An algebraic–elliptic algorithm for boundary orthogonal grid generation

Costanza Conti a, Rossana Morandi a, Rosa Maria Spitaleri b,*

a Dipartimento di Energetica, University of Florence, Via C. Lombroso 6/17, Florence 50133, Italy
b Istituto per le Applicazioni del Calcolo-CNR, Viale del Policlinico 137, Rome 00161, Italy

Abstract

To produce grids conforming to the boundary of a physical domain with boundary orthogonality features, algebraic methods like transfinite interpolating schemes can be profitably used. Moreover, the coupling of Hermite-type (also interpolating prescribed boundary direction) schemes with elliptic methods turns out to be effective to overcome the drawback of both algebraic and elliptic strategies. Thus, in this paper, we present an algorithm for the generation of boundary orthogonal grids which couples a mixed Hermite algebraic method with a boundary orthogonal elliptic scheme. Numerical tests on domains with classical geometries show satisfactory performances of the algorithm and coupling effectiveness in achieving grid boundary orthogonality and smoothness.

© 2004 Elsevier Inc. All rights reserved.

1. Introduction

It is well known that partial differential equation (PDE) systems are more and more often able to describe complex physical phenomena. On the other hand, computational processes for numerical system solution require powerful discretization methods based on the use of appropriate grids, that is a discrete set of points well representing the geometry of the definition domain. It has
been proved that grid generation strategies can strongly affect the efficiency of the whole numerical solution method of PDE systems (see for example [8,9,14] and reference quoted therein).

Generally speaking, the grids are required to satisfy some good properties like smoothness and boundary orthogonality which affect the solution accuracy [7]. In particular, orthogonality near the boundary is often necessary to achieve proper results, for example in simulation of boundary layer phenomena.

One possible approach for such a numerical grid generation is given by algebraic schemes which are the simplest methods to obtain interior grid points from boundary information via powerful interpolation methods based on transfinite interpolation [6]. More in detail, Lagrange transfinite interpolation methods provide conformity to the boundary of the physical domain and Hermite transfinite interpolation methods also provide grid lines orthogonal to the boundary both with a low computational cost. Unfortunately, these methods do not get much degrees of freedom to obtain workable meshes. Thus, they are profitably combined with tensor product techniques which, involving control points, allow us a control of the grid in the interior of the physical domain. However, algebraic grids can have some problems with smoothness, boundary overlapping and grid-line folding [14] while differential approaches avoid overlapping of coordinate lines and have a good control of the interior of the domain. Nevertheless, in case of complex geometries, the most used elliptic grid methods can be expensive with respect to human and computer time demanding. Since they usually involve iterative methods, a suitable algebraic grid as starting point for the elliptic generator can be useful to save efforts. Thus, an algebraic–elliptic coupling seems to be effective thinking of either the algebraic providing a good starting grid for the elliptic generator or the elliptic acting as smoother of the algebraic algorithm. This is the reason why in this paper we present an algebraic–elliptic algorithm which combines a mixed Hermite algebraic method with a boundary orthogonal elliptic grid generator. This way smoothness and boundary orthogonality are achieved. It should be noted that, basic ideas and preliminary results of the coupling are in [3]. Here, we highlight all the strategy steps by a detailed algorithm and we set the computational parameters.

The other four sections of the paper are organized as follows. In Section 2 we introduce the elliptic generation. In Section 3 we describe the mixed algebraic generation method that is the transfinite interpolation component and the tensor product component. Next, in Section 4 we specify the computational parameters and we present the detailed algorithm for algebraic–elliptic coupling. Finally, in Section 5 we illustrate numerical tests on four domains with classical geometries and we discuss the performances of the algorithm. The figures show grids with boundary orthogonality and smoothness in appropriate balance, computed by a reduced iteration number in the elliptic generation phase.
2. Boundary orthogonal elliptic generation

The most simple elliptic partial differential system on a given two dimensional physical domain \( \Omega \), with boundary \( \partial \Omega \) is the Laplace system with Dirichlet boundary conditions i.e.
\[
\Delta \xi = 0, \quad \Delta \eta = 0, \quad (x, y) \in \Omega, \\
\xi(x, y) = f(x, y), \quad \eta(x, y) = g(x, y), \quad (x, y) \in \partial \Omega.
\]

