheme on a rectangular grid. We show how it is efficiently applied to the solution of the discrete biharmonic problem.

In the second part of the paper, a fourth order accurate extension of the Stephenson scheme is introduced. The new fourth order discrete biharmonic equation is solved using a modified version of the second order algorithm. The key feature of the scheme consists of simple representation of the capacitance matrix in the eigenfunction basis of the Laplacian. It enables us to efficiently compute the capacitance matrix and to solve the resulting low-rank linear system using a diagonal preconditioned conjugate-gradient method. In addition, the gradient and the Laplacian of the solution of the biharmonic problem are obtained as byproducts of the solver.

To put this paper in the context of existing literature, we make the following remarks.

- A fourth order version of the Stephenson scheme was introduced in [34]. A multigrid solver for this scheme is devised in [1]. In the context of the equation $\Delta^2\psi = f$, it uses the values of f at five points (in the nine point stencil). In contrast, our scheme uses only the value of f at the center point. When f itself is a function of ψ (e.g., $\Delta\psi$), this leads to a significant simplification of the algorithm.
- A scheme based on orthogonal spline collocation has been investigated in [35], [26], [5]. This scheme is fourth order accurate and the linear system is solvable at a cost of $O(N^2 \operatorname{Log}(N))$ operations, where N is the number of grid points in one direction. An almost block diagonal solver, [16], [2], is used as a basis of the fast solver in [5].
- The algorithm presented here runs parallel the lines of the capacitance matrix principle of Golub (appendix of [15]), Buzzbee and Dorr [10](thirteen points scheme, $O(N^3)$ solver), Bjørstad [8](thirteen points scheme, $O(N^2)$ solver), Bjørstad and Tjøstheim [9], and Legendre or Hermite Galerkin scheme of Shen [30], [31], $O(N^3)$ solver. It consists basically of decomposition of the matrix, which represents the scheme, into a sum of a diagonal operator in the eigenfunction basis of the (five point) Laplacian, and of a low-rank perturbation, called the *capacitance matrix* ([11]). Our algorithm is based on two ingredients: (i) application of the Sherman-Morrison formula with a low-rank capacitance matrix, (ii) resolution of the diagonal component in the inversion formula by the FFT method (see [7] for a similar application). We emphasize the fact that our algorithm is a direct (whereas in [21], [22] it is an iterative) solver.

The outline of the paper is as follows. In Section 2 we recall the second order Stephenson scheme, using the notation introduced in [4], [3]. In Section 3, we develop our fast solver for the biharmonic problem, including detailed algebraic derivation for the nine point second order Stephenson operator. Section 4 is devoted to the new fourth order scheme, for which a similar fast algorithm is developed. Finally, in Section 5, we represent numerical results for both the second order and the fourth order schemes. The problems dealt with are mixed biharmonic-Laplacian problems, subject to non-homogeneous Dirichlet boundary conditions. Computing efficiency are reported for several fourth order problems. The computational cost appears to be $O(N^2 \operatorname{Log}(N))$, where N is the number of grid points in each direction. We note that a typical computing time of the solver for a 1024×1024 grid is ten seconds on a PC.

Finally, let us mention the work [29] for a comprehensive history of the biharmonic problem in two dimensions.

2. Notation.

2.1. Finite difference operators. We consider a square $\Omega = (0, L)^2$ with a uniform grid $x_i = ih$, $y_j = jh$, $i, j = 0, ...N$, $h = 1/N$. The internal points are $x_i = ih$, $y_j = jh$, for $1 \le i, j \le N-1$. Denote by $l^2_{h,0}$ the space of one-dimensional discrete functions (u_i) defined on $x_i = ih$, $1 \le i \le N-1$. In two dimensions, $L^2_{h,0}$ is the space of discrete functions $(\psi_{i,j})$ defined on $x_i = ih$, $y_j = jh$, for $1 \le i, j \le N-1$. The spaces $l^2_{h,0}$, $L^2_{h,0}$ are respectively equipped with the scalar products

$$(u, v)_h = h \sum_{i=1}^{N-1} u_i v_i, \ \forall u, v \in l^2_{h,0} \ ; \ (u, v)_h = h^2 \sum_{i,j=1}^{N-1} u_{i,j} v_{i,j}, \ \forall u, v \in L^2_{h,0}. \tag{2.1}$$

For discrete function in $L^2_{h,0}$ we define the following centered operators

$$\begin{cases} \delta_x \psi_{i,j} = \dfrac{\psi_{i+1,j} - \psi_{i-1,j}}{2h} & , \quad \delta_x^2 \psi_{i,j} = \dfrac{\psi_{i+1,j} + \psi_{i-1,j} - 2\psi_{i,j}}{h^2} \\ \delta_y \psi_{i,j} = \dfrac{\psi_{i,j+1} - \psi_{i,j-1}}{2h} & , \quad \delta_y^2 \psi_{i,j} = \dfrac{\psi_{i,j+1} + \psi_{i,j-1} - 2\psi_{i,j}}{h^2}. \end{cases} \tag{2.2}$$

and the mixed discrete derivative operator δ_{xy}

$$\delta_{xy} \psi_{i,j} = \delta_x \delta_y \psi_{i,j} = \frac{\psi_{i+1,j+1} - \psi_{i-1,j+1} - \psi_{i+1,j-1} + \psi_{i-1,j-1}}{4h^2} \quad , \quad 1 \le i, j \le N-1. \tag{2.3}$$

Consider the following fourth order partial differential problem in the square $\Omega = (0, L)^2$.

$$\begin{cases} \big(-a\Delta + b\Delta^2\big)\psi(x, y) = f \quad , \quad (x, y) \in \Omega \\ \psi(x, y) = 0 \quad , \quad \dfrac{\partial \psi}{\partial n}(x, y) = 0 \quad , \quad (x, y) \in \partial\Omega, \end{cases} \tag{2.4}$$

where $a \ge 0$, $b > 0$ are two real constants. Problem (2.4) is approximated by the scheme

$$\begin{cases} \big(-a\Delta_h + b\Delta_h^2\big)\psi_{i,j} = f_{i,j}, & \text{at interior points } (i, j) \\ \psi_{i,j} = 0 \quad , \quad \left(\dfrac{\partial \psi}{\partial n}\right)_{i,j} = 0, & \text{at boundary points } (i, j), \end{cases} \tag{2.5}$$

where

- $\Delta_h \psi_{i,j}$ is the five points discrete Laplacian

$$\Delta_h \psi_{i,j} = \delta_x^2 \psi_{i,j} + \delta_y^2 \psi_{i,j}, \quad 1 \le i, j \le N-1. \tag{2.6}$$

- and Δ_h^2 is the nine points discrete biharmonic (Stephenson approximation), defined by

$$\Delta_h^2 \psi_{i,j} = \delta_x^4 \psi_{i,j} + \delta_y^4 \psi_{i,j} + 2\delta_x^2 \delta_y^2 \psi_{i,j}. \tag{2.7}$$

$\delta_x^2\delta_y^2$ is the nine points operator defined by $\delta_x^2\delta_y^2 = \delta_x^2 \circ \delta_y^2$. The finite difference operators δ_x^4, δ_y^4 (introduced in [4], [3]), approximating $\frac{\partial^4}{\partial x^4}\psi$ and $\frac{\partial^4}{\partial y^4}\psi$ are the one-dimensional Stephenson operators

$$\begin{cases} \delta_x^4\psi = \dfrac{12}{h^2}\left(\delta_x\psi_x - \delta_x^2\psi\right) \\ \delta_y^4\psi = \dfrac{12}{h^2}\left(\delta_y\psi_y - \delta_y^2\psi\right). \end{cases} \tag{2.8}$$

For any $\psi \in L^2_{h,0}$, the *Hermitian gradient* $(\psi_x, \psi_y) \in (L^2_{h,0})^2$ in (2.8) is defined by the following relations between ψ and ψ_x and between ψ and ψ_y.

$$\begin{cases} (I + \dfrac{h^2}{6}\delta_x^2)\psi_{x,i,j} = \delta_x\psi_{i,j} \quad , \quad 1 \le i,j \le N-1 \\ (I + \dfrac{h^2}{6}\delta_y^2)\psi_{y,i,j} = \delta_y\psi_{i,j} \quad , \quad 1 \le i,j \le N-1. \end{cases} \tag{2.9}$$

The latter form fourth order approximation for ψ_x for given values of ψ and fourth order approximations for ψ_y for given values of ψ. The two operators $\delta_x^4\psi$, $\delta_y^4\psi$ are fourth order approximations for $\frac{\partial^4}{\partial x^4}\psi$ and $\frac{\partial^4}{\partial y^4}\psi$ in the Fourier sense (see [3]). Then, the numerical scheme (2.5) may be written in the following way. Find $\psi_{i,j} \in L^2_{h,0}$ such that

$$\left(-a\Delta_h + b\Delta_h^2\right)\psi_{i,j} = f_{i,j}. \tag{2.10}$$

The operator $-a\Delta_h + b\Delta_h^2$ is second order accurate because of the second order accuracy of the mixed term $\delta_x^2\delta_y^2$ in (2.7) and of the five points Laplacian (2.6). However, we will see in Section 4 that the order of accuracy can be easily increased to fourth order by a slight modification of Δ_h^2 and Δ_h, keeping a pointwise source term $f_{i,j}$.

2.2. Matrix operators. We relate the bidimensional finite difference operators acting in $L^2_{h,0}$ with matrix operators of size $(N-1)\times(N-1)$ $(N \ge 2)$, acting on a vector $u_{i,j} \in L^2_{h,0}$. Most of those operators are obtained as Kronecker products of $(N-1)\times(N-1)$ matrices.

We use the indexing of $u_{i,j} \in L^2_{h,0}$ as the column vector

$$U = \Big[u_{1,1}, ...u_{1,N-1}; u_{2,1}, ...u_{2,N-1}; ...; u_{N-1,1}, ...u_{N-1,N-1}\Big]^T \in \mathbb{R}^{(N-1)^2} \tag{2.11}$$

The bottom ordering of vector $U \in \mathbb{R}^{(N-1)^2}$ is obtained by letting the index j vary first. The notation "vect" stands indifferently for the operator which transforms any one-dimensional discrete function $u_i \in l^2_{h,0}$ to the vector $U = [u_1, .., u_{N-1}]^T \in \mathbb{R}^{N-1}$ or any two-dimensional discrete function $u_{i,j} \in L^2_{h,0}$ to the vector U defined by (2.11) (see [25]). [1] Recall that the Kronecker product of the matrices $A \in \mathbb{M}_{m,n}$ and $B \in \mathbb{M}_{p,q}$ is the matrix $A \otimes B \in \mathbb{M}_{mp,nq}$ defined by

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & ... & a_{1,n}B \\ ... & & & \\ ... & & & \\ a_{m,1}B & a_{(N-1),2}B & ... & a_{m,n}B \end{bmatrix}. \tag{2.12}$$

[1] Actually , in [25], the ordering of U is by lines.

Let us define several $(N-1)\times(N-1)$ matrices that will be useful for the representation of our fast algorithm.

- *Centered one-dimensional Laplacian matrix*
  The symmetric positive definite matrix T is the standard tridiagonal matrix related to the 3 points Laplacian

$$T_{i,m} = \begin{cases} 2, & m = i \\ -1, & |m-i| = 1 \\ 0, & |m-i| \geq 2 \end{cases} \quad T = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}. \tag{2.13}$$

- The symmetric positive definite matrix P is deduced from T by

$$P = 6I - T, \tag{2.14}$$

  or equivalently by

$$P_{i,m} = \begin{cases} 4, & m = i \\ 1, & |m-i| = 1 \\ 0, & |m-i| \geq 2 \end{cases} \quad , \quad P = \begin{bmatrix} 4 & 1 & 0 & \dots & 0 \\ 1 & 4 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & 1 & 4 & 1 \\ 0 & \dots & 0 & 1 & 4 \end{bmatrix}. \tag{2.15}$$

  We also define $\bar{P} = I - \frac{1}{6}T$.

