Università degli Studi di Catania

Facoltà di Scienze Matematiche, Fisiche e Naturali

Dipartimento di Matematica e Informatica

Dottorato di Ricerca in Matematica e Informatica

XXXI Ciclo

Ghost-point methods for Elliptic and Hyperbolic Equations

Supervisor

Prof. Giovanni Russo

Candidata

Santina Chiara Stissi

Anno Accademico 2017-2018

A Irene

"Chi ha coraggio di ridere, è padrone del mondo"
G. Leopardi

Acknowledgments

Sono giunta alla fine di questo meraviglioso percorso, che ha accompagnato in modo intenso questi ultimi 3 anni. Ci sono stati momenti faticosi ma anche momenti di soddisfazione per la conquista di ogni piccolo tassello da aggiungere al risultato finale.

Ringrazio profondamente il mio tutor prof. Giovanni Russo per l'aiuto fornitomi in tutti questi anni e per la grande conoscenza e la passione per la ricerca che mi ha trasmesso. Ringrazio Armando Coco sempre pronto ad aiutarmi e a chiarire ogni mio dubbio.

Ringrazio i miei genitori e mia sorella, mio punto di riferimento, che con il loro instancabile sostegno mi hanno permesso di arrivare fin qui.

Ringrazio i miei amici/colleghi per il loro supporto durante i momenti di sconforto, per il loro valido aiuto durante l'attività di ricerca e per le gioie condivise. Grazie a Daniele, Lorenzo, Nicola, Giovanni, Greta, Clarissa, Gabriella e Beatrice.

Contents

Introduction

The work of this thesis is devoted to the elliptic solvers in arbitrary regions on a regular Cartesian grid. In particular, the thesis concerns the analysis and the application to Gas Dynamics, of the Coco-Russo method [9, 10], which represents a generalization of the finite-difference method for the elliptic equations on arbitrary domains.

We are interested in the numerical study of elliptic equations in arbitrary domains because they arise in a wide variety of applications, such as diffusion phenomena, fluid dynamics, charge transport in semiconductors, crystal growth, electromagnetism.

Depending on the elliptic problem we want to solve it is possible to impose several kind of boundary conditions (Dirichlet, Neumann, mixed). Because of the arbitrariness of the domain, if it is immersed in a grid, the border may not be aligned with the grid itself. Special treatment is therefore required to impose at discrete level the boundary conditions and in this regard several techniques have been developed.
In the Finite Element Methods (FEM) [25], discretization is carried out through the creation of a grid composed of finite elements of coded form (triangles and quadrilaterals for 2D domains, tetrahedra and hexahedrons for 3D domains) to fit better the boundary. However, this method has disadvantages. In the case in which we consider moving boundary, a regeneration of the grid is necessary in each temporal phase, which makes the method expensive. For this reason a Cartesian grid method is preferred along with a

level-set approach [23, 27] to keep track of the boundary in each time step.
There are methods that make use of irregular Cartesian grids. For example, in the Shortley-Weller discretization [30], the authors discretize the Laplacian operator with the usual central difference, which makes use of a symmetrical stencil, for the internal points away from the border. For internal points near the boundary, the symmetrical stencil is replaced by a non-symmetrical stencil. In this case the boundary conditions are imposed on the points coming from the intersection of the grid with the boundary of the domain. The Shortley-Weller method produces second-order accurate solutions for the Dirichlet problem, but it has a limit. Indeed, it cannot be immediately applied in presence of Neumann boundary conditions. In [13] the author develop a second-order method for the variable coefficient Poisson equation on an irregular domain with a simple discretization that leads a symmetric matrix. The value at the ghost nodes is assigned by linear extrapolation. This method, however, is valid only for Dirichlet problems.
We present the method in [9, 10], in which the authors, who use a moving boundary, propose a technique that makes use of a regular Cartesian grid, in order to overcome the limits present in [25]. Unlike the method of Shortley-Weller, to maintain the symmetry of the stencil even for internal points close to the boundary, they use extra grid points, called ghost points because they do not belong to the domain but are involved in the discretization of the elliptic equation in question. For each ghost point we consider its projection F on the border of the domain. It is precisely in the point F that we impose the boundary conditions through a polynomial interpolation procedure [22], whose interpolation error can influence the accuracy order of the method in the case in which we impose mixed boundary conditions. Various numerical tests confirm the convergence of the numerical method in one, two and three dimensional cases. Hence the idea of applying the method to the numerical schemes of Gas Dynamics to solve Euler equations in domains with obstacles with complex geometries.
Already, in [8] the authors construct a explicit finite-difference shock-capturing scheme for the numerical solution of the Euler equation of gas dynamics on domain with circular obstacle. The method is based on discretizing the Euler equation on a regular Cartesian grid. However, this method is affected by the spatial restriction imposed by the CFL condition to guarantee the stability of the numerical method.
In [3] the authors propose a family of simple second order accurate semi-

implicit schemes for the numerical solution of Euler equations of gas dynamics on bidimensional domains in absence of obstacles. These methods are (linearly) implicit in the acoustic waves and thus they eliminate the acoustic CFL restriction on the time step. The idea leading to the construction of these semi-implicit schemes is that in the low Mach number regimes the acoustic waves carry a negligible amount of energy and thus have a negligible influence on the solution, but impose a very restrictive CFL condition if one uses an explicit scheme to solve them. A implicit scheme to solve the system would be more difficult to solve and would have the drawback to introduce an excessive dissipation on the slow waves. Therefore, a semi-implicit scheme avoids the CFL condition for the acoustic waves and maintains a good accuracy on the more important features of the flow. In this method the differential operators in space relative to convective or material fluxes are explicitely discretized with Lax-Friedrichs fluxes and the operators relative to acoustic waves are discretized implicitely with central differences.
These two works [3, 8] have suggested us the construction of a semi-implicit method for Euler equations on bidimensional domain in presence of obstacles with complex geometries, eliminating the restriction on the time step present in [8]. We have also provided an alternative technique to the iterative method proposed in [8], which allows us to impose the boundary conditions of the obstacle by solving a linear system.

As already mentioned, in both applications (elliptic problems and Gas Dynamics) the spatial discretization is performed on a uniform Cartesian grid, while the boundaries can have an arbitrary shape.

The thesis is therefore divided into two parts. In the first part (Chapter 1, Chapter 2 and Chapter 3) there is an analysis of the convergence of the method applied to the Poisson equation, while in the second part (Chapter 4 and Chapter 5) the method is applied at the Semi-Implicit schemes for Gas dynamics on domains with obstacles.

The structure of the thesis is as follows.

Chapter 1 is a description of the Coco-Russo method in one, two and three spatial dimensions. To impose the boundary conditions on the projection of the Ghost points we use the interpolation procedures, which can be linear or

quadratic. We have impose several kinds of boundary conditions: Dirichlet boundary conditions or mixed boundary conditions. The interpolation error is computed in all cases. At the end of this first chapter, we also presented how it is possible to symmetrize the discretization matrix and how the symmetrized matrix can somehow influence the stability of the numerical method.

In **Chapter 2** we present the numerical results both for the Dirichlet problem and for the mixed problem, whose aim is to highlight the stability and convergence characteristics of the numerical method. For each problem (Dirichlet or mixed) were compared the numerical results obtained by performing linear interpolations and quadratic interpolations to impose the boundary conditions. Therefore, several features are studied with different norms (L^1, L^2 and L^∞): the asymptotic behavior of the norm of the inverse matrix of the method, of its spectral radius and of the inverse of the Schur complement of the matrix to highlight the stability characteristics of the Coco-Russo method, the asymptotic behavior of the consistency error τ_h and of the error e_h to highlight the characteristics of consistency and convergence of the numerical method, respectively. For Dirichlet problems the Coco-Russo method is second-order accurate. Depending on the discretization of the boundary conditions for mixed problems (linear or quadratic interpolation) the Coco-Russo method can be first or second-order accurate.
In 2D we have chosen two different arbitrary domains: a square, that represents a simple extension of the one-dimensional case and a circular domain. In 3D we have chosen only a spherical domain, that it is an extension of the 2D circular domain.

The **Chapter 3** consists in a theoretical convergence analysis of the Coco-Russo method. The consistency error are computed both for Dirichlet problems and mixed problems with linear and quadratic interpolations in one, two and three spatial dimensions. In one-dimensional space we present two convergence proofs of the numerical method for the Dirichlet problem, one of which makes use of M-matrices, which confirm the second order accuracy, and a proof which makes use of Toeplitz matrices which confirms the stability of the Coco-Russo method in ∞-norm, already highlighted by the numerical results presented in Chapter 2. We have proved instead that in 2D the discretization matrix is not a M-matrix as in the one-dimensional case. Thus the stability and convergence of the Coco-Russo method in 2D is still

an open problem.

Chapter 4 is devoted to a review on numerical methods for hyperbolic systems of conservation laws. In particular, we stressed the importance of constructing non-oscillatory semi-discrete schemes for Isentropic and Compressible Gas Dynamics. At the end of this chapter we also present an application of the Coco-Russo method for the construction of a finite-difference semi-discrete scheme for Euler equations on a two-dimensional domain with obstacle [8]. However, these schemes are explicit and therefore suffer from the restriction on the time step imposed by the CFL condition to guarantee the stability of the numerical method.

In **Chapter 5** we present the idea of [3] for the construction of Semi-Implicit Methods for the Gas Dynamics that do not suffer the usual CFL stability restriction of explicit schemes. Various numerical tests present in [6, 8, 20] have been presented to highlight the characteristics of the method. Furthermore, we have implemented a semi-implicit version of the method proposed in [8] and we presented numerical results first considering the simplest case of a square obstacle and then the more complex case of a circular obstacle.

1

Coco-Russo method for the Poisson Equation

In this Chapter we deal with the description of the Coco-Russo method in arbitrary regions on a regular Cartesian grid in one, two and three-dimensional cases. As we will see, this method uses a polynomial interpolation technique to impose the boundary conditions. For each problem (Dirichlet or mixed) we describe the Coco-Russo method both by performing the linear interpolation procedure and by performing the quadratic interpolation procedure, calculating the interpolation errors, which will help us, as we will see in Chapter 3, to evaluate the consistency of the numerical method.

One of the advantages of using Cartesian grids lies in the fact that second-order accuracy can be achieved in a simple way. In the case of a moving domain is very useful to use regular Cartesian grids. Indeed, for example, in the Finite Element Methods the discretization is carried out through the creation of a grid composed of finite elements of coded form to fit better the boundary. But, in presence of moving boundaries, a grid regeneration is needed at each temporal phase, which makes the method expensive. For this reason a Cartesian grid method is preferred together with a level-set approach to keep track of the boundary at each time step. However, in this work we have considered the case of a fixed domain.

We introduce the theory for the elliptic operators [15].

Elliptic operators

Let L be a linear real second order partial differential operator defined on a bounded and connected open subset Ω of $\mathbb{R}^d$ $(d = 1, 2, \ldots)$ with boundary

$\Gamma = \partial\Omega$. For any $u \in \mathrm{C}^2(\Omega)$, we define

$$Lu(x) = -\sum_{i,j=1}^{d} a_{ij}(x)\frac{\partial^2 u}{\partial x_i \partial x_j}(x) + \sum_{i=1}^{d} a_i(x)\frac{\partial u}{\partial x_i}(x) + a(x)u(x), \text{ for all } x \in \Omega, \quad (1.1)$$

for some real-valued functions a_{ij} $(1 \leq i, j \leq d)$, a_i $(1 \leq i \leq d)$ and a defined over Ω.
In the case in which L is the elliptic operator: $Lu(x) = -\Delta u(x)$, and thus in (1.1) we assume: $a_{ij}(x) = 0$ $(i \neq j)$, $a_i(x) = 0$ and $a(x) = 0$.

To guarantee the uniqueness of the solution of the Dirichlet elliptic problems

$$\begin{cases} Lu(x) = f(x) & x \in \Omega, \\ u(x) = g(x) & x \in \Gamma \end{cases},$$

with f and g assigned functions over Ω, it is necessary that L satisfies the *maximum principle*: given any $\phi \in \mathrm{C}^0(\bar{\Omega}) \cap \mathrm{C}^2(\Omega)$ with $L\phi \leq 0$, for all $x \in \Omega$, it follows that

$$\max\left\{\phi(x); x \in \bar{\Omega}\right\} \leq \max\left\{0, \max\left\{\phi(x); x \in \Gamma\right\}\right\},$$

and the solution, if it exists, is the function $\phi \in \mathrm{C}^0(\Gamma) \cap \mathrm{C}^2(\Omega)$ given by

$$\phi(x) = \int_{\Omega} G(x,\xi) f(\xi) d\xi + \int_{\Gamma} G_{\partial}(x,\xi) g(\xi) d\sigma_{\xi}, \ x \in \bar{\Omega},$$

where G e G_{∂} are the associated Green's functions.

There is a discrete analogue of the maximum principle, known as the *discrete maximum principle*, which ensures the stability of the finite-difference method for the Dirichlet elliptic problems on non-arbitrary two-dimensional domains. For more details see [21].

We present now the Coco-Russo method suitable for arbitrary domains.

Coco-Russo method

Let $d \geq 1$ an integer, the prototype of elliptic equations is the *Poisson equation*:

$$-\Delta u = f \text{ in } \Omega \subset \mathbb{R}^d,$$

where $u : \Omega \longrightarrow \mathbb{R}$ and $f : \Omega \longrightarrow \mathbb{R}$ are the unknown function and the source respectively, and in which we can impose several boundary conditions:

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ Bu = g & \text{in } \Gamma \end{cases}, \tag{1.2}$$

where $g : \Gamma \longrightarrow \mathbb{R}$ is an assigned function and B is an operator that depends on the boundary conditions we impose.

We are interested in studying the numerical solution u_h of the problem (1.2), applying the *Coco-Russo method*. The Coco-Russo method is a finite-difference method modified in the case of an arbitrary domain. We describe the numerical method below.

We discretize the domain Ω with a regular grid consisting of the point $\mathrm{P}_{i_1 i_2 \ldots i_d}$ called *grid points* and characterized by a spatial step h along each axis of $\mathbb{R}^d$. We assume that $\Omega \subset [0,1]^d$. The following figure (Figure 1.1) shows the discretization of a domain Ω in two spatial dimensions. The black points are the grid points.

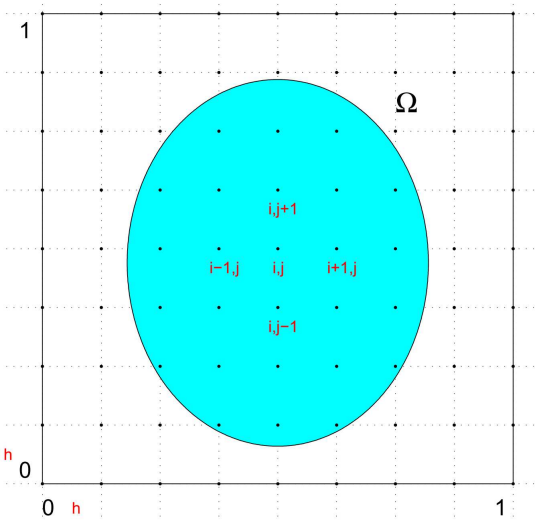
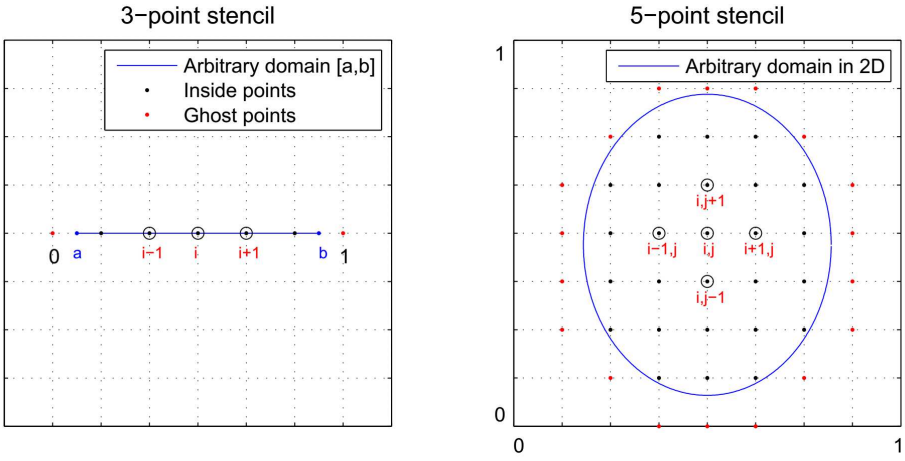


Figure 1.1: Discretization of a domain $\Omega \subset [0,1]^2$, choosing $h = 0.1250$. The grid consisting of the point $\mathrm{P}_{ij} = (x_i, y_j)$ and $h = x_{i+1} - x_i = y_{j+1} - y_j$, for all i, j

Each grid point inside the domain Ω is called *internal point*. For each internal points we consider its $2d$ neighbors. For example in the Figure 1.1 the internal point $\mathrm{P}_{ij} = (x_i, y_j)$ has 4 neighbors: $\mathrm{P}_{i-1j} = (x_{i-1}, y_j)$, $\mathrm{P}_{i+1j} = (x_{i+1}, y_j)$, $\mathrm{P}_{ij-1} = (x_i, y_{j-1})$ and $\mathrm{P}_{ij+1} = (x_i, y_{j+1})$. Each internal point $\mathrm{P}_{ij} = (x_i, y_j)$ with its neighbors consitutes a 5-point stencil. In general in d spatial dimension we have $2d + 1$-point stencil. The following figure shows, in the left the 3-point stencil (in one spatial dimension) centered in the point P_i, and in the right the 5-point stencil (in two spatial dimension) centered in the point P_{ij}.



If a neighbor does not belong to the domain Ω it is called *ghost point*. Thus a ghost point is a grid point that it belongs to a stencil centered in an internal point. We highlight in previous figure the internal and the ghost points. The black dots are the internal points and the red dots are the ghost points. We call Ω_h^{I} the set of internal points and Ω_h^{G} the set of ghost points.

To discretize the problem (1.2) we proceed in two different ways.
In the internal points we use the usual finite difference method that approximates the second derivative of the solution u considering the Taylor Series expansion. The approximation of the Laplacian operator (second derivative of u) in $\mathrm{P}_{i_1,\ldots,i_d}$ involves the value of the numerical solution in the nodes that they belong to the stencil centered in $\mathrm{P}_{i_1,\ldots,i_d}$.
We use the notation $u_{i_1,\ldots,i_d}$ and $f_{i_1,\ldots,i_d}$ to denote the numeric value of the functions u and f calculated at the grid point $\mathrm{P}_{i_1,\ldots,i_d}$.

If we consider only the equations for the internal points, we obtain a linear system of $|\Omega_h^{\mathrm{I}}|$ equations and $|\Omega_h^{\mathrm{I}}| + |\Omega_h^{\mathrm{G}}|$ unknowns.
To close the linear system we must write $|\Omega_h^{\mathrm{G}}|$ equations for the ghost points. For each ghost point G we consider its projection F on the border and a polynomial interpolation procedure at the node G and at its neighbors that depending on the accuracy we want to obtain and also on the dimension d. We will describe the problem of polynomial interpolation in more detail in the following chapters.
The following figure (Figure 1.2) shows the projection of each ghost point on the border; in this case the domain is a circumference ($d = 2$).

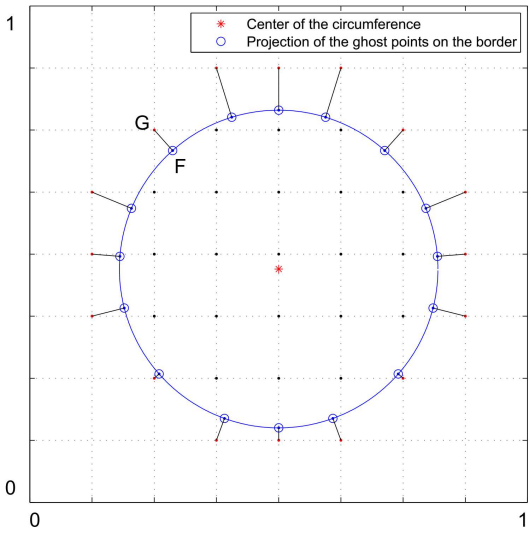


Figure 1.2: Projection of the ghost point G on the border (in 2D)

We call Ω_h^{F} the set of points F which are the projections of the ghost points G along the border of the domain Ω. In particular, we call $\Omega_h^{\mathrm{F}_D}$ the set of points F in which we impose Dirichlet conditions and we call $\Omega_h^{\mathrm{F}_N}$ the set of points F in which we impose Neumann conditions.

Proceeding as described we obtain the discretization of the problem (1.2):

$$\begin{cases} -\Delta_h u_h = f_h & \text{in } \Omega_h^{\mathrm{I}} \\ B_h u_h = g_h & \text{in } \Omega_h^{\mathrm{F}} \end{cases},$$

or in compact form $\mathrm{A}_h u_h = \mathrm{F}_h$.
The aim is to prove that the numerical method is consistent, stable and con-

vergent. In this regard, we recall the following result:

Given a linear problem with initial conditions, well-posed; a necessary and sufficient condition so that a consistent finite difference problem is convergent is that it is stable.

1.1 Coco-Russo method in 1D

In order to study some properties of the discretization we start to analyze the easier 1D problem.

The elliptic problem in 1D is:

$$\begin{cases} -u_{xx} = f & \text{in }]\mathbf{a}, \mathbf{b}[\subset [0,1] \\ Bu = g & \text{in } \Gamma \end{cases} . \tag{1.3}$$

1.1.1 Dirichlet boundary conditions

We solve numerically the problem (1.3) imposing Dirichlet boundary conditions. The problem becomes:

$$\begin{cases} -u_{xx} = f & \text{in }]\mathbf{a}, \mathbf{b}[\\ u(\mathbf{a}) = \bar{u}_{\mathrm{a}} \\ u(\mathbf{b}) = \bar{u}_{\mathrm{b}} \end{cases} . \tag{1.4}$$

Discretization of the domain

First, we discretize the domain $[\mathbf{a}, \mathbf{b}] \subset [0,1]$. We discretize the interval $[0,1]$ with $n+1$ grid points; we call $x_0, x_1, \ldots, x_n$ the grid points and $h = \frac{1}{n}$ the spatial step. We place the bottom extremity $\mathbf{a}$ between x_0 and x_1. We can choose to place the upper extremity $\mathbf{b}$ between x_{n-1} and x_n or fix it equal to $1 = x_n$. In the first case we have two ghost points x_0 and x_n. In the second case we have only a ghost point x_0, because $\mathbf{b}$ is a grid point. The following figure (Figure 1.3) shows the discretization of the domain $[0,1]$ and hence the discretization of the domain $[\mathbf{a}, \mathbf{b}]$ in the case in which $\mathbf{b}$ is a grid point.

Figure 1.3: Discretization of the domain $[\mathbf{a}, \mathbf{b}]$ with $n = 5$: the black dots are the *internal points*, the red dot is the only *ghost point*. We impose the Dirichlet boundary conditions in the blue dots $\mathbf{a}$ and $\mathbf{b}$=x_5=1

Discretization of the problem

Now, we discretize the problem (1.4), starting with the case in which **b** is a grid point.
In the *internal points* we use the finite-difference approximation for $-u_{xx}$ using the 3-point stencil:

$$-u_{xx}(x_i) \approx -\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}, \ i = 1, \dots, n-1, \tag{1.5}$$

where u_i indicates the value that the numerical solution assumes at the point x_i. In this way we obtain a linear system of $n-1$ equations and n unknowns $u_0, u_1, \dots, u_{n-1}$ (in x_n we know the exact solution $u(x_n) = \bar{u}_{\mathrm{b}}$). To close the system we impose the Dirichlet boundary condition in the point **a**.

Discretization of boundary conditions Since the point **a** isn't a grid point we perform a *linear interpolation* procedure at the nodes x_0 and x_1 or a *quadratic interpolation* procedure at the nodes x_0, x_1 and x_2 and we impose that in the point **a** the numerical solution (obtained through the linear interpolation procedure or the quadratic interpolation procedure) coincides with exact solution.

Second order accuracy on the border

Linear interpolation

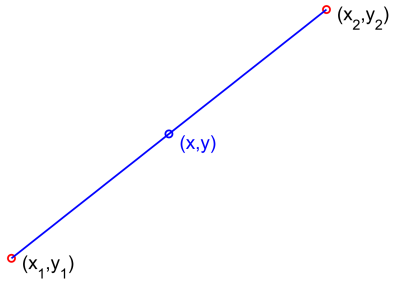


Figure 1.4: Given the two red dots, the blue line is the linear interpolant between the two dots

The linear interpolation is a polynomial interpolation procedure in the case in which the interpolating nodes are two. If the two known data points are (x_1, y_1), (x_2, y_2), the linear interpolant is the straight line passing through these points. Another way to obtain the formula of linear interpolation, apart from the method that consider the straight

line through the two points, is the Lagrange interpolation procedure to approximate the value that assumes a given function $y = y(x)$ in a generic node $x \in]x_1, x_2[$:

$$y = y_1 \frac{x - x_2}{x_1 - x_2} + y_2 \frac{x - x_1}{x_2 - x_1}.$$

Applying the previous formula to our case (we want approximate the value that assumes the function u at the point **a**; we can observe the position of the nodes in the Figure 1.3), we have:

$$\begin{aligned} u(\mathrm{a}) \approx \ & \frac{\mathrm{a} - x_1}{x_0 - x_1} u_0 + \frac{\mathrm{a} - x_0}{x_1 - x_0} u_1 = \frac{x_1 - \mathrm{a}}{h} u_0 + \frac{\mathrm{a} - x_0}{h} u_1 \\ & = \theta_{\mathrm{a}} u_0 + (1 - \theta_{\mathrm{a}}) u_1, \end{aligned} \tag{1.6}$$

where $0 < \theta_{\mathrm{a}} = \frac{x_1 - \mathrm{a}}{h} < 1$.

The formula (1.6) represents the formula of the linear interpolation at the nodes x_0 and x_1, to approximate the value that the function u assumes at the point **a**.

Linear interpolation error

If the function u to be interpolated admits at least derivative of order two ($u \in \mathrm{C}^2([x_0, x_1])$), the error that is made in linear interpolation satisfies, for all x, the following equation:

$$u(x) = P(x) + \frac{1}{2}(x - x_0)(x - x_1) u''(\xi), \tag{1.7}$$

where $\xi \in [\min\{x_0, x_1, x\}, \max\{x_0, x_1, x\}]$ and $P(x)$ is the interpolation polynomial.
In particular on point **a** the formula (1.7) becomes:

$$u(\mathrm{a}) = P(\mathrm{a}) + \frac{1}{2}(1 - \theta_{\mathrm{a}})(-\theta_{\mathrm{a}}) h^2 u''(\xi).$$

Therefore, the error $\mathcal{E} = -\frac{1}{2}(1 - \theta_{\mathrm{a}})\theta_{\mathrm{a}} h^2 u''(\xi) \sim \mathrm{O}(h^2)$.

The *numerical problem* of the problem (1.4), in the case we impose the Dirichlet boundary condition through a linear interpolation procedure, is:

$$\begin{cases} \theta_{\mathrm{a}} u_0 + (1-\theta_{\mathrm{a}}) u_1 = \bar{u}_{\mathrm{a}} \\ -\frac{u_{i-1}-2u_i+u_{i+1}}{h^2} = f_i \qquad i = 1, \ldots, n-1\,, \\ u(\mathbf{b}) = u(x_n) = \bar{u}_{\mathrm{b}} \end{cases} \tag{1.8}$$

that in matrix form becomes $\mathrm{A}_h u_h = \mathrm{F}_h$, where:

$$\mathrm{A}_h = \begin{pmatrix} \theta_{\mathrm{a}} & 1-\theta_{\mathrm{a}} & & & \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} \\ & & & -\frac{1}{h^2} & \frac{2}{h^2} \end{pmatrix} \in \mathbb{R}^{n\times n}$$

is the matrix of coefficients,

$$u_h = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{pmatrix} \in \mathbb{R}^n$$

is the vector of the unknowns (*numerical solution*),

$$\mathrm{F}_h = \begin{pmatrix} \bar{u}_{\mathrm{a}} \\ f_1 \\ \vdots \\ \vdots \\ f_{n-1} + \frac{u(x_n)}{h^2} \end{pmatrix} \in \mathbb{R}^n$$

is the vector of the known terms.

Symmetry of the matrix A_h

We can transform the system (1.8) into an equivalent system in which the matrix A_h is symmetric. To do this we divide the first equation of the system by $-(1-\theta_{\mathrm{a}})h^2$:

$$\theta_{\mathrm{a}} u_0 + (1-\theta_{\mathrm{a}}) u_1 = \bar{u}_{\mathrm{a}} \Longrightarrow -\frac{\theta_{\mathrm{a}}}{(1-\theta_{\mathrm{a}})h^2} u_0 - \frac{1}{h^2} u_1 = -\frac{\bar{u}_{\mathrm{a}}}{(1-\theta_{\mathrm{a}})h^2}. \tag{1.9}$$

Third order accuracy on the border

Quadratic interpolation

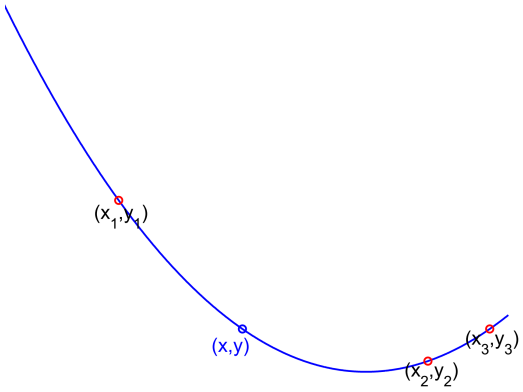


Figure 1.5: Given the three red dots, the blue line is the quadratic interpolant between the dots

If we want a third order accuracy on the border, we use a quadratic interpolation procedure instead of a linear interpolation procedure. The quadratic interpolation is a polynomial interpolation procedure in the case in which the interpolating nodes are three. If the three known data points are (x_1, y_1), (x_2, y_2), (x_3, y_3) and they are not aligned, the quadratic interpolant is the parabola that passes through these points. To obtain the formula of quadratic interpolation we consider the Lagrange interpolation procedure:

$$y = y_1 \frac{x - x_2}{x_1 - x_2}\frac{x - x_3}{x_1 - x_3} + y_2 \frac{x - x_1}{x_2 - x_1}\frac{x - x_3}{x_2 - x_3} + y_3 \frac{x - x_1}{x_3 - x_1}\frac{x - x_2}{x_3 - x_2}.$$

Applying the previous formula to our case (we want approximate the value that assumes the function u at the point $\mathbf{a}$ if we know (x_0, u_0), (x_1, u_1), (x_2, u_2); we can observe the position of the nodes in the Figure 1.3), we have:

$$u(\mathrm{a}) \approx \frac{1}{2}\theta_{\mathrm{a}}(1 + \theta_{\mathrm{a}})u_0 + (1 - \theta_{\mathrm{a}})(1 + \theta_{\mathrm{a}})u_1 - \frac{1}{2}\theta_{\mathrm{a}}(1 - \theta_{\mathrm{a}})u_2, \qquad (1.10)$$

where $0 < \theta_{\mathrm{a}} = \frac{x_1 - \mathrm{a}}{h} < 1$.

The formula (1.10) represents the formula of the quadratic interpolation at the nodes x_0, x_1 and x_2, to approximate the value that the function u assumes in the point $\mathbf{a}$.

Quadratic interpolation error

If the function u to be interpolated admits at least derivative of order three ($u \in \mathrm{C}^3([x_0, x_2])$), the error that is made in quadratic interpolation satisfies, for all x, the following equation:

$$u(x) = P(x) + \frac{1}{6}(x - x_0)(x - x_1)(x - x_2)u'''(\xi), \tag{1.11}$$

where $\xi \in [\min\{x_0, x_1, x_2, x\}, \max\{x_0, x_1, x_2, x\}]$ and $P(x)$ is the interpolation polynomial.
In particular on point **a** the formula (1.11) becomes:

$$u(\mathrm{a}) = P(\mathrm{a}) + \frac{1}{6}(1 - \theta_\mathrm{a})(-\theta_\mathrm{a})(-(1 + \theta_\mathrm{a}))h^3u'''(\xi).$$

Therefore, the error $\mathcal{E} = \frac{1}{6}(1 - \theta_\mathrm{a})(\theta_\mathrm{a})(1 + \theta_\mathrm{a})h^3u'''(\xi) \sim \mathrm{O}(h^3)$.

The *numerical problem* of the problem (1.4), in the case we impose the Dirichlet boundary condition through a quadratic interpolation procedure, is:

$$\begin{cases} \frac{\theta_\mathrm{a}(1+\theta_\mathrm{a})}{2}u_0 + (1 - \theta_\mathrm{a})(1 + \theta_\mathrm{a})u_1 - \frac{\theta_\mathrm{a}(1-\theta_\mathrm{a})}{2}u_2 = \bar{u}_\mathrm{a} \\ -\frac{u_{i-1}-2u_i+u_{i+1}}{h^2} = f_i \qquad\qquad i = 1, \ldots, n-1\,, \\ u(x_n) = \bar{u}_\mathrm{b} \end{cases} \tag{1.12}$$

and the matrix A_h of the coefficients is:

$$\mathrm{A}_h = \begin{pmatrix} \frac{\theta_\mathrm{a}(1+\theta_\mathrm{a})}{2} & (1 - \theta_\mathrm{a})(1 + \theta_\mathrm{a}) & \frac{-\theta_\mathrm{a}(1-\theta_\mathrm{a})}{2} & & \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} \\ & & & -\frac{1}{h^2} & \frac{2}{h^2} \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

We proceed with the case in which **b** is not a grid point. The following figure (Figure 1.6) shows the discretization of the domain $[\mathbf{a}, \mathbf{b}]$ in this case.

Figure 1.6: Discretization of the domain $[\mathbf{a}, \mathbf{b}]$ with $n = 5$: the black dots are the *internal points*, the red dots are the two *ghost points*. We impose the Dirichlet boundary conditions in the blue dots **a** and **b**

We discretize the problem (1.4). In the internal points we use the finite-difference approximation for $-u_{xx}$ using the 3-point stencil (see formula (1.5)). In this way we obtain a linear system of $n-1$ equations and $n+1$ unknowns $u_0, u_1, \ldots, u_n$. To close the system we impose the Dirichlet boundary conditions in the points **a** and **b**. Since the two points aren't grid points we perform a linear interpolation procedure at the nodes x_0 and x_1 or a quadratic interpolation procedure at the nodes x_0, x_1 and x_2 and we impose that in the point **a** the numerical solution coincides with exact solution and a linear interpolation procedure at the nodes x_{n-1} and x_n or a quadratic interpolation procedure at the nodes x_{n-2}, x_{n-1} and x_n and we impose that in the point **b** the numerical solution coincides with exact solution.

If we use a linear interpolation procedure, the *numerical problem* of the problem (1.4) is:

$$\begin{cases} \theta_{\mathrm{a}} u_0 + (1-\theta_{\mathrm{a}}) u_1 = \bar{u}_{\mathrm{a}} \\ -\frac{u_{i-1}-2u_i+u_{i+1}}{h^2} = f_i \qquad\qquad i = 1, \ldots, n-1\,, \\ (1-\theta_{\mathrm{b}}) u_{n-1} + \theta_{\mathrm{b}} u_n = \bar{u}_{\mathrm{b}} \end{cases} \tag{1.13}$$

with $0 < \theta_{\mathrm{a}} = \frac{x_1 - \mathrm{a}}{h}$, $\theta_{\mathrm{b}} = \frac{\mathrm{b} - x_{n-1}}{h} < 1$.

The system (1.13) in matrix form becomes $\mathrm{A}_h u_h = \mathrm{F}_h$, where:

$$\mathrm{A}_h = \begin{pmatrix} \theta_{\mathrm{a}} & 1-\theta_{\mathrm{a}} & & & \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} \\ & & & 1-\theta_{\mathrm{b}} & \theta_{\mathrm{b}} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}$$

is the matrix of coefficients,

$$u_h = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} \in \mathbb{R}^{n+1}$$

is the vector of the unknowns (*numerical solution*),

$$\mathrm{F}_h = \begin{pmatrix} \bar{u}_\mathrm{a} \\ f_1 \\ \vdots \\ f_{n-1} \\ \bar{u}_\mathrm{b} \end{pmatrix} \in \mathbb{R}^{n+1}$$

is the vector of the known terms.
Also in this case it is possible to symmetrize the matrix A_h by applying the formula (1.9) to both the first and last rows.

We can observe that in this case we have an approximation of the second order in the internal points and on the border. If we want to obtain an approximation of the third order on the border we use, to close the linear system, a quadratic interpolation procedure. The *numerical problem* of the problem (1.4), in this case, is:

$$\begin{cases} \frac{\theta_\mathrm{a}(1+\theta_\mathrm{a})}{2} u_0 + (1-\theta_\mathrm{a})(1+\theta_\mathrm{a}) u_1 - \frac{\theta_\mathrm{a}(1-\theta_\mathrm{a})}{2} u_2 = \bar{u}_\mathrm{a} \\ -\frac{u_{i-1}-2u_i+u_{i+1}}{h^2} = f_i \qquad\qquad i = 1, \ldots, n-1\,, \\ -\frac{\theta_\mathrm{b}(1-\theta_\mathrm{b})}{2} u_{n-2} + (1-\theta_\mathrm{b})(1+\theta_\mathrm{b}) u_{n-1} + \frac{\theta_\mathrm{b}(1+\theta_\mathrm{b})}{2} u_n = \bar{u}_\mathrm{b} \end{cases} \tag{1.14}$$

and the matrix A_h of coefficients is:

$$\mathrm{A}_h = \begin{pmatrix} \frac{\theta_\mathrm{a}(1+\theta_\mathrm{a})}{2} & (1-\theta_\mathrm{a})(1+\theta_\mathrm{a}) & \frac{-\theta_\mathrm{a}(1-\theta_\mathrm{a})}{2} & & \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} \\ & & \frac{-\theta_\mathrm{b}(1-\theta_\mathrm{b})}{2} & (1-\theta_\mathrm{b})(1+\theta_\mathrm{b}) & \frac{\theta_\mathrm{b}(1+\theta_\mathrm{b})}{2} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}.$$

1.1.2 Mixed boundary conditions

Now, we solve numerically the problem (1.3) imposing mixed boundary conditions. In particular, we impose Dirichlet boundary condition in **a** ($x_0 < \mathbf{a} < x_1$) and Neumann boundary condition in **b** ($x_{n-1} < \mathbf{b} < x_n$).