Nevertheless, a better control of the coordinate line distribution in \( \Omega \) can be exercised by using the elliptic generating system
\[
\Delta \xi = P(\xi, \eta), \quad \Delta \eta = Q(\xi, \eta), \quad (x, y) \in \Omega, \\
\xi(x, y) = f(x, y), \quad \eta(x, y) = g(x, y), \quad (x, y) \in \partial \Omega,
\]
where the so called "control functions" \( P \) and \( Q \) can be chosen, for example, to control the grid spacing and the orientation of the coordinate lines (see [14] for details). It is not difficult to see that, by interchanging variables, the previous system of equations in the computational domain \([0, 1]^2\) reads as
\[
\alpha \xi_{\xi\xi} - 2\beta \xi_{\eta\eta} + \gamma \xi_{\eta\eta} + J^2 (P \xi_{\xi} + Q \xi_{\eta}) = 0 \tag{1} \\
\alpha \eta_{\eta\eta} - 2\beta \eta_{\eta\eta} + \gamma \eta_{\eta\eta} + J^2 (P \eta_{\xi} + Q \eta_{\eta}) = 0 \tag{2}
\]
where
\[
\begin{align*}
\alpha &= x^2_\eta + y^2_\eta, & \beta &= x_\xi x_\eta + y_\xi y_\eta, \\
\gamma &= x^2_\xi + y^2_\xi, & J &= x_\xi y_\eta - x_\eta y_\xi. \tag{3, 4}
\end{align*}
\]

In order to guarantee the boundary orthogonality of the grid, we keep boundary points fixed and we adjust control functions, that is we deal with the so called Dirichlet orthogonality [7] approach. We choose the control functions at the boundary being as
\[
P|_{[0,1]^2} = -\frac{\alpha}{\gamma J^2} (X_{\xi\xi} \cdot X_{\xi}) - \frac{1}{J^2} (X_{\eta\eta} \cdot X_{\xi}), \\
Q|_{[0,1]^2} = -\frac{\gamma}{\alpha J^2} (X_{\eta\eta} \cdot X_{\eta}) - \frac{1}{J^2} (X_{\xi\xi} \cdot X_{\eta}), \tag{5}
\]
where we used the notations \( X(\xi, \eta) = (x(\xi, \eta), y(\xi, \eta)) \) and \( \partial [0, 1]^2 \) for the boundary of the computational domain.

As we will see in Section 4, these orthogonal control functions are evaluated at the boundary by the help of ghost points and defined all over \([0, 1]^2\) via transfinite interpolation [14].

With an obvious meaning of the symbols, the differential system (1) and (2), along with the associated boundary conditions can be written in a more compact form as
\[ LX = 0 \quad \text{on} \quad [0, 1]^2 \quad AX = \Phi \quad \text{on} \quad \partial [0, 1]^2 \]

so that its discrete version becomes

\[ L^h X^h = 0 \quad \text{on} \quad \mathcal{G}^h, \quad A^h X^h = \Phi^h \quad \text{on} \quad \Gamma^h, \quad (6) \]

where \( h = (h_x, h_y) \) is the meshsize of the chosen discretization. Now, let \( X_0^h \) be an initial approximation of the solution \( X^h \) of the problem (6) on the grid \( \mathcal{G}^h \). In order to obtain the required approximated solution, say \( X^h \), we can use a one grid algorithm, even as a base for future extension to multigrid computation [12], which applies an appropriate number \( v \) of sweeps of a relaxation procedure \( \mathcal{R} \) suitable for nonlinear problems [11,13], that is

\[ X^h = \mathcal{R}^v (X_0^h; L^h, A^h, \Phi^h). \quad (7) \]

Obviously, the goodness of the solution of the elliptic grid generator (6) depends on the required grid fineness and on the domain geometry.