- The antisymmetric matrix $K = (K_{i,m})_{1\leq i,m\leq N-1}$ is given by

$$K_{i,m} = \begin{cases} \operatorname{sgn}(m-i), & |m-i| = 1 \\ 0, & |m-i| \neq 1. \end{cases} \quad , \quad K = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & -1 & 0 & 1 \\ 0 & \dots & 0 & -1 & 0 \end{bmatrix}. \tag{2.16}$$

- For all $1 \leq i \leq N-1$, $e_i \in \mathbb{R}^{N-1}$ is the column vector of the canonical basis of $\mathbb{R}^{N-1}$. The $(N-1)\times(N-1)$ matrix E_i is defined by

$$E_i = e_i e_i^T,\ 1 \leq i \leq N-1 \tag{2.17}$$

  For all $u \in l^2_{h,0}$ and $U = \text{vect}(u) \in \mathbb{R}^{N-1}$,

$$(E_i U)_j = \delta_{i,j} U_j,\ 1 \leq i, j \leq N-1 \tag{2.18}$$

with $\delta_{i,j}$ the Kronecker symbol. If $U = \text{vect}(u), U_x = \text{vect}(u_x) \in \mathbb{R}^{N-1}$ then the following identities hold.

- *One-dimensional second order derivative* δ_x^2

$$-\text{vect}(\delta_x^2 u) = \frac{1}{h^2} TU. \tag{2.19}$$

- *One-dimensional centered gradient* δ_x

$$\text{vect}(\delta_x u) = \frac{1}{2h} KU. \tag{2.20}$$

- *One-dimensional hermitian gradient* u_x

$$\mathrm{vect}(u_x) = \frac{3}{h} P^{-1} K U. \tag{2.21}$$

- *One-dimensional Stephenson operator*

$$\mathrm{vect}(\delta_x^4 u) = \frac{12}{h^2} \left[\frac{3}{2h^2} K P^{-1} K + \frac{1}{h^2} T \right] U = \frac{6}{h^4} \left[3 K P^{-1} K + 2T \right] U. \tag{2.22}$$

Remark: The isomorphism operator "vect" which maps the discrete function $u \in L^2_{h,0}$ to the vector $U = \mathrm{vect}(u) \in \mathbb{R}^{(N-1)^2}$, induces a natural isomorphism of space operators. Let us denote by $< L, u >$ the resulting discrete function obtained by the operation of a x linear operator L on a discrete function u, We associate to L an operator $\tilde{L}$, which acts on a vector U, such that

$$< L, u > = < \tilde{L}, U > . \tag{2.23}$$

To simplify notation, from now on we identify the operators L and $\tilde{L}$ in cases where this notation does not lead to any confusion. For example (2.19) will be written

$$-\delta_x^2 = \frac{1}{h^2} T. \tag{2.24}$$

Now we apply the same identification between operators in the bidimensional framework. We identify the discrete functions $u \in L^2_{h,0}$ and the vector $U \in \mathbb{R}^{(N-1)^2}$. The following bidimensional operators are expressed as Kronecker product operators:

- *Bidimensional second order derivation operators* δ_x^2, δ_y^2

$$-\delta_x^2 = \frac{1}{h^2} T \otimes I \quad , \quad -\delta_y^2 = \frac{1}{h^2} I \otimes T. \tag{2.25}$$

- *Bidimensional first order derivation operators* δ_x, δ_y

$$\delta_x = \frac{1}{2h} K \otimes I \quad , \quad \delta_y = \frac{1}{2h} I \otimes K. \tag{2.26}$$

- *Bidimensional Hermitian gradient*

$$U_x = \frac{3}{h} \left[P^{-1} K \otimes I \right] U. \qquad U_y = \frac{3}{h} \left[I \otimes P^{-1} K \right] U. \tag{2.27}$$

- *Mixed derivative* $\delta_x^2 \delta_y^2$

$$\delta_x^2 \delta_y^2 = \frac{1}{h^4} T \otimes T. \tag{2.28}$$

- *Fourth order derivation operators in two dimensions*

$$\begin{cases} \delta_x^4 = \dfrac{12}{h^2} \left[\dfrac{3}{2h^2} K P^{-1} K + \dfrac{1}{h^2} T \right] \otimes I \\ \delta_y^4 = \dfrac{12}{h^2} I \otimes \left[\dfrac{3}{2h^2} K P^{-1} K + \dfrac{1}{h^2} T \right]. \end{cases} \tag{2.29}$$

The following summarizes several preliminary results about matrices K,P,T

LEMMA 2.1. *(i) The commutator of P and K is*

$$[P,K] = PK - KP = 2\left(E_{N-1} - E_1\right). \tag{2.30}$$

(ii) The commutator of P^{-1} and K is

$$[P^{-1},K] = P^{-1}K - KP^{-1} = -2P^{-1}\big(E_{N-1} - E_1\big)P^{-1}. \tag{2.31}$$

(iii) The symmetric matrix K^2 is related to T by

$$K^2 = T^2 - 4T + 2(E_1 + E_{N-1}). \tag{2.32}$$

Proof. *(i)* is easily verified and *(ii)* is deduced from *(i)* by conjugasion by P^{-1}. *(iii)*: The following identities are simply verified.

$$\begin{cases} (K^2U)_i = U_{i+2} + U_{i-2} - 2U_i \quad , \quad 2 \le i \le N-2 \\ (K^2U)_1 = U_3 - U_1 \\ (K^2U)_{N-1} = -U_{N-1} + U_{N-3} \end{cases} \tag{2.33}$$

$$\begin{cases} (T^2U)_i = U_{i+2} - 4U_{i+1} + 6U_i - 4U_{i-1} + U_{i-2} \quad , \quad 2 \le i \le N-2 \\ (T^2U)_1 = -(TU)_2 + 2(TU)_1 = U_3 - 4U_2 + 5U_1 \\ (T^2U)_{N-1} = U_{N-3} - 4U_{N-2} + 5U_{N-1}. \end{cases} \tag{2.34}$$

Therefore,

$$\begin{cases} \big((K^2 - T^2)U\big)_i = 4U_{i+1} - 8U_i + 4U_{i-1} = -4(TU)_i \quad , \quad 2 \le i \le N-2 \\ \big((K^2 - T^2)U\big)_1 = -4(TU)_1 + 2U_1 \\ \big((K^2 - T^2)U\big)_{N-1} = -4(TU)_{N-1} + 2U_{N-1}, \end{cases} \tag{2.35}$$

which gives (2.32). ∎

3. A fast FFT solver for the Stephenson biharmonic.

3.1. The FFT Poisson solver. We recall here briefly the standard FFT algorithm for the discrete Laplacian according to [6], [27]. Consider the Poisson problem in the square $\Omega = (0,L)^2$

$$\begin{cases} -\Delta u(x,y) = f, \qquad (x,y) \in \Omega = (0,L)^2 \\ u(x,y) = 0, \qquad (x,y) \in \partial\Omega, \end{cases} \tag{3.1}$$

Its discrete form is, (see (2.6)),

$$\begin{cases} -\Delta_h u_{i,j} = f_{i,j} \quad , \quad 1 \le i,j \le N-1 \\ u_{i,j} = 0 \quad , \quad i \in \{0,N\} \text{ or } j \in \{0,N\}. \end{cases} \tag{3.2}$$

The eigenvectors of $-\delta_x^2$, which form a basis of $l^2_{h,0}$, are $z^k \in l^2_{h,0}$ defined by, (L is the length of the interval).

$$z_j^k = \left(\frac{2}{L}\right)^{1/2} \sin\frac{kj\pi h}{L} \quad , \quad 1 \le k,j \le N-1, \tag{3.3}$$

They form an orthonormal basis for the one-dimensional scalar product $(.,.)_h$

$$(z^k, z^l)_h = \delta_{k,l}, \quad 1 \le k, l \le N-1. \tag{3.4}$$

Cast in vector form, we introduce the column vector $Z^k \in \mathbb{R}^{N-1}$ and row vector $Z_j \in \mathbb{R}^{N-1}$ defined by

$$Z^k = h^{1/2} z^k \quad , \quad Z_j = h^{1/2} z_j, \quad 1 \le k, j \le N-1 \tag{3.5}$$

$$Z_j^k = \left(\frac{2}{N}\right)^{1/2} \sin \frac{kj\pi}{N} \quad , \quad 1 \le k, j \le N-1, \tag{3.6}$$

The matrix $Z \in \mathbb{M}_{N-1}(\mathbb{R})$ whose $k-$th column is Z^k and $j-$th row is Z_j is a symmetric positive definite unitary matrix, thus

$$Z^2 = ZZ^T = I_{N-1}. \tag{3.7}$$

The eigenvalues of the matrix T are given by, (see (2.19),

$$\lambda_k = 4 \sin^2 \left(\frac{k\pi}{2N}\right). \tag{3.8}$$

In matrix form, the scheme (3.2) reduces to the linear system with right-hand side $F = h^2 \operatorname{vect}(f) \in \mathbb{R}^{(N-1)^2}$ and unknown $U = \operatorname{vect}(u) \in \mathbb{R}^{(N-1)^2}$

$$(T \otimes I + I \otimes T) U = F. \tag{3.9}$$

The orthonormal basis (in $\mathbb{R}^{(N-1)^2}$ of $T \otimes I + I \otimes T$ is $(Z^k \otimes Z^l)_{1 \le k,l \le N-1}$, with eigenvalues $(\lambda_k + \lambda_l)_{1 \le k,l \le N-1}$.

The algorithm of the fast Poisson solver is in 3 steps (see [27, 6] for more details).

Algorithm 1 (Fast FFT Poisson solver).

- **Step 1:**
  Decompose the source term $F = h^2 \operatorname{vect}(f)$ on the orthonormal basis $Z^k \otimes Z^l$. This step consists of computing the coefficients $F_{k,l}^Z = (F, Z^k \otimes Z^l)$ and is performed by FFT (actually the fast sine transform).
- **Step 2 :**
  Solve system (3.9) in the Fourier space by

$$u_{k,l}^Z = \frac{F_{k,l}^Z}{\lambda_k + \lambda_l} \tag{3.10}$$

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

$$U_{i,j} = \sum_{k,l=1}^{N-1} U_{k,l}^Z Z_i^k Z_j^l. \tag{3.11}$$

The grid function $u \in L_{h,0}^2$ is such that $U = \operatorname{vect}(u)$, therefore

$$u_{i,j} = U_{i,j}, \quad 1 \le i, j \le N-1 \tag{3.12}$$

Steps 1 and 3 are $O(N^2 \operatorname{Log}(N))$, and Step 2 is $O(N^2)$, which gives a $O(N^2 \operatorname{Log}(N))$ algorithm. For the complexity analysis of the FFT, we refer to [25] or [23].

3.2. The Stephenson operator. Let us begin by representing the finite difference operator δ_x^4 (2.8) in matrix form.

LEMMA 3.1. *The operator $P\delta_x^4 u$ has the following matrix form*

$$P\delta_x^4 = \frac{6}{h^4}T^2 + \frac{36}{h^4}\big[e_1(e_1 + KP^{-1}e_1)^T + e_{N-1}(e_{N-1} - KP^{-1}e_{N-1})^T\big]. \qquad (3.13)$$

Observe that $P\delta_x^4 u$ is not symmetric.

Proof. We use systematically that for all column vectors $u, v \in \mathbb{R}^{N-1}$, then $u^T, v^T \in \mathbb{R}^{N-1}$ are row vectors. For A a $(N-1)\times(N-1)$ matrix, the following relation holds

$$u(v^T A) = (uv^T)A = u(A^T v)^T \qquad (3.14)$$

The finite difference operator δ_x^4 reads in matrix form (see (2.22))

$$\delta_x^4 = \frac{6}{h^4}\big[3KP^{-1}K + 2T\big]. \qquad (3.15)$$

Multiplying on the left by P gives, using (2.30),

$$\begin{aligned} P\delta_x^4 &= \frac{6}{h^4}\Big[3PKP^{-1}K + 2PT\Big] \\ &= \frac{6}{h^4}\Big[3[P,K]P^{-1}K + 3KPP^{-1}K + 2PT\Big] \\ &= \frac{6}{h^4}\Big[6(E_{N-1} - E_1)P^{-1}K + 3K^2 + 2(6I - T)T\Big]. \end{aligned}$$