The problem becomes:

$$\begin{cases} -u_{xx} = f & \text{in }]\mathbf{a}, \mathbf{b}[\\ u(\mathbf{a}) = \bar{u}_{\mathrm{a}} & \\ \frac{\partial u}{\partial \mathbf{n}}(\mathbf{b}) = \bar{g}_{\mathrm{b}} & \end{cases}, \tag{1.15}$$

where **n** is the outward unit normal.

The discretization of the domain $[\mathbf{a}, \mathbf{b}]$ and of the problem (1.15) are done as in Section 1.1.1, except in point **b** in which we impose the Neumann condition. We can see the Figure 1.6 to see the discretization of the domain $[\mathbf{a}, \mathbf{b}] \subset [0, 1]$.

Discretization of boundary conditions

First order accuracy on the border

If we use a linear interpolation procedure to impose Neumann boundary condition in **b**, since we know that $u(x) \approx (1 - \theta_x)u_{n-1} + \theta_x u_n$, where $\theta_x = \frac{x - x_{n-1}}{h}$, we have

$$u'(\mathbf{b}) \approx -\frac{1}{h}u_{n-1} + \frac{1}{h}u_n. \tag{1.16}$$

It's obvious that, if in the formula (1.6) of the linear interpolation procedure the error $\mathcal{E} \sim \mathrm{O}(h^2)$, in the formula (1.16) we have $\mathcal{E} \sim \mathrm{O}(h)$.

Therefore, the *numerical problem* of the problem (1.15) in the case in which we impose the boundary conditions through a linear interpolation procedure both in **a** and in **b** is:

$$\begin{cases} \theta_{\mathrm{a}} u_0 + (1 - \theta_{\mathrm{a}})u_1 = \bar{u}_{\mathrm{a}} & \\ -\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = f_i & i = 1, \ldots, n-1 \\ -\frac{1}{h}u_{n-1} + \frac{1}{h}u_n = \bar{g}_{\mathrm{b}} & \end{cases}, \tag{1.17}$$

that in matrix form becomes $\mathrm{A}_h u_h = \mathrm{F}_h$, where:

$$\mathrm{A}_h = \begin{pmatrix} \theta_\mathrm{a} & 1-\theta_\mathrm{a} & & & \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} \\ & & & -\frac{1}{h} & \frac{1}{h} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}$$

is the matrix of coefficients,

$$u_h = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_n \end{pmatrix} \in \mathbb{R}^{n+1}$$

is the vector of the unknowns (*numerical solution*),

$$\mathrm{F}_h = \begin{pmatrix} \bar{u}_\mathrm{a} \\ f_1 \\ \vdots \\ \vdots \\ f_{n-1} \\ \bar{g}_\mathrm{b} \end{pmatrix} \in \mathbb{R}^{n+1}$$

is the vector of the known terms.

Symmetry of the matrix A_h

We can transform the system (1.17) into an equivalent system in which the matrix A_h is symmetric. To do this we divide the first equation of the system by $-(1-\theta_\mathrm{a})h^2$ as in the Dirichlet problem (see formula (1.9)) and the last equation of the system by h:

$$-\frac{1}{h}u_{n-1} + \frac{1}{h}u_n = \bar{g}_\mathrm{b} \Longrightarrow -\frac{1}{h^2}u_{n-1} + \frac{1}{h^2}u_n = \frac{\bar{g}_\mathrm{b}}{h}. \tag{1.18}$$

Second order accuracy on the border

Instead, if we use a quadratic interpolation procedure to impose Neumann boundary condition in **b**, from the formula (1.10) we have:

$$u'(\mathbf{b}) \approx \frac{1}{2h}(2\theta_{\mathrm{b}} - 1)u_{n-2} - \frac{2}{h}\theta_{\mathrm{b}}u_{n-1} + \frac{1}{2h}(2\theta_{\mathrm{b}} + 1)u_n.$$

In this case $\mathcal{E} \sim \mathrm{O}(h^2)$.

Remark 1.1: We can observe that if we use a quadratic interpolation procedure to impose the Neumann boundary condition in **b** we obtain a second order accuracy also on the border as well as on internal points.

1.2 Coco-Russo method in 2D

In this Section we show the discretization in two-dimensional case.

The elliptic problem is:

$$\begin{cases} -(u_{xx} + u_{yy}) = f & \text{in } \Omega \subset \mathbb{R}^2 \\ Bu = g & \text{in } \Gamma \end{cases}, \tag{1.19}$$

where the domain Ω has a particular geometry in a two-dimensional space.

1.2.1 Dirichlet boundary conditions

We solve numerically the problem (1.19) imposing Dirichlet boundary conditions.

First, we discretize the domain Ω (for simplicity, we suppose that $\Omega \subset [0,1]^2$) determining internal points and ghost points.

Now, we discretize the problem (1.19).
In the *internal points* we use the finite-difference approximation for $-(u_{xx} + u_{yy})$ using the 5-point stencil:

$$-u_{xx}(x_i, y_j) - u_{yy}(x_i, y_j) \approx -\frac{u_{i-1,j} + u_{i+1,j} - 4u_{i,j} + u_{i,j+1} + u_{i,j-1}}{h^2}, \ \forall (i,j) : (x_i, y_j) \in \Omega_h^{\mathrm{I}}. \tag{1.20}$$

Discretization of boundary conditions To close the linear system, for each *ghost point* G we perform the projection $\mathrm{F}(x_{\mathrm{F}}, y_{\mathrm{F}})$ of the point G on the border of the domain Ω. In this point F we impose the Dirichlet conditions through a *bilinear interpolation* procedure (if we want obtain a second-order accuracy on the border) or through a *biquadratic interpolation* procedure (if we want obtain a third-order accuracy on the border).

Second order accuracy on the border

Bilinear interpolation

The bilinear interpolation is an extension of linear interpolation for interpolating functions of two variables (x and y) on a square 2D grid.

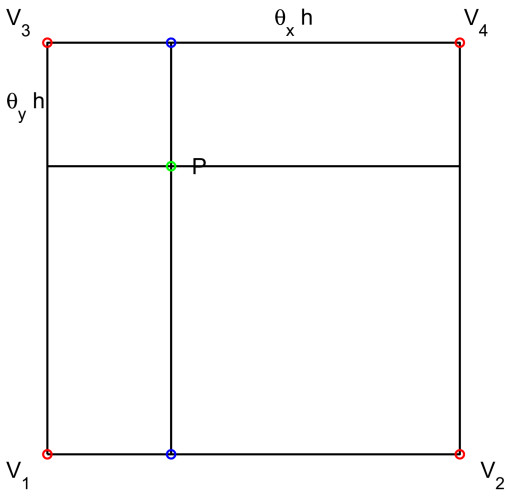


Figure 1.7: The four red circles are the points used to perform the bilinear interpolation and the green circle is the point at which we want to interpolate

The left figure (Figure 1.7) represents the position of the four nodes (the red circles $\mathrm{V}_i = \mathrm{V}(x_i, y_i)$, $i = 1, \ldots, 4$) used to perform the bilinear interpolation to approximate the value that a given function u assumes at green point $\mathrm{P} = (x_\mathrm{P}, y_\mathrm{P})$.

To obtain the formula of bilinear interpolation we first perform two linear interpolations in one direction and then one linear interpolation in the other direction. The formula of bilinear interpolation does not depend of the order of the interpolation steps along the two axes (x or y).

For example, if we indicate by

$$\theta_x = \frac{x_4 - x_\mathrm{P}}{h} \text{ and } \theta_y = \frac{y_3 - y_\mathrm{P}}{h},$$

we can perfom:

- the linear interpolation at the nodes V_1 and V_2:

$$u(x_\mathrm{P}, y_1) \approx \theta_x u(x_1, y_1) + (1 - \theta_x) u(x_2, y_2),$$

- the linear interpolation at the nodes V_3 and V_4:

$$u(x_{\mathrm{P}}, y_3) \approx \theta_x u(x_3, y_3) + (1-\theta_x)u(x_4, y_4),$$

- the linear interpolation at the nodes (x_{P}, y_1) and (x_{P}, y_3) (the blue circles in Figure 1.7) to approximate the value that the function u assumes in the green point P:

$$u(x_{\mathrm{P}}, y_{\mathrm{P}}) \approx \ \theta_y u(x_{\mathrm{P}}, y_1) + (1-\theta_y)u(x_{\mathrm{P}}, y_3) = \theta_x\theta_y u(\mathrm{V}_1) +$$
$$(1-\theta_x)\theta_y u(\mathrm{V}_2) + \theta_x(1-\theta_y)u(\mathrm{V}_3) + (1-\theta_x)(1-\theta_y)u(\mathrm{V}_4). \quad (1.21)$$

The formula (1.21) represents the formula of the bilinear interpolation at the nodes V_1, V_2, V_3 and V_4 to approximate the value that the function u assumes in the point P.

Bilinear interpolation error

To calculate the bilinear interpolation error we must note that the bilinear interpolation polynomial was constructed using three linear interpolation polynomials and thus we must consider the errors associated with these interpolation polynomials.

Based on the formula (1.7), if $u \in \mathrm{C}^2$,

$$u(x_{\mathrm{P}}, y_1) = P(x_{\mathrm{P}}, y_1) + \frac{1}{2}h^2(1-\theta_{x_{\mathrm{P}}})(-\theta_{x_{\mathrm{P}}})u''(\xi_1, \bar{\mu}_1),$$

for some $\xi_1 \in [\min\{x_1, x_2, x_{\mathrm{P}}\}, \max\{x_1, x_2, x_{\mathrm{P}}\}]$,

$$u(x_{\mathrm{P}}, y_3) = P(x_{\mathrm{P}}, y_3) + \frac{1}{2}h^2(1-\theta_{x_{\mathrm{P}}})(-\theta_{x_{\mathrm{P}}})u''(\xi_2, \bar{\mu}_2),$$

for some $\xi_2 \in [\min\{x_1, x_2, x_{\mathrm{P}}\}, \max\{x_1, x_2, x_{\mathrm{P}}\}]$, and

$$u(x_{\mathrm{P}}, y_{\mathrm{P}}) = \theta_{y_{\mathrm{P}}}u(x_{\mathrm{P}}, y_1) + (1-\theta_{y_{\mathrm{P}}})u(x_{\mathrm{P}}, y_3) + \frac{1}{2}h^2(1-\theta_{y_{\mathrm{P}}})(-\theta_{y_{\mathrm{P}}})u''(\bar{\xi}, \mu)$$
$$= P(x_{\mathrm{P}}, y_{\mathrm{P}}) + \frac{1}{2}h^2\left[\theta_{y_{\mathrm{P}}}(1-\theta_{x_{\mathrm{P}}})(-\theta_{x_{\mathrm{P}}})u''(\xi_1, \bar{\mu}_1) + \right.$$
$$\left.(1-\theta_{y_{\mathrm{P}}})(1-\theta_{x_{\mathrm{P}}})(-\theta_{x_{\mathrm{P}}})u''(\xi_2, \bar{\mu}_2) + (1-\theta_{y_{\mathrm{P}}})(-\theta_{y_{\mathrm{P}}})u''(\bar{\xi}, \mu)\right], \quad (1.22)$$

for some $\mu \in [\min\{y_1, y_3, y_{\mathrm{P}}\}, \max\{y_1, y_3, y_{\mathrm{P}}\}]$.

Therefore, the error $\mathcal{E} \sim \mathrm{O}(h^2)$.

Third order accuracy on the border

Biquadratic interpolation

In the biquadratic interpolation to calculate the value that the function u assumes in the point on the border F we involving 9 nodes rather than 4 nodes. The following figure (Figure 1.8) shows the position of the nine nodes (the red circles $\mathrm{V}_{i,j} = \mathrm{V}(x_i, y_j)$, $i, j = 1, \ldots, 9$) used to perfom the biquadratic interpolation to approximate the value that the function u assumes at green point $\mathrm{P} = (x_\mathrm{P}, y_\mathrm{P})$.

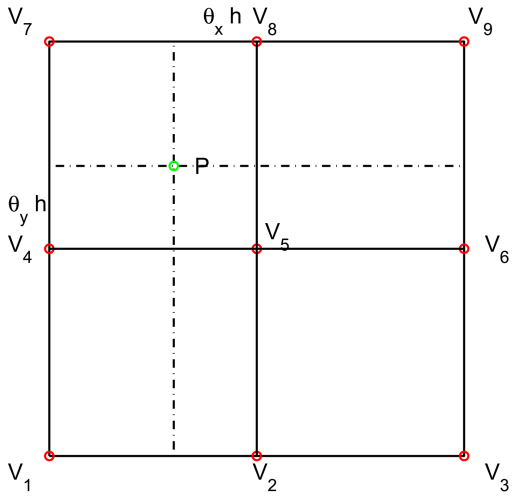


Figure 1.8: The nine red circles are the points used to perform the biquadratic interpolation and the green circle is the point at which we want to interpolate

If we indicate by

$$\theta_x = \frac{x_8 - x_\mathrm{P}}{h} \text{ and } \theta_y = \frac{y_\mathrm{P} - y_4}{h},$$

proceeding as in the case of bilinear interpolation we obtain:

$$u(x_\mathrm{P}, y_\mathrm{P}) \approx \frac{1}{4}\theta_x(1+\theta_x)\theta_y(\theta_y-1)u(\mathrm{V}_1) - \frac{1}{2}(\theta_x-1)(1+\theta_x)\theta_y(\theta_y-1)u(\mathrm{V}_2) +$$

$$\frac{1}{4}(\theta_x-1)\theta_x\theta_y(\theta_y-1)u(\mathrm{V}_3) - \frac{1}{2}\theta_x(1+\theta_x)(\theta_y-1)(1+\theta_y)u(\mathrm{V}_4) +$$

$$(\theta_x-1)(1+\theta_x)(\theta_y-1)(1+\theta_y)u(\mathrm{V}_5) - \frac{1}{2}(\theta_x-1)\theta_x(\theta_y-1)(1+\theta_y)u(\mathrm{V}_6) +$$

$$\frac{1}{4}\theta_x(1+\theta_x)(\theta_y+1)\theta_y u(\mathrm{V}_7) - \frac{1}{2}(\theta_x-1)(1+\theta_x)(\theta_y+1)\theta_y u(\mathrm{V}_8) +$$

$$\frac{1}{4}(\theta_x - 1)\theta_x(\theta_y + 1)\theta_y u(\mathrm{V}_9). \tag{1.23}$$

The formula (1.23) represents the formula of the biquadratic interpolation at the nodes V_i, $i = 1, \ldots, 9$, to approximate the value that the function u assumes in the point P.

Biquadratic interpolation error

To calculate the biquadratic interpolation error we must note that the biquadratic interpolation polynomial was constructed using four linear interpolation polynomials and thus we must consider the errors associated with these interpolation polynomials.

Based on the formula (1.11), if $u \in \mathrm{C}^3$,

$$u(x_{\mathrm{P}}, y_1) = P(x_{\mathrm{P}}, y_1) - \frac{1}{6}h^3(\theta_{x_{\mathrm{P}}} - 1)\theta_{x_{\mathrm{P}}}(1 + \theta_{x_{\mathrm{P}}})u'''(\xi_1, \bar{\mu}_1),$$

for some $\xi_1 \in [\min\{x_1, x_2, x_3, x_{\mathrm{P}}\}, \max\{x_1, x_2, x_3, x_{\mathrm{P}}\}]$,

$$u(x_{\mathrm{P}}, y_4) = P(x_{\mathrm{P}}, y_4) - \frac{1}{6}h^3(\theta_{x_{\mathrm{P}}} - 1)\theta_{x_{\mathrm{P}}}(1 + \theta_{x_{\mathrm{P}}})u'''(\xi_2, \bar{\mu}_2),$$

for some $\xi_2 \in [\min\{x_1, x_2, x_3, x_{\mathrm{P}}\}, \max\{x_1, x_2, x_3, x_{\mathrm{P}}\}]$,

$$u(x_{\mathrm{P}}, y_7) = P(x_{\mathrm{P}}, y_7) - \frac{1}{6}h^3(\theta_{x_{\mathrm{P}}} - 1)\theta_{x_{\mathrm{P}}}(1 + \theta_{x_{\mathrm{P}}})u'''(\xi_3, \bar{\mu}_3),$$

for some $\xi_3 \in [\min\{x_1, x_2, x_3, x_{\mathrm{P}}\}, \max\{x_1, x_2, x_3, x_{\mathrm{P}}\}]$, and

$$\begin{aligned}
u(x_{\mathrm{P}}, y_{\mathrm{P}}) &= \frac{1}{2}\theta_{y_{\mathrm{P}}}(\theta_{y_{\mathrm{P}}} - 1)u(x_{\mathrm{P}}, y_1) - (\theta_{y_{\mathrm{P}}} - 1)(1 + \theta_{y_{\mathrm{P}}})u(x_{\mathrm{P}}, y_4) + \\
&\quad \frac{1}{2}(\theta_{y_{\mathrm{P}}} + 1)\theta_{y_{\mathrm{P}}}u(x_{\mathrm{P}}, y_7) + \frac{1}{6}h^3(\theta_{y_{\mathrm{P}}} - 1)\theta_{y_{\mathrm{P}}}(1 + \theta_{y_{\mathrm{P}}})u'''(\bar{\xi}, \mu) \\
&= P(x_{\mathrm{P}}, y_{\mathrm{P}}) + h^3\left[-\frac{1}{12}(\theta_{x_{\mathrm{P}}} - 1)\theta_{x_{\mathrm{P}}}(1 + \theta_{x_{\mathrm{P}}})\theta_{y_{\mathrm{P}}}(\theta_{y_{\mathrm{P}}} - 1)u'''(\xi_1, \bar{\mu}_1)\right. \\
&\qquad + \frac{1}{6}(\theta_{x_{\mathrm{P}}} - 1)\theta_{x_{\mathrm{P}}}(1 + \theta_{x_{\mathrm{P}}})(\theta_{y_{\mathrm{P}}} - 1)(1 + \theta_{y_{\mathrm{P}}})u'''(\xi_2, \bar{\mu}_2) \\
&\qquad - \frac{1}{12}(\theta_{x_{\mathrm{P}}} - 1)\theta_{x_{\mathrm{P}}}(1 + \theta_{x_{\mathrm{P}}})(\theta_{y_{\mathrm{P}}} + 1)\theta_{y_{\mathrm{P}}}u'''(\xi_3, \bar{\mu}_3) \\
&\qquad \left. + \frac{1}{6}(\theta_{y_{\mathrm{P}}} - 1)\theta_{y_{\mathrm{P}}}(1 + \theta_{y_{\mathrm{P}}})u'''(\bar{\xi}, \mu)\right],
\end{aligned}$$

for some $\mu \in [\min\{y_1, y_4, y_7, y_{\mathrm{P}}\}, \max\{y_1, y_4, y_7, y_{\mathrm{P}}\}]$.

Therefore, the error $\mathcal{E} \sim \mathrm{O}(h^3)$.

The *numerical problem* of the problem (1.19), in the case we impose Dirichlet conditions, is:

$$\begin{cases} -\frac{u_{i-1j}+u_{ij-1}-4u_{ij}+u_{i+1j}+u_{ij+1}}{h^2} = f_{ij} & \text{in } \Omega_h^{\mathrm{I}} \\ u(\mathrm{F}) = g_D(\mathrm{F}) & \text{in } \Omega_h^{\mathrm{F}_D} \end{cases}, \tag{1.24}$$

where the value of $u(\mathrm{F})$ is approximated with the formula (1.21) of the bilinear interpolation or with the formula (1.23) of the biquadratic interpolation and g_D is the value of exact solution in F.

We choose two different domains. We begin choosing a simple domain like a *square*, which represents a simple extension of the one-dimensional case. Then, we continued our analysis with a more complex domain, that is a *circumference*.

Square domain

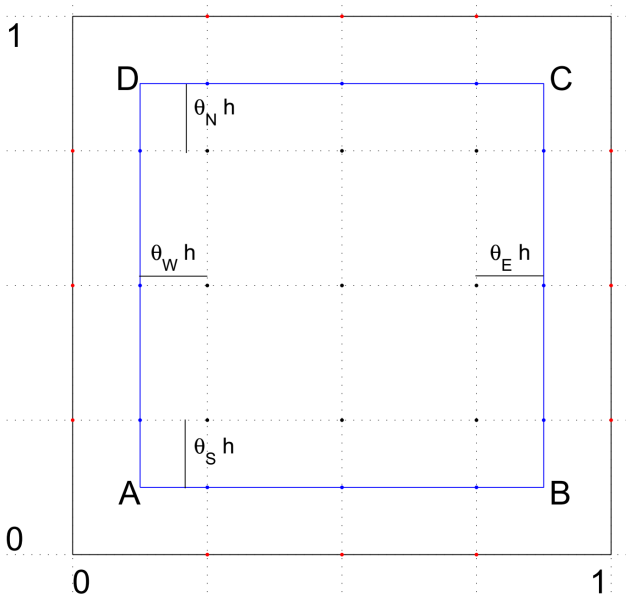


Figure 1.9: Discretization of the square: in the blue dots we impose the boundary conditions

We discretize the unit square with a regular grid of size h along the two axes respectively. We indicate by $N+1$ the number of grid points along the two axes respectively and let $h = \frac{1}{N}$ is the spatial step along the two axes. In total we have $(N+1)^2$ grid points. In this way, the square of vertices $\mathrm{A}(x_{\mathrm{A}}, y_{\mathrm{A}})$, $\mathrm{B}(x_{\mathrm{B}}, y_{\mathrm{B}})$, $\mathrm{C}(x_{\mathrm{C}}, y_{\mathrm{C}})$, $\mathrm{D}(x_{\mathrm{D}}, y_{\mathrm{D}})$ is also discretized. We indicate by n the total number of unknowns

(internal points + ghost points).

The figure above (Figure 1.9) represents the discretization of the unit square with $N = 4$ and so the discretization of the square, showing the internal points (black dots) and the ghost points (red dots) ($n = 21$).

We denote by $\mathrm{G_S}$ and $\mathrm{I_S}$, respectively, the ghost points and the internal points

$$\mathrm{G_S} = (ih, 0)\,,\ \mathrm{I_S} = (ih, h)\,,\ i = 1, \ldots, N-1.$$

We denote by $\mathrm{G_N}$ and $\mathrm{I_N}$, respectively, the ghost points and the internal points

$$\mathrm{G_N} = (ih, 1)\,,\ \mathrm{I_N} = (ih, 1-h)\,,\ i = 1, \ldots, N-1.$$

We denote by $\mathrm{G_W}$ and $\mathrm{I_W}$, respectively, the ghost points and the internal points

$$\mathrm{G_W} = (0, jh)\,,\ \mathrm{I_W} = (h, jh)\,,\ j = 1, \ldots, N-1.$$

We denote by $\mathrm{G_E}$ and $\mathrm{I_E}$, respectively, the ghost points and the internal points

$$\mathrm{G_E} = (1, jh)\,,\ \mathrm{I_E} = (1-h, jh)\,,\ j = 1, \ldots, N-1.$$

In the case of the square the bilinear (or biquadratic) interpolations are replaced by linear (or quadratic) interpolations. We show now the structure of the matrix of the coefficients in the easier case in which we use the linear interpolation procedures on the nodes $\mathrm{I_K}$ and $\mathrm{G_K}$, $\mathrm{K} = \mathrm{S}, \mathrm{N}, \mathrm{W}, \mathrm{E}$.

We denote by $\theta_{\mathrm{S}} = \frac{h - y_{\mathrm{B}}}{h}$ and $1 - \theta_{\mathrm{S}}$ the linear interpolation coefficients on the nodes $\mathrm{G_S}$ and $\mathrm{I_S}$, thus:

$$\theta_{\mathrm{S}} u_{\mathrm{G_S}} + (1 - \theta_{\mathrm{S}}) u_{\mathrm{I_S}} = u(ih, y_{\mathrm{B}}),\ i = 1, \ldots, N-1.$$

In the same way, we denote by $\theta_{\mathrm{N}} = \frac{y_{\mathrm{D}} - (1-h)}{h}$ and $1 - \theta_{\mathrm{N}}$ the linear interpolation coefficients on the nodes $\mathrm{G_N}$ and $\mathrm{I_N}$, thus:

$$\theta_{\mathrm{N}} u_{\mathrm{G_N}} + (1 - \theta_{\mathrm{N}}) u_{\mathrm{I_N}} = u(ih, y_{\mathrm{D}}),\ i = 1, \ldots, N-1.$$

We denote by $\theta_{\mathrm{W}} = \frac{h - x_{\mathrm{A}}}{h}$ and $1 - \theta_{\mathrm{W}}$ the linear interpolation coefficients on the nodes $\mathrm{G_W}$ and $\mathrm{I_W}$, thus:

$$\theta_{\mathrm{W}} u_{\mathrm{G_W}} + (1 - \theta_{\mathrm{W}}) u_{\mathrm{I_W}} = u(x_{\mathrm{A}}, jh),\ j = 1, \ldots, N-1.$$

We denote by $\theta_{\mathrm{E}} = \frac{x_{\mathrm{B}}-(1-h)}{h}$ and $1-\theta_{\mathrm{E}}$ the linear interpolation coefficients on the nodes $\mathrm{G_E}$ and $\mathrm{I_E}$, thus:

$$\theta_{\mathrm{E}} u_{\mathrm{G_E}} + (1-\theta_{\mathrm{E}}) u_{\mathrm{I_E}} = u(x_{\mathrm{B}}, jh),\ j = 1, \ldots, N-1.$$

Using a total lexicographical order the matrix of coefficients that we obtain is a block matrix with the following structure:

$$\mathrm{A}_h = \left(\begin{array}{c|c|c|c|c|c|c} \theta_{\mathrm{S}}\mathrm{I}_{N-1} & \mathrm{I_S} & & & & & \\ \hline \tilde{\mathrm{B}} & \mathrm{G} & \mathrm{B} & & & & \\ \hline & \mathrm{B} & \mathrm{G} & \mathrm{B} & & & \\ \hline & & \ddots & \ddots & \ddots & & \\ \hline & & & \mathrm{B} & \mathrm{G} & \mathrm{B} & \\ \hline & & & & \mathrm{B} & \mathrm{G} & \tilde{\mathrm{B}} \\ \hline & & & & & \mathrm{I_N} & \theta_{\mathrm{N}}\mathrm{I}_{N-1} \end{array}\right) \in \mathbb{R}^{n\times n},$$

where

$$\theta_{\mathrm{S}}\mathrm{I}_{N-1} \in \mathbb{R}^{(N-1)\times(N-1)},\ \theta_{\mathrm{N}}\mathrm{I}_{N-1} \in \mathbb{R}^{(N-1)\times(N-1)},$$

$$\mathrm{I_S} = \left(\begin{array}{c|c|c} \underline{0}_{N-1} & (1-\theta_{\mathrm{S}})\mathrm{I}_{N-1} & \underline{0}_{N-1} \end{array}\right) \in \mathbb{R}^{(N-1)\times(N+1)},$$

$$\mathrm{I_N} = \left(\begin{array}{c|c|c} \underline{0}_{N-1} & (1-\theta_{\mathrm{N}})\mathrm{I}_{N-1} & \underline{0}_{N-1} \end{array}\right) \in \mathbb{R}^{(N-1)\times(N+1)},$$

with $\underline{0}_{N-1}$ column vector of size $N-1$,

$$\tilde{\mathrm{B}} = \left(\begin{array}{c} \underline{0}^{\mathrm{T}}_{N-1} \\ \hline -\frac{\mathrm{I}_{N-1}}{h^2} \\ \hline \underline{0}^{\mathrm{T}}_{N-1} \end{array}\right) \in \mathbb{R}^{(N+1)\times(N-1)},\ \mathrm{B} = \left(\begin{array}{c|c|c} 0 & \underline{0}^{\mathrm{T}}_{N-1} & 0 \\ \hline \underline{0}_{N-1} & -\frac{\mathrm{I}_{N-1}}{h^2} & \underline{0}_{N-1} \\ \hline 0 & \underline{0}^{\mathrm{T}}_{N-1} & 0 \end{array}\right) \in \mathbb{R}^{(N+1)\times(N+1)},$$

with $\underline{0}^{\mathrm{T}}_{N-1}$ row vector of size $N-1$,

$$\mathrm{G} = \begin{pmatrix} \theta_{\mathrm{W}} & 1-\theta_{\mathrm{W}} & & & \\ -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} \\ & & & 1-\theta_{\mathrm{E}} & \theta_{\mathrm{E}} \end{pmatrix} \in \mathbb{R}^{(N+1)\times(N+1)}.$$

The number of matrices G is $N-1$.

Circular domain

A circumference of radius R and center C $= (x_C, y_C)$ is contained in the unit square. The value of the radius and the position of the center can change.

We discretize the unit square with a regular grid of size h along the two axes respectively. We indicate by $N + 1$ $\left(h = \frac{1}{N}\right)$ the number of grid points along the two axes respectively. In total we have $(N + 1)^2$ grid points. In this way, the circumference is also discretized. We indicate by n the total number of unknowns (internal points + ghost points).

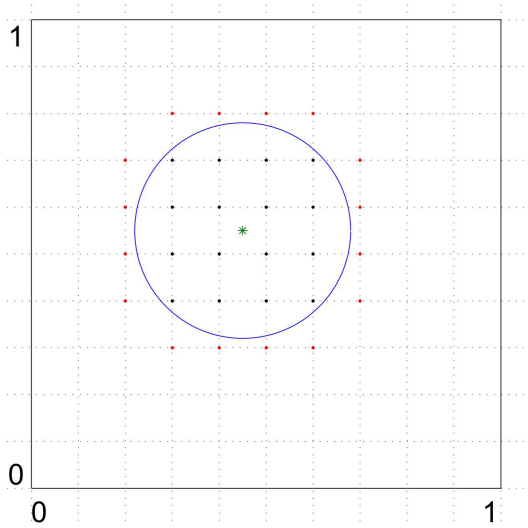


Figure 1.10: Discretization of the circumference

The right figure (Figure 1.10) represents the discretization of the unit square with $N = 10$ and thus the discretization of the circumference, showing the internal points (black dots) and the ghost points (red dots) ($n = 32$).

In this case there is not a well-defined structure of the matrix A_h as in the case of the square domain. It is enough to change the center or the radius of the circumference or the value of N, to change internal points and ghost points and therefore the structure of the matrix.

The following figure (Figure 1.11) highlights (with red circles) the nodes used in bilinear interpolation (left panel) and in biquadratic interpolation (right panel) to impose boundary condition at point F.

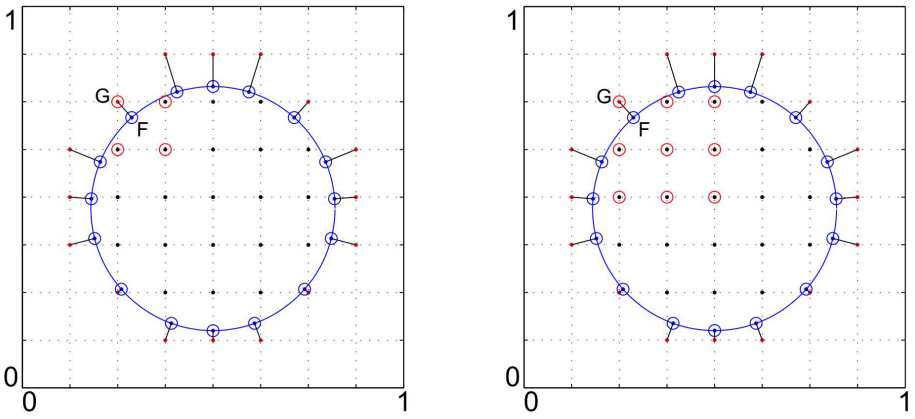


Figure 1.11: Bilinear interpolation (left panel) and biquadratic interpolation (right panel)

1.2.2 Mixed boundary conditions

Now, we solve numerically the problem (1.19) imposing mixed boundary conditions.

The discretization of the domain Ω, of the problem in the internal points and of the boundary conditions in the points in which we impose Dirichlet conditions are done as in Section 1.2.1.

Discretization of boundary conditions

First order accuracy on the border

If we use a bilinear interpolation procedure to impose Neumann boundary conditions, from the formula (1.21), in each point $\mathrm{F} = (x_{\mathrm{F}}, y_{\mathrm{F}})$ we have:

$$\frac{\partial u}{\partial \mathbf{n}}(x_{\mathrm{F}}, y_{\mathrm{F}}) \approx \left(-\frac{1}{h}\theta_y u(\mathrm{V}_1) + \frac{1}{h}\theta_y u(\mathrm{V}_2) - \frac{1}{h}(1-\theta_y)u(\mathrm{V}_3) + \frac{1}{h}(1-\theta_y)u(\mathrm{V}_4)\right)\mathrm{n}_x$$
$$+ \left(-\frac{1}{h}\theta_x u(\mathrm{V}_1) - \frac{1}{h}(1-\theta_x)u(\mathrm{V}_2) + \frac{1}{h}\theta_x u(\mathrm{V}_3) + \frac{1}{h}(1-\theta_x)u(\mathrm{V}_4)\right)\mathrm{n}_y, \tag{1.25}$$

where $\mathbf{n} = (\mathrm{n}_x, \mathrm{n}_y)$ is the outward unit normal, $\theta_x = \frac{x_4 - x_{\mathrm{F}}}{h}$ and $\theta_y = \frac{y_3 - y_{\mathrm{F}}}{h}$ (see Figure 1.7).

It's obvious that, if in the formula (1.21) of the bilinear interpolation procedure the error $\mathcal{E} \sim \mathrm{O}(h^2)$, in the formula (1.25) we have $\mathcal{E} \sim \mathrm{O}(h)$.

Second order accuracy on the border

If we use a biquadratic interpolation procedure to impose Neumann boundary conditions, from the formula (1.23), in each point $\mathrm{F} = (x_{\mathrm{F}}, y_{\mathrm{F}})$ we have:

$$\begin{aligned}\frac{\partial u}{\partial \mathbf{n}}(x_{\mathrm{F}}, y_{\mathrm{F}}) \approx \Big(&-\frac{1}{4h}(2\theta_x+1)\theta_y(\theta_y-1)u(\mathrm{V}_1) + \frac{1}{h}\theta_x\theta_y(\theta_y-1)u(\mathrm{V}_2) - \\ &\frac{1}{4h}(2\theta_x-1)\theta_y(\theta_y-1)u(\mathrm{V}_3) + \frac{1}{2h}(1+2\theta_x)(\theta_y-1)(1+\theta_y)u(\mathrm{V}_4) - \\ &\frac{2}{h}\theta_x(\theta_y-1)(1+\theta_y)u(\mathrm{V}_5) + \frac{1}{2h}(2\theta_x-1)(\theta_y-1)(1+\theta_y)u(\mathrm{V}_6) - \\ &\frac{1}{4h}(1+2\theta_x)(\theta_y+1)\theta_y u(\mathrm{V}_7) + \frac{1}{h}\theta_x(\theta_y+1)\theta_y u(\mathrm{V}_8) - \\ &\frac{1}{4h}(2\theta_x-1)(\theta_y+1)\theta_y u(\mathrm{V}_9)\Big)\,\mathrm{n}_x + \Big(-\frac{1}{4h}\theta_x(1+\theta_x)(1-2\theta_y)u(\mathrm{V}_1) + \\ &\frac{1}{2h}(\theta_x-1)(1+\theta_x)(1-2\theta_y)u(\mathrm{V}_2) - \frac{1}{4h}(\theta_x-1)\theta_x(1-2\theta_y)u(\mathrm{V}_3) - \\ &\frac{1}{h}\theta_x(1+\theta_x)\theta_y u(\mathrm{V}_4) + \frac{2}{h}(\theta_x-1)(1+\theta_x)\theta_y u(\mathrm{V}_5) - \frac{1}{h}(\theta_x-1)\theta_x\theta_y u(\mathrm{V}_6) + \\ &\frac{1}{4h}\theta_x(1+\theta_x)(2\theta_y+1)u(\mathrm{V}_7) - \frac{1}{2h}(\theta_x-1)(1+\theta_x)(2\theta_y+1)u(\mathrm{V}_8) + \\ &\frac{1}{4h}(\theta_x-1)\theta_x(2\theta_y+1)u(\mathrm{V}_9)\Big)\,\mathrm{n}_y, \end{aligned} \tag{1.26}$$

where $\theta_x = \frac{x_8 - x_{\mathrm{P}}}{h}$ and $\theta_y = \frac{y_{\mathrm{P}} - y_4}{h}$ (see Figure 1.8).

It's obvious that, if in the formula (1.23) of the bilinear interpolation procedure the error $\mathcal{E} \sim \mathrm{O}(h^3)$, in the formula (1.26) we have $\mathcal{E} \sim \mathrm{O}(h^2)$.

The *numerical problem* of the problem (1.19), in the case we impose mixed conditions, is:

$$\begin{cases} -\frac{u_{i-1j}+u_{ij-1}-4u_{ij}+u_{i+1j}+u_{ij+1}}{h^2} = f_{ij} & \text{in } \Omega_h^{\mathrm{I}} \\ u(\mathrm{F}) = g_D(\mathrm{F}) & \text{in } \Omega_h^{\mathrm{F}D} \\ \frac{\partial u}{\partial \mathbf{n}}(\mathrm{F}) = g_N(\mathrm{F}) & \text{in } \Omega_h^{\mathrm{F}N} \end{cases}, \tag{1.27}$$

where the values of $u(\mathrm{F})$ and $\frac{\partial u}{\partial \mathbf{n}}(\mathrm{F})$ are approximated respectively with the formulas (1.21) and (1.25) of the bilinear interpolation or with the formulas (1.23) and (1.26) of the biquadratic interpolation.

1.3 Coco-Russo method in 3D

Finally, we present the most complex three-dimensional case.

The elliptic problem is:

$$\begin{cases} -(u_{xx} + u_{yy} + u_{zz}) = f & \text{in } \Omega \subset \mathbb{R}^3 \\ Bu = g & \text{in } \Gamma \end{cases}, \tag{1.28}$$

where the domain Ω has a particular geometry in a three-dimensional space.

1.3.1 Dirichlet boundary conditions

We begin to study the case in which we impose Dirichlet boundary conditions.

Discretization of the problem

First, we discretize the domain Ω (for simplicity, we suppose that $\Omega \subset [0,1]^3$) determining internal points and ghost points.

Now, we discretize the problem (1.28).