3. Transfinite orthogonal algebraic generator

Let \( \Omega \subset \mathbb{R}^2 \) be such that \( \partial \Omega = \bigcup_{i=1}^4 \partial \Omega_i \), with \( \partial \Omega_1 \cap \partial \Omega_2 = \emptyset \) \( \partial \Omega_2 \cap \partial \Omega_4 = \emptyset \) where \( \partial \Omega_1, \partial \Omega_2, \partial \Omega_3, \partial \Omega_4 \) are the supports of four regular curves \( \gamma_i : [0, 1] \rightarrow \partial \Omega_i, \quad i = 1, \ldots, 4 \), taken counterclockwise. Furthermore, we assume that the curve intersections occur only at the end points of the boundary curves \( \gamma_i, \quad i = 1, \ldots, 4 \) i.e.

\[ \gamma_1(0) = \gamma_4(1), \quad \gamma_1(1) = \gamma_2(0), \quad \gamma_2(1) = \gamma_3(0), \quad \gamma_4(0) = \gamma_3(1). \]

We define \( \phi_1(\xi) := \gamma_1(\xi), \quad \phi_2(\xi) := \gamma_3(1 - \xi), \quad \xi \in [0, 1] \) and \( \psi_1(\eta) := \gamma_4(1 - \eta), \quad \psi_2(\eta) := \gamma_2(\eta), \quad \eta \in [0, 1] \) and four additional curves by computing the derivatives of the \( \phi \) and \( \psi \)-curves, i.e.

\[ \phi_{i+2}(\xi) = \frac{\mathcal{H}^i_{2+2}}{\| \phi_i \|_2} \left( -\left( \phi_i'(\xi) \right)', \left( \phi_i'(\xi) \right)' \right), \quad i = 1, 2, \]
\[ \psi_{j+2}(\eta) = \frac{\mathcal{H}^j_{2+2}}{\| \psi_j \|_2} \left( -\left( \psi_j'(\eta) \right)', \left( \psi_j'(\eta) \right)' \right), \quad j = 1, 2, \quad (8) \]

denoting by \( \phi^x, \phi^y \) and \( \psi^x, \psi^y \) the components of the \( \phi \)-curves and \( \psi \)-curves, respectively. The symbol \( \| \cdot \|_2 \) stands for the Euclidean norm and \( \mathcal{H}^i, \mathcal{H}^j, \quad l = 3, 4 \) are suitable constant values (see Section 4 for a specific definition). As we are going to deal with orthogonal lines emanating from the boundary of the domain, we assume the following conditions on the boundary curves
where $u_1 = 0$, $u_2 = 1$.

Then, we introduce the linear operators

$$P_1[\phi](\xi, \eta) := \sum_{i=1}^{4} \alpha_i(\eta) \phi_i(\xi), \quad P_2[\psi](\xi, \eta) := \sum_{j=1}^{4} \alpha_j(\xi) \psi_j(\eta),$$

$$P_1P_2[\phi, \psi](\xi, \eta) := \sum_{i=1}^{2} \left( \alpha_i(\eta) P_2[\psi](\xi, u_i) + \alpha_{i+2}(\eta) \frac{\partial P_2[\psi](\xi, u_i)}{\partial \eta} \right).$$

The functions $\alpha_j(\xi)$, $j = 1, \ldots, 4$ and $\alpha_i(\eta)$, $i = 1, \ldots, 4$, in (10) are the so-called blending functions satisfying cardinal interpolation

$$\alpha_{j+1}(l) = \delta_j, \quad \alpha'_{j+1}(l) = 0, \quad j = 0, 1, \quad l = 0, 1,$$

$$\alpha_{j+3}(l) = 0, \quad \alpha'_{j+3}(l) = \delta_j, \quad j = 0, 1, \quad l = 0, 1.$$

Thus, the Hermite blending function surface is

$$(P_1 \oplus P_2)[\phi, \psi](\xi, \eta) = P_1[\phi](\xi, \eta) + P_2[\psi](\xi, \eta) - P_1P_2[\phi, \psi](\xi, \eta),$$

where $(P_1 \oplus P_2)$ is the so-called Boolean sum operator. As well known, because of the blending function properties, it holds

$$(P_1 \oplus P_2)(u_j, \eta) = \psi_j(\eta), \quad j = 1, 2, \quad (P_1 \oplus P_2)(\xi, u_i) = \phi_i(\xi), \quad i = 1, 2,$$

$$\frac{\partial (P_1 \oplus P_2)(u_{j-2}, \eta)}{\partial \xi} = \psi_j(\eta), \quad j = 3, 4,$$

$$\frac{\partial (P_1 \oplus P_2)(\xi, u_{i-2})}{\partial \eta} = \phi_i(\xi), \quad i = 3, 4.$$