Using the expression of K^2 as a function of T, (see (2.32)) and $(P^{-1})^T = P^{-1}$, $K^T = -K$,

$$\begin{aligned} P\delta_x^4 = \frac{6}{h^4}\Big[&6e_{N-1}e_{N-1}^T P^{-1}K - 6e_1 e_1^T P^{-1}K \\ &+ 3(T^2 - 4T + 2e_1e_1^T + 2e_{N-1}e_{N-1}^T) + 12T - 2T^2\Big] \\ = \frac{6}{h^4}\Big[&-6e_{N-1}e_{N-1}^T(P^{-1})^T K^T + 6e_1e_1^T(P^{-1})^T K^T + T^2 + 6e_1e_1^T + 6e_{N-1}e_{N-1}^T)\Big] \\ = \frac{6}{h^4}&T^2 + \frac{36}{h^4}\Big[e_1(e_1 + KP^{-1}e_1)^T + e_{N-1}(e_{N-1} - KP^{-1}e_{N-1})^T\Big], \end{aligned}$$

which is the result. ∎

LEMMA 3.2. *The symmetric positive definite operator δ_x^4 (see (2.22) has the alternative matrix form*

$$\delta_x^4 = \frac{6}{h^4}P^{-1}T^2 + \frac{36}{h^4}\left(v_1v_1^T + v_2v_2^T\right), \qquad (3.16)$$

where the vectors v_1, v_2 are

$$\begin{cases} v_1 = (\alpha-\beta)^{1/2}P^{-1}\big[\frac{\sqrt{2}}{2}e_1 - \frac{\sqrt{2}}{2}e_{N-1}\big] \\ v_2 = (\alpha+\beta)^{1/2}P^{-1}\big[\frac{\sqrt{2}}{2}e_1 + \frac{\sqrt{2}}{2}e_{N-1}\big]. \end{cases} \qquad (3.17)$$

The matrix P is given in (2.14), and the constants α, β are

$$\begin{cases} \alpha = 2(2 - e_1^T P^{-1} e_1) \\ \beta = 2e_{N-1}^T P^{-1} e_1. \end{cases} \tag{3.18}$$

Proof. Applying P^{-1} to both sides of (3.13), we obtain

$$\begin{aligned} \delta_x^4 &= P^{-1}(P\delta_x^4) \\ &= \frac{6}{h^4}P^{-1}T^2 + \frac{36}{h^4}\Big\{P^{-1}e_1(e_1 + KP^{-1}e_1)^T + P^{-1}e_{N-1}(e_{N-1} - KP^{-1}e_{N-1})^T\Big\}. \end{aligned}$$

Therefore, the term in braces, referred as (I),

$$(I) = P^{-1}e_1(e_1 + KP^{-1}e_1)^T + P^{-1}e_{N-1}(e_{N-1} - KP^{-1}e_{N-1})^T \tag{3.19}$$

is expanded as

$$\begin{aligned} (I) &= P^{-1}e_1e_1^T - P^{-1}e_1e_1^TP^{-1}K + P^{-1}e_{N-1}e_{N-1}^T + P^{-1}e_{N-1}e_{N-1}^TP^{-1}K \\ &= P^{-1}e_1e_1^T - P^{-1}e_1e_1^TKP^{-1} + P^{-1}e_1e_1^T[K, P^{-1}] + P^{-1}e_{N-1}e_{N-1}^T \\ &+ P^{-1}e_{N-1}e_{N-1}^TKP^{-1} + P^{-1}e_{N-1}e_{N-1}^T[P^{-1}, K]. \end{aligned}$$

Using the value of the commutator (2.31)

$$[K, P^{-1}] = -2P^{-1}(E_1 - E_{N-1})P^{-1} = -2P^{-1}\left(e_1e_1^T - e_{N-1}e_{N-1}^T\right)P^{-1} = -[P^{-1}, K],$$

we obtain that (I) is the conjugate of (II) by P^{-1}, i.e.,

$$(I) = P^{-1}(II)P^{-1}, \tag{3.20}$$

with (II) defined as

$$\begin{aligned} (II) &= e_1e_1^TP - e_1e_1^TK - 2e_1e_1^TP^{-1}e_1e_1^T + 2e_1e_1^TP^{-1}e_{N-1}e_{N-1}^T \\ &+ e_{N-1}e_{N-1}^TP + e_{N-1}e_{N-1}^TK + 2e_{N-1}e_{N-1}^TP^{-1}e_1e_1^T - 2e_{N-1}e_{N-1}^TP^{-1}e_{N-1}e_{N-1}^T. \end{aligned}$$

Therefore, (I) rewrites

$$(I) = P^{-1}\big\{(S) + (S')\big\}P^{-1}, \tag{3.21}$$

where (S) and (S') are the matrices defined by

$$\begin{aligned} (S) &= -2e_1e_1^TP^{-1}e_1e_1^T - 2e_{N-1}e_{N-1}^TP^{-1}e_{N-1}e_{N-1}^T \\ &\quad + 2\left(e_1e_1^TP^{-1}e_{N-1}e_{N-1}^T + e_{N-1}e_{N-1}^TP^{-1}e_1e_1^T\right) \\ (S') &= e_1[(P+K)e_1]^T + e_{N-1}[(P-K)e_{N-1}]^T. \end{aligned}$$

The matrix (S) is clearly symmetric. In addition, we verify easily that

$$\begin{cases} (P+K)e_1 = 4e_1 \\ (P-K)e_{N-1} = 4e_{N-1}. \end{cases} \tag{3.22}$$

Therefore, the matrix (S') reduces to

$$(S') = 4e_1e_1^T + 4e_{N-1}e_{N-1}^T \tag{3.23}$$

and is as well symmetric. We deduce from (3.21) that (I) can be written as

$$(I) = P^{-1}\big[-2e_1e_1^TP^{-1}e_1e_1^T - 2e_{N-1}e_{N-1}^TP^{-1}e_{N-1}e_{N-1}^T \\ + 2e_1e_1^TP^{-1}e_{N-1}e_{N-1}^T + 2e_{N-1}e_{N-1}^TP^{-1}e_1e_1^T + 4e_1e_1^T + 4e_{N-1}e_{N-1}^T\big]P^{-1},$$

or

$$(I) = P^{-1}\Big[e_1, e_{N-1}\Big]\begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix}\begin{bmatrix} e_1^T \\ e_{N-1}^T \end{bmatrix}P^{-1}, \tag{3.24}$$

with

$$\begin{cases} \alpha = 4 - 2e_1^TP^{-1}e_1 = 4 - 2e_{N-1}^TP^{-1}e_{N-1} \\ \beta = 2e_1^TP^{-1}e_{N-1} = 2e_{N-1}^TP^{-1}e_1. \end{cases} \tag{3.25}$$

Using, in (3.24), that

$$\begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{-\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}\begin{bmatrix} \alpha-\beta & 0 \\ 0 & \alpha+\beta \end{bmatrix}\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}, \tag{3.26}$$

we obtain that (I) is the symmetric matrix

$$(I) = \Big[v_1, v_2\Big]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}, \tag{3.27}$$

where the vectors v_1, v_2 are

$$\begin{cases} v_1 = (\alpha-\beta)^{1/2}P^{-1}\left(\frac{\sqrt{2}}{2}e_1 - \frac{\sqrt{2}}{2}e_{N-1}\right) \\ v_2 = (\alpha+\beta)^{1/2}P^{-1}\left(\frac{\sqrt{2}}{2}e_1 + \frac{\sqrt{2}}{2}e_{N-1}\right). \end{cases} \tag{3.28}$$

which gives (3.16). ∎

Cast in the matrix framework, formula (3.16) is a decomposition of the one-dimensional Stephenson biharmonic operator δ_x^4 in two parts,

$$A = h^4\delta_x^4 = \underbrace{6P^{-1}T^2}_{B} + \underbrace{36[v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}}_{C}. \tag{3.29}$$

We note that $B \in \mathrm{Span}(T)$, since $P = 6I - T$ and that C is a rank 2 matrix. We can therefore use the *Sherman-Morrison* formula.

Formula 3.1 (Sherman-Morrison, [18], Chap.2, p. 50). *Suppose that $A, B \in \mathbb{M}_N(\mathbb{R})$ are two invertible matrices such that*

$$A = B + RS^T, \tag{3.30}$$

with $R, S \in \mathbb{M}_{N,n}(\mathbb{R})$, $n \leq N$, then the inverse of the matrix A can be written as

$$A^{-1} = B^{-1} - B^{-1}R(I + S^TB^{-1}R)^{-1}S^TB^{-1}. \tag{3.31}$$

provided that the matrix $I + S^TB^{-1}R \in \mathbb{M}_n(\mathbb{R})$ be invertible. When $n << N$, the matrix A is a low-rank perturbation of the matrix B. Hence, in the case that B is easily invertible, (3.31) provides an efficient way to invert A. In the following section (3.31) is used to solve the biharmonic problem in a rectangle.

3.3. Solution procedure. We now turn to the study of the discrete differential operators in the two-dimensional setting.

PROPOSITION 3.3. *The Stephenson discrete biharmonic operator Δ_h^2 can be expressed as (see the Remark after (2.22)).*

$$\Delta_h^2 = \frac{1}{h^4}\left[6P^{-1}T^2 \otimes I + 6I \otimes P^{-1}T^2 + 2T \otimes T\right] \\ + \frac{36}{h^4}[v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \otimes I_{N-1} + \frac{36}{h^4} I_{N-1} \otimes [v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}. \tag{3.32}$$

Proof. This is a simple consequence of the definition (2.7) of Δ_h^2 and of (2.29,3.16). Recall that x-operators act as left factors in Kronecker products, and that y-factor operate as right factors. The term $2T \otimes T$ corresponds to the mixed derivative $\delta_x^2 \delta_y^2$. ■

We decompose now the bidimensional discrete operator $h^4\Delta_h^2$ according to (3.32) to a diagonal part (with respect to the basis $Z^k \otimes Z^l$, (3.3)), and a perturbation part, which will turn out to be lower dimensional. We therefore write

$$\mathcal{A} = \mathcal{B} + \mathcal{C}, \tag{3.33}$$

where the matrices $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are specified as

- $\mathcal{A}$ is the matrix corresponding to $h^4\Delta_h^2$ in (3.32).
- $\mathcal{B}$ is the $(N-1)^2 \times (N-1)^2$ matrix

$$\mathcal{B} = 6P^{-1}T^2 \otimes I_{N-1} + 6I_{N-1} \otimes P^{-1}T^2 + 2T \otimes T. \tag{3.34}$$

  $\mathcal{B}$ is diagonal in the basis $Z^k \otimes Z^l$ and will be referred for convenience as the "diagonal" part of matrix $\mathcal{A}$.
- $\mathcal{C}$ is the $(N-1)^2 \times (N-1)^2$ matrix, (see (3.28) for the definition of v_1, v_2),

$$\mathcal{C} = 36\left([v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \otimes I_{N-1} + I_{N-1} \otimes [v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}\right). \tag{3.35}$$

Denoting by Z^k, Z_l respectively the column and line vectors of the unitary matrix (3.5), we replace in (3.35) the identity matrix I_{N-1} by

$$I_{N-1} = ZZ^T = \left[Z^1, Z^2, .., Z^{N-1}\right]\begin{bmatrix} Z_1 \\ \cdot \\ \cdot \\ Z_{N-1} \end{bmatrix}. \tag{3.36}$$

The interest of the decomposition of the identity operator (3.36), instead of the trivial one, will appear in the resolution of the capacitance system, see Appendix A.

The matrix in braces in (3.35) is therefore

$$\underbrace{[v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \otimes \left[Z^1, Z^2, .., Z^{N-1}\right]\begin{bmatrix} Z_1 \\ \cdot \\ \cdot \\ Z_{N-1} \end{bmatrix}}_{(a)} + \underbrace{\left[Z^1, Z^2, .., Z^{N-1}\right]\begin{bmatrix} Z_1 \\ \cdot \\ \cdot \\ Z_{N-1} \end{bmatrix} \otimes [v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}}_{(b)}. \tag{3.37}$$

At this point we use several rules of the Kronecker product alegebra, (see [25]), namely
(i) For all matrices A, B, C, D,

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{3.38}$$

assuming that the ordinary products AC and BD are defined.
(ii) For all matrices A, B,

$$(A \otimes B)^T = A^T \otimes B^T \tag{3.39}$$

Applying (3.38), (3.39), and the definition of the Kronecker product (2.12), the term (a) in (3.37) can be rewritten as

$$(a) = \left([v_1, v_2] \otimes \left[Z^1, Z^2, .., Z^{N-1}\right]\right) \left(\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \otimes \begin{bmatrix} Z_1 \\ . \\ . \\ Z_{N-1} \end{bmatrix} \right)$$

$$= \left[v_1 \otimes Z^1, .., v_1 \otimes Z^{N-1}\right] \begin{bmatrix} v_1^T \otimes Z_1 \\ . \\ . \\ v_1^T \otimes Z_{N-1} \end{bmatrix} + \left[v_2 \otimes Z^1, .., v_2 \otimes Z^{N-1}\right] \begin{bmatrix} v_2^T \otimes Z_1 \\ . \\ . \\ v_2 \otimes Z_{N-1} \end{bmatrix}$$

Combining this with similar calculations applied to term (b) in (3.37), it turns out that the matrix $\mathcal{C}$ in (3.35) can be expressed as a low-rank $(N-1)^2 \times (N-1)^2$ matrix in the form

$$\mathcal{C} = 36\mathcal{R}\mathcal{R}^T, \tag{3.40}$$

where $\mathcal{R}$ is a matrix $(N-1)^2 \times 4(N-1)$, written in the form

$$\mathcal{R} = [R_1, R_2, R_3, R_4]\,. \tag{3.41}$$

In the sequel, we refer to $\mathcal{C}$ as the “perturbation” (or *capacitance*) part of $\bar{\mathcal{A}}$.