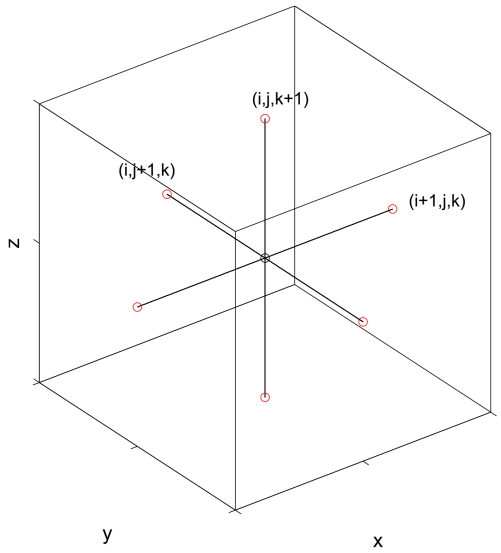


Figure 1.12: 7-point stencil

In the *internal points* we use the finite-difference approximation for $-(u_{xx} + u_{yy} + u_{zz})$ using the 7-point stencil.

The left figure (Figure 1.12) shows a 7-point stencil for the approximation of the second derivative $-\Delta u_{ijk}$ in the black internal point (x_i, y_j, z_k).

For each internal point (x_i, y_j, z_k), we have:

$$-u_{xx}(x_i, y_j, z_k) - u_{yy}(x_i, y_j, z_k) - u_{zz}(x_i, y_j, z_k) \approx$$
$$-\frac{u_{i-1jk} + u_{ij-1k} + u_{ijk-1} - 6u_{ijk} + u_{i+1jk} + u_{ij+1k} + u_{ijk+1}}{h^2}$$

Discretization of boundary conditions For each *ghost point* G we perfom the projection $\mathrm{F}(x_\mathrm{F}, y_\mathrm{F}, z_\mathrm{F})$ of the point G on the border of the domain Ω. In this point F we impose the Dirichlet condition through a *trilinear interpolation* procedure.

Second order accuracy on the border

Trilinear interpolation

The following figure (Figure 1.13) represents the position of the eight nodes (the blue circles $\mathrm{V}_i = \mathrm{V}_i(x_i, y_i, z_i)$, $i = 1, \ldots, 8$) used to perfom the trilinear interpolation to approximate the value that a given function u assumes at red point $\mathrm{C}(x_\mathrm{c}, y_\mathrm{c}, z_\mathrm{c})$.

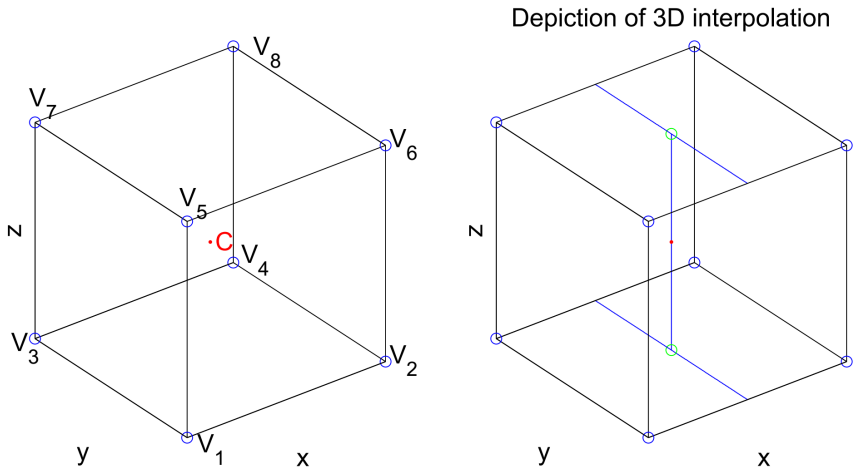


Figure 1.13: The eight blue circles are the points used to perform the trilinear interpolation and the red point C is the point at which we want to interpolate

To obtain the formula of trilinear interpolation we consider two bilinear interpolations combined with a linear interpolation. The formula of trilinear

interpolation does not depend of the order of the interpolation steps along the three axes.
If we indicate by

$$\theta_x = \frac{x_2 - x_{\mathrm{C}}}{h},\ \theta_y = \frac{y_3 - y_{\mathrm{C}}}{h} \text{ and } \theta_z = \frac{z_7 - z_{\mathrm{C}}}{h},$$

we can perfom:

- the bilinear interpolation at the nodes V_1, V_2, V_3 and V_4:

$$u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_1) \approx \theta_x\theta_y u(\mathrm{V}_1) + (1-\theta_x)\theta_y u(\mathrm{V}_2) +$$

$$\theta_x(1-\theta_y)u(\mathrm{V}_3) + (1-\theta_x)(1-\theta_y)u(\mathrm{V}_4),$$

- the bilinear interpolation at the nodes V_5, V_6, V_7 and V_8:

$$u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_5) \approx \theta_x\theta_y u(\mathrm{V}_5) + (1-\theta_x)\theta_y u(\mathrm{V}_6) +$$

$$\theta_x(1-\theta_y)u(\mathrm{V}_7) + (1-\theta_x)(1-\theta_y)u(\mathrm{V}_8),$$

- the linear interpolation at the nodes $(x_{\mathrm{C}}, y_{\mathrm{C}}, z_1)$ and $(x_{\mathrm{C}}, y_{\mathrm{C}}, z_5)$ (the green nodes in the right panel of the Figure 1.13) to approximate the value that the function u assumes in the red point C:

$$u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_{\mathrm{C}}) \approx\ \theta_z u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_1) + (1-\theta_z)u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_5) =$$

$$\theta_x\theta_y\theta_z u(\mathrm{V}_1) + (1-\theta_x)\theta_y\theta_z u(\mathrm{V}_2) + \theta_x(1-\theta_y)\theta_z u(\mathrm{V}_3) +$$
$$(1-\theta_x)(1-\theta_y)\theta_z u(\mathrm{V}_4) + \theta_x\theta_y(1-\theta_z)u(\mathrm{V}_5) +$$
$$(1-\theta_x)\theta_y(1-\theta_z)u(\mathrm{V}_6) + \theta_x(1-\theta_y)(1-\theta_z)u(\mathrm{V}_7) +$$
$$(1-\theta_x)(1-\theta_y)(1-\theta_z)u(\mathrm{V}_8). \tag{1.29}$$

The formula (1.29) represents the formula of the trilinear interpolation at the nodes V_1, V_2, V_3, V_4, V_5, V_6, V_7 and V_8 to approximate the value that the function u assumes in the point C.

Trilinear interpolation error

To calculate the trilinear interpolation error we must note that the trilinear interpolation polynomial was constructed using two bilinear interpolation polynomials and one linear interpolation and thus we must consider the errors associated with these interpolation polynomials.

Based on the formula (1.22), if $u \in \mathrm{C}^2$,

$$u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_1) = P(x_{\mathrm{C}}, y_{\mathrm{C}}, z_1) + \frac{1}{2}h^2 \left[\theta_{y_{\mathrm{P}}}(1-\theta_{x_{\mathrm{P}}})(-\theta_{x_{\mathrm{P}}})u''(\xi_1, \bar{\mu}_1, \bar{\zeta}_1) + \right.$$

$$\left.(1-\theta_{y_{\mathrm{P}}})(1-\theta_{x_{\mathrm{P}}})(-\theta_{x_{\mathrm{P}}})u''(\xi_2, \bar{\mu}_2, \bar{\zeta}_2) + (1-\theta_{y_{\mathrm{P}}})(-\theta_{y_{\mathrm{P}}})u''(\bar{\xi}_3, \mu_3, \bar{\zeta}_3)\right],$$

for some $\xi_1, \xi_2 \in [\min\{x_1, x_2, x_{\mathrm{C}}\}, \max\{x_1, x_2, x_{\mathrm{C}}\}]$,
$\mu_3 \in [\min\{y_1, y_3, y_{\mathrm{C}}\}, \max\{y_1, y_3, y_{\mathrm{C}}\}]$,

$$u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_5) = P(x_{\mathrm{C}}, y_{\mathrm{C}}, z_5) + \frac{1}{2}h^2 \left[\theta_{y_{\mathrm{P}}}(1-\theta_{x_{\mathrm{P}}})(-\theta_{x_{\mathrm{P}}})u''(\xi_4, \bar{\mu}_4, \bar{\zeta}_4) + \right.$$

$$\left.(1-\theta_{y_{\mathrm{P}}})(1-\theta_{x_{\mathrm{P}}})(-\theta_{x_{\mathrm{P}}})u''(\xi_5, \bar{\mu}_5, \bar{\zeta}_5) + (1-\theta_{y_{\mathrm{P}}})(-\theta_{y_{\mathrm{P}}})u''(\bar{\xi}_6, \mu_6, \bar{\zeta}_6)\right],$$

for some $\xi_4, \xi_5 \in [\min\{x_5, x_6, x_{\mathrm{C}}\}, \max\{x_5, x_6, x_{\mathrm{C}}\}]$,
$\mu_6 \in [\min\{y_5, y_7, y_{\mathrm{C}}\}, \max\{y_5, y_7, y_{\mathrm{C}}\}]$,

$$u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_{\mathrm{C}}) \approx \theta_z u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_1) + (1-\theta_z)u(x_{\mathrm{C}}, y_{\mathrm{C}}, z_5) - \frac{1}{2}(1-\theta_z)\theta_z h^2 u''(\bar{\xi}, \bar{\mu}, \zeta) =$$

$$P(x_{\mathrm{C}}, y_{\mathrm{C}}, z_{\mathrm{C}}) + \mathrm{O}(h^2),$$

for some $\zeta \in [\min\{z_1, z_5, z_{\mathrm{C}}\}, \max\{z_1, z_5, z_{\mathrm{C}}\}]$.

Therefore, the error $\mathcal{E} \sim \mathrm{O}(h^2)$.

The *numerical problem* of the problem (1.28), in the case we impose Dirichlet conditions, is:

$$\begin{cases} -\frac{u_{i-1jk}+u_{ij-1k}+u_{ijk-1}-6u_{ijk}+u_{i+1jk}+u_{ij+1k}+u_{ijk+1}}{h^2} = f_{ijk} & \text{in } \Omega_h^{\mathrm{I}} \\ u(\mathrm{F}) = g_D(\mathrm{F}) & \text{in } \Omega_h^{\mathrm{F}_D} \end{cases}, \tag{1.30}$$

where the value of $u(\mathrm{F})$ is approximated with the formula (1.29) of the trilinear interpolation and g_D is the value of exact solution in F.

1.3.2 Mixed boundary conditions

Now, we solve numerically the problem (1.28) imposing mixed boundary conditions.

The discretization of the domain Ω, of the problem in the internal points and of the boundary conditions in the points in which we impose Dirichlet conditions are done as in Section 1.3.1.

Discretization of boundary conditions

First order accuracy on the border

If we use a trilinear interpolation procedure to impose Neumann boundary conditions, from the formula (1.29), in each point $\mathrm{F} = (x_{\mathrm{F}}, y_{\mathrm{F}})$ we have:

$$\begin{aligned}
\frac{\partial u}{\partial \mathbf{n}}(\mathrm{F}) \approx \Big(&-\frac{1}{h}\theta_y\theta_z u(\mathrm{V}_1) + \frac{1}{h}\theta_y\theta_z u(\mathrm{V}_2) - \frac{1}{h}(1-\theta_y)\theta_z u(\mathrm{V}_3) + \frac{1}{h}(1-\theta_y)\theta_z u(\mathrm{V}_4)\\
&- \frac{1}{h}\theta_y(1-\theta_z)u(\mathrm{V}_5) + \frac{1}{h}\theta_y(1-\theta_z)u(\mathrm{V}_6) - \frac{1}{h}(1-\theta_y)(1-\theta_z)u(\mathrm{V}_7)\\
&+ \frac{1}{h}(1-\theta_y)(1-\theta_z)u(\mathrm{V}_8)\Big)\,\mathrm{n}_x + \Big(-\frac{1}{h}\theta_x\theta_z u(\mathrm{V}_1) - \frac{1}{h}(1-\theta_x)\theta_z u(\mathrm{V}_2)\\
&+ \frac{1}{h}\theta_x\theta_z u(\mathrm{V}_3) + \frac{1}{h}(1-\theta_x)\theta_z u(\mathrm{V}_4) - \frac{1}{h}\theta_x(1-\theta_z)u(\mathrm{V}_5) - \frac{1}{h}(1-\theta_x)(1-\theta_z)u(\mathrm{V}_6)\\
&- \frac{1}{h}(1-\theta_x)(1-\theta_z)u(\mathrm{V}_6) + \frac{1}{h}\theta_x(1-\theta_z)u(\mathrm{V}_7) + \frac{1}{h}(1-\theta_x)(1-\theta_z)u(\mathrm{V}_8)\Big)\,\mathrm{n}_y\\
&+ \Big(-\frac{1}{h}\theta_x\theta_y u(\mathrm{V}_1) - \frac{1}{h}(1-\theta_x)\theta_y u(\mathrm{V}_2) - \frac{1}{h}\theta_x(1-\theta_y)u(\mathrm{V}_3)\\
&- \frac{1}{h}(1-\theta_x)(1-\theta_y)u(\mathrm{V}_4) + \theta_x\theta_y u(\mathrm{V}_5) + \frac{1}{h}(1-\theta_x)\theta_y u(\mathrm{V}_6)\\
&+ \frac{1}{h}\theta_x(1-\theta_y)u(\mathrm{V}_7) + \frac{1}{h}(1-\theta_x)(1-\theta_y)u(\mathrm{V}_8)\Big)\,\mathrm{n}_z, \qquad (1.31)
\end{aligned}$$

where $\mathbf{n} = (\mathrm{n}_x, \mathrm{n}_y, \mathrm{n}_z)$ is the outward unit normal, $\theta_x = \frac{x_2 - x_{\mathrm{C}}}{h}$, $\theta_y = \frac{y_3 - y_{\mathrm{C}}}{h}$ and $\theta_z = \frac{z_7 - z_{\mathrm{C}}}{h}$ (see Figure 1.13).

It's obvious that, if in the formula (1.29) of the trilinear interpolation procedure the error $\mathcal{E} \sim \mathrm{O}(h^2)$, in the formula (1.31) we have $\mathcal{E} \sim \mathrm{O}(h)$.

The *numerical problem* of the problem (1.28), in the case we impose mixed conditions, is:

$$\begin{cases} -\frac{u_{i-1jk}+u_{ij-1k}+u_{ijk-1}-6u_{ijk}+u_{i+1jk}+u_{ij+1k}+u_{ijk+1}}{h^2} = f_{ijk} & \text{in } \Omega_h^{\mathrm{I}} \\ u(\mathrm{F}) = g_D(\mathrm{F}) & \text{in } \Omega_h^{\mathrm{F}D} \\ \frac{\partial u}{\partial \mathbf{n}}(\mathrm{F}) = g_N(\mathrm{F}) & \text{in } \Omega_h^{\mathrm{F}N} \end{cases}, \tag{1.32}$$

where the values of $u(\mathrm{F})$ and $\frac{\partial u}{\partial \mathbf{n}}(\mathrm{F})$ are approximated with the formulas (1.29) and (1.31) respectively.

As in 1D and 2D, also in 3D it is possible to perform the *triquadratic interpolation procedures* on the ghost points which allow a third-order accuracy at the border in the case in which we impose the Dirichlet conditions and a second-order accuracy at the border in the case in which we impose Neumann boundary conditions. Such interpolations are used in the next Chapter.

Symmetrization method

Let A_h be the matrix in 1D, 2D or 3D. It is possible to symmetrize the matrix and observe the behaviors of the inverse matrix and of the consistency error. It is clear that the behavior of the error in the case in which the matrix A_h is symmetric coincides with the behavior of the error in the case in which the matrix A_h is not symmetric.
We provide below a generally valid symmetrization method for the matrices A_h in 1D, 2D or 3D.

We write the linear system in this way:

$$A_h u_h = \begin{pmatrix} A_{ii} & A_{ig} \\ A_{gi} & A_{gg} \end{pmatrix} \begin{pmatrix} u_i \\ u_g \end{pmatrix} = \begin{pmatrix} f_i \\ f_g \end{pmatrix} = f, \tag{1.33}$$

where u_g and u_i are the vectors of numerical solution related to ghost points and to internal points respectively.

We look for a matrix C such that CA_h is symmetric. Choosing:

$$C = \begin{pmatrix} I & b \\ a^T & c \end{pmatrix},$$

the system (1.33) becomes:

$$CA_h u_h = \begin{pmatrix} A_{ii} + bA_{gi} & A_{ig} + bA_{gg} \\ a^T A_{ii} + cA_{gi} & a^T A_{ig} + cA_{gg} \end{pmatrix} \begin{pmatrix} u_i \\ u_g \end{pmatrix} = \begin{pmatrix} f_i + bf_g \\ a^T f_i + cf_g \end{pmatrix} = Cf. \tag{1.34}$$

For CA_h to be symmetrical it is necessary that:

- $A_{ii} + bA_{gi}$ must be symmetric $\Rightarrow b = \alpha A_{gi}^T$;
- $a^T A_{ii} + cA_{gi}$ and $A_{ig} + bA_{gg}$ must be the one transposing the other, thus you must have $A_{ig} + \alpha A_{gi}^T A_{gg} = (a^T A_{ii} + cA_{gi})^T \Rightarrow c = \alpha A_{gg}^T$ and $a = (A_{ii}^T)^{-1} A_{ig}$;
- $a^T A_{ig} + cA_{gg}$ must be symmetric. With these choices of a and c also the matrix $a^T A_{ig} + cA_{gg}$ is symmetric.

Remark 1.2: Our aim is now to verify if the numerical results about the convergence, the consistency and the stability of the method can depend on the symmetry of the matrix A_h.

In 1D in the case in which we impose the boundary conditions by performing the linear interpolations it is possible to symmetrize the matrix A_h both with the symmetrization method (1.34) just described and using formulas (1.9) and (1.18). If we perform the quadratic interpolations it is possible to symmetrize the matrix A_h only with the symmetrization method (1.34) for different values of the parameter $\alpha = h^m$ ($m = \ldots, -1, 0, 1, \ldots$).

We can observe that the convergence results of the Coco-Russo method do not depend on the symmetrization of the matrix A_h, since the numerical solution obtained by performing the symmetrization of the matrix coincides with the solution of the system in which the matrix A_h is not symmetric. Instead, since the consistency error is given by $\tau_h = A_h e_h$ it is clear that the consistency of the method may depend on the symmetrization of the matrix A_h. Depends on the symmetrization of the matrix A_h also the stability being connected to the behavior of the inverse matrix of A_h.
In both cases (symmetrization method (1.34) and formulas (1.9) and (1.18)) is not convenient to symmetrize the matrix A_h.

If we symmetrize the matrix with the formulas (1.9) and (1.18) in the matrix A_h the term $-\frac{\theta_a}{(1-\theta_a)h^2}$ (and/or $-\frac{\theta_b}{(1-\theta_b)h^2}$) appears. It is clear that when $\theta_a \sim 1$ (or $\theta_b \sim 1$) we have consistency problems. The stability of the numerical method is guaranteed, indeed, the norm of the inverse matrix of A_h tends to a constant for large values of n.

If we use the symmetrization method (1.34) there are consistency problems for values of m less than a certain threshold that is around -1, -2, -3. Above these values we haven't consistency problems not even for large values of θ_a (or of θ_b), but the accuracy of the second-order for the consistency error is not assured. Moreover, in this case the stability is not always guaranteed because there is values of the parameter m for which the norm of the inverse matrix of A_h grows for large value of n.

On the basis of this observation we did not apply this symmetrization meth-

ods to two-dimensional and three-dimensional cases. We will go into more detail on this topic in the next chapter, applying it to various problems (Dirichlet or mixed) in 1D.

2

Numerical tests for the Coco-Russo method

This Chapter provides numerical tests in 1D, 2D and 3D. Our aim is to prove numerically the convergence and the consistency of the method and obtaining numerical evidence on the stability in different norms.
From inequality

$$\|e_h\|_{L^p} \leq \|\mathrm{A}_h^{-1}\|_p \|\tau_h\|_{L^p},\ p = 1, 2, \infty$$

we can observe that to test the convergence and the consistency of the method is sufficient to study the behaviors of $\|e_h\|_{L^p}$ and $\|\tau_h\|_{L^p}$ respectively. But in this way we do not get any information on the stability of the method, thus we must also study $\|\mathrm{A}_h^{-1}\|_p$.
For each spatial dimension, we will present numerical tests for both Dirichlet problem and mixed problem, performing both linear interpolations and quadratic interpolations to impose boundary conditions on points F that belongs to the border but they are not grid points.
We will see that in the Dirichlet problem both the linear interpolations and the quadratic interpolations provide a second-order accuracy, while in mixed problem only the quadratic interpolations provide such accuracy. This is because the linear interpolations give us a first-order accuracy at the border in the case we impose Neumann conditions.

In all the following figures for any quantity $g(n)$ the red line is the straight best-fit line in logarithmic scale; we indicate with $sl(g) \approx \frac{\mathrm{d}\log(g)}{\mathrm{d}N}$, so that, for example, the norm of the inverse matrix of A_h is approximated by $\|\mathrm{A}_h^{-1}\| \approx c\ N^{sl(\|\mathrm{A}_h^{-1}\|)}$. For error and truncation errors we expect negative values of $sl(g)$.

2.1 Numerical tests in 1D

We start with the one-dimensional case. For each example we will compare the numerical results obtained performing the linear interpolations with those obtained performing the quadratic interpolations, both for Dirichlet problem and mixed problem.

2.1.1 Dirichlet boundary conditions - Only one ghost point

We start showing the numerical results in the case in which only the extreme **a** is not a grid point (**b**=1) imposing Dirichlet boundary conditions both in **a** and in **b**.

Behavior of the inverse matrix of $\mathbf{A}_h$

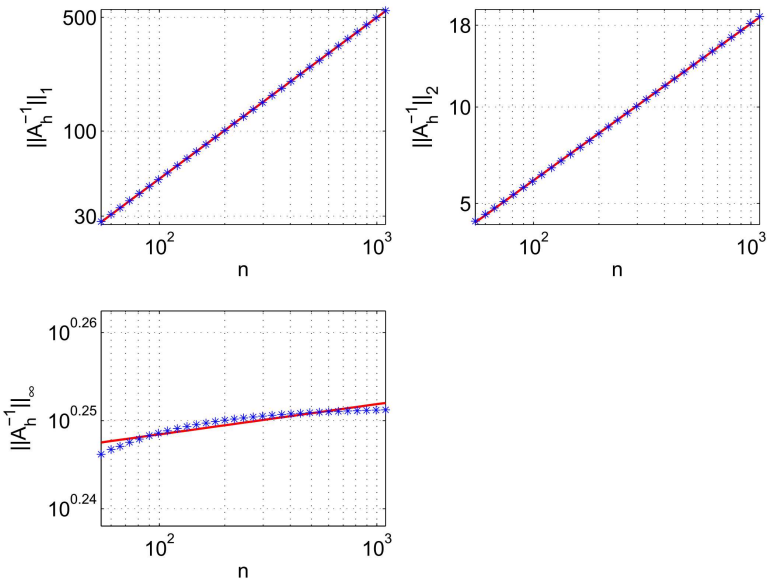


Figure 2.1: p - Norm of the inverse matrix of A_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left), $\theta_\mathrm{a} = 0.5$

The Figure 2.1 shows the behavior of the norm of A_h^{-1} as a function of the number of the grid points in the case in which we perform the linear inter-

polation procedure to impose the boundary condition in **a** and the domain is $[\mathbf{a}, \pi] \subset [0, \pi]$. We can read the values of the best-fit slope in the table below:

$n_{\min} = 54,\ n_{\max} = 1096$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
	0.9923	0.4892	$0.3429\ 10^{-2}$

The position of the extremity **a** is $\mathbf{a} = \frac{h}{2}$, but the same accuracy is confirmed also for different positions of **a** in $]0, h[$.

The following tables show the numerical results in the following cases:

- the domain is $[\mathbf{a}, 1] \subset [0, 1]$ and we have performed the linear interpolation procedure to impose the boundary condition in **a**:

$n_{\min} = 54,\ n_{\max} = 1096$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
$\mathbf{a} = \frac{h}{2}$	0.9923	0.4935	$-0.3876\ 10^{-2}$

- the domain is $[\mathbf{a}, \pi] \subset [0, \pi]$ and we have performed the quadratic interpolation procedure to impose the boundary condition in **a**:

$n_{\min} = 54,\ n_{\max} = 1096$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
$\mathbf{a} = \frac{h}{2}$	0.9923	0.4892	$0.3463\ 10^{-2}$

- the domain is $[\mathbf{a}, 1] \subset [0, 1]$ and we have performed the quadratic interpolation procedure to impose the boundary condition in **a**:

$n_{\min} = 54,\ n_{\max} = 1096$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
$\mathbf{a} = \frac{h}{2}$	0.9923	0.4935	$-0.3885\ 10^{-2}$

In all cases, the numerical results show that $\|\mathrm{A}_h^{-1}\|_p \sim \mathrm{O}(n^{\frac{1}{p}})$, $p = 1, 2, \infty$, thus we have stability in the ∞-norm.

Behavior of the spectral radius of $\mathbf{A}_h^{-1}$

Now we show the behavior of the spectral radius of the matrix A_h^{-1}. The study of the spectral radius of A_h^{-1} combined with the study of the inverse matrix of A_h gives us information on the stability of the numerical method.

The figure on the left shows the behavior of the reciprocal of the eigenvalue of minimum module of the matrix A_h as a function of n. The figure is obtained choosing $\mathbf{a} = \frac{h}{2}$, but we have the same result also with a different choice of the position of $\mathbf{a}$ in $]0, h[$. The figure on the right shows the behavior of the reciprocal of the eigenvalue of minimum module of the matrix A_h as a function of the position of $\mathbf{a}$, in the case in which we perform the linear interpolation procedure to impose the boundary condition in $\mathbf{a}$ and the domain is $[\mathbf{a}, \pi] \subset [0, \pi]$.

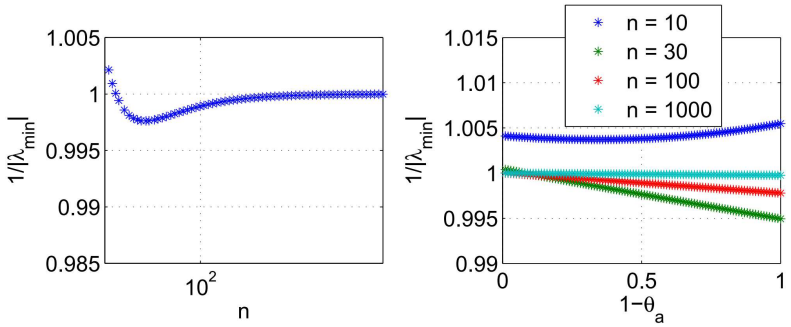


Figure 2.2: Dependence of the reciprocal of the eigenvalue of minimum module of the matrix A_h ($|\lambda_{\min}|$) on the number n of the grid points ($\theta_a = 0.5$) (left panel) and on the parameter $1 - \theta_a$ ($n = 100$) (right panel)

The two figures are in agreement and they show that the reciprocal of the eigenvalue of minimum module of the matrix A_h essentially does not depend on the position of $\mathbf{a}$ in $]0, h[$ and it tends to a constant for large values of n. In particular, the figures show that this constant is 1.

We have proved that we have the same result ($\rho(A_h^{-1}) \to 1$)) even in cases in which:

- the domain is $[\mathbf{a}, 1] \subset [0, 1]$ and we have performed the linear interpolation procedure to impose the boundary condition in $\mathbf{a}$;
- the domain is $[\mathbf{a}, \pi] \subset [0, \pi]$ and we have performed the quadratic interpolation procedure to impose the boundary condition in $\mathbf{a}$;
- the domain is $[\mathbf{a}, 1] \subset [0, 1]$ and we have performed the quadratic interpolation procedure to impose the boundary condition in $\mathbf{a}$.

Now, we evaluate the behaviors of the consistency error τ_h and of the error e_h on different functions f.

We have approximated $\|\tau_h\|_{L^p}$ and $\|e_h\|_{L^p}$ $(p = 1, 2, \infty)$ in this way:

$$\|a_h\|_{L^1} = h\sum_{i=0}^{n-1}|a_h(x_i)|,\ \|a_h\|_{L^2} = \sqrt{h\sum_{i=0}^{n-1}|a_h(x_i)|^2},\ \|a_h\|_{L^\infty} = \max_{i=0,\dots,n-1}|a_h(x_i)|,$$

where a_h is τ_h or e_h.

First example

We solve numerically the equation

$$-u_{xx} = \sin(x) \text{ in }]\mathbf{a}, \pi[\subset [0, \pi], \tag{2.1}$$

in which we impose Dirichlet boundary conditions in $\mathbf{a}$ and in $\mathbf{b} = \pi$. We known the exact solution of (2.1) that it is $u = \sin(x)$. Therefore, the problem to be solved numerically is:

$$\begin{cases} -u_{xx} = \sin(x) & \text{in }]\mathbf{a}, \pi[\subset [0, \pi] \\ u(\mathbf{a}) = \sin(\mathbf{a}) \\ u(\pi) = \sin(\pi) \end{cases}. \tag{2.2}$$

We discretize the interval $[0, \pi]$ and the problem (2.2) as we have described in the Section 1.1.1. Thus, we have $n+1$ grid points that we call $x_0, x_1, \dots, x_n$ and we suppose that $\mathbf{a}$ isn't a grid point but it is placed between x_0 and x_1. The spatial step is $h = \frac{\pi}{n}$.

Behavior of the consistency error τ_h

Linear interpolation procedure

The following figure (Figure 2.3) shows the behavior of the norm of the consistency error τ_h as a function of the number of the grid points, in the case we perform the linear interpolation procedure. We can read the values of the best-fit slope in the table below, which shows that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0012	-1.9996	-1.9910

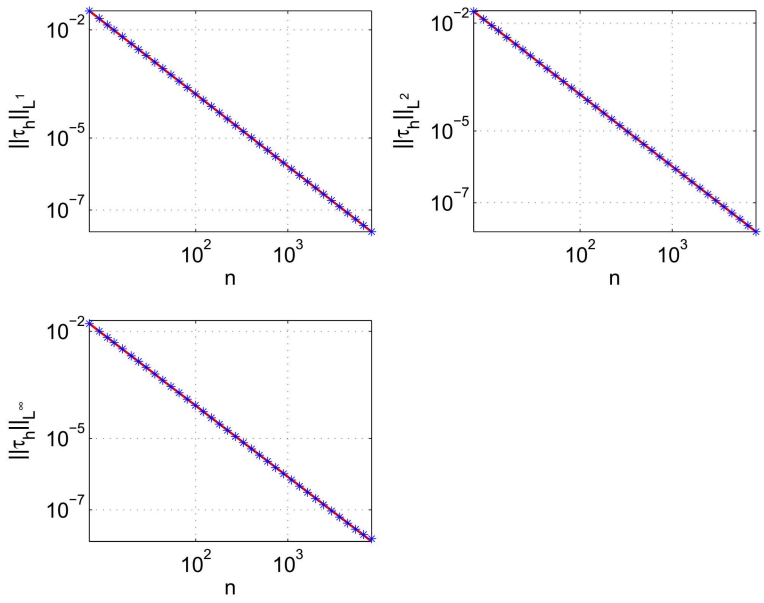


Figure 2.3: p - Norm of the consistency error τ_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

Quadratic interpolation procedure

The following table reports the values of the best-fit slope in the case we perform the quadratic interpolation procedure, which shows that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0009	-1.9996	-1.9916

In both cases (linear and quadratic interpolations), the position of the extremity **a** is not fixed but varies in $]0, h[$. The results show that the norm of τ_h does not depend on the position of **a**.

Behavior of the error e_h

Linear interpolation procedure

The following figure (Figure 2.4) shows the behavior of the norm of the error e_h as a function of the number of the grid points in the case in which

we perform the linear interpolation procedure. We can read the values of the best-fit slope in the table below, which shows that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9920	-1.9925	-1.9930

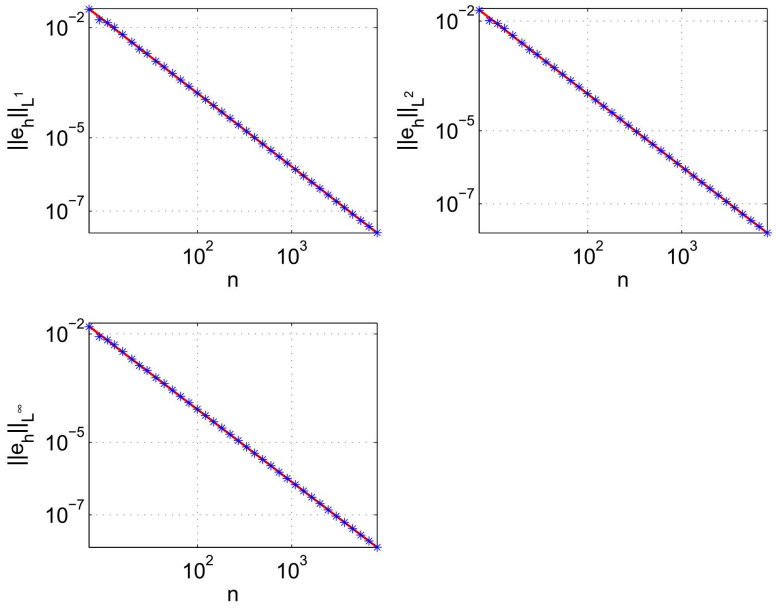


Figure 2.4: p - Norm of the error e_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

Quadratic interpolation procedure

Also in the case in which we perform the quadratic interpolation procedure we obtain that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9755	-1.9766	-1.9792

In both cases, the position of the extremity **a** is not fixed but varies in $]0, h[$. The results show that the norm of e_h does not depend clearly on the position of **a**.

Second example

We solve numerically the equation

$$-u_{xx} = e^x \text{ in }]\mathbf{a}, 1[\subset [0,1], \tag{2.3}$$

in which we impose Dirichlet boundary conditions in $\mathbf{a}$ and in $\mathbf{b} = 1$. We known the exact solution of (2.3) that it is $u = -e^x$. Therefore, the problem to be solved numerically is:

$$\begin{cases} -u_{xx} = e^x & \text{in }]\mathbf{a}, 1[\subset [0,1] \\ u(\mathbf{a}) = -e^{\mathbf{a}} & \\ u(1) = -e \end{cases} . \tag{2.4}$$

We discretize the interval $[0,1]$ and the problem (2.4) as we have described in the Section 1.1.1. The spatial step is $h = \frac{1}{n}$.

Behavior of the consistency error τ_h

In the following tables we can read the values of the best-fit slope of the norm of τ_h as a function of the number of the grid points. They show that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

Linear interpolation procedure

$n_{\min} = 7,\ n_{\max} = 2980$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-1.9975	-1.9954	-1.9653

Quadratic interpolation procedure

$n_{\min} = 7,\ n_{\max} = 2980$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-1.9824	-1.9885	-1.9685

In both cases, the results about the norm of the consistency error were obtained varying the position of $\mathbf{a}$ in $]0, h[$. The norm of the consistency error does not depend on the position of $\mathbf{a}$.

Remark 2.1 - on the behavior of the consistency error: We can observe, if we perform the linear interpolation procedure, in both **First example** and **Second example**, the consistency error is constant as a function of the position of **a** in $]0, h[$. It is clear that these are particular examples, since is possible to prove that in general the L^∞-norm of the consistency error depends on the position of **a**. For example if we consider the Dirichlet problem $-u_{xx} = \cos(x)$ in $[\mathbf{a}, \pi]$, the L^∞-norm of τ_h is smaller if $\theta_\mathrm{a} \sim 0$ or $\theta_\mathrm{a} \sim 1$ and it is larger if $\theta_\mathrm{a} \sim 0.5$. Only in the case in which $\theta_\mathrm{a} = 0.5$ also the L^∞-norm is constant as a function of the position of **a**. Instead, if we perform the quadratic interpolation procedure, the L^p-norm of τ_h $(p = 1, 2, \infty)$ is constant as a function of the position of **a**.

Thus in general the behavior of the consistency error is as follows: L^1- and L^2-norms are constant as a function of the position of **a**, L^∞-norm depends instead on the position of **a** in $]0, h[$ if we perform the linear interpolation. We can justify this result. The consistency error is a local error (it depends only on the values of the solution at that specific grid point and on the neighbors, unlike the numerical error e_h that depends globally on all grid values). This means that the consistency error is the same inside the grid (3-point finite difference of the second derivative) instead it is represented by the consistency error of the interpolation error on the ghost points. This is why L^1- and L^2-norms are almost constants (they are a kind of average on all grid points, and the number of ghost points is negligible with respect to the number of internal grid points) and L^∞ is not (it is the highest error between internal points and ghost points). The reason why L^∞-norm of the consistency error does not depends on **a** in the case of quadratic interpolations is because in this case the consistency error on the ghost points is much smaller $(\mathrm{O}(h^3))$ than the interior consistency error $(\mathrm{O}(h^2))$, so the maximum error is always the inner consistency error, which is almost constant overall.

Behavior of the error e_h

In the following tables we can read the values of the best-fit slope concerning the norm of e_h as a function of the number of the grid points. They show that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

LINEAR INTERPOLATION PROCEDURE

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|\|e_h\|\|_{L^1})$	$sl(\|\|e_h\|\|_{L^2})$	$sl(\|\|e_h\|\|_{L^\infty})$
	-2.0338	-2.0294	-2.0156

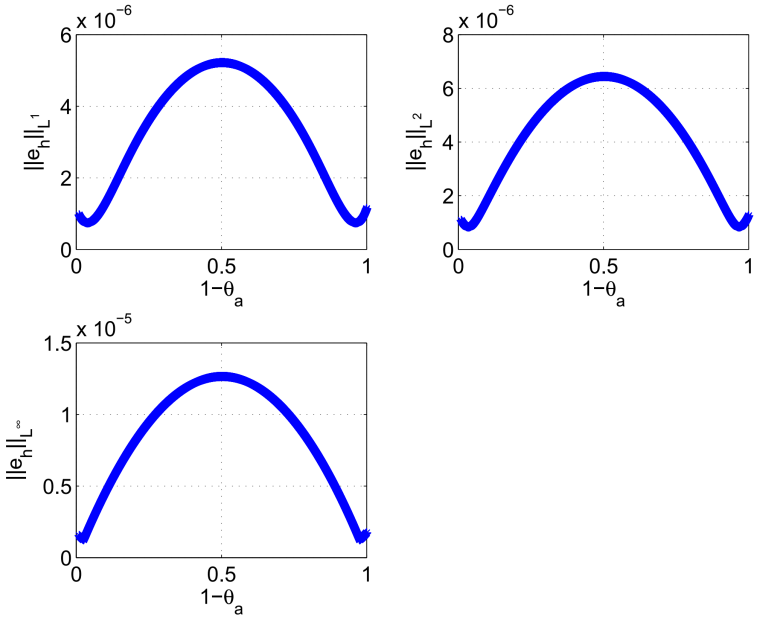


Figure 2.5: p - Norm of the error e_h as a function of the position of **a** ($n = 100$): $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

QUADRATIC INTERPOLATION PROCEDURE

$n_{\min} = 7,\ n_{\max} = 2980$	$sl(\|\|e_h\|\|_{L^1})$	$sl(\|\|e_h\|\|_{L^2})$	$sl(\|\|e_h\|\|_{L^\infty})$
	-2.0107	-2.0086	-2.0056

In both cases, the results about the norm of the error were obtained varying the position of **a** in $]0, h[$. In the case in which we perform the quadratic interpolation procedure, the norm of the error does not depend clearly on the position of **a**. Instead, if we perform the linear interpolation procedure, the norm of the error is smaller if $\theta_a \sim 0$ or $\theta_a \sim 1$ and it is larger if $\theta_a = 0.5$, as we can observe in Figure 2.5, which shows how $\|e_h\|_{L^p}$ varies as a function of the position of **a**. We have chosen $n = 100$.