3.1. Mixed orthogonal algebraic method

As already noticed, transfinite method do not get much degrees of freedom to obtain workable meshes. Thus, as suggested in [4] and further investigated in [5,10], they can be profitably combined with tensor product techniques which, involving control points, allow us a control of the grid on the interior of the domain $\Omega$. Thus, the idea is to work with blending functions $\alpha_i(\xi)$, $i = 1, \ldots, 4$, $\alpha_j(\eta)$, $j = 1, \ldots, 4$, with local support on $[0, 1]$. Here, they are defined as the dilated versions of the classical Hermite bases with support on $I_0^\xi := [0, \tilde{u}_\xi]$ and on $I_1^\xi := [1 - \tilde{u}_\xi, 1]$ being $0 < \tilde{u}_\xi < 1$ and $0 < \tilde{u}_\xi < 1$, i.e.
\[ x_1(\xi) := \left(1 + 2 \frac{\xi}{u_\xi}\right) \left(1 - \frac{\xi}{u_\xi}\right), \quad x_3(\xi) := \xi \left(1 - \frac{\xi}{u_\xi}\right)^2, \quad \xi \in I_0^x; \]

\[ x_2(\xi) := \left(3 - 2 \frac{\xi + \bar{u}_\xi - 1}{\bar{u}_\xi}\right) \left(\frac{\xi + \bar{u}_\xi - 1}{\bar{u}_\xi}\right)^2, \quad (13) \]

\[ x_4(\xi) := (\xi - 1) \left(\frac{\xi + \bar{u}_\xi - 1}{\bar{u}_\xi}\right)^2, \quad \xi \in I_1^x. \]

The blending functions \( x_j(\eta), j = 1, \ldots, 4 \) are analogously defined with support on \( I_0^x := [0, \bar{u}_\eta] \) and on \( I_1^x := [1 - \tilde{u}_\eta, 1] \). Next, we define the linear transformation \( X : [0,1]^2 \rightarrow \mathbb{R}^2, X(\xi, \eta) = (x(\xi, \eta), y(\xi, \eta)) \) as

\[ X(\xi, \eta) := T_p(\xi, \eta) + (P_1 \oplus P_2) ([\phi, \psi] - T_p)(\xi, \eta), \quad (14) \]

where

\[ T_p(\xi, \eta) := \sum_{j=1}^{m_p} \sum_{j=1}^{n_p} Q_{ij} B_{i3}(\xi) B_{j3}(\eta) \quad (15) \]

with \( B_{i3}(\xi) \) and \( B_{j3}(\eta) \) denoting the usual cubic B-splines with knots \( \{\xi_{i-2}, \xi_{i-1}, \xi_i, \xi_{i+1}, \xi_{i+2}\}, \{n_{j-2}, n_{j-1}, n_j, n_{j+1}, n_{j+2}\} \), respectively. The set \( \mathcal{Q} = \{Q_{ij}\}_{i,j=1}^{m_p,n_p} \) is the set of control points. For the B-spline knot choice and the control point choice the reader can refer to [1,2].

The Boolean sum operator \( (P_1 \oplus P_2) \) in (14) is also acting on the surface \( T_p(\xi, \eta) \) taking into account the eight boundary curves \( T_p(0, \eta), T_p(1, \eta), T_p(\xi, 0), T_p(\xi, 1), \frac{\partial T_p(0, \eta)}{\partial \xi}, \frac{\partial T_p(1, \eta)}{\partial \xi}, \frac{\partial T_p(\xi, 0)}{\partial \eta}, \frac{\partial T_p(\xi, 1)}{\partial \eta} \).