The four $(N-1)^2 \times (N-1)$ matrices R_k are

$$\begin{cases} R_1 = [v_1 \otimes Z^1, v_1 \otimes Z^2, ..., v_1 \otimes Z^{N-1}] \\ R_2 = [v_2 \otimes Z^1, v_2 \otimes Z^2, ..., v_2 \otimes Z^{N-1}] \\ R_3 = [Z^1 \otimes v_1, Z^2 \otimes v_1, ..., Z^{N-1} \otimes v_1] \\ R_4 = [Z^1 \otimes v_2, Z^2 \otimes v_2, ..., Z^{N-1} \otimes v_2]. \end{cases} \tag{3.42}$$

The interest of this decomposition (instead of the one using the canonical factorization of the identity operator), will appear clearly in Subsection A. Applying the Sherman-Morrison formula to the matrix

$$\mathcal{A} = \mathcal{B} + 36\mathcal{R}\mathcal{R}^T, \tag{3.43}$$

allows to express $\mathcal{A}^{-1}$ as

$$\mathcal{A}^{-1} = \mathcal{B}^{-1} - 36\mathcal{B}^{-1}\mathcal{R}\left[I_{4(N-1)} + 36\mathcal{R}^T\mathcal{B}^{-1}\mathcal{R}\right]^{-1}\mathcal{R}^T\mathcal{B}^{-1}. \tag{3.44}$$

We use now (3.44) to solve the system

$$\mathcal{A}U = F \tag{3.45}$$

with $\mathcal{A} = h^4\Delta_h^2$, $F = h^4\,\mathrm{vect}(f)$.

The following algorithm summarizes the solution procedure according to formula (3.31). Indications of the computing complexity are given at each step of the algorithm.

Algorithm 2 (Fast FFT algorithm for the biharmonic problem).

- **Step1: Solve $\mathcal{B}g = F$.** *Let $F \in \mathbb{R}^{(N-1)^2}$ be the source-term vector $F = h^4\,\mathrm{vect}(f)$. The linear system for $g \in \mathbb{R}^{(N-1)^2}$*

$$\mathcal{B}g = F, \tag{3.46}$$

is solved using the FFT transform as follows. First F is decomposed on the basis $Z^k \otimes Z^l$ where the vector Z^k, $k = 1, .., N-1$ are the eigenfunctions (3.5) of matrix T, as

$$F = \sum_{k,l=1}^{N-1} F^Z_{k,l} Z^k \otimes Z^l, \tag{3.47}$$

where the coefficients $F^Z_{k,l}$ are

$$F^Z_{k,l} = (F, Z^k \otimes Z^l) \quad , \quad 1 \le k, l \le N-1. \tag{3.48}$$

(3.48) is computed by FFT. The eigenvalues of $\mathcal{B}$ are

$$\mu_{k,l} = \frac{\lambda_k^2}{(1-\lambda_k/6)} + \frac{\lambda_l^2}{(1-\lambda_l/6)} + 2\lambda_k\lambda_l, \tag{3.49}$$

and $g = \mathcal{B}^{-1}F$ is given by

$$g = \sum_{k,l=1}^{N-1} \frac{F^Z_{k,l}}{\mu_{k,l}} Z^k \otimes Z^l. \tag{3.50}$$

The vector $g \in \mathbb{R}^{(N-1)^2}$ is stored for Step 7. The FFT computation (3.48) is $O(N^2\,\mathrm{Log}(N))$ and (3.50) is $O(N^2)$.

- **Step2:**
Compute the vector $\mathcal{R}^T g \in \mathbb{R}^{4(N-1)}$,

$$\mathcal{R}^T g = \begin{bmatrix} R_1^T g \\ R_2^T g \\ R_3^T g \\ R_4^T g \end{bmatrix}. \tag{3.51}$$

We have for example

$$R_1^T g = \begin{bmatrix} (v_1 \otimes Z^1)^T g \\ \cdot \\ \cdot \\ (v_1 \otimes Z^{N-1})^T g \end{bmatrix}. \tag{3.52}$$

The $l-$ component of vector $R_1^T g$ in (3.52) is

$$(v_1 \otimes Z^l)^T g = \sum_{i=1}^{N-1} (v_1)_i \sum_{j=1}^{N-1} g_{i,j} Z^j_l. \tag{3.53}$$

Each term

$$\sum_{j=1}^{N-1} g_{i,j} Z_l^j, \quad 1 \leq i \leq N-1 \tag{3.54}$$

is computed by FFT (actually the fast sine transform) via

$$\sum_{j=1}^{N-1} g_{i,j} Z_l^j = \left(\frac{2}{N}\right)^{1/2} \sum_{j=1}^{N-1} g_{i,j} \sin\frac{lj\pi}{N}. \tag{3.55}$$

Similarly, the k-th component in $R_3^T g$ is

$$(R_3^T g)_k = (Z^k \otimes v_1)^T g = \sum_{j=1}^{N-1} (v_1)_j \sum_{i=1}^{N-1} g_{i,j} Z_k^i. \tag{3.56}$$

The FFT is used to compute

$$\sum_{i=1}^{N-1} g_{i,j} Z_k^i = \left(\frac{2}{N}\right)^{1/2} \sum_{i=1}^{N-1} g_{i,j} \sin\frac{ik\pi}{N}. \tag{3.57}$$

The expressions of $R_2^T g$ and $R_4^T g$ are similar, replacing v_1 by v_2. As for the counting complexity, of Step 2, (3.54) is $O(N\,\mathrm{Log}(N))$ for each value of i, which gives $O(N^2\,\mathrm{Log}(N))$ in all. Then (3.52) is $O(N^2)$ using (3.53). The same is true for each of the four components of $\mathcal{R}^T g$ in (3.51).

- **Step 3:**
  Assemble the $4(N-1) \times 4(N-1)$ capacitance matrix matrix in brackets in formula (3.44)

$$I_{4(N-1)} + 36\mathcal{R}^T\mathcal{B}^{-1}\mathcal{R}. \tag{3.58}$$

We refer to the Appendix A for the detailed structure of the symmetric matrix (3.58), as well as for the $O(N^2)$ computing complexity of its assembling. Note that matrix (3.58) is computed once for all.

- **Step 4:**
  Solve the $4(N-1) \times 4(N-1)$ linear system

$$\left(I_{4(N-1)} + 36\mathcal{R}^T\mathcal{B}^{-1}\mathcal{R}\right) s = \mathcal{R}^T g. \tag{3.59}$$

The computing complexity of the whole algorithm relies on the efficiency of this solving. It is performed by the preconditionned conjugate gradient method. Numerical evidence displays a $O(N^2\,\mathrm{Log}(N))$ computing cost. We refer to Appendix A for more details. The solution $s \in \mathbb{R}^{4(N-1)}$ is decomposed in

$$s = [s_1, s_2, s_3, s_4]^T, \quad s_1, s_2, s_3, s_4 \in \mathbb{R}^{N-1} \tag{3.60}$$

- **Step 5:**
  Perform the product of $t = \mathcal{R}s$, $s \in \mathbb{R}^{4(N-1)}$, $t \in \mathbb{R}^{(N-1)^2}$.

$$t = t_1 + t_2 + t_3 + t_4, \tag{3.61}$$

with

$$\begin{cases} t_1 = \left(v_1 \otimes [Z^1, ..., Z^{N-1}]\right) s_1 \\ t_2 = \left(v_2 \otimes [Z^1, ..., Z^{N-1}]\right) s_2 \\ t_3 = \left([Z^1, ..., Z^{N-1}] \otimes v_1\right) s_3 \\ t_4 = \left([Z^1, ..., Z^{N-1}] \otimes v_2\right) s_4. \end{cases} \tag{3.62}$$

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

$$\begin{cases} (t_1)_{i,j} = (v_1)_i \sum_{l=1}^{N-1} (s_1)_l Z_j^l \\ (t_2)_{i,j} = (v_2)_i \sum_{l=1}^{N-1} (s_2)_l Z_j^l \\ (t_3)_{i,j} = (v_1)_j \sum_{k=1}^{N-1} (s_3)_k Z_i^k \\ (t_4)_{i,j} = (v_2)_j \sum_{k=1}^{N-1} (s_4)_k Z_i^k. \end{cases} \tag{3.63}$$

Each sum in each right-hand-side in (3.63) is computed by FFT, which gives a cost of $O(N\,\mathrm{Log}(N))$. *The computation of the vector* $t \in \mathbb{R}^{(N-1)^2}$ *in (3.61) is therefore* $O(N^2\,\mathrm{Log}(N))$.

- **Step 6:**
  Resolution of the linear system in $\mathbb{R}^{(N-1)^2}$

$$v = \mathcal{B}^{-1} t \tag{3.64}$$

via the fast FFT solver as in Step 1. The cost is $O(N^2\,\mathrm{Log}(N))$.

- **Step 7:**
  Assemble the solution $\psi \in \mathbb{R}^{(N-1)^2} \overset{\text{vect}}{\simeq} L^2_{h,0}$ *of the biharmonic problem (2.5) (with* $(a, b) = (0, 1)$*) by*

$$\psi = g - 36v \tag{3.65}$$

where $g, v \in \mathbb{R}^{(N-1)^2}$ *are given in (3.50,3.64). The cost is* $O(N^2)$.

- **Step 8:**
  Compute the hermitian gradient $\psi_x, \psi_y \in \mathbb{R}^{(N-1)^2}$ *as a post-processing of the grid values of* ψ *by*

$$\begin{cases} \psi_x = \left(\dfrac{3}{h} P^{-1} K \otimes I\right) \psi \\ \psi_y = \left(\dfrac{3}{h} I \otimes P^{-1} K\right) \psi \end{cases} \tag{3.66}$$

Computation (3.66) is performed using the one-dimensional FFT for a global cost $O(N^2\,\mathrm{Log}(N))$.

The overall computing cost of Algorithm 2 is therefore $O(N^2)\,\mathrm{Log}(N)$ under the assumption that Step 4 is at most $O(N^2\,\mathrm{Log}(N))$. See Appendix A. Let us conclude this Section by considering now the case of problem (2.4) discretized by (2.5), with $a \geq 0, b > 0$. The matrix form of the finite difference operator $-a\Delta_h + b\Delta_h^2$ is

$$-a\Delta_h + b\Delta_h^2 = \frac{a}{h^2}\left[T \otimes I + I \otimes T\right] + \frac{b}{h^4}\left(\mathcal{B} + \mathcal{C}\right). \tag{3.67}$$

Defining

$$\mathcal{B}^{a,b} = ah^2 \left[T \otimes I + I \otimes T\right] + b\mathcal{B}, \tag{3.68}$$

we have

$$h^4(-a\Delta_h + b\Delta_h^2) = \mathcal{B}^{a,b} + b\mathcal{C}. \tag{3.69}$$

It turns out that the algorithm that solves problem (2.5) is now exactly the same as Algorithm 2, replacing the eigenvalues $\mu_{k,l}$ of $\mathcal{B}$ by

$$\mu_{k,l}^{a,b} = ah^2(\lambda_k + \lambda_l) + b\left(\frac{\lambda_k^2}{(1-\lambda_k/6)} + \frac{\lambda_l^2}{(1-\lambda_l/6)} + 2\lambda_k\lambda_l\right). \tag{3.70}$$

In addition $\mathcal{C}$ is replaced by $b\mathcal{C}$ in (3.40, 3.44).