Conclusions

We itemize the numerical results on the *Dirichlet problems* (1.8) and (1.12) in $[\mathbf{a}, 1]$. We have chosen not to fix the position of $\mathbf{a}$, but to move $\mathbf{a}$ in $]0, h[$, so as to see if the results may or may not depend on its position. We consider two distinct cases: A_h not symmetric and A_h symmetric.

A_h not symmetric

In the case in which A_h is not symmetric, the numerical results are the following:

- $\|\mathrm{A}_h^{-1}\|_p \sim \mathrm{O}(n^{\frac{1}{p}}),\ p = 1, 2, \infty;$
- $\rho(\mathrm{A}_h^{-1}) \to 1;$
- $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty;$
- $\|e_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty.$

The norm of the consistency error and the norm of the error **can depend on the position of a** in the case in which we perform the linear interpolation to impose the boundary condition in $\mathbf{a}$, which provides a second-order accuracy at the border as on the internal points.
In particular we have observed that L^1- and L^2-norms of the consistency error are constant as a function of $\mathbf{a}$, while L^∞-norm of the consistency error is not. In **Remark 2.1** we have provided an explanation for this behavior of the consistency error.

Instead, the norm of the consistency error and the norm of the error **do not depend on the position of a** in the case in which we perform the quadratic interpolation to impose the boundary condition in $\mathbf{a}$, which provides a third-order accuracy at the border.

A_h symmetric

In the case in which the matrix A_h is symmetric the behaviors of the norms of the consistency error and of the inverse matrix of A_h depend on the symmetrization method used.

If we symmetrize the matrix A_h through the symmetrization method (1.34) the norm of τ_h does not depends on the position of **a** and

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h),$$

thus we haven't consistency problems. However, there is a threshold of negative values of m below which problems of consistency are starting to be verified. These values are around -1, -2, -3, depending on the test examined.
The spectral radius of A_h^{-1} and $\|\mathrm{A}_h^{-1}\|_p$ grow as a function of n for values of $m \geq 0$ and tend to a constant for values $m < 0$. Therefore, we have stability problems in the case in which $m \geq 0$.
Based on the above, is not convenient to symmetrize the matrix A_h.

If we symmetrize the matrix A_h through the formula (1.9) (thus only in the case in which we perform the linear interpolation procedure to impose the boundary condition in **a**) the order of accuracy of τ_h depends on the position of **a**. Performing various numerical tests we have observed that

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{1}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h^0),$$

therefore the consistency of the method is not guaranteed in L^∞-norm.
Only if $\theta_{\mathrm{a}} \to 0$, $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$. While, if $\theta_{\mathrm{a}} \to 1$ we have consistency problems.
The spectral radius of A_h^{-1} and $\|\mathrm{A}_h^{-1}\|_p$ tend to a constant for large values of n.
Therefore, also in this case is not convenient to symmetrize the matrix A_h.

2.1.2 Dirichlet boundary conditions - Two ghost points

We proceed providing the same numerical tests in the case in which both extremes **a** and **b** aren't grid points imposing Dirichlet boundary conditions both in **a** and in **b**.

Behavior of the inverse matrix of $\mathbf{A}_h$ and of the spectral radius of $\mathbf{A}_h^{-1}$

The behaviors of the inverse matrix of A_h and of its the spectral radius coincide with the case in which **b** is a grid point, that is $\|\mathrm{A}_h^{-1}\|_p \sim \mathrm{O}(n^{\frac{1}{p}})$ $(p = 1, 2, \infty)$ and $\rho(\mathrm{A}_h^{-1}) \to 1$, both in the case in which we perform the linear interpolation procedures and in the case in which we perform the quadratic interpolation procedures. The following tables show the numerical results about the norm of the inverse matrix in the following cases:

- the domain is $[\mathbf{a}, \mathbf{b}] \subset [0, \pi]$ and we have performed the linear interpolation procedures to impose the boundary conditions in **a** and in **b**:

$n_{\min} = 54,\ n_{\max} = 1096$	$sl(\|\mathrm{A}_h^{-1}\|_1)$	$sl(\|\mathrm{A}_h^{-1}\|_2)$	$sl(\|\mathrm{A}_h^{-1}\|_\infty)$
$\mathbf{a} = \frac{h}{2}$, $\mathbf{b} = \pi - \frac{h}{2}$	0.9947	0.4937	$0.5699\ 10^{-2}$

- the domain is $[\mathbf{a}, \mathbf{b}] \subset [0, 1]$ and we have performed the linear interpolation procedures to impose the boundary conditions in **a** and in **b**:

$n_{\min} = 54,\ n_{\max} = 1096$	$sl(\|\mathrm{A}_h^{-1}\|_1)$	$sl(\|\mathrm{A}_h^{-1}\|_2)$	$sl(\|\mathrm{A}_h^{-1}\|_\infty)$
$\mathbf{a} = \frac{h}{2}$, $\mathbf{b} = 1 - \frac{h}{2}$	0.9947	0.4974	$0.1139\ 10^{-2}$

- the domain is $[\mathbf{a}, \mathbf{b}] \subset [0, \pi]$ and we have performed the quadratic interpolation procedures to impose the boundary conditions in **a** and in **b**:

$n_{\min} = 54,\ n_{\max} = 1096$	$sl(\|\mathrm{A}_h^{-1}\|_1)$	$sl(\|\mathrm{A}_h^{-1}\|_2)$	$sl(\|\mathrm{A}_h^{-1}\|_\infty)$
$\mathbf{a} = \frac{h}{2}$, $\mathbf{b} = \pi - \frac{h}{2}$	0.9947	0.4937	$0.5744\ 10^{-2}$

- the domain is $[\mathbf{a}, \mathbf{b}] \subset [0, 1]$ and we have performed the quadratic interpolation procedures to impose the boundary conditions in **a** and in **b**:

$n_{\min} = 54,\ n_{\max} = 1096$	$sl(\|\|A_h^{-1}\|\|_1)$	$sl(\|\|A_h^{-1}\|\|_2)$	$sl(\|\|A_h^{-1}\|\|_\infty)$
$\mathbf{a} = \frac{h}{2},\ \mathbf{b} = 1 - \frac{h}{2}$	0.9947	0.4974	$0.1148\ 10^{-2}$

First example

We solve numerically the equation

$$-u_{xx} = \sin(x) \text{ in }]\mathbf{a}, \mathbf{b}[\subset [0, \pi],$$

in which we impose Dirichlet boundary conditions in **a** and in **b**. The problem to be solved numerically is:

$$\begin{cases} -u_{xx} = \sin(x) & \text{in }]\mathbf{a}, \mathbf{b}[\subset [0, \pi] \\ u(\mathbf{a}) = \sin(\mathbf{a}) \\ u(\mathbf{b}) = \sin(\mathbf{b}) \end{cases}.$$

We place $\mathbf{b} = \pi - \mathbf{a}$.

Behavior of the consistency error τ_h

The following tables report the values of the best-fit slope concerning the norm of τ_h as a function of the number of the grid points, and they show that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

Linear interpolation procedure

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|\|\tau_h\|\|_{L^1})$	$sl(\|\|\tau_h\|\|_{L^2})$	$sl(\|\|\tau_h\|\|_{L^\infty})$
	-2.0061	-2.0006	-1.9922

Quadratic interpolation procedure

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|\|\tau_h\|\|_{L^1})$	$sl(\|\|\tau_h\|\|_{L^2})$	$sl(\|\|\tau_h\|\|_{L^\infty})$
	-1.9985	-1.9993	-1.9908

In both cases, the position of the extremity **a** is not fixed but varies in $]0, h[$, thus the position of **b** varies in $]\pi - h, \pi[$. The results show that the norm of τ_h does not depend on the position of **a** and on the position of **b**.

Behavior of the error e_h

The following tables report the values of the best-fit slope concerning the norm of e_h as a function of the number of the grid points, and they show that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

LINEAR INTERPOLATION PROCEDURE

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9926	-1.9914	-1.9915

QUADRATIC INTERPOLATION PROCEDURE

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9437	-1.9434	-1.9446

In both cases, the positions of the extremes **a** and **b** are not fixed but varies in $]0, h[$ and in $]\pi - h, \pi[$ respectively. The results show that the norm of e_h does not depend on the position of **a** and on the position of **b**.

Second example

We solve numerically the equation

$$-u_{xx} = e^x \text{ in }]\mathbf{a}, \mathbf{b}[\subset [0, 1],$$

in which we impose Dirichlet boundary conditions in **a** and in **b**. The problem to be solved numerically is:

$$\begin{cases} -u_{xx} = e^x & \text{in }]\mathbf{a}, \mathbf{b}[\subset [0, 1] \\ u(\mathbf{a}) = -e^{\mathbf{a}} \\ u(\mathbf{b}) = -e^{\mathbf{b}} \end{cases}.$$

We place $\mathbf{b} = 1 - \mathbf{a}$.

Behavior of the consistency error τ_h

LINEAR INTERPOLATION PROCEDURE

The norm of τ_h depends only on the position of **b**. Indeed, if we fix the position of **b** in $]1-h, 1[$, $\|\tau_h\|_{L^p}$ $(p = 1, 2, \infty)$ is constant as a function of the position of **a** in $]0, h[$. This does not happen if we fix the position of **a** in $]0, h[$ and varies only the position of **b** in $]1-h, 1[$. In particular, the L^1- and L^2-norms of τ_h are constant, the L^∞-norm is smaller if $\theta_b \sim 0$ or $\theta_b \sim 1$, as the following figures (Figure 2.6, Figure 2.7) show:

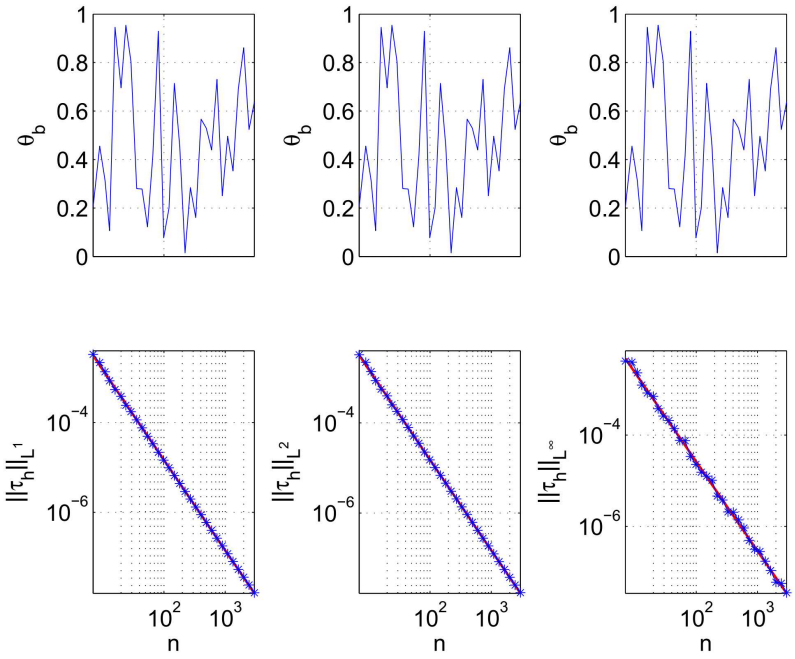


Figure 2.6: p - Norm of the consistency error τ_h as a function of the number of the grid points: $p = 1$ (left), $p = 2$ (center), $p = \infty$ (right)

We can read the values of the best-fit slope in the table below, which shows that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 2980$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0156	-2.0101	-1.9538

The numerical results (and the figures) were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]1-h, 1[$.

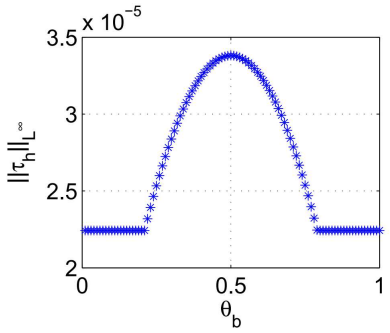


Figure 2.7: L^∞-norm of the consistency error τ_h as a function of the position of **b** ($n = 100$)

QUADRATIC INTERPOLATION PROCEDURE

$n_{\min} = 7,\ n_{\max} = 2980$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-1.9833	-1.9886	-1.9687

In these numerical results the position of **a** varies in $]0, h[$ and thus the position of **b** varies in $]1-h, 1[$. The norm of τ_h does not depend on the position of **a** and on the position of **b**.

Remark 2.2 - on the behavior of the consistency error: We can observe a behavior of the consistency error similar to that observed in the previous section in which **b** is a grid point. In general, if we perform the linear interpolation procedures to impose the boundary conditions in **a** and in **b** the L^1- and L^2-norms of the consistency error are constant and the L^∞-norm depends on the position of **a** and on the position of **b**. Only if $\theta_{\mathrm{a}} = 0.5$ or $\theta_{\mathrm{b}} = 0.5$ also the L^∞-norm is constant. We can draw the same conclusions of the previous case (see **Remark 2.1**).

Behavior of the error e_h

LINEAR INTERPOLATION PROCEDURE

The norm of the error depends on the position of **a** and on the position of **b** in the case in which we perform the linear interpolation procedures. It is smaller if $\theta_{\mathrm{a}} \sim 0$ or $\theta_{\mathrm{a}} \sim 1$ and at the same time $\theta_{\mathrm{b}} \sim 0$ or $\theta_{\mathrm{b}} \sim 1$ (see Figure

2.8). Only in the case in which $\mathbf{b} = 1 - \frac{h}{2}$ we observe a different behavior of $\|e_h\|_{L^p}$: in $\|e_h\|_{L^1}$ and $\|e_h\|_{L^2}$ the dipendence from $\mathbf{a}$ is less evident, while $\|e_h\|_{L^\infty}$ is constant to vary of the position of $\mathbf{a}$.

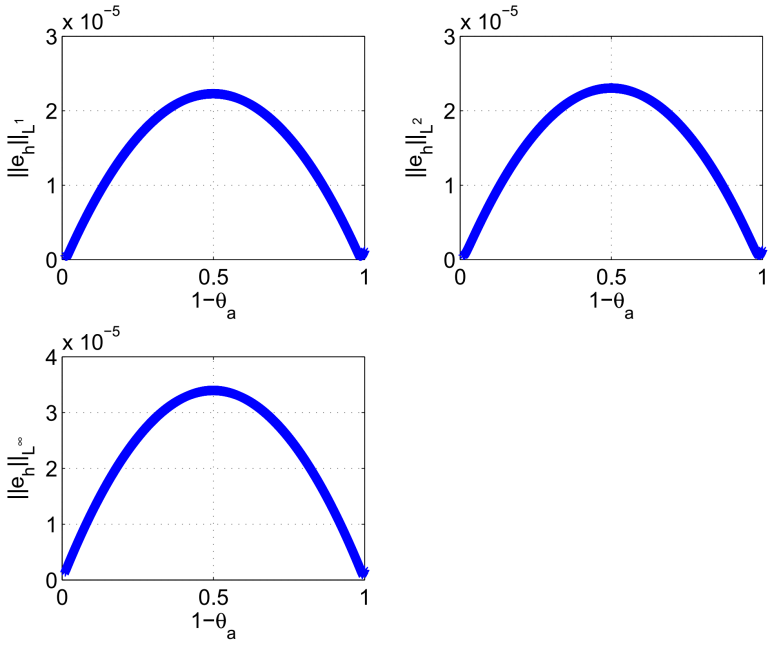


Figure 2.8: p - Norm of the error e_h as a function of the position of $\mathbf{a}$ ($n = 100$): $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left). The position of $\mathbf{a}$ varies in $]0, h[$ and thus the position of $\mathbf{b}$ varies in $]1 - h, 1[$

We can obtain a perfectly symmetrical figure with respect to that shown in Figure (2.8) if we consider the norm of e_h as a function of the position of $\mathbf{b}$.

In any case, $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$. We can read the values of the best-fit slope in the following table:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9797	-1.9735	-1.9715

These values were obtained varying the position of $\mathbf{a}$ in $]0, h[$ and thus the position of $\mathbf{b}$ in $]1 - h, 1[$.

Quadratic interpolation procedure

Also in the case in which we perform the quadratic interpolation procedures we obtain that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$:

$n_{\min} = 7,\ n_{\max} = 2980$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9590	-1.9533	-1.9594

These values were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]1-h, 1[$. The norm of e_h does not depend on the position of **a** and on the position of **b**.

Remark 2.3: It is interesting to underline the behaviors regarding the consistency error and the error as a function of the positions of **a** and of **b**. In general we can observe, both for the consistency error and for the error that the behavior of the L^p-norm as a function of the position of **a** is perfectly symmetric to the behavior as a function of the position of **b**. In the **First Example** the problem is perfectly symmetric, because $\mathbf{b} = \pi - \mathbf{a}$ and $f(x) = \sin(x)$ is symmetric around $\frac{\pi}{2}$. Thus the fact that we observe a symmetric behavior in **a** and **b** is obvious. In the **Second Example** the problem is not symmetric, because $f(x) = e^x$ is not symmetric, thus the fact that we observe a symmetric behavior in **a** and **b** is an interesting result.

Conclusions

We itemize the numerical results on the *Dirichlet problems* (1.13) and (1.14) in $[\mathbf{a}, \mathbf{b}] \subset [0, 1]$. We have chosen not to fix the position of **a** and the position of **b**, but to move **a** in $]0, h[$ and **b** in $]1-h, 1[$, so as to see if the results may or may not depend on their positions. We obtained results similar to the previous case, that is the case in which **b** is a grid point. We distinguish two cases: A_h not symmetric and A_h symmetric.

A_h not symmetric

In the case in which A_h is not symmetric, the numerical results are the following:

- $\|\mathrm{A}_h^{-1}\|_p \sim \mathrm{O}(n^{\frac{1}{p}}),\ p = 1, 2, \infty;$
- $\rho(\mathrm{A}_h^{-1}) \to 1;$
- $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty;$
- $\|e_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty.$

The norm of the consistency error and the norm of the error **can depend on the positions of a and of b** in the case in which we perform the linear interpolations to impose the boundary conditions in **a** and in **b**, which provide a second-order accuracy at the border as on the internal points.
In particular we have observed that L^1- and L^2-norms of the consistency error are constant as a function of **a** and of **b**, while L^∞-norm of the consistency error is not. In **Remark 2.1** we have provided an explanation for this behavior of the consistency error.

Instead, the norm of the consistency error and the norm of the error **do not depend on the position of a and on the position of b** in the case in which we perform the quadratic interpolations to impose the boundary conditions in **a** and in **b**, which provide a third-order accuracy at the border.

A_h symmetric

In the case in which the matrix A_h is symmetric the behaviors of the norms of the consistency error and of the inverse matrix of A_h depend on the symmetrization method used.

If we symmetrize the matrix A_h through the symmetrization method (1.34) the norm of τ_h does not depends on the position of **a** and on the position of **b** and

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h),$$

thus we haven't consistency problems. However, there is a threshold of negative values of m below which problems of consistency are starting to be verified. These values are around -1, -2, -3, depending on the test examined.
The spectral radius of A_h^{-1} and $\|\mathrm{A}_h^{-1}\|_p$ grow as a function of n for values of $m \geq 0$ and tend to a constant for values $m < 0$. Therefore, we have stability problems in the case in which $m \geq 0$.

Based on the above, is not convenient to symmetrize the matrix A_h.

If we symmetrize the matrix A_h through the formula (1.9) (thus only in the case in which we perform the linear interpolation procedures to impose the boundary conditions in **a** and in **b**) the norm of τ_h depends on the positions of **a** and of **b** and for this reason the order of accuracy itself depends on their positions. Performing various numerical tests we have observed that

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{1}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h^0),$$

therefore the consistency of the method is not guaranteed in L^∞-norm.
Only if $\theta_\mathrm{a} \to 0$ (or $\theta_\mathrm{b} \to 0$) , $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$. Furthemore, if $\theta_\mathrm{a} \to 1$ (or $\theta_\mathrm{b} \to 1$) we have consistency problems.
The spectral radius of A_h^{-1} and $\|\mathrm{A}_h^{-1}\|_p$ tend to a constant for large values of n.
Therefore, also in this case is not convenient to symmetrize the matrix A_h.

2.1.3 Mixed boundary conditions

Now we consider the same numerical tests for the mixed problem.
We impose Dirichlet boundary condition in **a** and Neumann boundary condition in **b**. We will consider the following cases:

- the case in which we perform the linear interpolation procedures to impose the boundary conditions both in **a** and in **b** (2D 2N);
- the case in which we perform the linear interpolation procedure to impose the boundary condition in **a** and the quadratic interpolation procedure to impose the boundary condition in **b** (2D 3N);
- the case in which we perform the quadratic interpolation procedures to impose the boundary conditions both in **a** and in **b** (3D 3N).

Behavior of the inverse matrix of $\mathbf{A}_h$ and of the spectral radius of $\mathbf{A}_h^{-1}$

The spectral radius of A_h^{-1} obviously does not depend on the position of **b** in the case in which we perform the linear interpolation procedure to impose the boundary condition in **b**. In all cases (2D 2N, 2D 3N, 3D 3N), the spectral radius of A_h^{-1} essentially does not depend on the position of **a** and on the position of **b**; it tends to a constant for large values of n. The following figure (Figure 2.9) shows the behavior of the spectral radius of A_h^{-1} choosing $\mathbf{a} = \frac{h}{2}$, $\mathbf{b} = \pi - \frac{h}{2}$ and the case 2D 2N. In the left panel we have chosen $n = 100$.

The behavior of the norm of the inverse matrix coincides with that of the Dirichlet Problems: $\|\mathrm{A}_h^{-1}\|_p \sim \mathrm{O}(n^{\frac{1}{p}})$, $p = 1, 2, \infty$. The following table shows the numerical results in the case 2D 2N, choosing $\mathbf{a} = \frac{h}{2}$ and $\mathbf{b} = \pi - \frac{h}{2}$:

$n_{\min} = 54,\ n_{\max} = 1096$	$sl(\|\mathrm{A}_h^{-1}\|_1)$	$sl(\|\mathrm{A}_h^{-1}\|_2)$	$sl(\|\mathrm{A}_h^{-1}\|_\infty)$
$\mathbf{a} = \frac{h}{2}$, $\mathbf{b} = \pi - \frac{h}{2}$	0.9999	0.4904	$0.6554\ 10^{-2}$

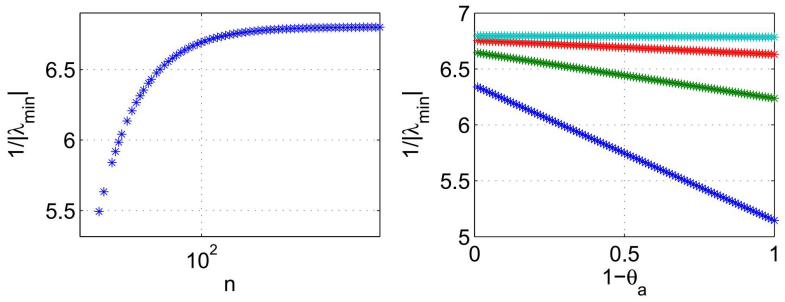


Figure 2.9: Dependence of the reciprocal of the eigenvalue of minimum module of the matrix A_h ($|\lambda_{\min}|$) on the number n of the grid points ($\theta_a = \theta_b = 0.5$) (left panel) and on the parameter $1 - \theta_a$ for different values of n: $n = 10$ (blue line), $n = 30$ (green line), $n = 100$ (red line), $n = 1000$ (cyan line) (right panel)

First example

We solve numerically the equation

$$-u_{xx} = \sin(x) \text{ in }]\mathbf{a}, \mathbf{b}[\subset [0, \pi],$$

in which we impose Dirichlet boundary condition in $\mathbf{a}$ and Neumann boundary condition in $\mathbf{b}$. The problem to be solved numerically is:

$$\begin{cases} -u_{xx} = \sin(x) & \text{in }]\mathbf{a}, \mathbf{b}[\subset [0, \pi] \\ u(\mathbf{a}) = \sin(\mathbf{a}) & \\ \frac{\partial u}{\partial \mathbf{n}}(\mathbf{b}) = \cos(\mathbf{b}) & \end{cases}.$$

We place $\mathbf{b} = \pi - \mathbf{a}$.

Behavior of the consistency error τ_h

2D 2N

The norm of the consistency error τ_h, in the case 2D 2N, depends only on the position of $\mathbf{b}$. Indeed, if we fix the position of $\mathbf{b}$ in $]\pi - h, \pi[$ the norm of the consistency error is constant as a function of the position of $\mathbf{a}$ in $]0, h[$. This does not happen if we fix the position of $\mathbf{a}$ and we observe its behavior as a function of the position of $\mathbf{b}$. In this latter case, the L^1- and L^2-norms are constant, the L^∞-norm is smaller if $\theta_b \sim 0.5$ and it is larger if $\theta_b \sim 0$ or $\theta_b \sim 1$ (see Figure 2.10).

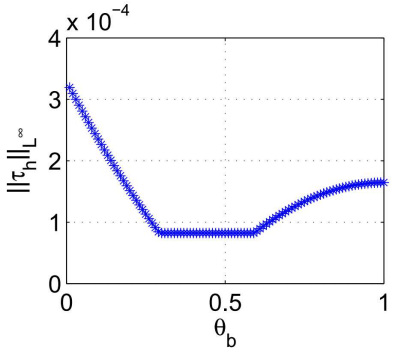


Figure 2.10: L^∞-norm of the consistency error τ_h as a function of the position of **b** ($n = 100$). The position of **a** varies in $]0, h[$ and $\mathbf{b} = \pi - \mathbf{a}$

Anyway $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$, as we can read in the table below:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0307	-2.0354	-1.9963

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]\pi - h, \pi[$.

2D 3N

Also in this case (2D 3N), the norm of the consistency error depends only on the position of **b**. In particular, the L^1- and L^2-norms are constant, the L^∞-norm is larger if $\theta_\mathrm{b} \sim 0$ or $\theta_\mathrm{b} \sim 1$ and it is smaller if $\theta_\mathrm{b} \sim 0.5$. The following figure (Figure 2.11) shows this behavior. In the figure, the position of **a** and the position of **b** vary in $]0, h[$ and in $]\pi - h, \pi[$ respectively.

Anyway $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$, as we can read in the table below:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0290	-2.0302	-1.9863

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]\pi - h, \pi[$.

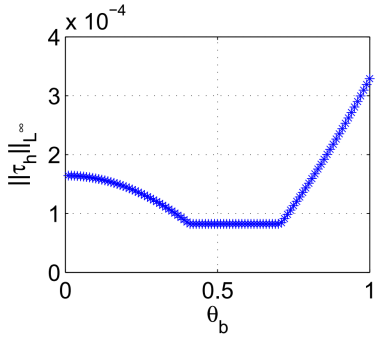


Figure 2.11: L^∞-norm of the consistency error τ_h as a function of the position of **b** ($n = 100$)

3D 3N

In this case (3D 3N), the behavior of the norm of τ_h coincides with that of the previous case (case 2D 3N). We can read the values of the best-fit slope in the table below, which shows that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0198	-2.0139	-1.9918

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]\pi - h, \pi[$.

Behavior of the error e_h

2D 2N

In the following table we show the numerical results about the norm of the error as a function of the number of the grid points in the case in which we impose boundary conditions both in **a** and in **b** performing linear interpolations. The numerical results show that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.0390	-2.0266	-2.0054

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]\pi - h, \pi[$.

The norm of the error depends only on the position of **b** in this example. Indeed, if we fix the position of **b** in $]\pi - h, \pi[$ the norm of the error is constant as a function of the position of **a** in $]0, h[$. This does not happen in the case we fix the position of **a** and we observe its behavior as a function of the position of **b**. In particular, it is larger if $\theta_b \sim 0$. This behavior can be highlighted in the following figure (Figure 2.12), which shows how varies the norm of the error as a function of the position of **b**. We have chosen $n = 100$. The position of **a** varies in $]0, h[$ and $\mathbf{b} = \pi - \mathbf{a}$.

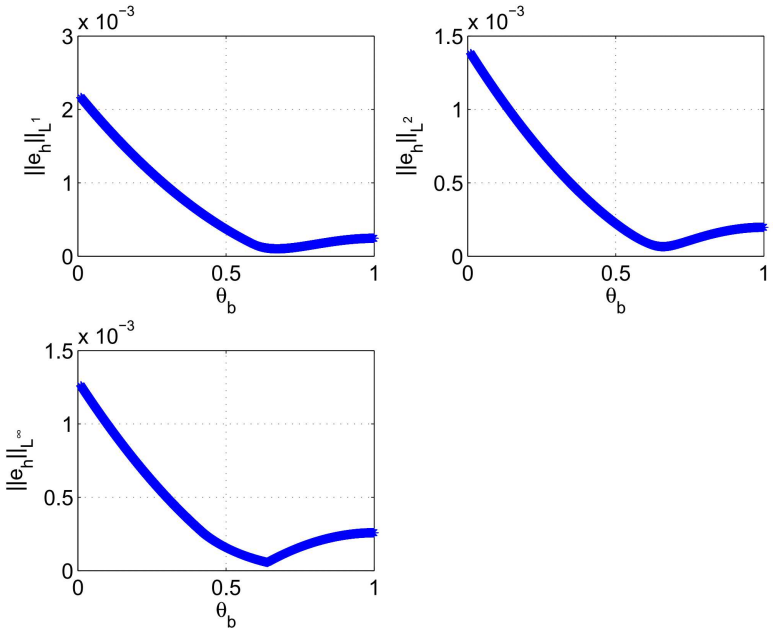


Figure 2.12: p - Norm of the error e_h as a function of the position of **b** ($n = 100$): $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

Remark 2.4 - on the behavior of the error: If we consider a mixed problem (case 2D 2N) we expect a loss of accuracy on the error with respect to the Dirichlet problem (**First example** in Subsection 2.1.2). This is because when we impose the Neumann condition on the extreme **b** by performing linear interpolation, from formula (1.16) we have $\mathcal{E} \sim \mathrm{O}(h)$, which is going to influence the global error. In fact, on the various numerical tests performed (just read the numerical results about the behavior of the error in the **Second example** of this Subsection), we obtain $\|e_h\|_{L^p} \sim \mathrm{O}(h)$ ($p = 1, 2, \infty$). We justify why in this case we obtain instead $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$.

If we consider the Taylor series of the function $u(x) = \sin(x)$ of center π we have:

$$u(\pi - h) = u(\pi) - u'(\pi)h + \frac{u''(\pi)}{2}h^2 - \frac{u'''(\pi)}{6}h^3 + \mathrm{O}(h^4),$$

from which

$$\frac{u(\pi) - u(\pi - h)}{h} = u'(\pi) - \frac{u''(\pi)}{2}h + \frac{u'''(\pi)}{6}h^2 + \mathrm{O}(h^3),$$

and being $u''(\pi) = 0$, we obtain $u'(\pi) \sim \mathrm{O}(h^2)$, as in the internal points. The same happens when choosing center 0.

For the same reason, a loss of accuracy in the consistency error does not occur, as instead happens in the **Second example** of this Subsection, in which:

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h).$$

The reason why this loss of accuracy affects the L^∞-norm most is in **Remark 2.1**.

2D 3N

Also in the case 2D 3N, the norm of the error depends only on the position of **b**. In particular, it is larger if $\theta_\mathrm{b} \sim 1$ (see Figure 2.13). Anyway, $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$, as we can read in the following table:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.0775	-2.0748	-2.0698

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]\pi - h, \pi[$.

3D 3N

In this latter case (3D 3N) the behavior of the norm of e_h coincides with that of the previous case (case 2D 3N). The following table shows that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.0354	-2.0206	-2.0037

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]\pi - h, \pi[$.

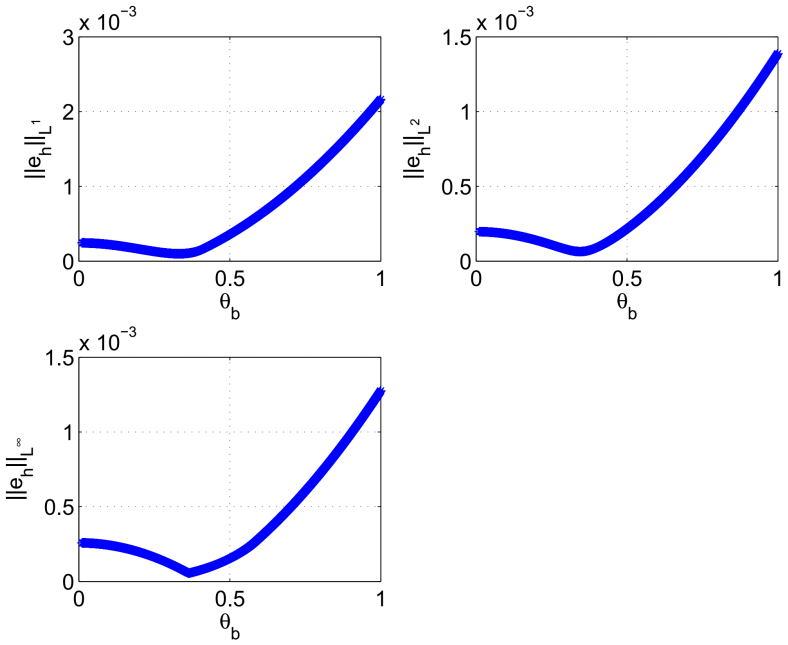


Figure 2.13: p - Norm of the error e_h as a function of the position of $\mathbf{b}$ ($n = 100$): $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left). The position of $\mathbf{a}$ varies in $]0, h[$ and $\mathbf{b} = \pi - \mathbf{a}$

Second example

We solve numerically the equation

$$-u_{xx} = e^x \text{ in }]\mathbf{a}, \mathbf{b}[\subset [0, 1],$$

in which we impose Dirichlet boundary condition in $\mathbf{a}$ and Neumann boundary condition in $\mathbf{b}$. The problem to be solved numerically is:

$$\begin{cases} -u_{xx} = e^x & \text{in }]\mathbf{a}, \mathbf{b}[\subset [0, 1] \\ u(\mathbf{a}) = -e^{\mathbf{a}} & \\ \frac{\partial u}{\partial \mathbf{n}}(\mathbf{b}) = -e^{\mathbf{b}} & \end{cases}.$$

We place $\mathbf{b} = 1 - \mathbf{a}$.

Behavior of the consistency error τ_h

2D 2N

In the case 2D 2N the numerical results show:

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2), \ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}), \ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h);$$

only in the case in which $\theta_b \sim 0.5$, $\|\tau_h\|_{L^p} \sim O(h^2)$, $p = 1, 2, \infty$:

$n_{min} = 7,\ n_{max} = 8103$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-1.9790	-1.5353	-1.0154

$n_{min} = 7,\ n_{max} = 8103$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
$\theta_b = 0.5$	-2.0021	-2.0025	-2.0059

In the table on the top the position of **a** and the position of **b** vary in $]0, h[$ and in $]1-h, 1[$ respectively, in the table on the bottom the position of **a** varies in $]0, h[$ and $\theta_b = 0.5$.

The norm of the consistency error τ_h, in this example, depends only on the position of **b**. Indeed, if we fix the position of **b** in $]1-h, 1[$ the norm of the consistency error is constant as a function of the position of **a** in $]0, h[$. This does not happen if we fix the position of **a** and we observe its behavior as a function of the position of **b**. It is larger if $\theta_b = 0$ or $\theta_b = 1$ and it is smaller if $\theta_b = 0.5$.

2D 3N

Also in the case 2D 3N, the norm of τ_h depends only on the position of **b**. In particular the L^1- and L^2-norms are constant, the L^∞-norm is larger if $\theta_b \sim 0$ or $\theta_b \sim 1$ and it is smaller if $\theta_b \sim 0.5$. This behavior is shown in the Figure 2.14.

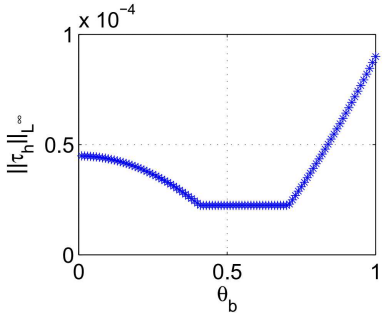


Figure 2.14: L^∞-norm of the consistency error τ_h as a function of the position of **b** ($n = 100$). The position of **a** varies in $]0, h[$ and $\mathbf{b} = 1 - \mathbf{a}$

The numerical results show that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 2980$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0188	2.0212	-1.9281

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]1 - h, 1[$.

3D 3N

In this latter case (3D 3N) the behavior of the norm of τ_h is equal to that of the previous case (case 2D 3N). The numerical results show that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 2980$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0116	-2.0152	-1.9372

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]1 - h, 1[$.

Behavior of the error e_h

Now, we show the behavior of the norm of the error as a function of the number of the grid points.

2D 2N

In the case 2D 2N the numerical results show:

$$\|e_h\|_{L^p} \sim \mathrm{O}(h),\ \ p = 1, 2, \infty;$$

only in the case in which $\theta_{\mathrm{b}} = 0.5$, $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.0505	-1.0483	-1.0412

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
$\theta_{\mathrm{b}} = 0.5$	-2.1123	-2.0941	-2.0632

In the table on the top the position of **a** and the position of **b** vary in $]0, h[$ and in $]1-h, 1[$ respectively, in the table on the bottom the position of **a** varies in $]0, h[$ and $\theta_{\mathrm{b}} = 0.5$.