As discussed in [2], \( X \) satisfies

\[ X(u_j, \eta) = \psi_j(\eta), \quad j = 1, 2, \quad X(\xi, u_i) = \phi_i(\xi), \quad i = 1, 2, \]

\[ \frac{\partial X(u_j, \eta)}{\partial \xi} = \psi_j(\eta), \quad j = 3, 4, \quad \frac{\partial X(\xi, u_i)}{\partial \eta} = \phi_i(\xi), \quad i = 3, 4. \quad (16) \]

Moreover, setting \( \bar{u}_\xi = \bar{u}_3 - \xi_2, \quad \bar{u}_\eta = n_3 - n_2, \quad \bar{\xi} = \xi_{m-1} - \xi_{m-2}, \quad \bar{\eta} = n_{m-1} - n_{m-2} \), because of the blending function locality, it holds \( X(\xi, \eta) = T_p(\xi, \eta) \) for all \( (\xi, \eta) \in [\xi_3, \xi_{m-2}] \times [n_3, n_{m-2}] \).

The grid \( X_0^k \) is then obtained by sampling \( X \) at a given set of parameter values \( \{(\xi_i, \eta_j)\}_{i,j=1}^{m_p,n_p} \) i.e.

\[ X_0^k := \{X(\xi_i, \eta_j) = (x(\xi_i, \eta_j), y(\xi_i, \eta_j))\}_{i,j=1}^{m_p,n_p}. \]
4. Numerical algorithm and computational parameters

Before to present the algorithm for constructing the algebraic–elliptic grid with orthogonal feature to the boundary, we need to make clearer how to set some of the parameters introduced in the previous sections.

4.1. Choice of B-splines knots

Concerning the knots of the B-splines involved in (15), that is \( \xi \) and \( \eta \), they are set with the property

\[
0 = \xi_{-1} = \xi_0 = \xi_1 = \xi_2 < \cdots < \xi_{m_p-1} = \xi_{m_p} = \xi_{m_p+1} = \xi_{m_p+2} = 1,
\]

with \( \xi_{i+1} - \xi_i = \frac{1}{m_p - 3}, \ i = 2, \ldots, m_p - 2. \) (17)

Similarly,

\[
0 = \eta_{-1} = \eta_0 = \eta_1 < \eta_2 < \cdots < \eta_{n_p-1} = \eta_{n_p} = \eta_{n_p+1} = \eta_{n_p+2} = 1,
\]

with \( \eta_{l+1} - \eta_l = \frac{1}{n_p - 3}, \ l = 2, \ldots, n_p - 2. \) (18)

4.2. Choice of the modulus of the boundary normal derivatives

Now, we discuss about the constants \( \mathcal{K}_i \), \( \mathcal{K}_j \), \( i = 1, 2 \) to be used in (8). By using the so called uniformity property (see [2] for a detailed discussion about this property) we arrive at the following choice

\[
\mathcal{K}_i = \frac{\sum_{r=3}^{m_p-2} (\psi_2^r(\xi_r) - \psi_1^r(\xi_r)) ||\phi_j||_2}{-\sum_{r=3}^{m_p-2} (\phi_1^r(\xi_r))'}, \ i = 3, 4
\] (19)

and

\[
\mathcal{K}_j = \frac{\sum_{r=3}^{n_p-2} (\psi_2^r(\eta_r) - \psi_1^r(\eta_r)) ||\psi_j'||_2}{\sum_{r=3}^{n_p-2} (\psi_j'(\eta_r))'}, \ j = 3, 4.
\] (20)

4.3. Choice of the control points

Let us consider the bilinear blending surface interpolating the four boundary curves

\[
L(s,t) = (1-t)\phi_1(s) + t\phi_2(s) + (1-s)\psi_1(t) + s\psi_2(t) - (1-t)((1-s)\psi_1(0)
+ s\psi_2(0)) - t((1-s)\psi_1(1) + s\psi_2(1))
\] (21)
and the parameter values, in $[0, 1]$ \{\sigma_i\}_{i=1}^{mp}$ and \{\tau_j\}_{j=1}^{np} defined as