3.4. Treatment of non-homogeneous boundary conditions. Let us first consider the modifications of $\delta_x^4 u$ at near boundary points. We write

$$\delta_x^4 u = \frac{12}{h^2}(\delta_x u_x - \delta_x^2 u). \tag{3.71}$$

Let U be a one dimensional vector of length $N-1$, associated with the values of the solution u for a fixed y. The vector U_x is associated with the approximated derivative of u. At near boundary point $x = x_1$ we have

$$(\delta_x u_x)_{x=x_1} = \frac{1}{2h}((U_x)_2 - (U_x)_0). \tag{3.72}$$

Similarly for $x = x_{N-1}$

$$(\delta_x u_x)_{x=x_{N-1}} = \frac{1}{2h}((U_x)_N - (U_x)_{N-2}). \tag{3.73}$$

Therefore, we can write

$$\delta_x u_x = \frac{1}{2h}KU_x + \frac{1}{2h}\begin{bmatrix} -(U_x)_0 \\ \cdot \\ \cdot \\ (U_x)_N \end{bmatrix}. \tag{3.74}$$

We replace KU_x by $KP^{-1}PU_x$ and get

$$\delta_x u_x = \frac{1}{2h}KP^{-1}PU_x + \frac{1}{2h}\begin{bmatrix} -(U_x)_0 \\ \cdot \\ \cdot \\ (U_x)_N \end{bmatrix}. \tag{3.75}$$

Now, for $i = 1,,,N-1$ we have $(U_x)_{i+1} + 4(U_x)_i + (U_x)_{i-1} = \frac{6}{2h}(U_{i+1} - U_{i-1})$. For $i = 1$ we have

$$(U_x)_2 + 4(U_x)_1 = \frac{6}{2h}U_2 - \frac{6}{2h}U_0 - (U_x)_0. \tag{3.76}$$

Similarly, for $i = N-1$

$$4(U_x)_{N-1} + (U_x)_{N-2} = -\frac{6}{2h}U_{N-2} + \frac{6}{2h}U_N - (U_x)_N. \tag{3.77}$$

Thus, it follows that

$$PU_x = \frac{6}{2h} KU. \tag{3.78}$$

For $i = 1$ we have

$$(U_x)_2 + 4(U_x)_1 = \frac{6}{2h} U_2 - \frac{6}{2h} U_0 - (U_x)_0. \tag{3.79}$$

Similarly, for $i = N - 1$

$$4(U_x)_{N-1} + (U_x)_{N-2} = -\frac{6}{2h} U_{N-2} + \frac{6}{2h} U_N - (U_x)_N. \tag{3.80}$$

Thus, it follows that

$$PU_x = \frac{6}{2h} KU + \begin{bmatrix} -\frac{6}{2h} U_0 - (U_x)_0 \\ \cdot \\ \cdot \\ \frac{6}{2h} U_N - (U_x)_N \end{bmatrix}. \tag{3.81}$$

Therefore,

$$\delta_x u_x = \frac{3}{2h^2} KP^{-1} KU + \frac{1}{2h} KP^{-1} \begin{bmatrix} -\frac{6}{2h} U_0 - (U_x)_0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ \frac{6}{2h} U_N - (U_x)_N \end{bmatrix} + \frac{1}{2h} \begin{bmatrix} -(U_x)_0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ (U_x)_N \end{bmatrix}. \tag{3.82}$$

We also have that

$$(\delta_x^2 u)_{x=x_1} = \frac{1}{h^2} (U_2 - 2U_1 + U_0). \tag{3.83}$$

Similarly for $x = x_{N-1}$

$$(\delta_x^2 u)_{x=x_1} = \frac{1}{h^2} (U_N - 2U_{N-1} + U_{N-2}). \tag{3.84}$$

Thus,

$$\delta_x^2 u = -\frac{T}{h^2} U + \frac{1}{h^2} \begin{bmatrix} U_0 \\ \cdot \\ \cdot \\ U_N \end{bmatrix}. \tag{3.85}$$

Therefore,

$$\delta_x^4 u = \frac{12}{h^2} \left(\left[\frac{3}{2h^2} KP^{-1} K + \frac{1}{h^2} T \right] U + \frac{1}{2h} KP^{-1} \begin{bmatrix} -\frac{6}{2h} U_0 - (U_x)_0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ \frac{6}{2h} U_N - (U_x)_N \end{bmatrix} + \frac{1}{2h} \begin{bmatrix} -(U_x)_0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ (U_x)_N \end{bmatrix} - \frac{1}{h^2} \begin{bmatrix} U_0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ U_N \end{bmatrix} \right). \tag{3.86}$$

Note that the previous expression is a perturbation of the operator δ_x^4 that we received in equation (2.22), where the additional terms come from the boundary. Since the value of the solution, along with its first order derivatives are known on the boundary, they may be transformed to the right hand side of the equation.

$\delta_y^4 u$ may be expressed in a similar way. The mixed derivative $\delta_x^2\delta_y^2 u$ yields modifications involving the values of U only at the boundary.

4. A fourth order compact scheme for biharmonic problems.

4.1. Fourth order accurate nine points compact schemes. In this subsection, we describe how to modify the 9 points biharmonic operator (2.7) and the 5 points Laplacian (2.6) in order to obtain fourth-order accuracy.

The 1-D operators $\delta_x^4\psi_{i,j}$, $\delta_y^4\psi_{i,j}$ in (4.1) are given as functions of ψ, ψ_x, ψ_y by

$$\delta_x^4\psi_{i,j} = \frac{12}{h^2}\Big[(\delta_x\psi_x)_{i,j} - (\delta_x^2\psi)_{i,j}\Big] \; ; \; \delta_y^4\psi_{i,j} = \frac{12}{h^2}\Big[(\delta_y\psi_y)_{i,j} - (\delta_y^2\psi)_{i,j}\Big]. \tag{4.1}$$

These two operators are fourth order accurate (see [3]). The second order accuracy of the operator Δ_h^2 is due only to the term $\delta_x^2\delta_y^2\psi_{i,j}$. Thus, the local truncation error is $\delta_x^2\delta_y^2\psi_{i,j} - \partial_x^2\partial_y^2\psi_{i,j} = O(h^2)$. The later may be checked using Taylor expansion of ψ around (x_i, y_j), or by checking the truncation error in $\delta_x^2\delta_y^2\psi_{i,j}$ by applying it to a Fourier mode $\psi = e^{i(kx+ly)}$. We refer to [3] for the study of the accuracy at near boundary points.

In order to derive a fourth order approximation to $\partial_x^2\psi$ we proceed as follows. The Taylor expansion of $\delta_x^2\psi$, $\delta_x\psi_x$ are

$$\delta_x^2\psi_i = \partial_x^2\psi_i + \frac{h^2}{12}\partial_x^4\psi_i + O(h^4), \tag{4.2}$$

$$\delta_x\psi_{x,i} = \partial_x^2\psi_i + \frac{h^2}{6}\partial_x^4\psi_i + O(h^4). \tag{4.3}$$

A linear combination of these two operators allows us to derive the fourth order accurate approximation $\tilde{\delta}_x^2$ to $\partial_x^2\psi$, eliminating the h^2 term in the following way

$$\tilde{\delta}_x^2\psi_i = 2\delta_x^2\psi_i - \delta_x\psi_{x,i} = \partial_x^2\psi_i + O(h^4). \tag{4.4}$$

In a similar way we derive a fourth-order accurate scheme for $\partial_y^2\psi$, i.e.,

$$\tilde{\delta}_y^2\psi_i = 2\delta_y^2\psi_i - \delta_y\psi_{y,i} = \partial_y^2\psi_i + O(h^4). \tag{4.5}$$

Therefore, a fourth order approximation for the Laplacian is

$$\tilde{\Delta}_h\psi_{i,j} = 2\delta_x^2\psi_{i,j} - \delta_x\psi_{x,i,j} + 2\delta_y^2\psi_{i,j} - \delta_y\psi_{y,i,j}. \tag{4.6}$$

Invoking

$$\delta_x\psi_x - \delta_x^2\psi = \frac{h^2}{12}\delta_x^4\psi, \quad \delta_y\psi_y - \delta_y^2\psi = \frac{h^2}{12}\delta_y^4\psi, \tag{4.7}$$

and applying it to (4.6), we find that the operator $\tilde{\Delta}_h$ may be rewritten as a perturbation of Δ_h in the following way

$$\tilde{\Delta}_h = \delta_x^2 - \frac{h^2}{12}\delta_x^4 + \delta_y^2 - \frac{h^2}{12}\delta_y^4 = \Delta_h - \frac{h^2}{12}(\delta_x^4 + \delta_y^4). \tag{4.8}$$

In a similar manner we construct a fourth-order accurate approximation to $\partial_x^2\partial_y^2\psi_{i,j}$. First we expand $\delta_y^2\delta_x\psi$ in powers of h

$$\delta_y^2\delta_x\psi_{x,i,j} = \partial_y^2\left[\partial_x^2\psi_{i,j}+\frac{h^2}{6}\partial_x^4\psi_{i,j}+O(h^4)\right]+\frac{h^2}{12}\partial_y^4\left[\partial_x^2\psi_{i,j}+\frac{h^2}{6}\partial_x^4\psi_{i,j}+O(h^4).\right] \quad (4.9)$$

Thus,

$$\delta_y^2\delta_x\psi_{x,i,j} = \partial_x^2\partial_y^2\psi_{i,j} + \frac{h^2}{6}\partial_y^2\partial_x^4\psi_{i,j} + \frac{h^2}{12}\partial_y^4\partial_x^2\psi_{i,j} + O(h^4). \quad (4.10)$$

Symmetrically,

$$\delta_x^2\delta_y\psi_{y,i,j} = \partial_x^2\partial_y^2\psi_{i,j} + \frac{h^2}{6}\partial_x^2\partial_y^4\psi_{i,j} + \frac{h^2}{12}\partial_x^4\partial_y^2\psi_{i,j} + O(h^4). \quad (4.11)$$

The Taylor expansion of the mixed operator $\delta_x^2\delta_y^2$ is therefore

$$\delta_x^2\delta_y^2\psi_{i,j} = \partial_x^2\partial_y^2\psi_{i,j} + \frac{h^2}{12}\partial_x^2\partial_y^4\psi_{i,j} + \frac{h^2}{12}\partial_y^2\partial_x^4\psi_{i,j} + O(h^4). \quad (4.12)$$

Combining (4.12), (4.11), (4.9), we define the new mixed finite-difference operator $\widetilde{\delta_x^2\delta_y^2}\psi_{i,j}$ as

$$\widetilde{\delta_x^2\delta_y^2}\psi_{i,j} = 3\delta_x^2\delta_y^2\psi_{i,j} - \delta_x^2\delta_y\psi_{y,i,j} - \delta_y^2\delta_x\psi_{x,i,j} = \partial_x^2\partial_y^2\psi_{i,j} + O(h^4). \quad (4.13)$$

Keeping δ_x^4 and δ_y^4 as before, we define the 4th order biharmonic operator $\tilde{\Delta}_h^2$ as

$$\tilde{\Delta}_h^2\psi_{i,j} = \delta_x^4\psi_{i,j} + \delta_y^4\psi_{i,j} + 2\widetilde{\delta_x^2\delta_y^2}\psi_{i,j}. \quad (4.14)$$

Invoking (4.7) again allows us to rewrite the 4th order operator $\tilde{\Delta}_h^2$ as a perturbation of Δ_h^2 in the following way.

$$\tilde{\Delta}_h^2 = \delta_x^4\left(I - \frac{h^2}{6}\delta_y^2\right) + \delta_y^4\left(I - \frac{h^2}{6}\delta_x^2\right) + 2\delta_x^2\delta_y^2. \quad (4.15)$$

Finally, the new fourth order scheme which approximates (2.4) is

$$\begin{cases} \left[-a\tilde{\Delta}_h + b\tilde{\Delta}_h^2\right]\psi_{i,j} = f_{i,j} \quad , \quad 1 \le i,j \le N-1 \\ \psi_{i,j} = 0 \quad , \quad \psi_{x,i,j} = \psi_{y,i,j} = 0, \quad , i \in \{0,N\}, j \in \{0,N\}. \end{cases} \quad (4.16)$$

As a consequence of (4.15) a fourth-order approximation to the biharmonic equation $\Delta^2\psi = f$ is

$$\tilde{\Delta}_h^2 = \delta_x^4\left(I - \frac{h^2}{6}\delta_y^2\right) + \delta_y^4\left(I - \frac{h^2}{6}\delta_x^2\right) + 2\delta_x^2\delta_y^2 = f. \quad (4.17)$$

Note that in the right-hand-side of (4.17) only the value of $f(i,j)$ is involved in the discretization of the differential equation at (x_i, y_j). This feature of the scheme is important in cases where the biharmonic problem has to be solved when the function f in unknown on the boundary, but is known only at interior points (see [3]). A different fourth-order accurate scheme for the biharmonic equation $\Delta^2\psi = f$ was presented in ([34], Sec.3.2) and a multigrid solver was designed for this scheme in [1]. The scheme in ([34], Sec.3.2) involves the five values of f, $f_{i,j}, f_{i+1,j}, f_{i-1,j}, f_{i,j+1}$ and $f_{i,j-1}$, in order to construct a fourth-order approximation to the biharmonic operator.