The norm of e_h does not depend on the position of **a**. Indeed, if you fix the position of **b** in $]1-h, 1[$ the norm of the error is constant as a function of the position of **a** in $]0, h[$. Only if $\mathbf{b} = 1 - \frac{h}{2}$ the norm is smaller if $\theta_{\mathrm{a}} \sim 0$ or $\theta_{\mathrm{a}} \sim 1$ and it is larger if $\theta_{\mathrm{a}} = 0.5$. The norm of e_h depends on the position of **b**: it is larger if $\theta_{\mathrm{b}} = 0$ or $\theta_{\mathrm{b}} = 1$ and it is smaller if $\theta_{\mathrm{b}} = 0.5$.

2D 3N

In the case 2D 3N the numerical results show that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9648	-1.9538	-1.9305

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]1-h, 1[$.

The behavior of the norm of the error is as follow:

- as a function of the position of **a** in $]0, h[$, for each position of **b** (except the case in which $\theta_{\mathrm{b}} \sim 1$) the norm of the error is smaller if $\theta_{\mathrm{a}} \sim 0$ or $\theta_{\mathrm{a}} \sim 1$ and it is larger if $\theta_{\mathrm{a}} \sim 0.5$. In the case in which $\theta_{\mathrm{b}} \sim 1$ the norm of the error is constant as a function of the position of **a**;
- as a function of the position of **b** in $]1-h, 1[$, for each position of **a**, the norm of the error is larger if $\theta_{\mathrm{b}} \sim 1$. The following figure (Figure 2.15) shows the behavior of $\|e_h\|_{L^p}$ as a function of the position of **b** (the position of **a** varies in $]0, h[$).

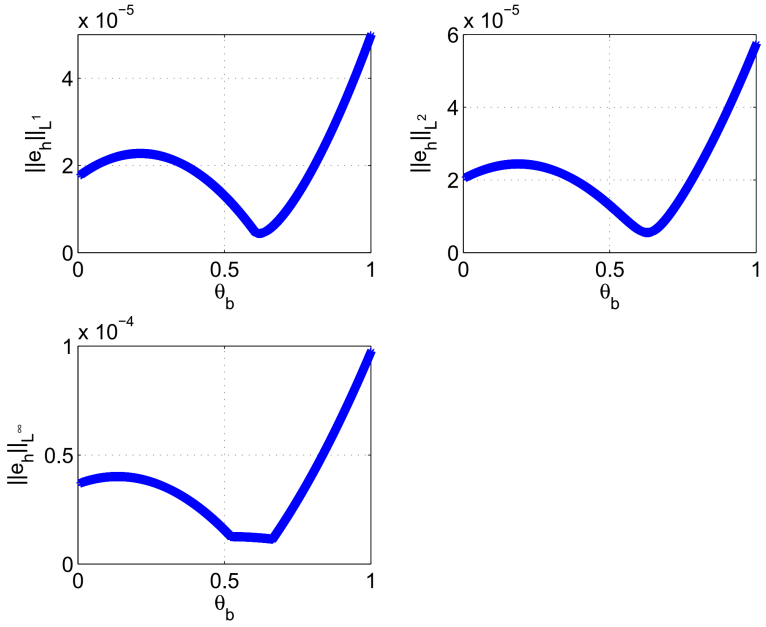


Figure 2.15: p - Norm of the error e_h as a function of the position of **b** ($n = 100$): $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

3D 3N

In the case 3D 3N the numerical results show that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$n_{\min} = 7,\ n_{\max} = 8103$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9390	-1.9353	-1.9193

These results were obtained varying the position of **a** in $]0, h[$ and thus the position of **b** in $]1 - h, 1[$.

The norm of the error depends only on the position of **b**. Indeed, if we fix the position of **b** in $]1 - h, 1[$ the norm of the error is constant as a function of the position of **a** in $]0, h[$. This does not happen if we fix the position of **a** and we observe its behavior as a function of the position of **b**. The results on the dependence of the norm from the position of **b** are equal to those of the previous case (case 2D 3N).

Remark 2.5 - on the behavior of the consistency error: The norm of the consistency error depends only on the position of **b**. Indeed, if we fix the position of **b** in $]1-h,1[$ the norm of the consistency error is constant as a function of the position of **a** in $]0,h[$. This does not happen if we fix the position of **a** and we observe its behavior as a function of the position of **b**. In particular, we can observe that, except for the 2D 2N case, the L^1- and L^2-norms of τ_h are constant as a function of the position of **b**, while the L^∞-norm is not. In the 2D 2N case not even the L^1- and L^2-norms are constant. The reason is similar to the explanation given in **Remark 2.1**.

Conclusions

We itemize the numerical results on the *mixed problems* in $[\mathbf{a},\mathbf{b}] \subset [0,1]$. We have chosen not to fix the position of **a** and the position of **b**, but to move **a** in $]0,h[$ and **b** in $]1-h,h[$, so as to see if the results may or may not depend on their positions. We distinguish two cases: A_h not symmetric and A_h symmetric.

A_h not symmetric

In the case in which A_h is not symmetric, the numerical results are the following:

- $\|\mathrm{A}_h^{-1}\|_p \sim \mathrm{O}(n^{\frac{1}{p}}),\ p = 1, 2, \infty$;
- $\rho(\mathrm{A}_h^{-1}) \to \mathrm{c}$, where c is a constant.

The accuracy orders of $\|\tau_h\|_{L^p}$ and of $\|e_h\|_{L^p}$ depend on the procedure with which we impose the boundary condition in **b**.
If we perform the linear interpolation procedure (case 2D 2N) we have:

- $\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h)$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h),\ p = 1, 2, \infty$;

only if $\theta_\mathrm{b} = 0.5$ we have a second-order accuracy for both the error and the consistency error. The loss of accuracy with respect to the Dirichlet problem is due to the fact that the linear interpolation provides a first-order accuracy in **b** in the case in which we impose Neumann condition.
If we perform the quadratic interpolation procedure (cases 2D 3N and 3D 3N), which provides a second-order accuracy on the border in the case in which we impose Neumann condition, we have:

- $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty;$
- $\|e_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty.$

The norm of the consistency error **can depend on the position of b** and the norm of the error **can depend on the position of a and on the position of b** in the case in which we perform the linear interpolation to impose the Dirichlet boundary condition in **a** (case 2D 2N and 2D 3N). They **depend only the position of b** if we perform the quadratic interpolation to impose the boundary condition in **a** (case 3D 3N). If we want that there is not even dependence from **b** we need to perform a cubic interpolation procedure to impose the Neumann condition in **b**, which provides a third-order accuracy on the border.

A_h symmetric

In the case in which the matrix A_h is symmetric the behaviors of the norms of the consistency error and of the inverse matrix of A_h depend on the symmetrization method used.

If we symmetrize the matrix A_h through the symmetrization method (1.34) the norm of τ_h does not depends on the position of **a** and on the position of **b** and

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h),$$

thus we haven't consistency problems. However, for $m \leq 0$ problems of consistency are starting to be verified.
The spectral radius of A_h^{-1} and $\|\mathrm{A}_h^{-1}\|_p$ grow as a function of n for all values of m. Based on the above, is not convenient to symmetrize the matrix A_h.

If we symmetrize the matrix A_h through the formulas (1.9) and (1.18) (thus only in the case 2D 2N) the norm of τ_h depends on the positions of **a** and of **b**. Performing various numerical tests we have observed that

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{1}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h^0),$$

therefore the consistency of the method is not guaranteed in L^∞-norm.
Furthemore, if $\theta_\mathrm{a} \to 1$ we have consistency problems.
The spectral radius of A_h^{-1} and $\|\mathrm{A}_h^{-1}\|_p$ tend to a constant for large values of n. Therefore, also in this case is not convenient to symmetrize the matrix A_h.

2.2 Numerical tests in 2D

We continue showing the numerical tests in two-dimensional case. We present two several cases: the case in which the domain is a square and the case in which the domain is a circumference.

2.2.1 Dirichlet boundary conditions - Square domain

We start to show the numerical results of the Dirichlet problem both in the case in which we perform the linear interpolations and in the case in which we perform the quadratic interpolations, choosing a square domain. We have chosen a square because it represents a simple extension of the one-dimensional case.

Behavior of the inverse matrix of $\mathbf{A}_h$

We show the behavior of the norm of the inverse matrix of A_h as a function of N. The norm of A_h^{-1} depends on the position of the vertices A, B, C and D (see Figure 1.9) of the square respect to the grid. The square domain within the uniform grid, it is indeed circumscribed to a square whose vertices are internal points and it is inscribed in a square whose vertices are ghost points.

Linear interpolation procedures

The numerical results in the case in which we perform the linear interpolations are listed in the following table:

$N_{\min} = 40,\ N_{\max} = 109$	$sl(\lVert \mathrm{A}_h^{-1} \rVert_1)$	$sl(\lVert \mathrm{A}_h^{-1} \rVert_2)$	$sl(\lVert \mathrm{A}_h^{-1} \rVert_\infty)$
$\theta_S = \theta_N = \theta_E = \theta_W = 0.5$	0.9292	0.4811	$-0.1641\ 10^{-3}$
$\theta_S = \theta_N = \theta_E = \theta_W = 0.9990$	0.9773	0.4864	$0.3109\ 10^{-4}$
$\theta_S = \theta_N = \theta_W = 0.9990,\ \theta_E = 0.001$	0.7878	-0.1400	$-0.2009\ 10^{-3}$
$\theta_S = \theta_N = \theta_E = 0.9990,\ \theta_W = 0.001$	0.7878	-0.1400	$-0.2009\ 10^{-3}$
$\theta_N = \theta_W = \theta_E = 0.9990,\ \theta_S = 0.001$	0.7878	-0.1400	$-0.2009\ 10^{-3}$
$\theta_S = \theta_W = \theta_E = 0.9990,\ \theta_N = 0.001$	0.7878	-0.1400	$-0.2009\ 10^{-3}$
$\theta_S = \theta_N = 0.9990,\ \theta_W = \theta_E = 0.001$	0.7725	-0.1128	$-0.2482\ 10^{-3}$
$\theta_S = \theta_N = 0.001,\ \theta_W = \theta_E = 0.9990$	0.7725	-0.1128	$-0.2482\ 10^{-3}$
$\theta_S = \theta_N = \theta_W = \theta_E = 0.001$	$0.6563\ 10^{-6}$	$0.1032\ 10^{-11}$	$-0.2585\ 10^{-6}$
$\theta_S = \theta_N = \theta_W = 0.001,\ \theta_E = 0.9990$	$0.6518\ 10^{-6}$	$0.1032\ 10^{-11}$	$-0.2585\ 10^{-6}$
$\theta_S = \theta_N = \theta_E = 0.001,\ \theta_W = 0.9990$	$0.6518\ 10^{-6}$	$0.1032\ 10^{-11}$	$-0.2585\ 10^{-6}$
$\theta_N = \theta_W = \theta_E = 0.001,\ \theta_S = 0.9990$	$0.6518\ 10^{-6}$	$0.1032\ 10^{-11}$	$-0.2585\ 10^{-6}$
$\theta_S = \theta_W = \theta_E = 0.001,\ \theta_N = 0.9990$	$0.6518\ 10^{-6}$	$0.1032\ 10^{-11}$	$-0.2585\ 10^{-6}$

Quadratic interpolation procedures

The numerical results in the case in which we perform the quadratic interpolations are listed in the following table:

$N_{\min} = 40,\ N_{\max} = 109$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = \theta_{\mathrm{W}} = 0.5$	0.9159	0.4805	$-0.1359\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = \theta_{\mathrm{W}} = 0.9990$	0.9773	0.4863	$0.3131\ 10^{-4}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = 0.9990,\ \theta_{\mathrm{E}} = 0.001$	0.7694	-0.1179	$-0.2486\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = 0.9990,\ \theta_{\mathrm{W}} = 0.001$	0.7694	-0.1179	$-0.2486\ 10^{-3}$
$\theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990,\ \theta_{\mathrm{S}} = 0.001$	0.7694	-0.1179	$-0.2486\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990,\ \theta_{\mathrm{N}} = 0.001$	0.7694	-0.1179	$-0.2486\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = 0.9990,\ \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.001$	0.7722	-0.1153	$-0.2478\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = 0.001,\ \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990$	0.7722	-0.1153	$-0.2478\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = \theta_{\mathrm{W}} = 0.001$	$0.1645\ 10^{-6}$	$0.1032\ 10^{-11}$	$-0.1295\ 10^{-6}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = 0.001,\ \theta_{\mathrm{E}} = 0.9990$	$0.1634\ 10^{-6}$	$0.729\ 10^{-12}$	$-0.1295\ 10^{-6}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = 0.001,\ \theta_{\mathrm{W}} = 0.9990$	$0.1634\ 10^{-6}$	$0.1179\ 10^{-11}$	$-0.1295\ 10^{-6}$
$\theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.001,\ \theta_{\mathrm{S}} = 0.9990$	$0.1634\ 10^{-6}$	$0.1188\ 10^{-11}$	$-0.1295\ 10^{-6}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.001,\ \theta_{\mathrm{N}} = 0.9990$	$0.1634\ 10^{-6}$	$0.742\ 10^{-12}$	$-0.1295\ 10^{-6}$

It is possible to notice that when all the sides of the domain are close to the ghost points the behavior coincides with that of the 1D case, that is, $\|\mathrm{A}_h^{-1}\|_p \sim \mathrm{O}(N^{\frac{1}{p}})$, $p = 1, 2, \infty$. When the sides are close to the internal points, we have:

- if one side or two sides are close to the internal points:

$$\|\mathrm{A}_h^{-1}\|_1 \sim \mathrm{O}(N),\ \|\mathrm{A}_h^{-1}\|_2 \sim \mathrm{O}(N^0),\ \|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0);$$

- if three or all sides are close to the internal points:

$$\|\mathrm{A}_h^{-1}\|_p \sim \mathrm{O}(N^0),\ p = 1, 2, \infty.$$

Thus, the stability is guaranteed in the ∞-norm whatever the position of the vertices of the domain with respect to the grid.

Behavior of the spectral radius of A_h^{-1}

Now, we show the behavior of the spectral radius of the matrix A_h^{-1}, in the case in which we perform the linear interpolations to impose the Dirichlet conditions:

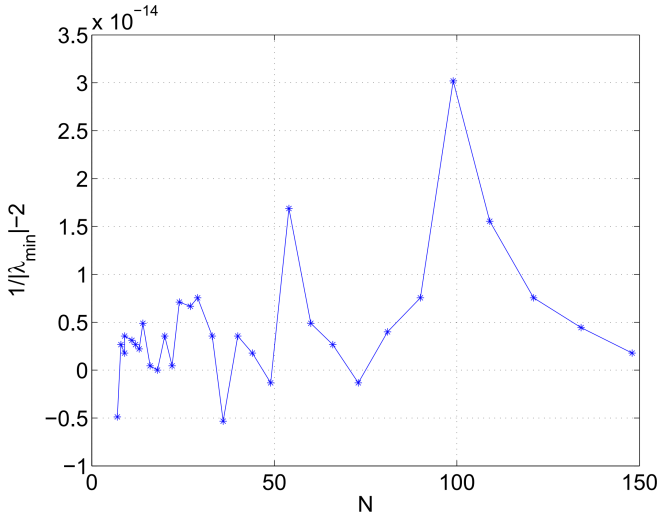


Figure 2.16: Dependence of the reciprocal of the eigenvalue of minimum module of the matrix A_h ($|\lambda_{\min}|$) on the number N of the grid points

These results were obtained choosing $x_{\mathrm{A}} = y_{\mathrm{A}} = \frac{h}{2}$ and $x_{\mathrm{B}} = y_{\mathrm{D}} = 1 - \frac{h}{2}$. The figure shows that the spectral radius of A_h^{-1} is constant for large values of N. This constant depends on the position of the vertices A, B, C and D of the square. In this case is 2.

We have the same results ($\rho(\mathrm{A}_h^{-1}) \to \mathrm{c}$) in the case in which we perform the quadratic interpolation procedures to impose the boundary conditions.

Behavior of the Schur complement of $\mathbf{A}_h$

We also show the behavior of the norm of the inverse of the Schur complement of A_h:

$$\mathrm{S}_h = \mathrm{A}_{\mathrm{gg}} - \mathrm{A}_{\mathrm{gi}}\mathrm{A}_{\mathrm{ii}}^{-1}\mathrm{A}_{\mathrm{ig}}.$$

The study of Schur complement is fundamental for evaluate the stability of the numerical method. We explain why.

We write the linear system in this way:

$$\mathrm{A}_h\mathrm{V} = \begin{pmatrix} \mathrm{A_{gg}} & \mathrm{A_{gi}} \\ \mathrm{A_{ig}} & \mathrm{A_{ii}} \end{pmatrix} \begin{pmatrix} \mathrm{V_g} \\ \mathrm{V_i} \end{pmatrix} = \begin{pmatrix} z_\mathrm{g} \\ z_\mathrm{i} \end{pmatrix},$$

where $\mathrm{V_g}$ and $\mathrm{V_i}$ are the vectors of the ghost points and of the internal points respectively. Being:

$$\mathrm{A_{ii}V_i} + \mathrm{A_{ig}V_g} = z_\mathrm{i} \Longrightarrow \mathrm{V_i} = \mathrm{A_{ii}^{-1}}(z_\mathrm{i} - \mathrm{A_{ig}V_g}), \tag{2.5}$$

$$\mathrm{A_{gi}V_i} + \mathrm{A_{gg}V_g} = z_\mathrm{g}, \tag{2.6}$$

replacing (2.5) in (2.6) we have

$$z_\mathrm{g} = (\mathrm{A_{gg}} - \mathrm{A_{gi}A_{ii}^{-1}A_{ig}})\mathrm{V_g} + \mathrm{A_{gi}A_{ii}^{-1}}z_\mathrm{i}.$$

The quantity $\mathrm{A_{gi}A_{ii}^{-1}}z_\mathrm{i}$ is negligible, therefore we focus only on quantity $\mathrm{S}_h = \mathrm{A_{gg}} - \mathrm{A_{gi}A_{ii}^{-1}A_{ig}}$ which is precisely the Schur complement of A_h. We have also verified that in the case of the square domain the Schur complement is a symmetric matrix. To get information on stability, we deal with the study of the norm of the inverse matrix of S_h.

We can read the values of the best-fit slope in the table below in the case in which we perform the linear interpolation procedures, choosing $x_\mathrm{A} = y_\mathrm{A} = \frac{h}{2}$ and $x_\mathrm{B} = y_\mathrm{D} = 1 - \frac{h}{2}$, and it shows that the norm and the spectral radius of S_h^{-1} are constant for large values of N:

$N_{\min} = 33,\ N_{\max} = 109$	$sl(\lVert \mathrm{S}_h^{-1} \rVert_1)$	$sl(\lVert \mathrm{S}_h^{-1} \rVert_2)$	$sl(\lVert \mathrm{S}_h^{-1} \rVert_\infty)$
	$-0.5329\ 10^{-7}$	$0.2937\ 10^{-3}$	$-0.5329\ 10^{-7}$

We obtain the same accuracy in the case in which we perform the quadratic interpolation procedures to impose the boundary conditions:

$N_{\min} = 33,\ N_{\max} = 109$	$sl(\lVert \mathrm{S}_h^{-1} \rVert_1)$	$sl(\lVert \mathrm{S}_h^{-1} \rVert_2)$	$sl(\lVert \mathrm{S}_h^{-1} \rVert_\infty)$
	$0.1506\ 10^{-6}$	$-0.5049\ 10^{-8}$	$-0.3079\ 10^{-7}$

Now, we proceed showing some numerical results about the behaviors of the consistency error and of the error.

We choose some functions f for our numerical tests:

- $-u_{xx} - u_{yy} = \frac{4x+x^3+\mathrm{xy}^2}{(x^2+y^2+1)^2\sqrt{x^2+y^2+1}} \Longrightarrow u = \frac{x}{\sqrt{x^2+y^2+1}}$
- $-u_{xx} - u_{yy} = -2e^{x^2+y^2}[2 + 2(x^2 + y^2)] \Longrightarrow u = e^{x^2+y^2}$
- $-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi y)\cos(2\pi x) \Longrightarrow u = 2 + \sin(2\pi y)\cos(2\pi x)$

We have approximated $\|\tau_h\|_{L^p}$ and $\|e_h\|_{L^p}$ $(p = 1, 2, \infty)$ in this way:

$$\|a_h\|_{L^1} = h^2 \sum_{i,j \,\in\, \Omega_h^{\mathrm{I}} \cup \Omega_h^{\mathrm{G}}} |a_h(x_i, y_j)|,$$

$$\|a_h\|_{L^2} = \sqrt{h^2 \sum_{i,j \,\in\, \Omega_h^{\mathrm{I}} \cup \Omega_h^{\mathrm{G}}} |a_h(x_i, y_j)|^2},$$

$$\|a_h\|_{L^\infty} = \max_{i,j \,\in\, \Omega_h^{\mathrm{I}} \cup \Omega_h^{\mathrm{G}}} |a_h(x_i, y_j)|,$$

where a_h is τ_h or e_h.

Behavior of the consistency error τ_h

Through our numerical tests we want showing that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$. We show below the numerical results obtained choosing $x_{\mathrm{A}} = y_{\mathrm{A}} = \frac{h}{2}$ and $x_{\mathrm{B}} = y_{\mathrm{D}} = 1 - \frac{h}{2}$.

In each table related to a certain function, in the first row we inserted the results obtained performing the linear interpolations, in the second row we inserted the results obtained performing the quadratic interpolations.

First example

$$-u_{xx} - u_{yy} = \frac{4x + x^3 + xy^2}{(x^2 + y^2 + 1)^2\sqrt{x^2 + y^2 + 1}}$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
Linear interpolation procedures	-1.9825	-1.9839	-1.9937
Quadratic interpolation procedures	-1.9738	-1.9834	-1.9937

The following figure (Figure 2.17) shows the behavior of the consistency error as a function of the number of the grid points, in the case in which we perform the linear interpolations. Moreover, we have verified that in this example the norm of the consistency error does not depend on the position of the vertices of the square, as well as in the case we perform the quadratic interpolations.

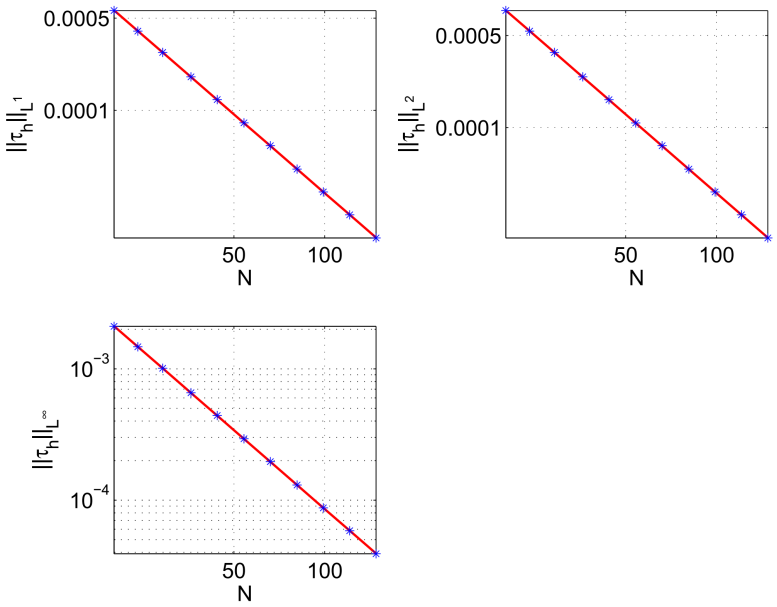


Figure 2.17: p - Norm of the consistency error τ_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

Second example

$$-u_{xx} - u_{yy} = -2e^{x^2+y^2}[2 + 2(x^2 + y^2)]$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
LINEAR INTERPOLATION PROCEDURES	-1.9417	-1.9414	-1.8756
QUADRATIC INTERPOLATION PROCEDURES	-1.9318	-1.9408	-1.8756

Third example

$$-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi y)\cos(2\pi x)$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
LINEAR INTERPOLATION PROCEDURES	-1.9596	-1.9767	-1.9979
QUADRATIC INTERPOLATION PROCEDURES	-1.9583	-1.9767	-1.9979

We have verified that in all examples the norm of the consistency error does not depend on the position of the vertices of the square both in the case we perform the linear interpolations and in the case we perform the quadratic interpolations.

Behavior of the error e_h

Now we show that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$. We show below the results obtained choosing $x_{\mathrm{A}} = y_{\mathrm{A}} = \frac{h}{2}$ and $x_{\mathrm{B}} = y_{\mathrm{D}} = 1 - \frac{h}{2}$.

First example

$$-u_{xx} - u_{yy} = \frac{4x + x^3 + xy^2}{(x^2 + y^2 + 1)^2\sqrt{x^2 + y^2 + 1}}$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
Linear interpolation procedures	-2.0428	-2.0265	-2.0154
Quadratic interpolation procedures	-1.8849	-1.8840	-1.9095

Second example

$$-u_{xx} - u_{yy} = -2e^{x^2+y^2}[2 + 2(x^2 + y^2)]$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
Linear interpolation procedures	-2.0358	-2.0265	-1.9469
Quadratic interpolation procedures	-1.8420	-1.8301	-1.8360

Third example

$$-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi y)\cos(2\pi x)$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
Linear interpolation procedures	-2.0604	-2.0502	-2.0217
Quadratic interpolation procedures	-1.9671	-1.9546	-1.9590

We have verified that in all examples the norm of the error depends on the position of the vertices of the square in the case we perform the linear interpolations. In the case we perform the quadratic interpolations the dependence on the position of the vertices of the domain is less evident. The following figure (Figure 2.18) shows the behavior of the norm of the error as a function

of the number of the grid points related to the **First example**, in the case in which we perform the linear interpolation procedures to impose the Dirichlet conditions.

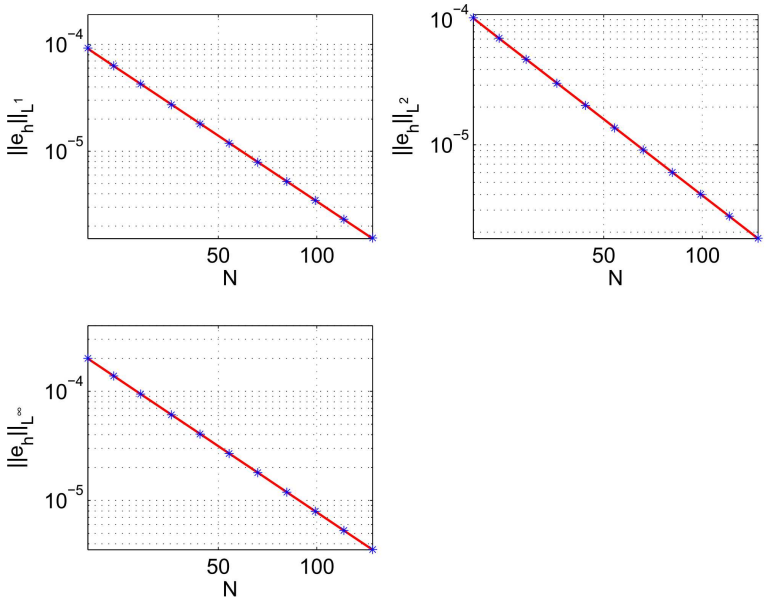


Figure 2.18: p - Norm of the error e_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

We conclude this Section showing a figure relative to the **Third example** in which the numerical solution overlaps the exact solution. We have chosen $N = 50$, the linear interpolations to impose the Dirichlet boundary conditions and $\theta_{\text{K}} = 0.5$ (K = S, N, W, E).

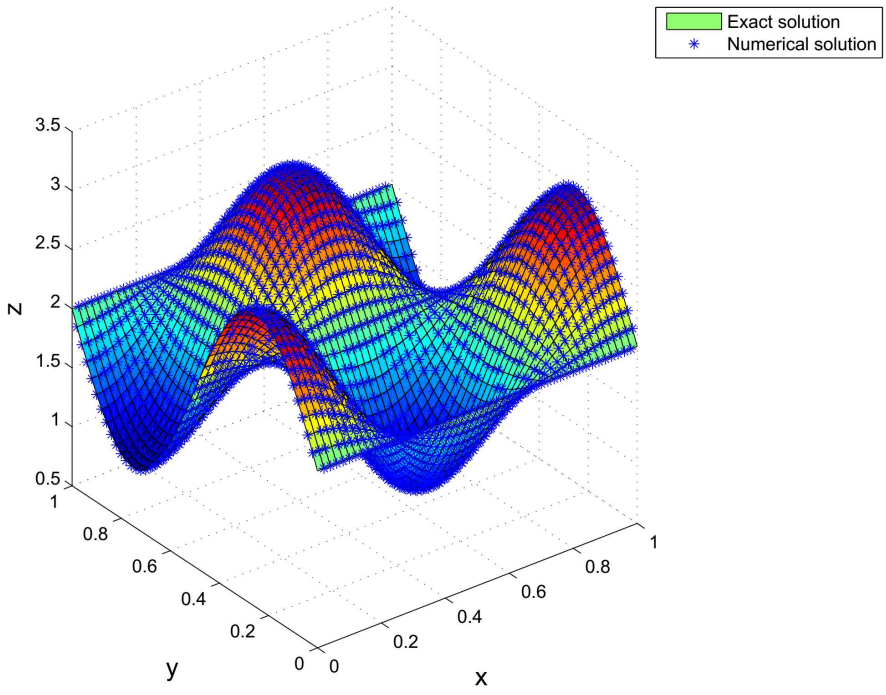


Figure 2.19: Solution for the **Third example** with Dirichlet boundary conditions

2.2.2 Mixed boundary conditions - Square domain

We proceed to show the numerical results of the mixed problem both in the case in which we perform the linear interpolations and in the case in which we perform the quadratic interpolations. In particular, we impose the Dirichlet boundary conditions in the points of the south (S) and west (W) zones and we impose the Neumann boundary conditions in the points of the nord (N) and est (E) zones.

Behavior of the inverse matrix of $\mathbf{A}_h$

We show the behavior of the norm of the inverse matrix of A_h as a function of N. The norm of A_h^{-1} depends on the position of the vertices A, B, C and D (see Figure 1.9) of the square respect to the grid.

Linear interpolation procedures

The numerical results in the case in which we perform the linear interpolations are listed in the following table:

$N_{\min} = 33,\ N_{\max} = 109$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = \theta_{\mathrm{W}} = 0.5$	0.9082	0.4762	$0.1287\ 10^{-1}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = \theta_{\mathrm{W}} = 0.9990$	0.9334	0.4476	$0.6506\ 10^{-2}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = 0.9990,\ \theta_{\mathrm{E}} = 0.001$	0.9334	0.4476	$0.6506\ 10^{-2}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = 0.9990,\ \theta_{\mathrm{W}} = 0.001$	0.8290	0.3727	$-0.5283\ 10^{-3}$
$\theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990,\ \theta_{\mathrm{S}} = 0.001$	0.8290	0.3727	$-0.5283\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990,\ \theta_{\mathrm{N}} = 0.001$	0.9334	0.4476	$0.6506\ 10^{-2}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = 0.9990,\ \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.001$	0.8290	0.3727	$-0.5283\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = 0.001,\ \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990$	0.8290	0.3727	$-0.5283\ 10^{-3}$
$\theta_{\mathrm{S}}, \theta_{\mathrm{N}}, \theta_{\mathrm{W}}, \theta_{\mathrm{E}} \sim 0$	$0.6229\ 10^{-8}$	$0.581\ 10^{-12}$	$0.3284\ 10^{-7}$
$\theta_{\mathrm{S}}, \theta_{\mathrm{N}}, \theta_{\mathrm{W}} \sim 0,\ \theta_{\mathrm{E}} \sim 1$	$0.6229\ 10^{-8}$	$0.581\ 10^{-12}$	$0.3284\ 10^{-7}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = 0.001,\ \theta_{\mathrm{W}} = 0.9990$	0.8290	0.3727	$-0.5283\ 10^{-3}$
$\theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.001,\ \theta_{\mathrm{S}} = 0.9990$	0.8290	0.3727	$-0.5283\ 10^{-3}$
$\theta_{\mathrm{S}}, \theta_{\mathrm{W}}, \theta_{\mathrm{E}} \sim 0,\ \theta_{\mathrm{N}} \sim 1$	$0.6152\ 10^{-8}$	$0.581\ 10^{-12}$	$0.7692\ 10^{-7}$

Quadratic interpolation procedures

The numerical results in the case in which we perform the quadratic interpolations are listed in the following table:

$N_{\min} = 33,\ N_{\max} = 109$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = \theta_{\mathrm{W}} = 0.5$	0.9031	0.4761	$-0.2875\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = \theta_{\mathrm{W}} = 0.9990$	0.9631	0.4699	$0.6589\ 10^{-2}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = 0.9990,\ \theta_{\mathrm{E}} = 0.001$	0.9552	0.4774	$0.6622\ 10^{-2}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = 0.9990,\ \theta_{\mathrm{W}} = 0.001$	0.8950	0.3691	$-0.5261\ 10^{-3}$
$\theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990,\ \theta_{\mathrm{S}} = 0.001$	0.8950	0.3691	$-0.5261\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990,\ \theta_{\mathrm{N}} = 0.001$	0.9552	0.4714	$0.6622\ 10^{-2}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = 0.9990,\ \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.001$	0.8881	0.3683	$-0.5275\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = 0.001,\ \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990$	0.8881	0.3683	$-0.5275\ 10^{-3}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.001$	$0.1531\ 10^{-6}$	$0.35\ 10^{-13}$	$-0.1655\ 10^{-6}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = 0.001,\ \theta_{\mathrm{E}} = 0.9990$	$0.1559\ 10^{-6}$	$0.35\ 10^{-13}$	$-0.1655\ 10^{-6}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{E}} = 0.001,\ \theta_{\mathrm{W}} = 0.9990$	0.7783	0.3517	$-0.4094\ 10^{-2}$
$\theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.001,\ \theta_{\mathrm{S}} = 0.9990$	0.7783	0.3517	$-0.4094\ 10^{-2}$
$\theta_{\mathrm{S}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.001,\ \theta_{\mathrm{N}} = 0.9990$	$0.1559\ 10^{-6}$	$0.35\ 10^{-13}$	$-0.1655\ 10^{-6}$

Based on the position of the vertices of the domain relative to the grid, the numerical results show that the following cases can occur:

- $\|\mathrm{A}_h^{-1}\|_1 \sim \mathrm{O}(N)$, $\|\mathrm{A}_h^{-1}\|_2 \sim \mathrm{O}(N^{\frac{1}{2}})$, $\|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$
- $\|\mathrm{A}_h^{-1}\|_1 \sim \mathrm{O}(N^0)$, $\|\mathrm{A}_h^{-1}\|_2 \sim \mathrm{O}(N^0)$, $\|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$.

As in the Dirichlet problem, stability is guaranteed in the ∞-norm whatever the position of the vertices of the domain with respect to the grid.

The behavior of the spectral radius of the matrix A_h^{-1} is identical to that of the Dirichlet problem, both in the case in which we perform the linear interpolations and in the case in which we perform the quadratic interpolations, that is, it tends to a constant for large values of N.

Behavior of the Schur complement of $\mathbf{A}_h$

The norm of the inverse of the Schur complement of A_h and its spectral radius tend to a constant for large values of N:

$N_{\min} = 33,\ N_{\max} = 109$	$sl(\|\mathrm{S}_h^{-1}\|_1)$	$sl(\|\mathrm{S}_h^{-1}\|_2)$	$sl(\|\mathrm{S}_h^{-1}\|_\infty)$
Linear interpolation procedures	0.1202	$0.3093\ 10^{-2}$	$0.1009\ 10^{-1}$
Quadratic interpolation procedures	0.1150	$0.1951\ 10^{-7}$	$-0.9570\ 10^{-4}$

These results were obtained choosing $\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.5$, but we have the same accuracy also for different values of the positions of the vertices of the square with respect to the grid.

Behavior of the consistency error τ_h

We show the behavior of the norm of the consistency error as a function of the number of the grid points, starting with the case in which we perform the linear interpolations.

LINEAR INTERPOLATION PROCEDURES

First example

$$-u_{xx} - u_{yy} = \frac{4x + x^3 + xy^2}{(x^2 + y^2 + 1)^2\sqrt{x^2 + y^2 + 1}}$$

The behavior of the consistency error in the case in which we perform the linear interpolations coincides with that of the one-dimensional case:

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h).$$

Only in the case in which at least $\theta_{\mathrm{N}} = \theta_{\mathrm{E}} = 0.5$ we obtain $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$. We can read the values of the best-fit slope in the tables below:

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-1.9795	-1.9836	-1.9937

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-1.9801	-1.5147	-1.0101

In the table on the top we have chosen $\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.5$ and in the table on the bottom we have chosen $\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990$.

Second example

$$-u_{xx} - u_{yy} = -2e^{x^2+y^2}[2 + 2(x^2 + y^2)]$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.5$	-1.9426	-1.9416	-1.8756
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990$	-1.9436	-1.4683	-0.9369

Third example

$$-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi y)\cos(2\pi x)$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
$\theta_S = \theta_N = \theta_W = \theta_E = 0.5$	-1.9603	-1.9767	-1.9979
$\theta_S = \theta_N = \theta_W = \theta_E = 0.9990$	-1.9665	-1.7848	-0.9964

In this last case there is a second-order accuracy also in the L^2-norm and it is enough that there is $\theta_E = 0.5$ for it to occur the second-order accuracy:

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
$\theta_E = 0.5,\ \theta_S = \theta_N = \theta_W = 0.9990$	-1.9634	-1.9772	-1.9979

We have verified that in all examples the norm of the consistency error depends on the position of the vertices of the square.

Quadratic interpolation procedures

In the case in which we perform the quadratic interpolations we show the numerical results only in the case in which $\theta_S = \theta_N = \theta_W = \theta_E = 0.5$, because $\|\tau_h\|_{L^p} \sim O(h^2)$ $(p = 1, 2, \infty)$ whatever is the position of the vertices of the domain with respect to the grid. Furthermore, $\|\tau_h\|_{L^p}$ does not depend on the position of the vertices of the square.

First example

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-1.9763	-1.9834	-1.9937

Second example

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-1.9414	-1.9416	-1.8756

Third example

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-1.9597	-1.9767	-1.9979

Behavior of the error e_h

We now examine the behavior of the norm of the error as a function of N, starting with the case in which we perform the linear interpolations.