$$
\begin{align*}
\sigma_1 &:= 0, \quad \sigma_2 := \sigma_1 + \frac{2}{3(m_p - 2)}, \quad \sigma_3 := \sigma_2 + \frac{3}{3(m_p - 2)}, \\
\sigma_{i+1} &:= \sigma_i + \frac{3}{3(m_p - 2)}, \quad i = 3, \ldots, m_p - 3, \\
\sigma_{mp-1} &:= \sigma_{mp-2} + \frac{2}{3(m_p - 2)}, \quad \sigma_{mp} := \sigma_{mp-1} + \frac{1}{3(m_p - 2)} = 1, \\
\tau_1 &:= 0, \quad \tau_2 := \tau_1 + \frac{2}{3(n_p - 2)}, \quad \tau_3 := \tau_2 + \frac{3}{3(n_p - 2)}, \\
\tau_{j+1} &:= \tau_j + \frac{3}{3(n_p - 2)}, \quad j = 3, \ldots, n_p - 3, \\
\tau_{np-1} &:= \tau_{np-2} + \frac{2}{3(n_p - 2)}, \quad \tau_{np} := \tau_{np-1} + \frac{1}{3(n_p - 2)} = 1.
\end{align*}
$$

The control points are obtained by sampling $L(s, t)$, at the previous parameter values that is

$$
Q_{ij} = L(\sigma_i, \tau_j), \quad i = 1, \ldots, m_p, \quad j = 1, \ldots, n_p.
$$

4.4. Choice of the ghost points

In order to impose boundary orthogonality to the elliptic generator of the algorithm, we exploit the boundary orthogonal grid constructed by the defined mixed Hermite method in step 2 to

(i) set the boundary grid point distribution as Dirichlet conditions for the problem (6) in step 3.1 (even taking into account physical boundary conditions from step 1 on),

(ii) compute the orthogonal control functions (5) in step 3.3.

The algebraic grid is assumed to have the first interior coordinate lines correctly positioned so that appropriate exterior curves of ghost points can be derived by orthogonal symmetry (algorithm—step 3.2). Leaving the ghost points unchanged allows us to use the information on boundary orthogonality in (5) and save it during the iterations. Permitting free positions to all the interior points allows us to achieve smoothness. Orthogonality at corner points is not imposed.

**Algorithm**

1. Input the boundary curves $\phi_1, \phi_2, \psi_1, \psi_2$, the grid size $m_g, n_g$, the number of control points $m_p, n_p$, the tolerances $\tau_d, \tau_r$ and $\tau_o$ and the max iteration number $n_{\text{max}}$
2. Construct the algebraic grid through the following steps:
   2.1. compute the orthogonal directions $\phi_3, \phi_4, \psi_3, \psi_4$ by means of (8) and (19), (20)
   2.2. compute the control points $\{Q_{ij}\}_{i,j=1}^{m,n}$ with the strategy in (23)
   2.3. compute the knots of the B-splines as in (17) and (18)
   2.4. use (14) to generate the mixed algebraic surface $\hat{X}(\hat{\xi}, \hat{\eta})$
   2.5. compute the discrete grid $X^h := \{X(\xi_i, \eta_j)\}_{i,j=1}^{m,n}$ with fineness $h = (h_\xi, h_\eta)$, where $h_\xi := \frac{1}{m-1}$, $h_\eta := \frac{1}{n-1}$ and $\xi_i := ih_\xi$, $\eta_j := jh_\eta$ for $i, j = 1, \ldots, m, n$.
   2.6. set the initial grid $X^h_0 := X^h$