Note finally that in (4.17) the gradient (ψ_x, ψ_y) is used at all the nine points of the stencil of the scheme. In a forthcoming paper, an analogous scheme for irregular domains will be derived, using directional derivatives at corner points.

4.2. Fast solution procedure. The fast solution procedure for solving (4.15) follows exactly the same lines as in Section 3. The matrix forms of the positive operators $\tilde{\Delta}_h^2$ and $-(\tilde{\Delta}_h)$ are

$$\begin{cases} \tilde{\Delta}_h^2 & = \frac{1}{h^4}\Big(6P^{-1}T^2 \otimes \left(I_{N-1} + \frac{T}{6}\right) + 6\left(I_{N-1} + \frac{T}{6}\right) \otimes P^{-1}T^2 + 2T \otimes T\Big) \\ & + \frac{36}{h^4}[v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \otimes \left(I_{N-1} + \frac{T}{6}\right) + \frac{36}{h^4}\left(I_{N-1} + \frac{T}{6}\right) \otimes [v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \\ -(\tilde{\Delta}_h) & = \frac{1}{h^2}\Big(T \otimes I + I \otimes T + \frac{1}{2}[P^{-1}T^2 \otimes I_{N-1} + I_{N-1} \otimes P^{-1}T^2]\Big) \\ & + \frac{3}{h^2}[v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \otimes I_{N-1} + \frac{3}{h^2} I_{N-1} \otimes [v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}. \end{cases}$$

The matrix form $\mathcal{A}'$ of the fourth order operator $h^4\left(-a\tilde{\Delta}_h + b\tilde{\Delta}_h^2\right)$ as a "diagonal" part $\mathcal{B}'$ and a "capacitance" part $\mathcal{C}'$, (see Section 3.3) is

$$\mathcal{A}' = \mathcal{B}' + \mathcal{C}' \tag{4.18}$$

with

$$\mathcal{B}' = ah^2\Big(T \otimes I + I \otimes T + \frac{1}{2}[P^{-1}T^2 \otimes I_{N-1} + I_{N-1} \otimes P^{-1}T^2]\Big) + b\Big(6P^{-1}T^2 \otimes \left(I_{N-1} + \frac{T}{6}\right) + 6\left(I_{N-1} + \frac{T}{6}\right) \otimes P^{-1}T^2 + 2T \otimes T\Big).$$

The eigenvectors of $\mathcal{B}'$ are $Z^k \otimes Z^l$ and the eigenvalues are

$$\mu'_{k,l} = ah^2\left((\lambda_k + \lambda_l) + \frac{1}{12}\frac{\lambda_k^2}{(1 - \lambda_k/6)} + \frac{1}{12}\frac{\lambda_l^2}{(1 - \lambda_l/6)}\right) + b\left(\lambda_k^2\frac{1 + \lambda_l/6}{1 - \lambda_k/6} + \lambda_l^2\frac{1 + \lambda_k/6}{1 - \lambda_l/6} + 2\lambda_k\lambda_l\right).$$

The capacitance part of $\mathcal{A}'$ is

$$\mathcal{C}' = 36\left([v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \otimes \left((\frac{ah^2}{12} + b)I_{N-1} + b\frac{T}{6}\right) + \left((\frac{ah^2}{12} + b)I_{N-1} + b\frac{T}{6}\right) \otimes [v_1, v_2]\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}\right). \tag{4.19}$$

The structure of $\mathcal{C}'$ is the same as precedingly,

$$\mathcal{C}' = 36\mathcal{R}'\mathcal{R}'^T \tag{4.20}$$

where $\mathcal{R}'$ is the $(N-1)^2 \times 4(N-1)$ matrix

$$\mathcal{R}' = [R'_1, R'_2, R'_3, R'_4], \tag{4.21}$$

and

$$\begin{cases} R'_1 = [v_1 \otimes Z'^{,1}, v_1 \otimes Z'^{,2}, ..., v_1 \otimes Z'^{,N-1}] \\ R'_2 = [v_2 \otimes Z'^{,1}, v_2 \otimes Z'^{,2}, ..., v_2 \otimes Z'^{,N-1}] \\ R'_3 = [Z'^{,1} \otimes v_1, Z'^{,2} \otimes v_1, ..., Z'^{,N-1} \otimes v_1] \\ R'_4 = [Z'^{,1} \otimes v_2, Z'^{,2} \otimes v_2, ..., Z'^{,N-1} \otimes v_2] \end{cases} \tag{4.22}$$

with

$$Z' = \left[Z'^{,1}, Z'^{,2}, .., Z'^{,N-1} \right], \quad Z'^{,j} = \left(\frac{ah^2}{12} + b(1 + \frac{\lambda_j}{6}) \right)^{1/2} Z^j, \quad 1 \le j \le N-1 \tag{4.23}$$

The numerical algorithm for (4.16) is the same as Algorithm 2, replacing $\mathcal{B}, \mathcal{C}, \mathcal{R}, Z$, by $\mathcal{B}', \mathcal{C}', \mathcal{R}', Z'$. Non homogeneous boundary conditions are treated as in Subsection 3.4.

5. Numerical results. The numerical results presented in the sequel have been obtained with a code written in FORTRAN90. The package *fftpack* of Swarztrauber [36] has been used for computing the FFT. In addition we have used the *g95* compiler [37] without optimization. The computations have been ran in double precision on a laptop with a processor Intel Pentium M, 2.13 GHZ, with 1GB memory.

5.1. Accuracy. We report numerical results obtained so far with the two versions of the scheme, the second order and the fourth order versions. The discrete L^2, and L^∞ errors are defined by

$$\begin{cases} \|\psi - \psi_h\|_h = \left[h^2 \sum_{i,j} (\psi(x_i, y_j) - \psi_{i,j})^2 \right]^{1/2} & (a), \\ \|\psi - \psi_h\|_{\infty,h} = \sup_{i,j=1,..N-1} |\psi(x_i, y_j) - \psi_{i,j}| & (c). \end{cases} \tag{5.1}$$

Taking into account non homogeneous boundary conditions gives an additional contribution to the right-hand side for near boundary points, whose value is simply deduced from the boundary values of ψ, ψ_x, ψ_y on the four edges, which appear in the expression of the discrete Laplacian and Biharmonic operators.

• *Example 1*

The problem solved is (2.5) with exact solution $\psi(x, y) = \sin^2(x) \sin^2(y)$ on the square $\Omega = [0, \pi]^2$. That test is the same as Example 2 in [1] and Example 1 in [5]. Table 5.1 reports the numerical results with the second order accurate scheme (2.5) (with $(a, b) = (0, 1)$. The source term is $\Delta^2 \psi(x, y)$ and the boundary conditions are zeros on the four sides of the square. We observe the second order accuracy of the scheme for ψ, the gradient (ψ_x, ψ_y) as well as for the Laplacian $\Delta\psi \simeq \Delta_h \psi$.

Table 5.2 reports the numerical results with the fourth order scheme (4.16). The scheme exhibits a fourth order accuracy, for ψ, (ψ_x, ψ_y) as well as for $\Delta\psi \simeq \tilde{\Delta}_h \psi$, up to a 512×512 grid, where the numerical accuracy of the computer is reached, (double precision).

• *Example 2*

We consider Problem (2.5) with zero source term and boundary conditions

$$\begin{cases} \Delta^2 \psi(x, y) = 0, \quad (x, y) \in \Omega \\ \psi(x, y) = 0 \\ \dfrac{\partial \psi}{\partial x}(0, y) = \dfrac{\partial \psi}{\partial x}(1, y) = \dfrac{\partial \psi}{\partial y}(x, 0) = 0 \\ \dfrac{\partial \psi}{\partial y}(x, 1) = -1 \end{cases} \tag{5.2}$$

This corresponds to the Stokes problem in pure streamfunction form for a driven cavity setting. See Example 6 in [5], Problem 3 in [1]. The fourth order scheme has been used.

N	$\|\psi-\psi_h\|_{\infty,h}$	$\|\psi_x-\psi_{x,h}\|_{\infty,h}$	$\|\psi-\psi_{y,h}\|_{\infty,h}$	$\|\Delta\psi-\Delta_h\psi_h\|_{\infty,h}$
$N=16$	6.46(-3)	6.59(-3)	6.59(-3)	2.24(-2)
conv. rate	2.00	1.98	1.98	2.00
$N=32$	1.61(-3)	1.67(-3)	1.67(-3)	5.58(-3)
conv. rate	1.99	1.98	1.98	2.00
$N=64$	4.04(-4)	4.22(-4)	4.22(-4)	1.39(-3)
conv. rate	2.00	1.99	1.99	1.99
$N=128$	1.01(-4)	1.06(-4)	1.06(-4)	3.49(-4)
conv. rate	1.99	2.00	2.00	2.00
$N=256$	2.53(-5)	2.65(-5)	2.65(-5)	8.72(-5)
conv. rate	2.00	2.00	2.00	2.00
$N=512$	6.32(-6)	6.61(-6)	6.61(-6)	2.18(-5)
conv. rate	2.00	2.00	2.00	1.99
$N=1024$	1.58(-6)	1.65(-6)	1.65(-6)	5.47(-6)

TABLE 5.1

Error and convergence rate for Test Case 1 with the second order scheme (2.5).

N	$\|\psi-\psi_h\|_{\infty,h}$	$\|\psi_x-\psi_{x,h}\|_{\infty,h}$	$\|\psi-\psi_{y,h}\|_{\infty,h}$	$\|\Delta\psi-\tilde{\Delta}_h\psi_h\|_{\infty,h}$
$N=16$	3.42(-5)	1.00(-4)	1.00(-4)	3.99(-4)
conv. rate	4.04	4.01	4.01	4.00
$N=32$	2.08(-6)	6.21(-6)	6.21(-6)	2.48(-5)
conv. rate	4.01	4.00	4.00	4.00
$N=64$	1.29(-7)	3.87(-7)	3.87(-7)	1.55(-6)
conv. rate	4.00	4.00	4.00	4.00
$N=128$	8.06(-9)	2.41(-8)	2.41(-8)	9.68(-8)
conv. rate	3.99	3.99	3.99	3.83
$N=256$	5.04(-10)	1.51(-9)	1.51(-9)	6.77(-9)
conv. rate	3.74	4.02	4.02	-0.22
$N=512$	3.76(-11)	9.27(-11)	9.07(-11)	7.90(-9)
conv. rate	-0.13	0.19	0.19	0.59
$N=1024$	4.12(-11)	8.09(-11)	8.09(-11)	5.22(-8)

TABLE 5.2

Error and convergence rate for Test Case 1 with the fourth order scheme (4.16)

N	max$\|\psi\|$	location (x,y)
64	0.1000803	$(0.5, 0.765625)$
128	0.1000767	$(0.5, 0.765625)$
256	0.1000759	$(0.5, 0.765625)$
Bialecki(N=128)[5]	0.100076276	$(0.5, 0.765)$
Altas *et al.*(N=64)[1]	0.10008	$(0.5, 0.766)$

TABLE 5.3

Maximum value and location of $\max|\psi|$ *in Stokes Problem (5.2) with the fourth order scheme (4.16).*

- *Example 3*

In order to demonstrate the efficiency of the fourth order scheme (4.16), we present several examples of problems (2.4) with coefficients $(a,b) = (1,2)$ for which we applied our fourth order scheme (4.16). For the first test problem

$$\psi(x,y) = (1+x^2)(1+y^2) \tag{5.3}$$

Here we received zero error up to the machine accuracy, according to the fact that the scheme is exact for fourth order polynomials.

• *Example 4*

Consider the case where the exact solution is

$$\psi(x,y) = (1-x^2)^2(1-y^2)^2, \quad -1 \le x,y \le 1. \tag{5.4}$$

This function solves the equation

$$-\Delta\psi + 2\Delta^2\psi = f(x,y),$$

where $f(x,y) = -\Delta\psi + 2\Delta^2\psi$ is the forcing term. In this case we have homogeneous boundary conditions.

Table 5.4 summarizes the errors, $e = \|\psi - \psi_h\|_h$, and the error in the x and y-derivatives $e_x = \|\partial_x\psi - \psi_{x,h}\|_h$, $e_y = \|\partial_y\psi - \psi_{y,h}\|_h$.

N	32	Rate	64	Rate	128	Rate	256
e	2.0763(−6)	4.03	1.2735(−7)	4.00	7.9604(−9)	4.08	4.9762(−10)
e_x	3.4466(−6)	4.00	2.1542(−7)	4.00	1.3465(−8)	4.00	8.4173(−10)
e_y	3.4466(−6)	4.00	2.1542(−7)	4.00	1.3465(−8)	4.00	8.4173(−10)

TABLE 5.4

Error and convergence rate in $\|.\|_h$ norm for $\psi(x,y) = (1-x^2)^2(1-y^2)^2$, $(a,b) = (1,2)$ with the fourth order scheme (4.16).