LINEAR INTERPOLATION PROCEDURES

First example

$$-u_{xx} - u_{yy} = \frac{4x + x^3 + xy^2}{(x^2 + y^2 + 1)^2\sqrt{x^2 + y^2 + 1}}$$

The behavior of the error in the case in which we perform the linear interpolations coincides with that of the one-dimensional case:

$$\|e_h\|_{L^p} \sim \mathrm{O}(h), \ p = 1, 2, \infty.$$

Only in the case in which at least $\theta_{\mathrm{N}} = \theta_{\mathrm{E}} = 0.5$ we obtain $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$. We can read the values of the best-fit slope in the tables below:

$N_{\min} = 20, \ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.0348	-2.0165	-1.9969

$N_{\min} = 20, \ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.0483	-1.0404	-1.0126

In the table on the top we have chosen $\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.5$ and in the table on the bottom we have chosen $\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990$.

Second example

$$-u_{xx} - u_{yy} = -2e^{x^2+y^2}[2 + 2(x^2 + y^2)]$$

$N_{\min} = 20, \ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.5$	-1.9270	-1.9002	-1.8915
$\theta_{\mathrm{S}} = \theta_{\mathrm{N}} = \theta_{\mathrm{W}} = \theta_{\mathrm{E}} = 0.9990$	-1.0228	-1.0107	-0.9592

Third example

$$-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi y)\cos(2\pi x)$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
$\theta_S = \theta_N = \theta_W = \theta_E = 0.5$	-2.0505	-2.0359	-2.0215
$\theta_S = \theta_N = \theta_W = \theta_E = 0.9990$	-1.0712	-1.0664	-1.0322

In this last example it is enough that there is $\theta_E = 0.5$ for it to occur the second-order accuracy.

We have verified that in all examples the norm of the error depends on the position of the vertices of the square with respect to the grid.

Quadratic interpolation procedures

In the case in which we perform the quadratic interpolations we show the numerical results only in the case in which $\theta_S = \theta_N = \theta_W = \theta_E = 0.5$, because $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$ $(p = 1, 2, \infty)$ whatever is the position of the vertices of the domain. Furthermore, $\|e_h\|_{L^p}$ depends on the position of the vertices of the square.

First example

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9046	-1.89704	-1.9139

Second example

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9280	-1.9196	-1.9102

Third example

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.0285	-2.0235	-2.0357

2.2.3 Dirichlet boundary conditions - Circular domain

To conclude the analysis in the 2D case, we present the numerical results in the case of a circular domain. It is clear that in this case, unlike the previous case, we can not choose the values of the θ_{ij} (coefficients of bilinear or biquadratic interpolations), but we can only fix the center and the radius of the circumference. In fact, the values of the θ_{ij} vary from ghost point to ghost point. We start imposing the Dirichlet boundary conditions.

We performed various numerical tests on different functions changing the position of the circumference with respect to the grid, to see if the accuracy can depend on it or not.

Behavior of the inverse matrix of $\mathbf{A}_h$

First, we show the behavior of the norm of the inverse matrix of A_h as a function of N. We have verified that the accuracy does not depend on the position of the circumference, therefore we present below the numerical results obtained choosing

$$x_{\mathrm{C}} = 0.4,\ y_{\mathrm{C}} = 0.4,\ \mathrm{R} = 0.3.$$

We can read the values of the best-fit slope in the table below, both in the case we perform the bilinear interpolations and in the case we perform the biquadratic interpolations, which shows that

$$\|\mathrm{A}_h^{-1}\|_1 \sim \mathrm{O}(N),\ \|\mathrm{A}_h^{-1}\|_2 \sim \mathrm{O}(N^0),\ \|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)\ :$$

$N_{\min} = 49,\ N_{\max} = 181$	$sl(\|\mathrm{A}_h^{-1}\|_1)$	$sl(\|\mathrm{A}_h^{-1}\|_2)$	$sl(\|\mathrm{A}_h^{-1}\|_\infty)$
Bilinear interpolation procedures	0.8737	0.1855	$0.3240\ 10^{-1}$
Biquadratic interpolation procedures	0.8238	0.1102	$0.7302\ 10^{-1}$

Also in the case of a circular domain, for the Dirichlet problem, the stability is guaranteed in the ∞-norm.

The spectral radius of the inverse matrix basically remains constant, both in the case we perform the bilinear interpolations and in the case we perform the biquadratic interpolations. The following figure (Figure 2.20) shows the spectral radius of A_h^{-1} in the case in which we perform the bilinear interpolations.

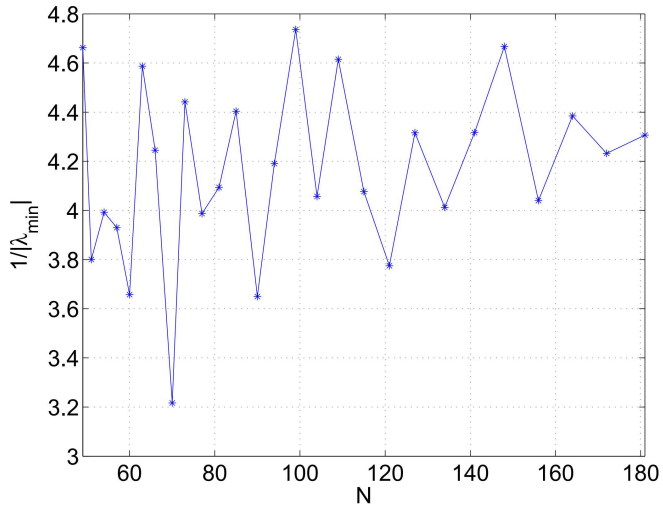


Figure 2.20: Spectral radius of A_h^{-1} as a function of N

Behavior of the Schur complement of $\mathbf{A}_h$

The norm of the inverse of the Schur complement of A_h and its spectral radius tend to a constant for large values of N. We have verified that the accuracy does not depend on the position of the circumference, therefore we present below the numerical results obtained choosing

$$x_{\mathrm{C}} = 0.4,\ y_{\mathrm{C}} = 0.4,\ \mathrm{R} = 0.3 :$$

$N_{\min} = 49,\ N_{\max} = 181$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_\infty)$
BILINEAR INTERPOLATION PROCEDURES	$0.4428\ 10^{-1}$	$0.2604\ 10^{-1}$	$0.1078\ 10^{-1}$
BIQUADRATIC INTERPOLATION PROCEDURES	$0.7937\ 10^{-1}$	0.1111	$0.7318\ 10^{-1}$

The following figure (Figure 2.21) shows the behavior of the norm of inverse matrix of the Schur complement of A_h and of its spectral radius as a function of N, in the case in which we perform the bilinear interpolations:

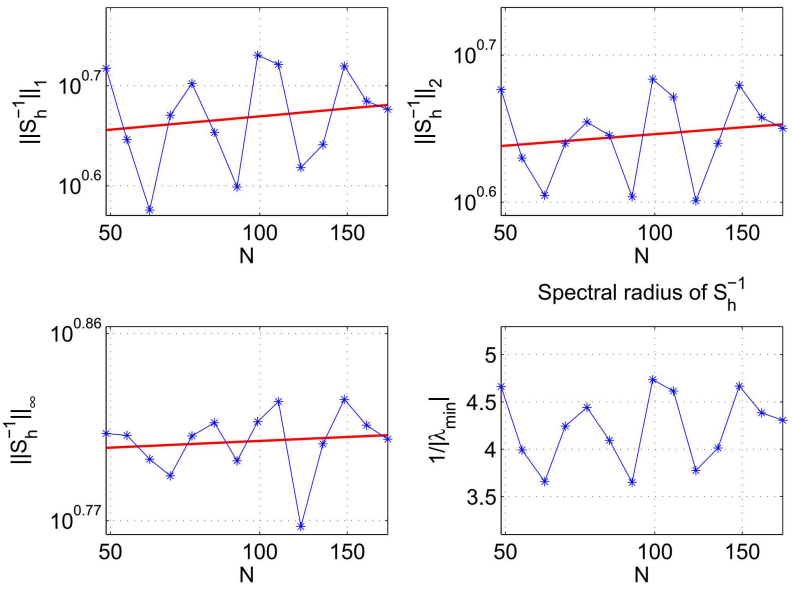


Figure 2.21: p - Norm and spectral radius of the inverse of the Schur complement of A_h as a function of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

Behavior of the consistency error τ_h

We proceed showing the behavior of the consistency error as a function of the number of the grid points. Also in this case the accuracy does not depend on the position of the circumference, therefore we present below the numerical results obtained choosing

$$x_C = 0.4, \; y_C = 0.4, \; R = 0.3.$$

We can read the values of the best-fit slope in the tables below, which show that $\|\tau_h\|_{L^p} \sim O(h^2)$, $p = 1, 2, \infty$.

First example

$$-u_{xx} - u_{yy} = \frac{4x + x^3 + xy^2}{(x^2 + y^2 + 1)^2\sqrt{x^2 + y^2 + 1}}$$

$N_{min} = 20, \; N_{max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
BILINEAR INTERPOLATION PROCEDURES	-2.0111	-1.9976	-1.9834
BIQUADRATIC INTERPOLATION PROCEDURES	-1.9966	-1.9966	-1.9834

Second example

$$-u_{xx} - u_{yy} = -2e^{x^2+y^2}[2 + 2(x^2 + y^2)]$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
BILINEAR INTERPOLATION PROCEDURES	-2.0124	-1.9993	-1.9808
BIQUADRATIC INTERPOLATION PROCEDURES	-1.9976	-1.9981	-1.9808

Third example

$$-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi y)\cos(2\pi x)$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
BILINEAR INTERPOLATION PROCEDURES	-1.9911	-1.9966	-1.9979
BIQUADRATIC INTERPOLATION PROCEDURES	-1.9885	-1.9966	-1.9979

We have verified that in all examples the norm of the consistency error does not depend on the values of the θ_{ij}, both in the case we perform the bilinear interpolations and in the case we perform the biquadratic interpolations.

Behavior of the error e_h

Now, we show the behavior of the norm of the error as a function of N.

BILINEAR INTERPOLATION PROCEDURES

First, we show the results in the case we perform the bilinear interpolation procedures. We have verified that the accuracy does not depend on the position of the circumference, therefore we present below the numerical results obtained choosing

$$x_{\text{C}} = 0.4,\ y_{\text{C}} = 0.4,\ \text{R} = 0.3.$$

We can read the values of the best-fit slope in the tables below, which show that $\|e_h\|_{L^p} \sim \text{O}(h^2)$, $p = 1, 2, \infty$.

First example

$$-u_{xx} - u_{yy} = \frac{4x + x^3 + xy^2}{(x^2 + y^2 + 1)^2\sqrt{x^2 + y^2 + 1}}$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.1149	-2.0942	-1.8843

Second example

$$-u_{xx} - u_{yy} = -2e^{x^2+y^2}[2 + 2(x^2 + y^2)]$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.1271	-2.1097	-1.7630

Third example

$$-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi y)\cos(2\pi x)$$

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.1721	-2.1543	-1.9570

In all examples the numerical results depend on the values of the θ_{ij}.

Biquadratic interpolation procedures

In the case in which we perform the biquadratic interpolations to impose the Dirichlet conditions we show the numerical results choosing $x_{\text{C}} = y_{\text{C}} = 0.50001$ and R = 0.45.

First example

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9526	-1.9348	-1.9565

Second example

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9435	-1.9211	-2.0922

Third example

$N_{\min} = 20,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.0124	-1.9645	-1.9619

In all examples the numerical results do not depend on the values of the θ_{ij}.

2.2.4 Mixed boundary conditions - Circular domain

Once the analysis of the Dirichlet problem is concluded, we proceed showing the numerical results of the mixed problem. In particular, we impose Dirichlet boundary conditions in the area to the right of the center and Neumann boundary conditions in the area to the left of the center.

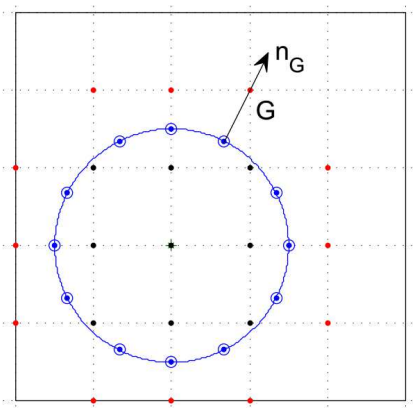


Figure 2.22: Outward unit normal vector of the ghost point G

We impose the Neumann boundary conditions in the point $\mathrm{F} = (x_{\mathrm{F}}, y_{\mathrm{F}})$ (projection of the ghost point G on the border) through the formula:

$$\frac{\partial u}{\partial \mathbf{n}}(x_{\mathrm{F}}, y_{\mathrm{F}}) = \frac{\partial u}{\partial x}(x_{\mathrm{F}}, y_{\mathrm{F}})\mathrm{n}_x + \frac{\partial u}{\partial y}(x_{\mathrm{F}}, y_{\mathrm{F}})\mathrm{n}_y,$$

in which n_x and n_y are the components of the outward unit normal $\mathbf{n_G}$ at the border in the point G (see Figure 2.22). We compute n_x and n_y in this way:

$$\mathrm{n}_x = \frac{x_{\mathrm{G}} - x_{\mathrm{C}}}{\mathrm{d}}, \ \mathrm{n}_y = \frac{y_{\mathrm{G}} - y_{\mathrm{C}}}{\mathrm{d}},$$

where $\mathrm{d} = \sqrt{(x_{\mathrm{G}} - x_{\mathrm{C}})^2 + (y_{\mathrm{G}} - y_{\mathrm{C}})^2}$ is the usual Euclidean distance.

Behavior of the inverse matrix of A_h

We start showing the behavior of the norm of the inverse matrix of A_h as a function of N. We present below the numerical results obtained choosing

$$x_{\mathrm{C}} = 0.4, \ y_{\mathrm{C}} = 0.4, \ \mathrm{R} = 0.3.$$

We can read the values of the best-fit slope in the tables below, which show that

$$\|\mathrm{A}_h^{-1}\|_1 \sim \mathrm{O}(N^{\frac{3}{2}}), \ \|\mathrm{A}_h^{-1}\|_2 \sim \mathrm{O}(N^{\frac{1}{2}}), \ \|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0).$$

Bilinear interpolation procedures

$N_{\min} = 49,\ N_{\max} = 181$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
	1.3789	0.5180	$0.3796\ 10^{-1}$

Biquadratic interpolation procedures

$N_{\min} = 49,\ N_{\max} = 181$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
	1.3906	0.3703	$0.7253\ 10^{-1}$

Also in the case of a circular domain, for the mixed problem, the stability is guaranteed in the ∞-norm.

The spectral radius of the inverse matrix basically remains constant, both in the case we perform the bilinear interpolations and in the case we perform the biquadratic interpolations.

Behavior of the Schur complement of $\mathbf{A}_h$

Now, we show the behavior of the norm of the inverse of the Schur complement of A_h as a function of N.

Bilinear interpolation procedures

In the case in which we perform the bilinear interpolations, the numerical results show that

$$\|\mathrm{S}_h^{-1}\|_1 \sim \mathrm{O}(N^{\frac{1}{2}}),\ \|\mathrm{S}_h^{-1}\|_2 \sim \mathrm{O}(N^0),\ \|\mathrm{S}_h^{-1}\|_\infty \sim \mathrm{O}(N^0).$$

We present below the numerical results obtained choosing

$$x_{\mathrm{C}} = 0.4,\ y_{\mathrm{C}} = 0.4,\ \mathrm{R} = 0.3:$$

$N_{\min} = 49,\ N_{\max} = 181$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_\infty)$
	0.2960	$0.3191\ 10^{-1}$	$0.2183\ 10^{-1}$

Biquadratic interpolation procedures

In the case in which we perform the biquadratic interpolations, the numerical results show that

$$\|\mathrm{S}_h^{-1}\|_1 \sim \mathrm{O}(N^{\frac{1}{2}}),\ \|\mathrm{S}_h^{-1}\|_2 \sim \mathrm{O}(N^0),\ \|\mathrm{S}_h^{-1}\|_\infty \sim \mathrm{O}(N^0),$$

but there are cases (for small rays) in which $\|\mathrm{S}_h^{-1}\|_p \sim \mathrm{O}(N^0),\ p = 1, 2, \infty$. We can read the values of the best-fit slope in the table below:

$N_{\min} = 49,\ N_{\max} = 181$	$sl(\|\mathrm{S}_h^{-1}\|_1)$	$sl(\|\mathrm{S}_h^{-1}\|_2)$	$sl(\|\mathrm{S}_h^{-1}\|_\infty)$
$\mathrm{C}((0.4, 0.4), 0.3)$	0.2786	$0.8237\ 10^{-1}$	$0.8445\ 10^{-1}$
$\mathrm{C}((0.77, 0.34), 0.1)$	$0.5901\ 10^{-1}$	$0.7096\ 10^{-1}$	$-0.8553\ 10^{-1}$

Behavior of the consistency error τ_h

We proceed showing the behavior of the norm of the consistency error as a function of the number of the grid points. We have verified that the accuracy does not depend on the position of the circumference, therefore we present below the numerical results obtained choosing

$$x_{\mathrm{C}} = 0.4,\ y_{\mathrm{C}} = 0.4,\ \mathrm{R} = 0.3.$$

We can read the values of the best-fit slope in the tables below, which show that

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h)$$

if we perform the bilinear interpolations, and $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$ $(p = 1, 2, \infty)$ if we perform the biquadratic interpolations.

First example

$$-u_{xx} - u_{yy} = \frac{4x + x^3 + xy^2}{(x^2 + y^2 + 1)^2\sqrt{x^2 + y^2 + 1}}$$

$N_{\min} = 36,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
Bilinear interpolation procedures	-2.0251	-1.6064	-1.0061
Biquadratic interpolation procedures	-1.9398	-1.9633	-1.9837

Second example

$$-u_{xx} - u_{yy} = -2e^{x^2+y^2}[2 + 2(x^2 + y^2)]$$

$N_{\min} = 36,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
BILINEAR INTERPOLATION PROCEDURES	-2.0415	-1.6445	-1.0098
BIQUADRATIC INTERPOLATION PROCEDURES	-1.9199	-1.9527	-1.9717

Third example

$$-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi y)\cos(2\pi x)$$

$N_{\min} = 36,\ N_{\max} = 148$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
BILINEAR INTERPOLATION PROCEDURES	-2.0014	-1.8859	-1.0061
BIQUADRATIC INTERPOLATION PROCEDURES	-1.9126	-1.9498	-1.9956

We have verified that in all examples the norm of the consistency error depend on the values of the θ_{ij} if we perform the bilinear interpolations and it does not depend on the values of the θ_{ij} if we perform the biquadratic interpolations.

Behavior of the error e_h

Finally, we show the behavior of the norm of the error as a function of the number of the grid points. We have verified that the accuracy does not depend on the position of the circumference, therefore we present below the numerical results obtained choosing

$$x_{\text{C}} = 0.4,\ y_{\text{C}} = 0.4,\ \text{R} = 0.3.$$

We can read the values of the best-fit slope in the tables below, which show that $\|e_h\|_{L^p} \sim \text{O}(h)$ $(p = 1, 2, \infty)$ if we perform the bilinear interpolations, and $\|e_h\|_{L^p} \sim \text{O}(h^2)$ $(p = 1, 2, \infty)$, if we perform the biquadratic interpolations.

First example

$$-u_{xx} - u_y = \frac{4x + x^3 + xy^2}{(x^2+y^2+1)^2\sqrt{x^2+y^2+1}}$$

$N_{\min} = 44,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
BILINEAR INTERPOLATION PROCEDURES	-1.1369	-1.1174	-1.0476
BIQUADRATIC INTERPOLATION PROCEDURES	-1.9439	-1.9280	-1.913

Second example

$$-u_{xx} - u_{yy} = -2e^{x^2+y^2}[2 + 2(x^2+y^2)]$$

$N_{\min} = 44,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
BILINEAR INTERPOLATION PROCEDURES	-1.1851	-1.1646	-1.1424
BIQUADRATIC INTERPOLATION PROCEDURES	-1.9445	-1.9206	-1.9203

Third example

$$-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi y)\cos(2\pi x)$$

$N_{\min} = 44,\ N_{\max} = 148$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
BILINEAR INTERPOLATION PROCEDURES	-1.1090	-1.0498	-1.0440
BIQUADRATIC INTERPOLATION PROCEDURES	-1.9927	-1.9579	-1.9640

In this last example, a second-order accuracy also occurs in the L^2-norm in the case in which we perform the bilinear interpolations. This behavior also occurred in the case of a square domain.

We have verified that in all examples the norm of the error depend on the values of the θ_{ij} if we perform the bilinear interpolations and it does not depend on the values of the θ_{ij} if we perform the biquadratic interpolations.

Conclusions

Dirichlet problem

We itemize the numerical results on the *Dirichlet problem* (1.24). The numerical results are the following:

- $\|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$;
- $\rho(\mathrm{A}_h^{-1}) \to \mathrm{c}$, where c is a constant;
- $\|\mathrm{S}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$;
- $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$.

The norm of the consistency error and the norm of the error **can depend on the values of $\theta_{\mathbf{ij}}$** in the case in which we perform the bilinear interpolations to impose the Dirichlet boundary conditions. They **do not depend on the values of $\theta_{\mathbf{ij}}$** in the case in which we perform the biquadratic interpolations.

Mixed problem

We itemize the numerical results on the *mixed problem* (1.27). The numerical results are the following:

- $\|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$;
- $\rho(\mathrm{A}_h^{-1}) \to \mathrm{c}$, where c is a constant;
- $\|\mathrm{S}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$.

The accuracy orders of $\|\tau_h\|_{L^p}$ and of $\|e_h\|_{L^p}$ depend on the procedure with which we impose the Neumann boundary conditions. If we perform the bilinear interpolation procedures we have:

- $\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h)$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h),\ p = 1, 2, \infty$,

because the Neumann conditions confers a first-order accuracy at the border in the case in which we perform the bilinear interpolations. If we perform the biquadratic interpolation procedures we have the second order accuracy both in the consistency error and in the error:

- $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$.

The norm of the consistency error and the norm of the error **can depend on the values of $\theta_{\mathbf{ij}}$** in the case in which we perform the bilinear interpolations to impose the boundary conditions. They **do not depend on the values of $\theta_{\mathbf{ij}}$** in the case in which we perform the biquadratic interpolations.

2.3 Numerical tests in 3D

We conclude this Chapter showing some numerical simulations to highlight the main features of the numerical method in three-dimensional case.

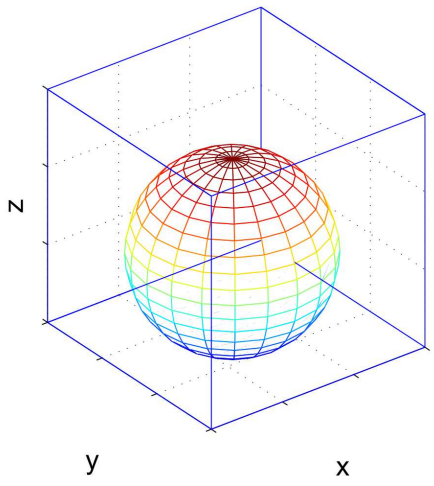


Figure 2.23: Spherical domain

We chose a *spherical domain*. A sphere of radius R and center $\mathrm{C}(x_\mathrm{C}, y_\mathrm{C}, z_\mathrm{C})$ is contained in the unit cube. The value of the radius and the position of the center can change. We discretize the unit cube with a regular grid of size h along the three axes respectively. We indicate by N the number of intervals with which we discretize the interval $[0, 1]$ along each axis $\left(N = \frac{1}{h}\right)$. In total we have $(N+1)^3$ grid points. In this way, the sphere is also discretized. We indicate by n the total number of grid points in the sphere. The figure above on the left (Figure 2.23) represents a sphere of radius $\mathrm{R} = 0.3$ and center $\mathrm{C}(0.42, 0.42, 0.43)$. It is for these values of the center and of the radius that we will show the numerical results, unless otherwise stated.

2.3.1 Dirichlet boundary conditions

We start showing the numerical tests on the Dirichlet problem. We start showing the behaviors of the norm of the inverse matrix of A_h and its spectral radius and the norm of the inverse of the Schur complement S_h as a function of N.

Behavior of the inverse matrix of $\mathbf{A}_h$

In the following tables we report the values of the best-fit slope of the norm of the inverse matrix as a function of N.

TRILINEAR INTERPOLATION PROCEDURES

$N_{\min} = 33,\ N_{\max} = 49$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
	0.4453	$0.7380\ 10^{-1}$	$0.5213\ 10^{-1}$

TRIQUADRATIC INTERPOLATION PROCEDURES

$N_{\min} = 22,\ N_{\max} = 40$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{A}_h^{-1}\|\|_\infty)$
	0.2705	$-0.1672\ 10^{-1}$	$-0.6804\ 10^{-1}$

In both cases, the numerical results show that

$$\|\mathrm{A}_h^{-1}\|_1 \sim \mathrm{O}(N^{\frac{1}{2}}),\ \|\mathrm{A}_h^{-1}\|_2 \sim \mathrm{O}(N^0),\ \|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0),$$

thus the stability is guaranteed in the ∞-norm, as in 1D and 2D cases.

The spectral radius of A_h^{-1} substantially tends to a constant for large values of N.

Behavior of the Schur complement of $\mathbf{A}_h$

The norm of the inverse of the Schur complement of A_h tends to a constant for large values of N. We can read the values of the best-fit slope in the tables below:

TRILINEAR INTERPOLATION PROCEDURES

$N_{\min} = 20,\ N_{\max} = 40$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_\infty)$
	$0.8045\ 10^{-1}$	$0.4898\ 10^{-1}$	$0.3268\ 10^{-1}$

TRIQUADRATIC INTERPOLATION PROCEDURES

$N_{\min} = 20,\ N_{\max} = 40$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_\infty)$
	-0.1973	$-0.6547\ 10^{-1}$	$-0.6950\ 10^{-1}$

We proceed by showing the behaviors of the norm of the consistency error and of the norm of the error as a function of N. We consider the following equation:

$$-u_{xx} - u_{yy} - u_{zz} = f,$$

with

$$f = \frac{[(y^2z^2 + x^2y^2 + x^2z^2)(x+y+z)^2 + 6]\sin(xyz) + 2(yz+xy+xz)\cos(xyz)(x+y+z)}{(x+y+z)^3}$$

and whose exact solution is $u = \frac{\sin(xyz)}{x+y+z}$.

We have approximated $\|\tau_h\|_{L^p}$ and $\|e_h\|_{L^p}$ $(p = 1, 2, \infty)$ in this way:

$$\|a_h\|_{L^1} = h^3 \sum_{i,j,k \ \in \ \Omega_h^I \cup \Omega_h^G} |a_h(x_i, y_j, z_k)|,$$

$$\|a_h\|_{L^2} = \sqrt{h^3 \sum_{i,j,k \ \in \ \Omega_h^I \cup \Omega_h^G} |a_h(x_i, y_j, z_k)|^2},$$

$$\|a_h\|_{L^\infty} = \max_{i,j,k \ \in \ \Omega_h^I \cup \Omega_h^G} |a_h(x_i, y_j, z_k)|,$$

where a_h is τ_h or e_h.

Behavior of the consistency error τ_h

We start showing the results about the norm of the consistency error as a function of N.

Trilinear interpolation procedures

The following figure (Figure 2.24) shows the behavior of the norm of τ_h as a function of N in the case in which we perform the trilinear interpolation procedures to impose the boundary conditions. We can read the values of the best-fit slope in the table below, which shows that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$N_{\min} = 24,\ N_{\max} = 60$	$sl(\|\tau_h\|_{L^1})$	$al(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0252	-2.0020	-1.9549

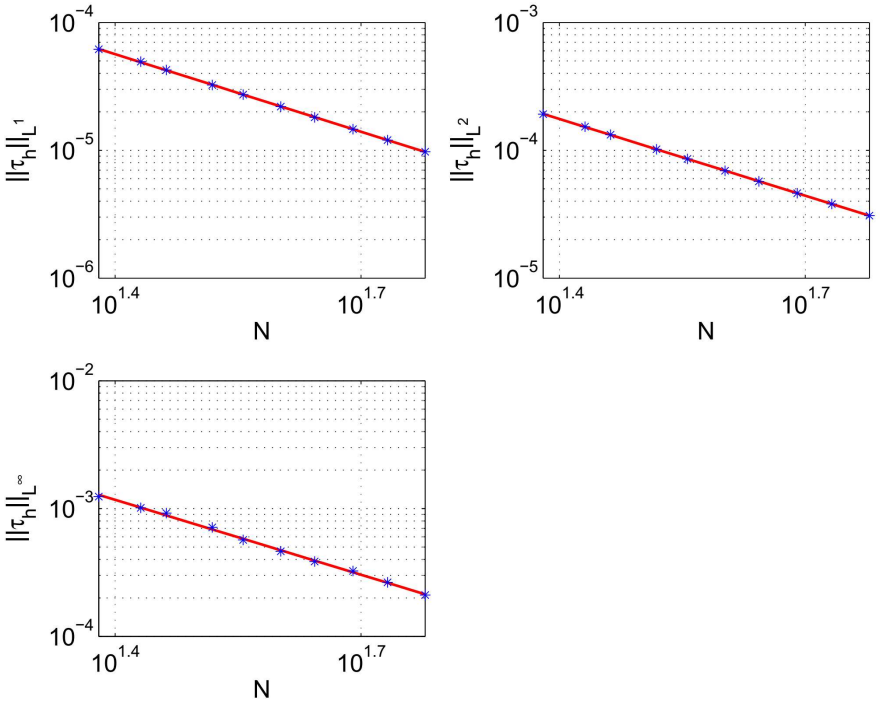


Figure 2.24: p - Norm of the consistency error τ_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

Triquadratic interpolation procedures

Also in the case in which we perform the triquadratic interpolations we obtain $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$N_{\min} = 24,\ N_{\max} = 60$	$sl(\|\tau_h\|_{L^1})$	$sl(\|\tau_h\|_{L^2})$	$sl(\|\tau_h\|_{L^\infty})$
	-2.0046	-2.0007	-1.9549

Behavior of the error e_h

We continue showing the results about the behavior of the norm of the error as a function of N. The numerical results were obtained choosing $x_C = y_C = z_C = 0.500001$ and $R = 0.3$.

Trilinear interpolation procedures

The following figure (Figure 2.25) shows the behavior of the norm of e_h as a function of N in the case in which we perform the trilinear interpolation procedures. We can read the values of the best-fit slope in the table below, which shows that $\|e_h\|_{L^p} \sim O(h^2)$, $p = 1, 2, \infty$:

$N_{min} = 40,\ N_{max} = 66$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.2942	-2.2248	-1.7875

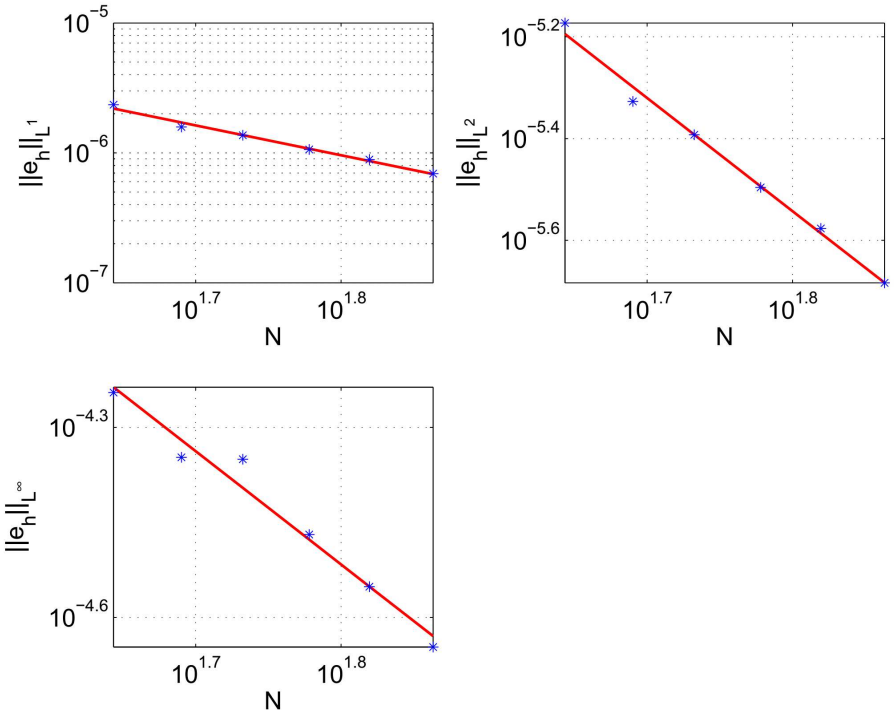


Figure 2.25: p - Norm of the error e_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

Triquadratic interpolation procedures

The following figure (Figure 2.26) shows the behavior of the norm of e_h as a function of N in the case in which we perform the triquadratic interpolation procedures. We can read the values of the best-fit slope in the table below, which shows that $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$N_{\min} = 44,\ N_{\max} = 73$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-2.0961	-2.0017	-2.4117

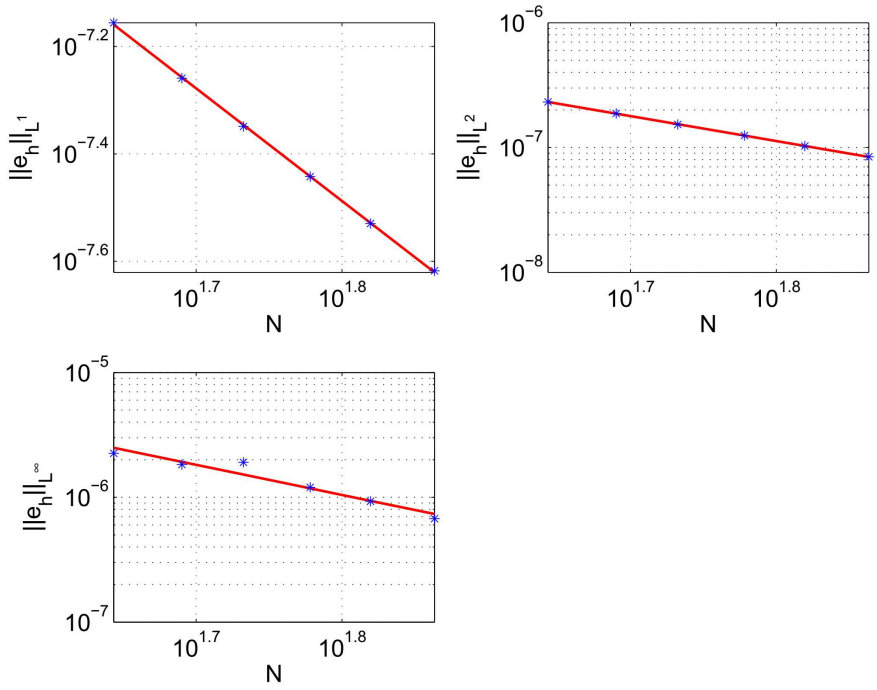


Figure 2.26: p - Norm of the error e_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

2.3.2 Mixed boundary conditions

We proceed by showing the results of the mixed problem both in the case in which we perform the trilinear interpolations and in the case in which we perform the triquadratic interpolations.

Behavior of the inverse matrix of A_h

In the following tables we report the values of the best-fit slope of the norm of the inverse matrix of A_h as a function of N:

Trilinear interpolation procedures

$N_{\min} = 22,\ N_{\max} = 40$	$sl(\|\mathrm{A}_h^{-1}\|_1)$	$sl(\|\mathrm{A}_h^{-1}\|_2)$	$sl(\|\mathrm{A}_h^{-1}\|_\infty)$
	1.2511	$0.5230\ 10^{-1}$	$0.3182\ 10^{-1}$

Triquadratic interpolation procedures

$N_{\min} = 22,\ N_{\max} = 40$	$sl(\|\mathrm{A}_h^{-1}\|_1)$	$sl(\|\mathrm{A}_h^{-1}\|_2)$	$sl(\|\mathrm{A}_h^{-1}\|_\infty)$
	0.9881	$0.6027\ 10^{-2}$	$-0.7371\ 10^{-1}$

In both cases, the numerical results show that

$$\|\mathrm{A}_h^{-1}\|_1 \sim \mathrm{O}(N),\ \|\mathrm{A}_h^{-1}\|_2 \sim \mathrm{O}(N^0),\ \|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0),$$

thus the stability is guaranteed in the ∞-norm, as in 1D and 2D cases.

The spectral radius of A_h^{-1} substantially tends to a constant for large values of N.

Behavior of the Schur complement of A_h

The numerical results about the norm of the inverse of the Schur complement of A_h as a function of N show that

$$\|\mathrm{S}_h^{-1}\|_1 \sim \mathrm{O}(N^{\frac{1}{2}}),\ \|\mathrm{S}_h^{-1}\|_2 \sim \mathrm{O}(N^0),\ \|\mathrm{S}_h^{-1}\|_\infty \sim \mathrm{O}(N^0),$$

in the case in which we perform the trilinear interpolations and it tend to a constant if we perform the triquadratic interpolations, as we can read ih the following tables:

TRILINEAR INTERPOLATION PROCEDURES

$N_{\min} = 20,\ N_{\max} = 40$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_\infty)$
	0.3022	0.1041	$0.4237\ 10^{-1}$

TRIQUADRATIC INTERPOLATION PROCEDURES

$N_{\min} = 20,\ N_{\max} = 40$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_1)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_2)$	$sl(\|\|\mathrm{S}_h^{-1}\|\|_\infty)$
s	$0.4195\ 10^{-1}$	$-0.2785\ 10^{-1}$	$-0.4022\ 10^{-1}$

Finally we show the behaviors of the consistency error and of the error, choosing the same equation of the Dirichlet problem, but imposing Dirichlet boundary conditions in the area to the right of the center and Neumann boundary conditions in the area to the left of the center.

Behavior of the consistency error τ_h

TRILINEAR INTERPOLATION PROCEDURES

The following figure (Figure 2.27) shows the behavior of the norm of τ_h as a function of N in the case in which we perform the trilinear interpolation procedures. We can read the values of the best-fit slope in the table below, which shows that

$$\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h).$$

$N_{\min} = 33,\ N_{\max} = 66$	$sl(\|\|\tau_h\|\|_{L^1})$	$sl(\|\|\tau_h\|\|_{L^2})$	$sl(\|\|\tau_h\|\|_{L^\infty})$
	-2.0711	-1.7737	-0.8195

TRIQUADRATIC INTERPOLATION PROCEDURES

In the case in which we perform the triquadratic interpolation procedures the table shows that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$:

$N_{\min} = 24,\ N_{\max} = 60$	$sl(\|\|\tau_h\|\|_{L^1})$	$sl(\|\|\tau_h\|\|_{L^2})$	$sl(\|\|\tau_h\|\|_{L^\infty})$
	-2.0080	-1.9981	-1.9194

These results were obtained choosing $x_{\mathrm{C}} = y_{\mathrm{C}} = z_{\mathrm{C}} = 0.500001$ and $\mathrm{R} = 0.3$.