3. Compute a smoother grid by the elliptic procedure:
   3.1. set prescribed boundary grid points as Dirichlet boundary conditions for the problem (6), that is $\Phi^h := \{X^h_0(1,j), X^h_0(m,g,j), X^h_0(i,1), X^h_0(i,n_g)\}_{i,j=1}^{m,g,n_g}$
   3.2. construct an exterior curve of points by orthogonal symmetry from $\{X^h_0(2,j), X^h_0(m_g - 1,j), X^h_0(i,2), X^h_0(i,n_g - 1)\}_{i,j=2}^{m_g - 1,n_g - 1}$
   3.3. compute $P^l|_{[0,1]^2}$ and $Q^l|_{[0,1]^2}$ using (5)
   3.4. extend $P$ and $Q$ on $[0,1]^2$ by bilinear transfinite interpolation
   3.5. Solve the system (6) by Gauss–Seidel relaxation procedure:
      3.5.1. for $k = 1, \ldots, n_{\text{max}}$
        * solve $X^h_k = R^k(X^h_{k-1}, L^h, A^h, \Phi^h)$ where $R^k$ is the $k$th iteration of the GS-relaxation procedure
        * compute $d_k := ||X^h_k - X^h_{k-1}||_2$ and the residual $r_k := ||L^h X^h_k||_2$
        * compute the vector $\text{Ort}_k$ containing the scalar product of the discrete boundary curves and the grid lines at the boundary
        * compute $\rho_k := ||\text{Ort}_k||_2$
        * if $d_k \leq \tau_d$ & $r_k \leq \tau_r$ & $\rho_k \leq \tau_o$ go to 4
   4. Set $\nu := k$ and $X^h_\nu := X^h_k$
   5. Print the final grid $X^h_\nu$ and the iteration number $\nu$.

5. Numerical results

In this section we present some numerical results to put in evidence the performance of coupling algebraic and elliptic generators. In particular, we compare the results obtained by using either the quite commonly used bilinear transfinite method (BTM) or the mixed Hermite method (MHM) to provide the starting grid $X^h_0$ for the elliptic generator with the orthogonal control functions (5). In order to generate BTM-elliptic grids, steps 1, 3–5 of the algorithm defined in the previous section are unchanged, while step 2 is substituted by
2. Construct the algebraic grid by the step:

2.1. compute the discrete grid $X^h := \{L(\zeta_i, \eta_j)\}_{i,j=1}^{m_x n_x}$, with fineness $h = (h_{\zeta}, h_{\eta})$, where $h_{\zeta} := \frac{1}{m_x-1}, h_{\eta} := \frac{1}{n_x-1}$ and $\zeta_i := ih_{\zeta}, \eta_j := jh_{\eta}$ for $i, j = 1, \ldots, m_x, n_x$ by means of (21)

2.2. set the initial grid $X_0^h := X^h$.

It is worthwhile to notice that in this case the algebraic grid gives no information on boundary orthogonality since the points of the first interior line are not correctly located, however step 3.2 is still carried out by recovering orthogonal directions to the boundary and computing ghost points again by orthogonal symmetry [7].

Commonly used geometries have been selected as test problems [7,14]. For the four displayed domains, we show the initial grid $X_0^h$ (on the left) and the final grid $X^h$ (on the right). As specified in the algorithm steps we assume three stop criteria. The control over the residual, the relative difference of two

![Fig. 1. BTM-elliptic grids, $v = 37, \tau_d = 10^{-3}, \tau_r = 10^{-3}, \tau_o = 10^{-4}$.](image1)

![Fig. 2. MHM-elliptic grids, $v = 11, \tau_d = 10^{-3}, \tau_r = 10^{-3}, \tau_o = 10^{-4}$.](image2)
consecutive iterations and the control of the orthogonality at the boundary have been used to evaluate the results. The specific values of the used tolerance

\[ \tau_d = 10^{-3}, \tau_e = 10^{-3}, \tau_o = 10^{-2}. \]

\[ \tau_d = 10^{-3}, \tau_e = 10^{-3}, \tau_o = 10^{-2}. \]

\[ \tau_d = 10^{-3}, \tau_e = 10^{-4}, \tau_o = 10^{-3}. \]
(τ_d, τ_r, τ_o respectively) and the iteration numbers (v), spent to reach all the three assigned tolerance values balancing smoothness and orthogonality requests, are given in the captions of Figs. 1–8.
6. Conclusions

Grid generation is a fundamental step in the numerical solution of PDEs. We believe that coupling advanced algebraic methods with elliptic generation leads to advantages in achieving grid properties. The algebraic–elliptic algorithm presented in this paper provides grids with balanced boundary orthogonality and smoothness by a reduced number of smoothing iterations. Algorithm performances, since promising, suggest several development directions. Open investigations deal with both geometric extensions to volume or surface generation and enlarging computational capabilities to multiblock or multigrid computation.

References