• *Example 5*

An additional example with non-zero boundary conditions is

$$\psi(x,y) = (x^4 + y^4)^2.$$

The results are summarized in Table 5.5.

N	32	Rate	64	Rate	128	Rate	256
e	2.5796(−4)	3.98	1.6385(−5)	3.99	1.0296(−6)	3.98	6.5118(−8)
e_x	1.4434(−4)	3.91	9.6212(−6)	3.96	6.1936(−7)	3.82	4.3760(−8)
e_y	1.4434(−4)	3.91	9.6212(−6)	3.96	6.1936(−7)	3.82	4.3760(−8)

TABLE 5.5

Error and convergence rate in $\|.\|_h$ norm for $\psi(x,y) = (x^4+y^4)^2$, $(a,b) = (1,2)$ with the fourth order scheme (4.16).

5.2. Computing efficiency. We report here the CPU time in seconds for the fourth order scheme (4.16) (FORTRAN90, Intel Pentium 2.13GHZ, 1GB memory). In Table 5.6 we display the CPU time for some of the results in Table 5.2. CPU_{tot} stands for the time which corresponds to obtain the complete solution of the linear system, while CPU_∞ stands for the CPU time for the solution procedure without the assembling the capacitance matrix (3.58) (Step 3 of algorithm 2). Note that the capacitance matrix does not depend on the right-hand-side of the system of equations, thus for a time-dependent problem it may be computed only once. In 5.2 we also report on the ratio $\mathrm{CPU}_{tot}/(N^2\,\mathrm{Log}(N))$. It seems from the computations that this ratio is slowly decreasing to a constant. In addition, we report on the number of the iterations in the CG algorithm for the capacitance linear system to converge within a prescribed accuracy of 10^{-20}. We refer the reader to Appendix A for some estimates on the condition number of the capacitance matrix. It can be observed in 5.2 that the number of iterations in the CG algorithm grow very slowly. In practice, we observe that the spectral part of the solution procedure is more demanding in computing resources compared to the capacitance matrix solving part.

N	N=64	N=128	N=256	N=512	N=1024	N=2048
CPU_{tot}	0.016s	0.11s	0.47s	1.91s	7.67s	33.52s
CPU_∞	0.016s	0.093s	0.38s	1.52s	6.28s	27.28s
$\mathrm{CPU}_{tot}/(N^2\,\mathrm{Log}(N))$	9.17(-7)	1.37(-6)	1.29(-6)	1.17(-6)	1.06(-6)	1.05(-6)
k_{CG}	17	18	19	19	21	23

Table 5.6
Indicative CPU time for results in Table 5.2.

In Table 5.7 we report on the computing efficiency, as well as the number of CG iterations, of our solver for a non-separable biharmonic problem in $\Omega = [0,1]^2$, [1], with exact solution

$$\psi(x,y) = x^3 \ln(1+y) + \frac{y}{1+x} \tag{5.5}$$

Observe that, for this example too, the number of iterations in the CG algorithm growth very slowly.

N	N=16	N=32	N=64	N=128	N=256	N=512
CPU_{tot}	0.00s	0.00s	0.06s	0.13s	0.56s	2.09s
$\mathrm{CPU}_{tot}/(N^2\,\mathrm{Log}(N))$	0.	0.	3.70(-6)	1.57(-6)	1.46(-6)	1.27(-6)
k_{CG}	23	26	28	30	32	34

Table 5.7
Indicative CPU time for the biharmonic problem with exact solution (5.5) and the fourth order scheme

6. Conclusion. The capacitance matrix method, applied to the second order Stephenson scheme (2.10) and to the fourth order scheme (4.16), appears to be efficient. It seems that it is competitive with the multigrid method reported by Altas *et al*, [1]. In the latter the solver is designed for the fourth order Stephenson scheme and the numerical results are limited to 128×128 grids. In addition, the design of our algorithm seems to be simpler than the fast solver presented in [5] for the OSC scheme.

In fact, we currently use the new algorithm to solve the time-dependent Navier-Stokes equation on fine grids. Finally, note that the extension of the solution procedure to problems with boundary condition on $\Delta\psi$ can be handled without major modifications. The latter can be carried out using the discretizations (2.6) or (4.6). In addition, the extension to three-dimensional problems appears to be tractable by a similar procedure.

Appendix

Appendix A. Resolution of the capacitance linear system. In this subsection we focus on the resolution of the capacitance system (3.59). Consider first the $4(N-1)\times 4(N-1)$ matrix $\mathcal{R}\mathcal{B}^{-1}\mathcal{R}^T$ which has a 4×4 block structure

$$\mathcal{R}^T\mathcal{B}^{-1}\mathcal{R} = \begin{bmatrix} R_1^T\mathcal{B}^{-1}R_1 & R_1^T\mathcal{B}^{-1}R_2 & R_1^T\mathcal{B}^{-1}R_3 & R_1^T\mathcal{B}^{-1}R_4 \\ R_2^T\mathcal{B}^{-1}R_1 & R_2^T\mathcal{B}^{-1}R_2 & R_2^T\mathcal{B}^{-1}R_3 & R_2^T\mathcal{B}^{-1}R_4 \\ R_3^T\mathcal{B}^{-1}R_1 & R_3^T\mathcal{B}^{-1}R_2 & R_3^T\mathcal{B}^{-1}R_3 & R_3^T\mathcal{B}^{-1}R_4 \\ R_4^T\mathcal{B}^{-1}R_1 & R_4^T\mathcal{B}^{-1}R_2 & R_4^T\mathcal{B}^{-1}R_3 & R_4^T\mathcal{B}^{-1}R_4 \end{bmatrix}, \tag{A.1}$$

where the four $(N-1)^2\times(N-1)$ matrices R_k are given in (3.42). By symmetry, only the upper diagonal part of (A.1) has to be computed. The vector $Z^j\in\mathbb{R}^{N-1}$ is given in (3.6) and the vectors $v_1, v_2\in\mathbb{R}^{N-1}$ in (3.28) are decomposed as a linear combination of Z^k by

$$\left\{\begin{array}{l} v_1 = (\alpha-\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k - Z_{N-1}^k}{6-\lambda_k}Z^k \\ v_2 = (\alpha+\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k + Z_{N-1}^k}{6-\lambda_k}Z^k. \end{array}\right. \tag{A.2}$$

Therefore for $i=1,..,N-1$,

$$\left\{\begin{array}{ll} v_1\otimes Z^i = (\alpha-\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k - Z_{N-1}^k}{6-\lambda_k}Z^k\otimes Z^i, & (i) \\ v_2\otimes Z^i = (\alpha+\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k + Z_{N-1}^k}{6-\lambda_k}Z^k\otimes Z^i, & (ii) \\ Z^i\otimes v_1 = (\alpha-\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k - Z_{N-1}^k}{6-\lambda_k}Z^i\otimes Z^k, & (iii) \\ Z^i\otimes v_2 = (\alpha+\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k + Z_{N-1}^k}{6-\lambda_k}Z^i\otimes Z^k. & (iv) \end{array}\right. \tag{A.3}$$

Operating with $\mathcal{B}^{-1}$ on the left in $(A.3)_{i,ii,iii,iv}$ gives for $j=1,..,N-1$

$$\left\{\begin{array}{ll} \mathcal{B}^{-1}(v_1\otimes Z^j) = (\alpha-\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k - Z_{N-1}^k}{(6-\lambda_k)\mu_{k,j}}Z^k\otimes Z^j, & (i) \\ \mathcal{B}^{-1}(v_2\otimes Z^j) = (\alpha+\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k + Z_{N-1}^k}{(6-\lambda_k)\mu_{k,j}}Z^k\otimes Z^j, & (ii) \\ \mathcal{B}^{-1}(Z^j\otimes v_1) = (\alpha-\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k - Z_{N-1}^k}{(6-\lambda_k)\mu_{k,j}}Z^k\otimes Z^j, & (iii) \\ \mathcal{B}^{-1}(Z^j\otimes v_2) = (\alpha+\beta)^{1/2}\dfrac{\sqrt{2}}{2}\displaystyle\sum_{k=1}^{N-1}\dfrac{Z_1^k + Z_{N-1}^k}{(6-\lambda_k)\mu_{k,j}}Z^k\otimes Z^j. & (iv) \end{array}\right. \tag{A.4}$$

Taking the $\mathbb{R}^{(N-1)^2}$ scalar product of (A.3) and (A.4), and using that

$$\left(Z^k\otimes Z^l, Z^{k'}\otimes Z^{l'}\right) = \delta_{k,k'}\delta_{l,l'}, \quad 1\le k,k',l,l'\le N-1, \tag{A.5}$$

yields the term (i,j) of the matrix $R_1^T\mathcal{B}^{-1}R_1$ is

$$(v_1\otimes Z^i)^T\mathcal{B}^{-1}(v_1\otimes Z^j)=\frac{\alpha-\beta}{2}\delta_{i,j}\sum_{k=1}^{N-1}\left(\frac{Z_1^k-Z_{N-1}^k}{6-\lambda_k}\right)^2\frac{1}{\mu_{k,j}}. \tag{A.6}$$

This proves that the $(N-1)\times(N-1)$ matrix $D_1=R_1^T\mathcal{B}^{-1}R_1$ is actually diagonal. Using

$$Z_{N-1}^k=\left(\frac{2}{N}\right)^{1/2}\sin\frac{k(N-1)\pi}{N}=(-1)^{k+1}\left(\frac{2}{N}\right)^{1/2}\sin\frac{k\pi}{N}=(-1)^{k+1}Z_1^k \tag{A.7}$$

in (A.6) yields that the diagonal coefficient $(D_1)_j$ is

$$\Big[R_1^T\mathcal{B}^{-1}R_1\Big]_{j,j}=\frac{4(\alpha-\beta)}{N}\sum_{k=1,k\text{ even}}^{N-1}\sin^2\left(\frac{k\pi}{N}\right)\frac{1}{(6-\lambda_k)^2\mu_{k,j}}. \tag{A.8}$$

Similarly, we find that the matrix $D_2=R_2^T\mathcal{B}^{-1}R_2$ is diagonal as well, with j-th coefficient

$$\begin{aligned}(v_2\otimes Z^j)^T\mathcal{B}^{-1}(v_2\otimes Z^j)&=\frac{1}{2}(\alpha+\beta)\sum_{k=1}^{N-1}\left(\frac{Z_1^k+Z_{N-1}^k}{6-\lambda_k}\right)^2\frac{1}{\mu_{k,j}}\\&=\frac{4(\alpha+\beta)}{N}\sum_{k=1,k\text{ odd}}^{N-1}\sin^2\left(\frac{k\pi}{N}\right)\frac{1}{(6-\lambda_k)^2\mu_{k,j}}.\end{aligned}$$

In addition, using that $\mu_{k,l}=\mu_{l,k}$ it is easy to verify that

$$\begin{cases}R_3^T\mathcal{B}^{-1}R_3=R_1^T\mathcal{B}^{-1}R_1\\R_4^T\mathcal{B}^{-1}R_4=R_2^T\mathcal{B}^{-1}R_2\end{cases} \tag{A.9}$$

and that

$$\begin{cases}R_1^T\mathcal{B}^{-1}R_2=0\\R_3^T\mathcal{B}^{-1}R_4=0.\end{cases} \tag{A.10}$$