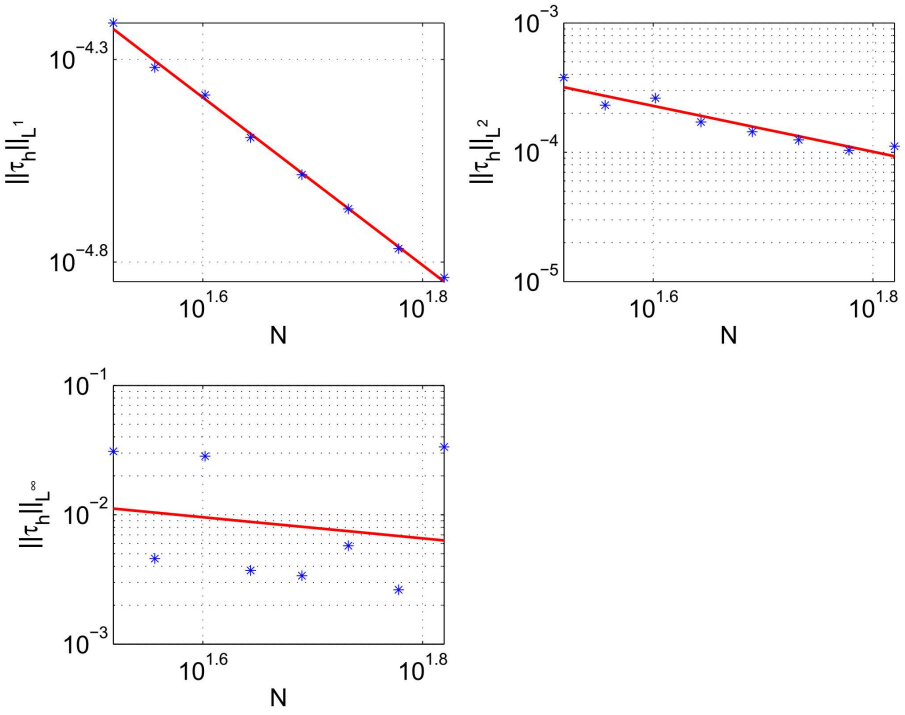


Figure 2.27: p - Norm of the consistency error τ_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

Behavior of the error e_h

We continue showing the numerical results on the norm of the error as a function of N, choosing $x_{\mathrm{C}} = y_{\mathrm{C}} = z_{\mathrm{C}} = 0.500001$ and $\mathrm{R} = 0.3$.

Trilinear interpolation procedures

The following figure (Figure 2.28) shows the behavior of the norm of e_h as a function of N in the case in which we perform the trilinear interpolation procedures. We can read the values of the best-fit slope in the table below, which shows that $\|e_h\|_{L^p} \sim \mathrm{O}(h)$, $p = 1, 2, \infty$:

$N_{\min} = 27,\ N_{\max} = 66$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.1307	-1.0496	-1.0444

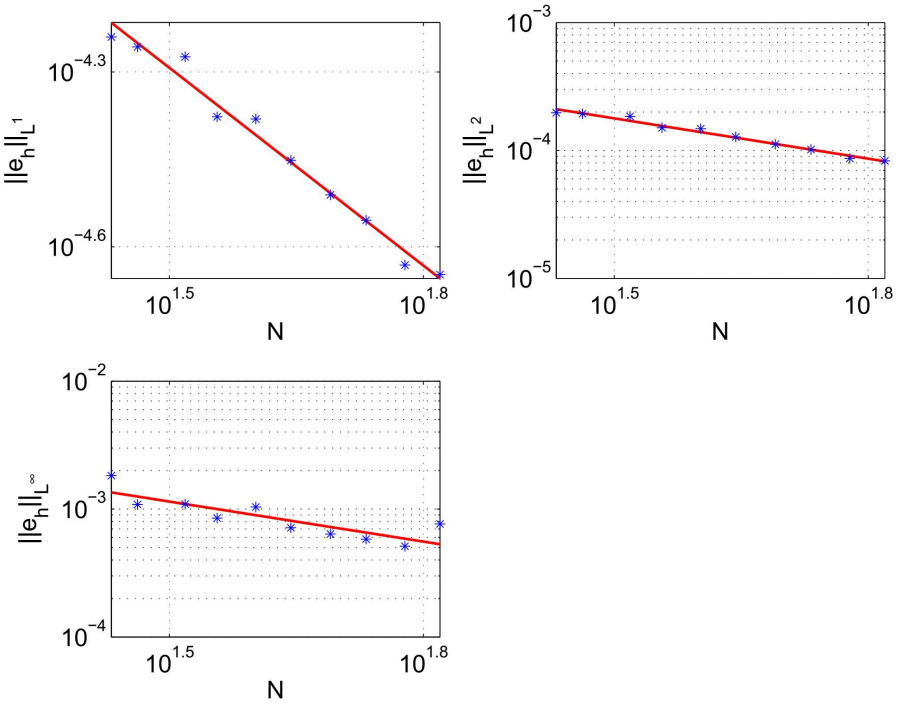


Figure 2.28: p - Norm of the error e_h as a function of the number of the grid points: $p = 1$ (top left), $p = 2$ (top right), $p = \infty$ (bottom left)

Triquadratic interpolation procedures

In the case we perform the triquadratic interpolation procedures we have $\|e_h\|_{L^p} \sim \mathrm{O}(h^2)$ $(p = 1, 2, \infty)$, as we can read in the following table:

$N_{\min} = 40,\ N_{\max} = 66$	$sl(\|e_h\|_{L^1})$	$sl(\|e_h\|_{L^2})$	$sl(\|e_h\|_{L^\infty})$
	-1.9855	-1.9463	-2.0210

Conclusions

Dirichlet problem

We itemize the numerical results on the *Dirichlet problem* (1.30). The numerical results are similar to the 2D case and are the following:

- $\|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$;
- $\rho(\mathrm{A}_h^{-1}) \to \mathrm{c}$, where c is a constant;
- $\|\mathrm{S}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$;
- $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$.

Mixed problem

We itemize the numerical results on the *mixed problem* (1.32). The numerical results are the following:

- $\|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$;
- $\rho(\mathrm{A}_h^{-1}) \to \mathrm{c}$, where c is a constant;
- $\|\mathrm{S}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$.

The accuracy orders of $\|\tau_h\|_{L^p}$ and of $\|e_h\|_{L^p}$ depend on the procedure with which we impose the Neumann boundary conditions. If we perform the trilinear interpolation procedures we have:

- $\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h)$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h),\ p = 1, 2, \infty$,

because the Neumann conditions confers a first-order accuracy at the border in the case in which we perform the trilinear interpolations. If we perform the triquadratic interpolation procedures we have:

- $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$.

3

Convergence analysis for the Coco-Russo method

We conclude the analysis of the Coco-Russo method addressing the question of the stability and convergence of the numerical method.

The consistency errors for Dirichlet and mixed problems are computed in one, two and three spatial dimensions. We have one proof of the stability and one of the convergence of the numerical method only in the one-dimensional case. We will see that the extension of the proofs to the two-dimensional case is quite complex due to the arbitrary nature of the domain in 2D.

We start analyzing the consistency, the convergence and the stability of the numerical method in the one-dimensional case.

3.1 Consistency for the Dirichlet problem in 1D

We consider the Dirichlet problems (1.8) and (1.12), in which the domain is $[\mathbf{a}, 1]$.

We prove that the numerical method is consistent and we determine that in the internal points the numerical solution is *consistent with second-order accuracy*.

We calculate the consistency error $\tau_h = \mathrm{A}_h u - \mathrm{A}u$.

In the internal points, if $u \in \mathrm{C}^4([x_0, x_n])$, we have:

$$\tau_i = \tau_h(x_i) = \Delta_h u(x_i) - \Delta u(x_i) = \frac{h^2}{12} u^{\mathrm{IV}}(\xi_i) \quad i = 1, \ldots, n-1,$$

for some $\xi_i \in [x_{i-1}, x_{i+1}]$. Therefore,

$$\begin{aligned}
\|\tau_h\|_{L^1} &= \int |\tau_h| \mathrm{d}x \approx \sum_{i=1}^{n-1} |\tau_h(x_i)| h = \frac{h^2}{12} \sum_{i=1}^{n-1} |u^{\mathrm{IV}}(\xi_i)| h \approx \frac{h^2}{12} \|u^{\mathrm{IV}}\|_{L^1} \ \sim \mathrm{O}(h^2), \\
\|\tau_h\|_{L^2} &= \left(\int |\tau_h|^2 \mathrm{d}x \right)^{\frac{1}{2}} \approx \left(\sum_{i=1}^{n-1} |\tau_h(x_i)|^2 h \right)^{\frac{1}{2}} = \frac{h^2}{12} \left(\sum_{i=1}^{n-1} |u^{\mathrm{IV}}(\xi_i)|^2 h \right)^{\frac{1}{2}} \\
&\approx \frac{h^2}{12} \|u^{\mathrm{IV}}\|_{L^2} \ \sim \mathrm{O}(h^2), \\
\|\tau_h\|_{L^\infty} &= \max_{[x_1, x_{n-1}]} |\tau_h| = \frac{h^2}{12} \max_{[x_1, x_{n-1}]} |u^{\mathrm{IV}}(\xi_i)| = \frac{h^2}{12} \|u^{\mathrm{IV}}\|_{L^\infty} \ \sim \mathrm{O}(h^2).
\end{aligned}$$

In the only ghost point x_0, if we impose the Dirichlet condition through the linear interpolation procedure, we have:

$$\tau_0 = \tau(x_0) = \left| -\frac{1}{2} \theta_{\mathrm{a}} (1 - \theta_{\mathrm{a}}) h^2 u''(\xi) \right| \ \sim \mathrm{O}(h^2),$$

for some $\xi \in [\min\{x_0, x_1, \mathrm{a}\}, \max\{x_0, x_1, \mathrm{a}\}]$, while if we impose the Dirichlet condition through the quadratic interpolation procedure, we have:

$$\tau_0 = \tau(x_0) = \left| \frac{1}{6} \theta_{\mathrm{a}} (1 - \theta_{\mathrm{a}}) (1 + \theta_{\mathrm{a}}) h^3 u'''(\xi) \right| \ \sim \mathrm{O}(h^3),$$

for some $\xi \in [\min\{x_0, x_1, x_2, \mathbf{a}\}, \max\{x_0, x_1, x_2, \mathbf{a}\}]$.

With the symmetry of the matrix A_h (through the formula (1.9)), in the case in which we impose the Dirichlet condition in $\mathbf{a}$ through the linear interpolation procedure, we have:

$$\tau_0 = \tau(x_0) = \frac{1}{2} \theta_{\mathrm{a}} u''(\xi) \ \sim \mathrm{O}(h^0),$$

for some $\xi \in [\min\{x_0, x_1, \mathbf{a}\}, \max\{x_0, x_1, \mathbf{a}\}]$.

If we consider the Dirichlet problems (1.13) and (1.14) in which the domain is $[\mathbf{a}, \mathbf{b}] \subset [0, 1]$, the consistency error in the ghost point x_n is calculated in a similar way to the consistency error in the ghost point x_0.

To study the stability of the numerical method just try that $\|\mathrm{A}_h^{-1}\|_p$ has an upper bound. We consider the simplest case in which the domain is $[\mathbf{a}, 1]$ and we perform the linear interpolation to impose the boundary condition in $\mathbf{a}$ (Problem (1.8)).

3.2 Stability for the Dirichlet problem in 1D

This proof will appear in the joint paper in preparation *Spectral and norm estimates for matrix sequences arising from a Finite Difference of approximation of elliptic operators* with Stefano Serra-Capizzano, Sven-Erik Ekström and Giovanni Russo. This is a proof of stability of the numerical method, since an approximation of the norm of the inverse matrix of the numerical method is given.

1. Introduction

Let T_n be a Toeplitz matrix of order n

$$\mathrm{T}_n = \begin{pmatrix} a_0 & \cdots & a_{-\alpha} & & & & \\ \vdots & \ddots & & \ddots & & & \\ a_\alpha & & \ddots & & \ddots & & \\ & \ddots & & \ddots & & a_{-\alpha} & \\ & & \ddots & & \ddots & \vdots & \\ & & & a_\alpha & \cdots & a_0 & \end{pmatrix},$$

where the coefficients a_k, $k = -\alpha, \ldots, \alpha$, are complex numbers and let $\alpha < n$ be a positive integer.

Let $f \in L^1(-\pi, \pi)$ and let $\mathrm{T}_n(f)$ be the Toeplitz matrix generated by f that is $(\mathrm{T}_n(f))_{s,t} = a_{s-t}(f)$, $s, t = 1, \ldots, n$, with f indicated as generating function of $\{\mathrm{T}_n(f)\}$ and with $a_k(f)$ being the k-th Fourier coefficient of f that is

$$a_k(f) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta)\, e^{-\mathbf{i}k\theta}\, d\theta, \quad \mathbf{i}^2 = -1,\ k \in \mathbb{Z}.$$

2. Problem formulation and notation in 1D

We can decompose A_h as follows:

$$\mathrm{A}_h = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} + \frac{1}{h^2}\begin{pmatrix} \theta_{\mathrm{a}}h^2-2 & (1-\theta_{\mathrm{a}})h^2+1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

$$= \frac{1}{h^2}\mathrm{T}_n(2-2\cos(\theta)) + \frac{1}{h^2}\begin{pmatrix} \theta_{\mathrm{a}}h^2-2 & (1-\theta_{\mathrm{a}})h^2+1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

$$= \frac{1}{h^2}\mathrm{T}_n(2-2\cos(\theta)) + \frac{1}{h^2}\begin{pmatrix} \mathbf{v}_h^{\mathrm{T}} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$$

$$= \mathrm{S}_n + \frac{1}{h^2}\mathbf{e}_1\mathbf{v}_h^{\mathrm{T}},$$

where $\mathrm{T}_n(f)$ is the Toeplitz matrix generated by f according to (3.2), with $f(\theta) = 2 - 2\cos(\theta)$ so that, in the matrix in (3.2), we have $\alpha = 1$, $a_0 = 2, a_1 = a_{-1} = -1$.

From the Sherman–Morrison–Woodbury formula [28, 29],we obtain an explicit expression of the inverse matrix of A_h:

$$\begin{aligned} \mathrm{A}_h^{-1} &= \left(\mathrm{I} + \frac{1}{h^2}\mathrm{S}_n^{-1}\mathbf{e}_1\mathbf{v}_h^{\mathrm{T}}\right)^{-1}\mathrm{S}_n^{-1} \\ &= \mathrm{S}_n^{-1} - \frac{\frac{1}{h^2}\mathrm{S}_n^{-1}\mathbf{e}_1\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}}{1+\frac{1}{h^2}\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}\mathbf{e}_1} \\ &= \mathrm{S}_n^{-1} - \mathrm{R}_n. \end{aligned}$$

We want to estimate quite accurately $\|\mathrm{A}_h^{-1}\|_p$ with $p \in [1,\infty]$ and with $\|\cdot\|_p$ being the matrix norm induced by the vector norm $\|\mathbf{x}\|_p = (\sum |x_j|^p)^{1/p}$. We consider only the cases $p = 1, 2, \infty$, since the other estimates can be obtained via classical interpolation techniques.

Being $\mathrm{A}_h^{-1} = \mathrm{S}_n^{-1} - \mathrm{R}_n$, we start by estimating $\|\mathrm{S}_n^{-1}\|_1$, $\|\mathrm{S}_n^{-1}\|_\infty$, $\|\mathrm{R}_n\|_1$, $\|\mathrm{R}_n\|_\infty$, that will be used to give quite precise bounds on $\|\mathrm{A}_h^{-1}\|_1$ and $\|\mathrm{A}_h^{-1}\|_\infty$. From the estimate on $\|\mathrm{A}_h^{-1}\|_1$ and $\|\mathrm{A}_h^{-1}\|_\infty$ and from the inequality $\|\mathrm{A}_h^{-1}\|_2 \leq \sqrt{\|\mathrm{A}_h^{-1}\|_1 \|\mathrm{A}_h^{-1}\|_\infty}$ we will give an estimate for $\|\mathrm{A}_h^{-1}\|_2$.

2.1. Estimating $\|S_n^{-1}\|_p$ with $p = 1, \infty$

We have $\mathrm{S}_n^{-1} = h^2 \mathrm{T}_n^{-1}$ where $\mathrm{T}_n = \mathrm{T}_n(2 - 2\cos(\theta))$ and

$$\mathrm{T}_n^{-1} = \begin{bmatrix} t_1^{(1)} & t_1^{(2)} & \dots & t_1^{(n)} \\ t_2^{(1)} & t_2^{(2)} & \dots & t_2^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ t_n^{(1)} & t_n^{(2)} & \dots & t_n^{(n)} \end{bmatrix} = \begin{bmatrix} \mathbf{t}^{(1)} & \mathbf{t}^{(2)} & \dots & \mathbf{t}^{(n)} \end{bmatrix},$$

with

$$\begin{aligned} t_k^{(r)} &= \frac{(n+1-r)k}{n+1}, \quad k = 1, \dots, r-1, \ r > 1, \\ t_k^{(r)} &= \frac{(n+1-k)r}{n+1}, \quad k = r, \dots, n, \end{aligned}$$

and symmetrically

$$\begin{aligned} t_k^{(r)} &= \frac{(n+1-k)r}{n+1}, \quad r = 1, \dots, k-1, \ k > 1, \\ t_k^{(r)} &= \frac{(n+1-r)k}{n+1}, \quad r = k, \dots, n. \end{aligned}$$

Therefore

$$\left(\mathrm{T}_n^{-1}\right)_{k,r} = t_k^{(r)}.$$

All terms of S_n^{-1} (and T_n^{-1}) are positive and real, and they are symmetric. Hence by using the explicit expressions of the considered norms, we find

$$\begin{aligned} \|\mathrm{S}_n^{-1}\|_\infty &= \max_k \left\{ \sum_{r=1}^n \left(\mathrm{S}_n^{-1}\right)_{k,r} \right\} = \max_k \left\{ h^2 \sum_{r=1}^n \left(\mathrm{T}_n^{-1}\right)_{k,r} \right\} \\ &= \max_r \left\{ h^2 \sum_{k=1}^n \left(\mathrm{T}_n^{-1}\right)_{k,r} \right\} = \max_r \left\{ \sum_{j=1}^n \left(\mathrm{S}_n^{-1}\right)_{k,r} \right\} = \|\mathrm{S}_n^{-1}\|_1, \end{aligned}$$

so we limit ourselves only to obtain an estimate for $\|\mathrm{S}_n^{-1}\|_\infty$.

Numerically it is obvious that the highest row sum for matrices T_n^{-1} with n even is for index $\frac{n}{2}$ (or $\frac{n}{2}+1$, they are equal). For odd n, the highest row sum is for row $\frac{n+1}{2}$.

Thus for n even:

$$\begin{aligned}
\|\mathrm{T}_n^{-1}\|_\infty &= \sum_{r=1}^{n} t_{n/2}^{(r)} = \sum_{r=1}^{n/2-1} \frac{(n+1-\frac{n}{2})r}{n+1} + \sum_{r=n/2}^{n} \frac{(n+1-r)\frac{n}{2}}{n+1} \\
&= \frac{n+2}{2(n+1)} \sum_{r=1}^{n/2-1} r + \frac{n}{2} \sum_{r=n/2}^{n} 1 - \frac{n}{2(n+1)} \sum_{r=n/2}^{n} r \\
&= \frac{n+2}{2(n+1)} \frac{\left(\frac{n}{2}-1\right)\frac{n}{2}}{2} + \frac{n}{2}\left(\frac{n}{2}+1\right) - \frac{n}{2(n+1)}\left(\frac{n(n+1)}{2} - \frac{\left(\frac{n}{2}-1\right)\frac{n}{2}}{2}\right) \\
&= \frac{n+2}{2(n+1)} \frac{(n-2)n}{8} + \frac{n(n+2)}{4} - \frac{n}{2(n+1)}\left(\frac{4n(n+1)}{8} - \frac{(n-2)n}{8}\right) \\
&= \frac{n+2}{2(n+1)} \frac{(n-2)n}{8} + \frac{n(n+2)}{4} - \frac{n}{2(n+1)} \frac{3n(n+2)}{8} \\
&= \frac{n(n+2)}{16(n+1)} \left((n-2) + 4(n+1) - 3n\right) \\
&= \frac{n(n+2)}{16(n+1)} 2(n+1) \\
&= \frac{n(n+2)}{8} \\
&= \frac{2h+1}{8h^2},
\end{aligned}$$

from which:

$$\begin{aligned}
\|\mathrm{S}_n^{-1}\|_\infty &= h^2 \|\mathrm{T}_n^{-1}\|_\infty \\
&= h^2 \frac{1+2h}{8h^2} \\
&= \frac{1+2h}{8},
\end{aligned}$$

and as $h \to 0$ we have $\|\mathrm{S}_n^{-1}\|_\infty \to \frac{1}{8}$.

While for n odd:

$$\begin{aligned}
\|\mathrm{T}_n^{-1}\|_\infty &= \sum_{r=1}^{n} t_{(n+1)/2}^{(r)} = \sum_{r=1}^{(n-1)/2} \frac{\left(n+1-\frac{n+1}{2}\right) r}{n+1} + \sum_{r=(n+1)/2}^{n} \frac{(n+1-r)\frac{n+1}{2}}{n+1} \\
&= \sum_{r=1}^{(n-1)/2} \frac{\frac{n+1}{2} r}{n+1} + \sum_{r=(n+1)/2}^{n} \frac{(n+1-r)\frac{n+1}{2}}{n+1} \\
&= \frac{1}{2} \sum_{r=1}^{(n-1)/2} r + \frac{n+1}{2} \sum_{r=(n+1)/2}^{n} 1 - \frac{1}{2} \sum_{r=(n+1)/2}^{n} r \\
&= \frac{1}{2} \frac{\frac{n-1}{2}\left(\frac{n-1}{2}+1\right)}{2} + \frac{n+1}{2}\left(n - \left(\frac{n+1}{2}-1\right)\right) \\
&\quad - \frac{1}{2}\left(\frac{n(n+1)}{2} - \frac{\frac{n+1}{2}\left(\frac{n+1}{2}-1\right)}{2}\right) \\
&= \frac{(n-1)(n+1)}{16} + \frac{(n+1)^2}{4} - \frac{1}{2}\left(\frac{4n(n+1)}{8} - \frac{(n+1)(n-1)}{8}\right) \\
&= \frac{(n-1)(n+1)}{16} + \frac{(n+1)^2}{4} - \frac{(3n+1)(n+1)}{16} \\
&= \frac{(n+1)^2}{4} - \frac{2(n+1)(n+1)}{16} \\
&= \frac{4(n+1)^2}{16} - \frac{2(n+1)^2}{16} \\
&= \frac{(n+1)^2}{8} = \frac{(h+1)^2}{8h^2},
\end{aligned}$$

from which:

$$\begin{aligned}
\|\mathrm{S}_n^{-1}\|_\infty &= h^2 \|\mathrm{T}_n^{-1}\|_\infty \\
&= h^2 \frac{(h+1)^2}{8h^2} \\
&= \frac{(h+1)^2}{8}
\end{aligned}$$

and as $h \to 0$ we have $\|\mathrm{S}_n^{-1}\|_\infty \to \frac{1}{8}$.

Therefore, from the symmetry, for all n we obtain:

$$\lim_{n\to\infty} \|\mathrm{S}_n^{-1}\|_\infty = \lim_{n\to\infty} \|\mathrm{S}_n^{-1}\|_1 = \frac{1}{8}.$$

2.2. Estimating $\|R_n\|_p$ for $p = 1, \infty$

Since $\mathrm{S}_n^{-1} = h^2\mathrm{T}_n^{-1}$ and $\mathrm{T}_n^{-1}\mathbf{e}_1 = \mathbf{t}^{(1)}$, we find that:

$$\begin{aligned}
\mathrm{R}_n &= \frac{\frac{1}{h^2}\mathrm{S}_n^{-1}\mathbf{e}_1\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}}{1+\frac{1}{h^2}\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}\mathbf{e}_1} \\
&= \frac{\mathrm{T}_n^{-1}\mathbf{e}_1\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}}{1+\mathbf{v}_h^{\mathrm{T}}\mathrm{T}_n^{-1}\mathbf{e}_1} \\
&= \frac{\mathbf{t}^{(1)}\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}}{1+\mathbf{v}_h^{\mathrm{T}}\mathbf{t}^{(1)}}.
\end{aligned}$$

Moreover:

$$\begin{aligned}
t_k^{(1)} &= \frac{n+1-k}{n+1} = 1 - \frac{k}{n+1}, \quad k = 1, \ldots, n, \\
\mathbf{v}_h^{\mathrm{T}} &= \begin{bmatrix} \theta_{\mathrm{a}}h^2 - 2 & (1-\theta_{\mathrm{a}})h^2 + 1 & 0 & \ldots & 0 \end{bmatrix} \\
&= \begin{bmatrix} v_1 & v_2 & 0 & \ldots & 0 \end{bmatrix}, \\
\mathbf{v}_h^{\mathrm{T}}\mathbf{t}^{(1)} &= v_1 t_1^{(1)} + v_2 t_2^{(1)} \\
&= (\theta_{\mathrm{a}}h^2 - 2)\left(1 - \frac{1}{n+1}\right) + \left((1-\theta_{\mathrm{a}})h^2 + 1\right)\left(1 - \frac{2}{n+1}\right) \\
&= \theta_{\mathrm{a}}h^2 - 2 - \frac{\theta_{\mathrm{a}}h^2}{n+1} + \frac{2}{n+1} + (1-\theta_{\mathrm{a}})h^2 + 1 \\
&\quad - \frac{2}{n+1}(1-\theta_{\mathrm{a}})h^2 - \frac{2}{n+1} \\
&= (-2+1) + (\theta_{\mathrm{a}} + 1 - \theta_{\mathrm{a}})h^2 + (-\theta_{\mathrm{a}} - 2 + 2\theta_{\mathrm{a}})\frac{h^2}{n+1} \\
&= (\theta_{\mathrm{a}} - 2)\frac{h^2}{n+1} + h^2 - 1, \\
1 + \mathbf{v}_h^{\mathrm{T}}\mathbf{t}^{(1)} &= 1 + v_1 t_1^{(1)} + v_2 t_2^{(1)} \\
&= (\theta_{\mathrm{a}} - 2)\frac{h^2}{n+1} + h^2 \\
&= \left(\frac{\theta_{\mathrm{a}} - 2}{n+1} + 1\right)h^2,
\end{aligned}$$

$$
\begin{aligned}
\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1} &= h^2\mathbf{v}_h^{\mathrm{T}}\mathrm{T}_n^{-1} = h^2\mathbf{z}_h^{\mathrm{T}},\\
\mathbf{z}_h^{\mathrm{T}} &= [v_1t_1^{(1)} + v_2t_2^{(1)} \quad v_1t_1^{(2)} + v_2t_2^{(2)} \quad \dots\\
&\qquad \dots \quad v_1t_1^{(r)} + v_2t_2^{(r)} \quad \dots \quad v_1t_1^{(n)} + v_2t_2^{(n)}],\\
\left(\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}\right)_r &= h^2\left(v_1t_1^{(r)} + v_2t_2^{(r)}\right),\\
\left(\mathbf{t}^{(1)}\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}\right)_{k,r} &= t_k^{(1)}h^2\left(v_1t_1^{(r)} + v_2t_2^{(r)}\right).
\end{aligned}
$$

Being:

$$
\begin{aligned}
t_k^{(1)} &= 1 - \frac{k}{n+1}, \quad k = 1,\dots,n,\ r = 1,\\
t_1^{(r)} &= 1 - \frac{r}{n+1}, \quad k = 1,\ r = 1,\dots,n,\\
t_2^{(1)} &= 1 - \frac{2}{n+1}, \quad k = 2,\ r = 1,\\
t_2^{(r)} &= 2 - 2\frac{r}{n+1}, \quad k = 2,\ r = 2,\dots,n,
\end{aligned}
$$

we obtain, for $r = 1$:

$$
\begin{aligned}
&\left(\mathbf{t}^{(1)}\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}\right)_{k,1} = t_k^{(1)}h^2\left(v_1t_1^{(1)} + v_2t_2^{(1)}\right) = t_k^{(1)}\mathbf{v}_h^{\mathrm{T}}\mathbf{t}^{(1)}\\
&= \left(1 - \frac{k}{n+1}\right)h^2\left[(\theta_{\mathrm{a}}h^2 - 2)\left(1 - \frac{1}{n+1}\right) + \left((1-\theta_{\mathrm{a}})h^2 + 1\right)\left(1 - \frac{2}{n+1}\right)\right]\\
&= \left(1 - \frac{k}{n+1}\right)h^2\left((\theta_{\mathrm{a}} - 2)\frac{h^2}{n+1} + h^2 - 1\right),
\end{aligned}
$$

and for $r > 1$:

$$
\begin{aligned}
&\left(\mathbf{t}^{(1)}\mathbf{v}_h^{\mathrm{T}}\mathrm{S}_n^{-1}\right)_{k,r} = t_k^{(1)}h^2\left(v_1t_1^{(r)} + v_2t_2^{(r)}\right)\\
&= \left(1 - \frac{k}{n+1}\right)h^2\left[(\theta_{\mathrm{a}}h^2 - 2)\left(1 - \frac{r}{n+1}\right) + \left((1-\theta_{\mathrm{a}})h^2 + 1\right)\left(2 - \frac{2r}{n+1}\right)\right]\\
&= -\left(1 - \frac{k}{n+1}\right)\left(1 - \frac{r}{n+1}\right)h^2(\theta_{\mathrm{a}} - 2)\\
&= -\left(1 - \frac{k}{n+1}\right)h^2(\theta_{\mathrm{a}} - 2)\frac{n+1-r}{n}h^2\left(1 - \frac{1}{n+1}\right).
\end{aligned}
$$

Thus:

$$
\begin{aligned}
\left(\mathrm{R}_n\right)_{k,1} &= \frac{\left(1-\frac{k}{n+1}\right)h^2\left((\theta_{\mathrm{a}}-2)\frac{h^2}{n+1}+h^2-1\right)}{\left(\frac{\theta_{\mathrm{a}}-2}{n+1}+1\right)h^2} && r=1\\
&= \frac{\left(1-\frac{k}{n+1}\right)\left((\theta_{\mathrm{a}}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1}\\
&= \left(1-\frac{k}{n+1}\right)\left(h^2-\frac{1}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1}\right),\\
\left(\mathrm{R}_n\right)_{k,r} &= -\frac{h^2\left(1-\frac{k}{n+1}\right)(\theta_{\mathrm{a}}-2)\left(1-\frac{1}{n+1}\right)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1}\frac{n+1-r}{n} && r>1\\
&= -\frac{h^2\left(1-\frac{k}{n+1}\right)(\theta_{\mathrm{a}}-2)\left(1-\frac{1}{n+1}\right)(n+1)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1}\frac{\left(1-\frac{r}{n+1}\right)}{n}.
\end{aligned}
$$

Numerically it is obvious that $\|\mathrm{R}_n\|_1$ and $\|\mathrm{R}_n\|_\infty$ are always for the first column and first row.

Now we compute $\|\mathrm{R}_n\|_1$.
We have:

$$
\begin{aligned}
\|\mathrm{R}_n\|_1 &= \sum_{k=1}^{n}\left|\frac{\left(1-\frac{k}{n+1}\right)\left((\theta_{\mathrm{a}}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1}\right|\\
&= \left|\frac{\left((\theta_{\mathrm{a}}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1}\right|\sum_{k=1}^{n}\left(1-\frac{k}{n+1}\right)\\
&= -\frac{\left((\theta_{\mathrm{a}}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1}\left(n-\frac{n(n+1)}{2(n+1)}\right)\\
&= -\frac{n}{2}\frac{\left((\theta_{\mathrm{a}}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1},
\end{aligned}
$$

because of

$$\begin{aligned} \frac{n}{2} &> 0, \\ (\theta_{\mathrm{a}}-2)\frac{h^2}{n+1} &< 0, \\ h^2-1 &< 0, \\ \frac{\theta_{\mathrm{a}}-2}{n+1}+1 &> 0. \end{aligned}$$

Now we can compute $\|\mathrm{A}_h^{-1}\|_1$.
Numerically it is obvious that $\|\mathrm{A}_h^{-1}\|_1$ is computed on the first column. Being $\mathrm{A}_h^{-1} = \mathrm{S}_n^{-1} - \mathrm{R}_n$ and since the first column of S_n^{-1} has all positive elements and the the first column of R_n has all negative elements, indeed, they are respectively:

$$(\mathrm{S}_n^{-1})_{k,1} = \left(1 - \frac{k}{n+1}\right) > 0, \quad k = 1, \ldots, n,$$

$$(\mathrm{R}_n)_{k,1} = \frac{\left(1-\frac{k}{n+1}\right)\left((\theta_{\mathrm{a}}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1} < 0,$$

we can compute the norm directly for A_h^{-1}.
The sum of the positive elements of the first column of T_n^{-1} is equal to $\frac{n}{2}$, and thus the sum for the first column of S_n^{-1} is $\frac{h^2 n}{2}$, therefore we have:

$$\begin{aligned} \|\mathrm{A}_h^{-1}\|_1 &= \frac{h^2 n}{2} - \frac{n}{2}\frac{\left((\theta_{\mathrm{a}}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1} \\ &= \frac{n}{2}\left(h^2 - \frac{\left((\theta_{\mathrm{a}}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1}\right) \\ &= \frac{n}{2}\frac{1}{\frac{\theta_{\mathrm{a}}-2}{n+1}+1} = \frac{n(n+1)}{2(n+\theta_{\mathrm{a}}-1)} \sim \mathrm{O}(n). \end{aligned}$$

We now compute $\|\mathrm{R}_n\|_\infty$, by taking into consideration that all coefficients are positive except the first in the first column.