Finally, we obtain that the matrices $M_{1,3}=R_1^T\mathcal{B}^{-1}R_3$, $M_{1,4}=R_1^T\mathcal{B}^{-1}R_4$, $M_{2,4}=R_2^T\mathcal{B}^{-1}R_4$ are given by

$$(R_1^T\mathcal{B}^{-1}R_3)_{i,j}=\begin{cases}\dfrac{4(\alpha-\beta)}{N}\dfrac{\sin\frac{i\pi}{N}\sin\frac{j\pi}{N}}{(6-\lambda_i)(6-\lambda_j)\mu_{i,j}} & \text{if } i \text{ even },j\text{ even}\\ 0\text{ if } i\text{ odd or } j\text{ odd}\end{cases} \tag{A.11}$$

$$(R_1^T\mathcal{B}^{-1}R_4)_{i,j}=\begin{cases}\dfrac{4(\alpha-\beta)^{1/2}(\alpha+\beta)^{1/2}}{N}\dfrac{\sin\frac{i\pi}{N}\sin\frac{j\pi}{N}}{(6-\lambda_i)(6-\lambda_j)\mu_{i,j}} & \text{if } i \text{ odd },j\text{ even}\\ 0\text{ if } i\text{ even or } j\text{ odd}\end{cases} \tag{A.12}$$

$$(R_2^T\mathcal{B}^{-1}R_3)_{i,j}=\begin{cases}\dfrac{4(\alpha-\beta)^{1/2}(\alpha+\beta)^{1/2}}{N}\dfrac{\sin\frac{i\pi}{N}\sin\frac{j\pi}{N}}{(6-\lambda_i)(6-\lambda_j)\mu_{i,j}} & \text{if } i \text{ even },j\text{ odd}\\ 0\text{ if } i\text{ odd or } j\text{ even}\end{cases} \tag{A.13}$$

$$(R_2^T\mathcal{B}^{-1}R_4)_{i,j} = \begin{cases} \dfrac{4(\alpha+\beta)}{N}\dfrac{\sin\frac{i\pi}{N}\sin\frac{j\pi}{N}}{(6-\lambda_i)(6-\lambda_j)\mu_{i,j}} & \text{if } i \text{ odd }, j \text{ odd }. \\ 0 \text{ if } i \text{ even or } j \text{ even} \end{cases} \tag{A.14}$$

Using that

$$(R_1^T\mathcal{B}^{-1}R_4)^T = (R_2^T\mathcal{B}^{-1}R_3), \tag{A.15}$$

it results that the $4(N-1)\times 4(N-1)$ matrix $\mathcal{R}^T\mathcal{B}^{-1}\mathcal{R}$ has the 4×4 block structure

$$\mathcal{R}^T\mathcal{B}^{-1}\mathcal{R} = \begin{bmatrix} D_1 & 0 & M_{1,3} & M_{1,4} \\ 0 & D_2 & M_{1,4}^T & M_{2,4} \\ M_{1,3} & M_{1,4} & D_1 & 0 \\ M_{1,4}^T & M_{2,4} & 0 & D_2 \end{bmatrix}, \tag{A.16}$$

Formulas (A.8, A.11, A.12, A.13, A.14) show that the computing cost for the assembling is $O(N^2)$. The capacitance matrix $I + 36\mathcal{R}^T\mathcal{B}^{-1}\mathcal{R}$ in (3.59) is therefore

$$I + 36\mathcal{R}^T\mathcal{B}^{-1}\mathcal{R} = \begin{bmatrix} I+36D_1 & 0 & 36M_{1,3} & 36M_{1,4} \\ 0 & I+36D_2 & 36M_{1,4}^T & 36M_{2,4} \\ 36M_{1,3} & 36M_{1,4} & I+36D_1 & 0 \\ 36M_{1,4}^T & 36M_{2,4} & 0 & I+36D_2 \end{bmatrix}. \tag{A.17}$$

This symmetric positive matrix has the form $I_{4(N-1)} + \tilde{Z}^T\tilde{Z} \geq I$, where

$$\tilde{Z} = 6(\mathcal{B}^{-1})^{1/2}\mathcal{R}. \tag{A.18}$$

The conjugate gradient method can be applied to (3.59). The simple diagonal preconditioning yields a linear system $\mathcal{M}g = g'$ with matrix of the form

$$\mathcal{M} = \begin{bmatrix} I_{2(N-1)} & \bar{M} \\ \bar{M}^T & I_{2(N-1)} \end{bmatrix}, \tag{A.19}$$

where $\bar{M} \in \mathbb{M}_{2(N-1)}(\mathbb{R})$ is defined by

$$\bar{M} = \begin{bmatrix} (\frac{1}{36}I + D_1)^{-1}M_{13} & (\frac{1}{36}I + D_1)^{-1}M_{14} \\ (\frac{1}{36}I + D_2)^{-1}M_{14}^T & (\frac{1}{36}I + D_2)^{-1}M_{24} \end{bmatrix}. \tag{A.20}$$

Solving (3.59) with the CG method for the matrix in (A.19) has been proved to be very efficient. Equivalently, one can split the linear system into two independent linear systems of size $2(N-1)$ and matrices $I_{2(N-1)} - \bar{M}^T\bar{M}$ and $I_{2(N-1)} - \bar{M}\bar{M}^T$, [10], each of them being solved by the CG algorithm.

The convergence analysis of the CG algorithm (see for example [13], chap.8, pp. 249 sqq.) yields that the norm of the error is reduced by a factor ε after k iterations. Here k is selected such that

$$k \geq \frac{1}{2}\sqrt{\kappa_2(I - \bar{M}\bar{M}^T)}\ln(2/\varepsilon), \tag{A.21}$$

where $\kappa_2(I - \bar{M}\bar{M}^T)$ is the condition number of $I - \bar{M}\bar{M}^T$.

The eigenvalues of $I - \bar{M}\bar{M}^T$ are

$$0 < \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{2(N-1)} \leq 1, \tag{A.22}$$

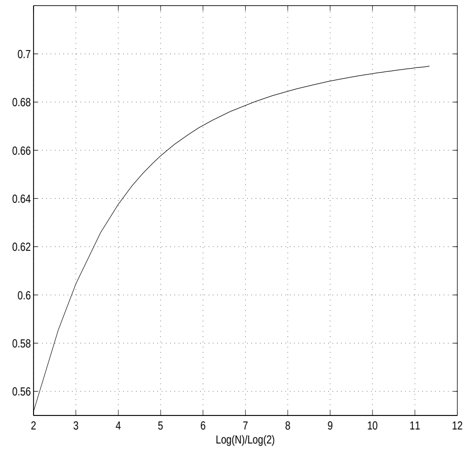

FIG. A.1. *Curve* $\mathrm{Log}_2(N) \mapsto \sigma_{\max}(\bar{M})$.

where $\lambda_k = 1 - \sigma_k^2$ and σ_k is the k^{th} singular value of $\bar{M}$. Therefore,

$$\kappa_2(I - \bar{M}\bar{M}^T) \leq \frac{1}{1 - \sigma_{\max}^2(\bar{M})}. \tag{A.23}$$

A full analytic estimate of $\sigma_{\max}(\bar{M})$ seems to be difficult to derive. Thus, we limit ourselves to the numerical study of the relation between $\mathrm{Log}_2(N)$ and $\sigma_{\max}(\bar{M})$. In Fig.1 we display the graph of $\mathrm{Log}_2(N) \mapsto \sigma_{\max}(\bar{M})$. Here N is the size of the problem and it ranges 2 to 2600. This value encompasses the number of grid points, which is usually picked for two-dimensional problems. As can be observed on Fig.1, one can infer a monotonic increase in the behaviour of $\sigma_{\max}(\bar{M})$ as a function of N, with a very slow growth for large N. The existence of a bound uniform in N , though very plausible, is not completely apparent. Anyway, we observe numerically that $\sigma_{max}(N) \leq 0.7$ for $N \leq 2600$. For a given tolerance error ε, the lower bound for the number of iterations k is independent in practice of the grid size, namely, $k \geq 0.7 \ln(2/\varepsilon)$, at least for $N \leq 2600$, (see (A.21)). Since each iteration of the CG algorithm is $O(N^2)$, we have that Step 4 in Algorithm 2 is in practice $O(N^2)$. In Tables 5.6, 5.7 we report on the number of iterations of the CG algorithm for the system (3.59) for the two specific examples. It indicates a very slow increasing behaviour of the number of iterations of the CG algorithm, corroborating Fig. 1.

REFERENCES

[1] I. Altas, J. Dym, M. M. Gupta, and R. P Manohar. Mutigrid solution of automatically generated high-order discretizations for the biharmonic equation. *SIAM J. Sci. Comput.*, 19:1575–1585, 1998.

[2] P. Amodio, J.R. Cash, G. Roussos, R.W. Wright, G. Fairweather, I. Gladwell, G.L. Kraut, and M. Paprzycki. Almost block diagonal linear systems: sequential and parallel solution techniques, and applications. *Numer. Linear Algebra with Applications*, 7:275–317, 2000.

[3] M. Ben-Artzi, J-P. Croisille, and D. Fishelov. Convergence of a compact scheme for the pure streamfunction formulation of the unsteady Navier-Stokes system. *SIAM J. Numer. Anal.*, 44,5:1997–2024, 2006.

[4] M. Ben-Artzi, J-P. Croisille, D. Fishelov, and S. Trachtenberg. A Pure-Compact Scheme for the Streamfunction Formulation of Navier-Stokes equations. *J. Comput. Phys.*, 205(2):640–664, 2005.

[5] B. Bialecki. A fast solver for the orthogonal spline collocation solution of the biharmonic Dirichlet problem on rectangles. *J. Comput. Phys.*, 191:601–621, 2003.

[6] 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.

[7] 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.

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

[9] P. Bjørstad and B.P. Tjøstheim. Efficient algorithms for solving a fourth order equation with the spectral Galerkin method. *SIAM J. Sci. Stat. Comput.*, 1997.

[10] B.L. Buzzbee 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.

[11] B.L. Buzzbee, F.W. Dorr, J.A. George, and G.H. Golub. The direct solution of the discrete Poisson equation on irregular regions. *SIAM J. Numer. Anal.*, 1971.

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

[13] P. Deuflhard and A. Hohmann. *Numerical Analysis in Modern Scientific Computing. An Introduction.* Springer-Verlag, TAM, 43, 2-nd edition, 2003.

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

[15] L. W. Ehrlich. Solving the biharmonic equation in a square: A direct versus a semidirect method. *Comm. ACM*, 16:711–714, 1973.

[16] G. Fairweather and I. Gladwell. Algorithms for almost block diagonal systems. *SIAM Review*, 46(1):49–58, 2004.

[17] D. Fishelov, M. Ben-Artzi, and J-P. Croisille. A compact scheme for the streamfunction formulation of Navier-Stokes equations. *Notes on Computer Science*, pages 809–817, 2003.

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

[19] J.W. Goodrich, K. Gustafson, and K. Halasi. Hopf bifurcation in the driven cavity. *J. Comput. Phys.*, 90:219–261, 1990.

[20] J.W. Goodrich and W. Y. Soh. Time-dependent viscous incompressible Navier-Stokes equations: The finite difference Galerkin formulation and streamfunction algorithms. *J. Comput. Phys.*, 84(1):207–241, 1989.

[21] M. M. Gupta. Direct Solution of the Biharmonic Equation using noncoupled approach. *J. Comput. Phys.*, 33:236–248, 1979.

[22] M. M. Gupta and J. C. Kalita. A new paradigm for solving Navier-Stokes equations: streamfunction-velocity formulation. *J. Comput. Phys.*, 207(2):52–68, 2005.

[23] R. Kress. *Numerical Analysis.* Graduate Texts in Mathematics 181. Springer, 1998.

[24] K. Kunz. *Numerical Analysis.* McGrawHill, 1958.

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

[26] Z.M. Lou, B. Bialecki, and G. Fairweather. Orthogonal spline collocation methods for biharmonic pronblems. *Numer. Math.*, 80:267–303, 1998.

[27] R.E. Lynch, J.R. Rice, and D.H. Thomas. Direct solution of partial difference equations by tensor product methods. *Numer. Math.*, 6:185–199, 1964.

[28] J.W. McLaurin. A general coupled equation approach for solving the biharmonic boundary value problem. *SIAM J. Numer. Anal.*, 11(1):14–33, 1974.

[29] V.V. Meleshko. Selected topics in the history of the two-dimensional biharmonic problem. *Appl. Mech. Review*, 56(1):33–85, 2003.

[30] J. Shen. Efficient Spectral-Galerkin method I. Direct solvers of second and fourth order equations using Legendre polynomials. *SIAM J. Sci. Comput.*, 15:1440–1451, 1994.

[31] J. Shen. Efficient Spectral-Galerkin method II. Direct solvers of second and fourth order equations using Chebyshev polynomials. *SIAM J. Sci. Comput.*, 16:74–87, 1995.

[32] J. Smith. The coupled equation approach to the numerical solution of the biharmonic equation by finite differences. I. *SIAM J. Numer. Anal.*, 5(2):323–339, 1968.

[33] J. Smith. The coupled equation approach to the numerical solution of the biharmonic equation by finite differences. II. *SIAM J. Numer. Anal.*, 7(1):104–111, 1970.

[34] J. W. Stephenson. Single cell discretizations of order two and four for biharmonic problems. *J. Comput. Phys.*, 55:65–80, 1984.

[35] W. Sun. Orthogonal collocation solution of biharmonic equations. *Int. Jour. Comput. Math*, 49:2221–232, 1993.

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

[37] A. Vaught. *g95* Manual. Technical report, http://www.g95.org.