We have:

$$
\begin{aligned}
(\mathrm{R}_n)_{1,1} &= \frac{\left(1-\frac{1}{n+1}\right)\left((\theta_\mathrm{a}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_\mathrm{a}-2}{n+1}+1},\\
(\mathrm{R}_n)_{1,r} &= \frac{\left(1-\frac{1}{n+1}\right)\left((\theta_\mathrm{a}-2)\frac{h^2}{n+1}-h^2\theta_\mathrm{a}+2h^2\right)}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\frac{n+1-r}{n},\\
\|\mathrm{R}_n\|_\infty &= -\frac{\left(1-\frac{1}{n+1}\right)\left((\theta_\mathrm{a}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\\
&\quad+\frac{\left(1-\frac{1}{n+1}\right)\left((\theta_\mathrm{a}-2)\frac{h^2}{n+1}-h^2\theta_\mathrm{a}+2h^2\right)}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\sum_{r=2}^{n}\frac{n+1-r}{n}\\
&= -\frac{\left(1-\frac{1}{n+1}\right)\left((\theta_\mathrm{a}-2)\frac{h^2}{n+1}+h^2-1\right)}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\\
&\quad+\frac{\left(1-\frac{1}{n+1}\right)\left((\theta_\mathrm{a}-2)\frac{h^2}{n+1}-h^2\theta_\mathrm{a}+2h^2\right)}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\left[\frac{(n+1)(n-1)}{n}-\frac{1}{n}\left(\frac{n(n+1)}{2}-1\right)\right]\\
&= -\left(1-\frac{1}{n+1}\right)h^2+\frac{1-\frac{1}{n+1}}{\frac{\theta_\mathrm{a}-2}{n+1}+1}+\frac{\left(1-\frac{1}{n+1}\right)\left((\theta_\mathrm{a}-2)\frac{h^2}{n+1}-h^2\theta_\mathrm{a}+2h^2\right)}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\left(\frac{n-1}{2}\right)\\
&= -\left(1-\frac{1}{n+1}\right)h^2+\frac{1-\frac{1}{n+1}}{\frac{\theta_\mathrm{a}-2}{n+1}+1}+\frac{\left(1-\frac{1}{n+1}\right)\left((\theta_\mathrm{a}-2)\frac{h^2}{n+1}+h^2\right)}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\left(\frac{n-1}{2}\right)\\
&\quad+\frac{\left(1-\frac{1}{n+1}\right)h^2(1-\theta_\mathrm{a})}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\left(\frac{n-1}{2}\right)\\
&= \frac{1-\frac{1}{n+1}}{\frac{\theta_\mathrm{a}-2}{n+1}+1}+\left(1-\frac{1}{n+1}\right)h^2\left(\frac{n-3}{2}\right)+\frac{\left(1-\frac{1}{n+1}\right)h^2(1-\theta_\mathrm{a})}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\left(\frac{n-1}{2}\right)\\
&= \left(1-\frac{1}{n+1}\right)\left[h^2\left(\frac{n-3}{2}\right)+\frac{2+h^2(1-\theta_\mathrm{a})(n-1)}{2\left(\frac{\theta_\mathrm{a}-2}{n+1}+1\right)}\right]\\
&= \left(1-\frac{1}{n+1}\right)\left[\frac{h^2(n-3)\left(\frac{\theta_\mathrm{a}-2}{n+1}+1\right)}{2\left(\frac{\theta_\mathrm{a}-2}{n+1}+1\right)}+\frac{2+h^2(1-\theta_\mathrm{a})(n-1)}{2\left(\frac{\theta_\mathrm{a}-2}{n+1}+1\right)}\right]\\
&= \frac{1-\frac{1}{n+1}}{2\left(\frac{\theta_\mathrm{a}-2}{n+1}+1\right)}\left[h^2(n-3)\left(\frac{\theta_\mathrm{a}-2}{n+1}+1\right)+2+h^2(1-\theta_\mathrm{a})(n-1)\right].
\end{aligned}
$$

Now we can compute $\|\mathrm{A}_h^{-1}\|_\infty$.
We know $\mathrm{A}_h^{-1} = \mathrm{S}_n^{-1} - \mathrm{R}_n$. Being:

$$
\begin{aligned}
t_1^{(r)} &= 1 - \frac{r}{n+1}, \quad r = 1, \dots, n, \\
\left(\mathrm{S}_n^{-1}\right)_{1,r} &= h^2 t_1^{(r)} = h^2 \left(1 - \frac{r}{n+1}\right), \\
(\mathrm{R}_n)_{1,1} &= \left(1 - \frac{1}{n+1}\right)\left(h^2 - \frac{1}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\right), \quad r = 1, \\
(\mathrm{R}_n)_{1,r} &= -\frac{h^2(\theta_\mathrm{a}-2)\left(1-\frac{1}{n+1}\right)^2}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\frac{\left(1-\frac{r}{n+1}\right)}{n}(n+1), \quad r = 2, \dots, n,
\end{aligned}
$$

we have:

$$
\begin{aligned}
\left(\mathrm{A}_h^{-1}\right)_{1,1} &= h^2 t_1^{(1)} - (\mathrm{R}_n)_{1,1} \\
&= h^2\left(1 - \frac{1}{n+1}\right) - \left(1 - \frac{1}{n+1}\right)\left(h^2 - \frac{1}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\right) \\
&= \frac{1-\frac{1}{n+1}}{\frac{\theta_\mathrm{a}-2}{n+1}+1},
\end{aligned}
$$

$$
\begin{aligned}
\left(\mathrm{A}_h^{-1}\right)_{1,r} &= h^2 t_r^{(1)} - (\mathrm{R}_n)_{1,r} \\
&= h^2\left(1 - \frac{r}{n+1}\right) + \frac{h^2(n+1)(\theta_\mathrm{a}-2)\left(1-\frac{1}{n+1}\right)^2}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\frac{\left(1-\frac{r}{n+1}\right)}{n},
\end{aligned}
$$

from which:

$$
\begin{aligned}
\|\mathrm{A}_h^{-1}\|_\infty &= \left(\mathrm{A}_h^{-1}\right)_{1,1} - \sum_{r=2}^{n}\left(\mathrm{A}_h^{-1}\right)_{1,r} \\
&= \frac{1-\frac{1}{n+1}}{\frac{\theta_\mathrm{a}-2}{n+1}+1} - \sum_{r=2}^{n}\left[h^2\left(1-\frac{r}{n+1}\right) + \frac{h^2(n+1)(\theta_\mathrm{a}-2)\left(1-\frac{1}{n+1}\right)^2}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\frac{\left(1-\frac{r}{n+1}\right)}{n}\right] \\
&= \frac{1-\frac{1}{n+1}}{\frac{\theta_\mathrm{a}-2}{n+1}+1} - \sum_{r=2}^{n} h^2\left(1-\frac{r}{n+1}\right)\left(1 + \frac{(\theta_\mathrm{a}-2)\left(1-\frac{1}{n+1}\right)}{\frac{\theta_\mathrm{a}-2}{n+1}+1}\right) \\
&= \frac{1-\frac{1}{n+1}}{\frac{\theta_\mathrm{a}-2}{n+1}+1} - \sum_{r=2}^{n} h^2\left(1-\frac{r}{n+1}\right)\frac{\theta_\mathrm{a}-1}{\frac{\theta_\mathrm{a}-2}{n+1}+1}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1-\frac{1}{n+1}}{\frac{\theta_a-2}{n+1}+1} - h^2(n-1)\frac{\theta_a-1}{\frac{\theta_a-2}{n+1}+1} + \sum_{r=2}^{n} r\frac{h^2}{n+1}\frac{\theta_a-1}{\frac{\theta_a-2}{n+1}+1} \\
&= \frac{1-\frac{1}{n+1}}{\frac{\theta_a-2}{n+1}+1} - h^2(n-1)\frac{\theta_a-1}{\frac{\theta_a-2}{n+1}+1} + \left(\frac{n(n+1)}{2}-1\right)\frac{h^2}{n+1}\frac{\theta_a-1}{\frac{\theta_a-2}{n+1}+1} \\
&= \frac{1}{\frac{\theta_a-2}{n+1}+1}\left(1-\frac{1}{n+1} - h^2(n-1)(\theta_a-1) + \frac{\left(nh^2-2\frac{h^2}{n+1}\right)(\theta_a-1)}{2}\right) \\
&= \frac{1}{\frac{\theta_a-2}{n+1}+1}\left(1-\frac{1}{n+1} + \frac{\left(2-n-\frac{2}{n+1}\right)h^2(\theta_a-1)}{2}\right) \\
&= \frac{1}{2\left(\frac{\theta_a-2}{n+1}+1\right)}\left\{2\left(1-\frac{1}{n+1}\right) + \left[2\left(1-\frac{1}{n+1}\right)-n\right]h^2(\theta_a-1)\right\} \\
&= \frac{\frac{n}{n+1}}{2\left(\frac{\theta_a-2}{n+1}+1\right)}\left(2+(2-n-1)h^2(\theta_a-1)\right) \sim \mathrm{O}(n^0).
\end{aligned}
$$

Finally we have:

$$
\begin{aligned}
\|\mathrm{A}_h^{-1}\|_2 &\leq \sqrt{\frac{n}{2\left(\frac{\theta_a-2}{n+1}+1\right)}\frac{1-\frac{1}{n+1}}{2\left(\frac{\theta_a-2}{n+1}+1\right)}\left(2+(2-n-1)h^2(\theta_a-1)\right)} \\
&= \frac{n}{2\left(\frac{\theta_a-2}{n+1}+1\right)}\sqrt{\frac{1}{n+1}\left(2+(2-n-1)h^2(\theta_a-1)\right)} \\
&= \frac{n}{2(\theta_a-1+n)}\sqrt{2(n+1)+(2-n-1)(n+1)h^2(\theta_a-1)} \sim \mathrm{O}(n^{\frac{1}{2}}).
\end{aligned}
$$

Remark 3.1: The estimates for $\|\mathrm{A}_h^{-1}\|_p$ are tight and the growth is like $n^{\frac{1}{p}}$. Thus, if we refer to the numerical results on the norm of the inverse matrix, we show a stability only in the infinite norm. However the numerical growth of the error seems to be bounded by a constant independently of p. This is because there are vectors, in our case e_h, which guarantee convergence despite $\|\mathrm{A}_h^{-1}\|_p$ grows.

Remark 3.2: These analytical results confirm the numerical results seen in the previous Chapter.

3.3 Convergence for the Dirichlet problem in 1D

We prove that the method is convergent and we determine that the numerical solution *converges with second-order accuracy* to the exact solution.

We always consider the case in which the domain is [a, 1] and we perform the linear interpolation to impose the boundary condition in **a** (Problem (1.8)). We provide two different convergence proof.

First proof

This proof is due to Lisl Weynans [1] and it is based on the following steps:

1. Prove that the matrix A_h is "almost" monotonic. That is, we want to prove that all coefficients of all the rows of A_h^{-1}, except the first row possibly, are positive.

2. Deduce from this "almost" monotonicity property a variant of a discrete maximum principle, which will lead us to obtain estimates of the coefficients of A_h^{-1}.

Remark 3.3: We cannot simply prove that A_h is monotonic, because the point x_0 is located outside of the domain Ω where the elliptic problem is defined, and thus no maximum principle related to this elliptic problem can be applied on x_0.

1. A_h is "almost" monotonic

We consider an array V such that $A_h V \geq 0$, coefficient by coefficient. We define the index i_0 such that V_{i_0} is the minimum of all coefficients V_i with $1 \leq i \leq n-1$ ($V_{i_0} = \min_{i=1,\dots,n-1} V_i$).

- If $2 \leq i_0 \leq n-2$ (*indices relative at internal points that have internal neighbors*):
  We have:
  $$\frac{-V_{i_0-1} + 2V_{i_0} - V_{i_0+1}}{h^2} \geq 0 \Longrightarrow V_{i_0-1} + V_{i_0+1} \geq 2V_{i_0} \geq V_{i_0-1} + V_{i_0+1}.$$

[1] IMB , INRIA Bordeaux Sud-Ouest, Bordeaux University, 351 cours de la Liberation, 33405 Talence

This is possible only if $V_{i_0-1} = V_{i_0} = V_{i_0+1}$. We apply the same reasoning on V_{i_0-1}, V_{i_0+1} and their successive neighbours and conclude that all V_i are equal. It leads us to the next case.

- If $i_0 = n-1$:
  We have:
  $$\frac{-V_{n-2} + 2V_{n-1}}{h^2} \geq 0 \Longrightarrow 2V_{n-2} \geq 2V_{n-1} \geq V_{n-2}.$$
  This is possible only if $V_{n-2} \geq 0$ and consequently $V_{n-1} \geq 0$.

- If $i_0 = 1$:
  We have:
  $$\frac{-V_0 + 2V_1 - V_2}{h^2} \geq 0 \Longrightarrow 2V_1 \geq V_0 + V_2, \tag{3.1}$$
  $$\theta_a V_0 + (1-\theta_a)V_1 \geq 0 \Longrightarrow V_0 \geq -\frac{1-\theta_a}{\theta_a} V_1. \tag{3.2}$$
  If $V_1 \leq 0$ from (3.2) we have $V_0 \geq 0$. Being $V_0 \geq V_1$, from (3.1) we have $2V_1 \geq V_1 + V_2$, so $V_1 \geq V_2$, which is not possible.

We conclude that $V_{i_0} \geq 0$, and consequently, all coefficients of V, except maybe V_0, are positive.
It means that all coefficients of all the rows of A_h^{-1}, except possibly the first row, are positive.

2. Discrete maximum principle

We denote by G^i the $i-th$ column of A_h^{-1} for $0 \leq i \leq n-1$. It satisfies:

$$A_h G^i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

where the element 1 is in $i-th$ location.

First, we define the array W such that $\mathrm{W}_j = \frac{y_j(1-y_j)}{2}$, for all j, with $y_j = \frac{x_j - x_0}{x_n - x_0}$. We can observe that $\mathrm{W} \geq 0$ and it satisfies:

$$\theta_\mathrm{a}\mathrm{W}_0 + (1-\theta_\mathrm{a})\mathrm{W}_1 \geq 0,$$

$$\frac{-\mathrm{W}_{j+1} + 2\mathrm{W}_j - \mathrm{W}_{j-1}}{h^2} = 1,\ 1 \leq j \leq n-2,$$

$$\frac{2\mathrm{W}_{n-1} - \mathrm{W}_{n-2}}{h^2} = 1.$$

Therefore:

$$\mathrm{A}_h\mathrm{W} \geq \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \mathrm{A}_h \left(\sum_{i=1}^{n-1} \mathrm{G}^i \right).$$

We have proven before that all the rows of A_h^{-1}, except the first one, contain only positive coefficients. Consequently, when we multiply the previous inequality by A_h^{-1}, we obtain for each row of the resulting array, except W_0:

$$\mathrm{W}_j \geq \sum_{i=1}^{n-1} \mathrm{G}_j^i,\ 1 \leq j \leq n-1,$$

$$\sum_{i=1}^{n-1} \mathrm{G}_j^i \leq \max_{1 \leq k \leq n-1} \mathrm{W}_k \leq \frac{1}{8},\ 1 \leq j \leq n-1.$$

To calculate $\max_{1 \leq k \leq n-1} \mathrm{W}_k$:
$\frac{\partial \mathrm{W}_k}{\partial y_j} = \frac{1-2y_j}{2} = 0 \Longrightarrow y_j = \frac{1}{2}$ so $\max_{1 \leq k \leq n-1} \mathrm{W}_k = \frac{1}{8}$.

We define the local error $e_i = u(x_i) - u_i$. It satisfies the same linear system as the numerical solution u_h, with the truncation error τ as a source term:

$$\mathrm{A}_h e = \tau,$$

and therefore:

$$e = \mathrm{A}_h^{-1}\tau.$$

If $1 \leq i \leq n-1$:

$$|e_j| = |u(x_j) - u_j| = \left| \sum_{i=0}^{n-1} \mathrm{G}_j^i \tau_j \right| \leq \sum_{i=1}^{n-1} \mathrm{G}_j^i \, |\tau_j| \leq \sum_{i=1}^{n-1} \mathrm{G}_j^i (\max_j |\tau_j|) \ \leq \frac{1}{8}\mathrm{O}(h^2).$$

Therefore, the numerical solution converges with second-order accuracy to the exact solution.

Second proof

Based on the previous proof, we provide an alternative proof of point **1** of the proof proposed by Lisl Weynans. This proof is based on the use of M-matrices.

We consider an array V such that $\mathrm{A}_h\mathrm{V} \geq 0$, we want to prove that $\mathrm{V} \geq 0$. We write the linear system in this way:

$$\mathrm{A}_h\mathrm{V} = \begin{pmatrix} \mathrm{A}_{\mathrm{gg}} & \mathrm{A}_{\mathrm{gi}} \\ \mathrm{A}_{\mathrm{ig}} & \mathrm{A}_{\mathrm{ii}} \end{pmatrix} \begin{pmatrix} \mathrm{V}_{\mathrm{g}} \\ \mathrm{V}_{\mathrm{i}} \end{pmatrix} = \begin{pmatrix} z_{\mathrm{g}} \\ z_{\mathrm{i}} \end{pmatrix} \geq 0,$$

where V_{g} and V_{i} are the vectors of the ghost points and of the internal points respectively. We have:

$$\mathrm{V}_{\mathrm{i}} = \begin{pmatrix} \mathrm{V}_1 \\ \vdots \\ \vdots \\ \mathrm{V}_{n-1} \end{pmatrix} \in \mathbb{R}^{n-1}, \ \mathrm{V}_{\mathrm{g}} = \mathrm{V}_0 \in \mathbb{R}, \ \mathrm{A}_{\mathrm{ig}} = \begin{pmatrix} -\frac{1}{h^2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{n-1},$$

$$\mathrm{A}_{\mathrm{gg}} = \theta_{\mathrm{a}} \in \mathbb{R}, \ \mathrm{A}_{\mathrm{gi}} = \begin{pmatrix} 1-\theta_{\mathrm{a}} & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{1\times(n-1)},$$

$$\mathrm{A}_{\mathrm{ii}} = \begin{pmatrix} \frac{2}{h^2} & -\frac{1}{h^2} & & & \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & \\ & -\frac{1}{h^2} & \frac{2}{h^2} & \ddots & \\ & & \ddots & \ddots & -\frac{1}{h^2} \\ & & & -\frac{1}{h^2} & \frac{2}{h^2} \end{pmatrix} \in \mathbb{R}^{(n-1)\times(n-1)}.$$

Therefore:

$$\mathrm{A}_h\mathrm{V} = \left(\begin{array}{c|cccc} \theta_{\mathrm{a}} & 1-\theta_{\mathrm{a}} & & & \\ \hline -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} \\ & & & -\frac{1}{h^2} & \frac{2}{h^2} \end{array}\right) \left(\begin{array}{c} \mathrm{V}_0 \\ \hline \mathrm{V}_1 \\ \vdots \\ \vdots \\ \mathrm{V}_{n-1} \end{array}\right) \geq 0.$$

Being:

$$\mathrm{A_{ii}V_i} + \mathrm{A_{ig}V_g} = z_\mathrm{i} \geq 0, \tag{3.3}$$

$$\mathrm{A_{gi}V_i} + \mathrm{A_{gg}V_g} = z_\mathrm{g} \geq 0 \Longrightarrow \mathrm{V_g} = \mathrm{A_{gg}^{-1}}(z_\mathrm{g} - \mathrm{A_{gi}V_i}), \tag{3.4}$$

replacing (3.4) in (3.3) we have

$$z_\mathrm{i} = (\mathrm{A_{ii}} - \mathrm{A_{ig}A_{gg}^{-1}A_{gi}})\mathrm{V_i} + \mathrm{A_{ig}A_{gg}^{-1}}z_\mathrm{g} \geq 0.$$

Since $\mathrm{A_{ig}A_{gg}^{-1}}z_\mathrm{g} = \begin{pmatrix} -\frac{z_\mathrm{g}}{\theta_\mathrm{a}h^2} & 0 & \cdots & 0 \end{pmatrix}^\mathrm{T}$ and $-\frac{z_\mathrm{g}}{\theta_\mathrm{a}h^2} \leq 0$, we have

$$(\mathrm{A_{ii}} - \mathrm{A_{ig}A_{gg}^{-1}A_{gi}})\mathrm{V_i} \geq 0,$$

that is:

$$\begin{pmatrix} \frac{1}{h^2}\left(2 + \frac{1-\theta_\mathrm{a}}{\theta_\mathrm{a}}\right) & -\frac{1}{h^2} & & & \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & \\ & -\frac{1}{h^2} & \frac{2}{h^2} & \ddots & \\ & & \ddots & \ddots & -\frac{1}{h^2} \\ & & & -\frac{1}{h^2} & \frac{2}{h^2} \end{pmatrix} \begin{pmatrix} \mathrm{V}_1 \\ \vdots \\ \vdots \\ \mathrm{V}_{n-2} \\ \mathrm{V}_{n-1} \end{pmatrix} \geq 0.$$

We denote this matrix with A and we observe that it is a tridiagonal, symmetric and strictly diagonally dominant matrix.
We put $\mathrm{W_i} = \mathrm{AV_i} \geq 0$, whence $\mathrm{V_i} = \mathrm{A^{-1}W_i}$, $\mathrm{i} = 2, \ldots, n-1$. Therefore, if we prove that A is an invertible matrix and $\mathrm{A^{-1}} \geq 0$, we have the thesis ($\mathrm{V_i} \geq 0$).

To prove this we use the following theorem:

Theorem 1 *Let A be a matrix $\in \mathbb{R}^{n\times n}$ with positive diagonal elements and let $D = diag(a_{11}, \ldots a_{nn})$. If $B = D - A \geq 0$ and $\rho(D^{-1}B) < 1$ then exists A^{-1} and $A^{-1} \geq 0$.*

We apply this result to our matrix A:

1. $a_{11} = 2 + \frac{1-\theta_\mathrm{a}}{\theta_\mathrm{a}} = \frac{\theta_\mathrm{a}+1}{\theta_\mathrm{a}} > 0$,
   $a_{ii} = \frac{2}{h^2} > 0$, $i \neq 1$;

2. $$\mathrm{D} = \begin{pmatrix} \frac{1}{h^2}\left(2 + \frac{1-\theta_\mathrm{a}}{\theta_\mathrm{a}}\right) & & & \\ & \frac{2}{h^2} & & \\ & & \ddots & \\ & & & \frac{2}{h^2} \end{pmatrix},$$

$$\mathrm{B} = \mathrm{D} - \mathrm{A} = \begin{pmatrix} 0 & \frac{1}{h^2} & & & \\ \frac{1}{h^2} & 0 & \frac{1}{h^2} & & \\ & \frac{1}{h^2} & 0 & \ddots & \\ & & \ddots & \ddots & \frac{1}{h^2} \\ & & & \frac{1}{h^2} & 0 \end{pmatrix} \geq 0;$$

3.

$$\mathrm{D}^{-1}\mathrm{B} = \begin{pmatrix} 0 & \frac{\theta_\mathrm{a}}{\theta_\mathrm{a}+1} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ & \frac{1}{2} & 0 & \ddots & \\ & & \ddots & \ddots & \frac{1}{2} \\ & & & \frac{1}{2} & 0 \end{pmatrix}.$$

For the First and the Third Gershgorin's Theorem we are sure that $\rho(\mathrm{D}^{-1}\mathrm{B}) < 1$. Indeed for the first theorem the eigenvalues of the matrix $\mathrm{D}^{-1}\mathrm{B}$ are located in the region of the complex plane identified by the intersection between the union of the row circles (which are circles centered in the origin and radius 1, $\frac{1}{2}$ and $\frac{\theta_\mathrm{a}}{\theta_\mathrm{a}+1} + \frac{1}{2}$) and the union of the column circles (which are circles centered in the origin and radius 1, $\frac{1}{2}$ and $\frac{\theta_\mathrm{a}}{\theta_\mathrm{a}+1}$), with $\frac{\theta_\mathrm{a}}{\theta_\mathrm{a}+1} < \frac{1}{2}$. Moreover, for the third Gershgorin theorem, the eigenvalues can not be on the border, thus $\rho(\mathrm{D}^{-1}\mathrm{B}) < 1$.

Therefore, we can conclude that $\mathrm{A}^{-1} \geq 0$, whence $\mathrm{V}_i \geq 0$.

3.4 Consistency for the mixed problem in 1D

Now, we consider the mixed problems and we prove that the method is consistent.

The consistent in the internal points and in the ghost point x_0 has already been described in the Section 3.1. Now we calculate the consistency error in the ghost point x_n.
In the case in which we perform the linear interpolation procedure to impose the boundary condition in $\mathbf{b}$, if $u \in \mathrm{C}^2([x_{n-1}, x_n])$:

$$\tau_n = \tau(x_n) = \frac{u(x_n) - u(x_{n-1})}{h} - u'(\mathrm{b}) = hu''(\xi)(1 - 2\theta_\mathrm{b}) \ \sim \mathrm{O}(h), \quad (3.5)$$

for some $\xi \in [x_{n-1}, x_n]$.

The formula (3.5) derived from:

$$u(x_n) = u(\mathbf{b}) + u'(\mathrm{b})(x_n - \mathbf{b}) + \frac{u''(\xi^+)}{2}(x_n - \mathbf{b})^2, \quad (3.6)$$

for some $\xi^+ \in [\mathbf{b}, x_n]$, and

$$u(x_{n-1}) = u(\mathbf{b}) + u'(\mathbf{b})(x_{n-1} - \mathbf{b}) + \frac{u''(\xi^-)}{2}(x_{n-1} - \mathbf{b})^2,$$

for some $\xi^- \in [x_{n-1}, \mathbf{b}]$, thus

$$u(x_{n-1}) = u(\mathbf{b}) - u'(\mathbf{b})(\mathbf{b} - x_{n-1}) + \frac{u''(\xi^-)}{2}(\mathbf{b} - x_{n-1})^2 \quad (3.7)$$

$$= u(\mathbf{b}) - u'(\mathbf{b})[h - (x_n - \mathbf{b})] + \frac{u''(\xi^-)}{2}(\mathbf{b} - x_{n-1})^2. \quad (3.8)$$

Subtracting member to member (3.6) and (3.8) we obtain for some $\xi \in [x_{n-1}, x_n]$:

$$\begin{aligned} u(x_n) - u(x_{n-1}) &= hu'(\mathbf{b}) + u''(\xi)\left[h^2(1 - \theta_\mathrm{b})^2 - h^2\theta_\mathrm{b}^2\right] \\ &= hu'(\mathbf{b}) + h^2u''(\xi)(1 - 2\theta_\mathrm{b}). \end{aligned}$$

With the symmetry of the matrix A_h (through the formula (1.18)) in the

case in which we impose the Neumann condition in $\mathbf{b}$ through the linear interpolation procedure, we have:

$$\tau_n = \tau(x_n) = u''(\xi)(1 - 2\theta_\mathrm{b}) \ \sim \mathrm{O}(h^0),$$

for some $\xi \in [x_{n-1}, x_n]$.

Now, we calculate the consistency error τ_n in the case in which we perform the quadratic interpolation procedure to impose the boundary condition in $\mathbf{b}$.
If $u \in \mathrm{C}^3([x_{n-2}, x_n])$:

$$\begin{aligned} u(x_{n-2}) &= u(\mathbf{b}) + u'(\mathbf{b})(x_{n-2} - \mathbf{b}) + \frac{u''(\mathbf{b})}{2}(x_{n-2} - \mathbf{b})^2 + \frac{u'''(\xi^-)}{6}(x_{n-2} - \mathbf{b})^3 \\ &= u(\mathbf{b}) - u'(\mathbf{b})(1 + \theta_\mathrm{b})h + \frac{u''(\mathbf{b})}{2}(1 + \theta_\mathrm{b})^2h^2 - \frac{u'''(\xi^-)}{6}(1 + \theta_\mathrm{b})^3h^3, \end{aligned}$$

for some $\xi^- \in [x_{n-2}, \mathbf{b}]$,

$$\begin{aligned} u(x_{n-1}) &= u(\mathbf{b}) + u'(\mathbf{b})(x_{n-1} - \mathbf{b}) + \frac{u''(\mathbf{b})}{2}(x_{n-1} - \mathbf{b})^2 + \frac{u'''(\xi^+)}{6}(x_{n-1} - \mathbf{b})^3 \\ &= u(\mathbf{b}) - u'(\mathbf{b})\theta_\mathrm{b}h + \frac{u''(\mathbf{b})}{2}\theta_\mathrm{b}^2h^2 - \frac{u'''(\xi^+)}{6}\theta_\mathrm{b}^3h^3, \end{aligned}$$

for some $\xi^+ \in [x_{n-1}, \mathbf{b}]$, and

$$\begin{aligned} u(x_n) &= u(\mathbf{b}) + u'(\mathbf{b})(x_n - \mathbf{b}) + \frac{u''(\mathbf{b})}{2}(x_n - \mathbf{b})^2 + \frac{u'''(\xi)}{6}(x_n - \mathbf{b})^3 \\ &= u(\mathbf{b}) + u'(\mathbf{b})(1 - \theta_\mathrm{b})h + \frac{u''(\mathbf{b})}{2}(1 - \theta_\mathrm{b})^2h^2 + \frac{u'''(\xi)}{6}(1 - \theta_\mathrm{b})^3h^3, \end{aligned}$$

for some $\xi \in [\mathbf{b}, x_n]$.

Thus we have:

$$\begin{aligned} \tau_n &= \frac{1}{2h}(2\theta_\mathrm{b} - 1)u(x_{n-2}) - \frac{2}{h}\theta_\mathrm{b}u(x_{n-1}) + \frac{1}{2h}(2\theta_\mathrm{b} + 1)u(x_n) - u'(\mathbf{b}) \\ &= -\frac{u'''(\xi)}{12}(2\theta_\mathrm{b} - 1)(1 + \theta_\mathrm{b})^3h^2 + \frac{u'''(\xi)}{3}\theta_\mathrm{b}^4h^2 + \frac{u'''(\xi)}{12}(2\theta_\mathrm{b} + 1)(1 - \theta_\mathrm{b})^3h^2 \\ &= \frac{u'''(\xi)}{12}\left(-(2\theta_\mathrm{b} - 1)(1 + \theta_\mathrm{b})^3 + 4\theta_\mathrm{b}^4 + (2\theta_\mathrm{b} + 1)(1 - \theta_\mathrm{b})^3\right)h^2 \ \sim \mathrm{O}(h^2), \end{aligned}$$

for some $\xi \in [x_{n-2}, x_n]$.

3.5 Consistency for the Dirichlet problem in 2D

We consider the Dirichlet problems in two-dimensional case and we prove that the method is consistent.

We calculate the consistency error $\tau_h = \mathrm{A}_h u - \mathrm{A}u$.

In the internal points, if $u \in \mathrm{C}^4(\Omega_h^{\mathrm{I}}, \Omega_h^{\mathrm{G}})$, we have:

$$\tau_{ij} = \tau_h(x_i, y_j) = \Delta_h u(x_i, y_j) - \Delta u(x_i, y_j) = \frac{h^2}{12} u^{\mathrm{IV}}(\xi_i, \mu_j), \quad (i,j) \in \Omega_h^{\mathrm{I}},$$

for some $\xi_i \in [x_{i-1}, x_{i+1}]$, $\mu_j \in [y_{j-1}, y_{j+1}]$.
Therefore,

$$\begin{aligned}
\|\tau_h\|_{L^1} &= \int\int |\tau_h| \mathrm{d}x\mathrm{d}y \approx \sum_{i,j \in \Omega_h^{\mathrm{I}}} |\tau_h(x_i, y_j)| h^2 = \frac{h^2}{12} \sum_{i,j \in \Omega_h^{\mathrm{I}}} |u^{\mathrm{IV}}(\xi_i, \mu_j)| h^2 \\
&\approx \frac{h^2}{12} \|u^{\mathrm{IV}}\|_{L^1} \sim \mathrm{O}(h^2), \\
\|\tau_h\|_{L^2} &= \left(\int\int |\tau_h|^2 \mathrm{d}x\mathrm{d}y \right)^{\frac{1}{2}} \approx \left(\sum_{i,j \in \Omega_h^{\mathrm{I}}} |\tau_h(x_i, y_j)|^2 h^2 \right)^{\frac{1}{2}} \\
&= \frac{h^2}{12} \left(\sum_{i,j \in \Omega_h^{\mathrm{I}}} |u^{\mathrm{IV}}(\xi_i, \mu_j)|^2 h^2 \right)^{\frac{1}{2}} \approx \frac{h^2}{12} \|u^{\mathrm{IV}}\|_{L^2} \sim \mathrm{O}(h^2),
\end{aligned}$$

$$\|\tau_h\|_{L^\infty} = \max_{i,j \in \Omega_h^{\mathrm{I}}} |\tau_h| = \frac{h^2}{12} \max_{i,j \in \Omega_h^{\mathrm{I}}} |u^{\mathrm{IV}}(\xi_i, \mu_j)| = \frac{h^2}{12} \|u^{\mathrm{IV}}\|_{L^\infty} \sim \mathrm{O}(h^2).$$

Proceeding as in the one-dimensional case, in each ghost point, if we perform the bilinear interpolation procedure, from the formula (1.22) of the interpolation error, we obtain:

$$\tau_{\mathrm{G}} \sim \mathrm{O}(h^2),$$

instead, if we perform the biquadratic interpolation procedure, from the formula of the interpolation error, we obtain:

$$\tau_{\mathrm{G}} \sim \mathrm{O}(h^3).$$

3.6 Attempts to proof of convergence for the Dirichlet problem in 2D

In this Section we show the attempts to proof the convergence of the numerical method for the Dirichlet problems in 2D. However, the problem is still open.

In a first attempt we tried to extend the proof of Lisl Weynans (Section 3.3) to two-dimensional case. As in the one-dimensional case, we consider an array V such that $A_h V \geq 0$ and we want to prove that $V \geq 0$. However, unlike the one-dimensional case, in this case the equation (3.2) for the ghost point is a system of coupled equations (because the equation for a ghost point can involve other ghost points) too complex to resolve.

In a second attempt we have tried to extend the convergence proof of Section 3.3 to two-dimensional case, which is based on the use of M-matrices. However, as we will see, in 2D the matrix A_h is not a M-matrix.
A matrix $B \in \mathbb{R}^{n\times n}$ is a M-matrix if $b_{i,j} \leq 0$ with $i \neq j$ and $B^{-1} \geq 0$.
Consider the simple case in which there is an only one ghost point G as shown in the following figure (Figure 3.1)

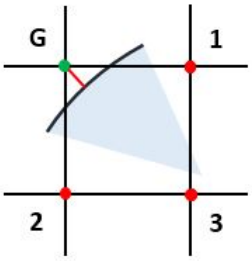


Figure 3.1: Portion of a two-dimensional arbitrary domain with only one ghost point (green dot)

and we write the system in this way:

$$\begin{pmatrix} A_{gg} & A_{gi} \\ A_{ig} & A_{ii} \end{pmatrix} \begin{pmatrix} u_g \\ u_i \end{pmatrix} = \begin{pmatrix} \bar{u}_G \\ f \end{pmatrix},$$

or equivalently

$$\mathrm{A}_{\mathrm{gg}} u_{\mathrm{g}} + \mathrm{A}_{\mathrm{gi}} u_{\mathrm{i}} = \bar{u}_{\mathrm{G}}$$

that it is the equation for the ghost point and

$$\mathrm{A}_{\mathrm{ig}} u_{\mathrm{g}} + \mathrm{A}_{\mathrm{ii}} u_{\mathrm{i}} = f$$

that are the equations for the internal points, where $\bar{u}_{\mathrm{G}}$ is the exact value that assumes the function u at the ghost point G.
In particular, the first of the two equations is equivalent to a linear combination of the values that the function u assumes at the points involved in the bilinear interpolation:

$$\alpha_{\mathrm{G}} u_{\mathrm{G}} + \alpha_1 u_1 + \alpha_2 u_2 + \alpha_3 u_3 = \bar{u}_{\mathrm{G}}, \text{ with } -1 < \alpha_{\mathrm{G}}, \alpha_1, \alpha_2, \alpha_3 < 1.$$

We have:

- $\mathrm{A}_{\mathrm{gg}} = (\alpha_{\mathrm{G}})$, therefore $\mathrm{A}_{\mathrm{gg}}^{-1} = \left(\frac{1}{\alpha_{\mathrm{G}}}\right)$
- $\mathrm{A}_{\mathrm{gi}} = \begin{pmatrix} \alpha_1 & 0 & \dots & 0 & \alpha_2 & \alpha_3 & 0 & \dots & \dots \end{pmatrix}$
- $\mathrm{A}_{\mathrm{ig}} = \begin{pmatrix} -\frac{1}{h^2} \\ 0 \\ \vdots \\ 0 \\ -\frac{1}{h^2} \\ 0 \\ \vdots \\ \vdots \end{pmatrix}$
- A_{ii} is the classical block matrix obtained on a two-dimensional domain in the absence of ghost points
- $\mathrm{A}_{\mathrm{ig}}\mathrm{A}_{\mathrm{gg}}^{-1}\mathrm{A}_{\mathrm{gi}} = \begin{pmatrix} -\frac{\alpha_1}{\alpha_{\mathrm{G}} h^2} & 0 & \dots & 0 & -\frac{\alpha_2}{\alpha_{\mathrm{G}} h^2} & -\frac{\alpha_3}{\alpha_{\mathrm{G}} h^2} & 0 & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ -\frac{\alpha_1}{\alpha_{\mathrm{G}} h^2} & 0 & \dots & 0 & -\frac{\alpha_2}{\alpha_{\mathrm{G}} h^2} & -\frac{\alpha_3}{\alpha_{\mathrm{G}} h^2} & 0 & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}$

- $\mathrm{A_{ii}} - \mathrm{A_{ig}}\mathrm{A_{gg}^{-1}}\mathrm{A_{gi}}$ has the elements $\frac{-\alpha_{\mathrm{G}}+\alpha_i}{\alpha_{\mathrm{G}}h^2}$ out of the main diagonal which may not be negative. Thus $\mathrm{A_{ii}} - \mathrm{A_{ig}}\mathrm{A_{gg}^{-1}}\mathrm{A_{gi}}$ isn't a M-matrix.

3.7 Consistency for the mixed problem in 2D

We consider the mixed problems in two-dimensional case and we prove that the method is consistent.

In the internal points we have already tried that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$. We have already also studied the consistency error in the ghost points in which we impose the Dirichlet boundary conditions.
For each ghost point G in which we impose the Neumann boundary conditions, from the formulas of the errors in the bilinear and biquadratic interpolations, we obtain:

$$\tau_{\mathrm{G}} \sim \mathrm{O}(h)$$

if we perform the bilinear interpolation procedure, and

$$\tau_{\mathrm{G}} \sim \mathrm{O}(h^2)$$

if we perform the biquadratic interpolation procedure.

3.8 Consistency for the Dirichlet problem in 3D

We proceed with the consistency of the method for the Dirichlet problems in three-dimensional case.

We calculate the consistency error $\tau_h = \mathrm{A}_h u - \mathrm{A}u$.

In the internal points, if $u \in \mathrm{C}^4(\Omega_h^{\mathrm{I}}, \Omega_h^{\mathrm{G}})$, we have:

$$\tau_{ijk} = \tau_h(x_i, y_j, z_k) = \Delta_h u_{ijk} - \Delta u_{ijk} = \frac{h^2}{12} u^{\mathrm{IV}}(\xi_i, \mu_j, \zeta_k), \qquad (i, j, k) \in \Omega_h^{\mathrm{I}},$$

for some $\xi_i \in [x_{i-1}, x_{i+1}]$, $\mu_j \in [y_{j-1}, y_{j+1}]$, $\zeta_k \in [z_{k-1}, z_{k+1}]$.

Therefore,

$$\begin{aligned}
\|\tau_h\|_{L^1} &= \int\int\int |\tau_h| \mathrm{d}x\mathrm{d}y\mathrm{d}z \approx \sum_{i,j,k\in\Omega_h^{\mathrm{I}}} |\tau_h(x_i,y_j,z_k)|h^3 \\
&= \frac{h^2}{12} \sum_{i,j,k\in\Omega_h^{\mathrm{I}}} |u^{\mathrm{IV}}(\xi_i,\mu_j,\zeta_k)|h^3 \approx \frac{h^2}{12}\|u^{\mathrm{IV}}\|_{L^1} \sim \mathrm{O}(h^2), \\
\|\tau_h\|_{L^2} &= \left(\int\int\int |\tau_h|^2 \mathrm{d}x\mathrm{d}y\mathrm{d}z\right)^{\frac{1}{2}} \approx \left(\sum_{i,j,k\in\Omega_h^{\mathrm{I}}} |\tau_h(x_i,y_j,z_k)|^2 h^3\right)^{\frac{1}{2}} \\
&= \frac{h^2}{12}\left(\sum_{i,j,k\in\Omega_h^{\mathrm{I}}} |u^{\mathrm{IV}}(\xi_i,\mu_j,\zeta_k)|^2 h^3\right)^{\frac{1}{2}} \approx \frac{h^2}{12}\|u^{\mathrm{IV}}\|_{L^2} \sim \mathrm{O}(h^2), \\
\|\tau_h\|_{L^\infty} &= \max_{i,j,k\in\Omega_h^{\mathrm{I}}} |\tau_h| = \frac{h^2}{12} \max_{i,j,k\in\Omega_h^{\mathrm{I}}} |u^{\mathrm{IV}}(\xi_i,\mu_j,\zeta_k)| = \frac{h^2}{12}\|u^{\mathrm{IV}}\|_{L^\infty} \sim \mathrm{O}(h^2).
\end{aligned}$$

Proceeding as in one-dimensional case, in each ghost point G, if we perform the trilinear interpolation procedure, from the formula of the interpolation error, we obtain:

$$\tau_{\mathrm{G}} \sim \mathrm{O}(h^2);$$

if we perform the triquadratic interpolation procedure we obtain:

$$\tau_{\mathrm{G}} \sim \mathrm{O}(h^3).$$

3.9 Consistency for the mixed problem in 3D

Finally, we prove the consistency of the method for the mixed problems in three-dimensional case.

In the internal points we have already tried that $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2)$, $p = 1, 2, \infty$. We have already also studied the consistency error in the ghost points in which we impose the Dirichlet boundary conditions.
For each ghost point G in which we impose the Neumann boundary condition, from the formula of the interpolation error in the trilinear interpolation procedure, we obtain:

$$\tau_{\mathrm{G}} \sim \mathrm{O}(h);$$

if we perform the triquadratic interpolation procedure, we have:

$$\tau_{\mathrm{G}} \sim \mathrm{O}(h^2).$$

4

Shock capturing schemes for compressible Gas Dynamics

In order to apply the Coco-Russo method to Full Euler equations of Gas Dynamics on two-dimensional arbitrary domains, in this Chapter we present explicit shock capturing schemes for the numerical resolution of hyperbolic systems of conservation laws.

We introduce the theory on hyperbolic systems of conservation laws [14, 17, 18, 34].

Hyperbolic systems of conservation laws

Introduction

A *system of conservation laws* is a system of partial differential equations which depends on time and which takes on a particular form.
In *one spatial dimension* the equations take the form

$$\frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}f(u(x,t)) = 0, \tag{4.1}$$

in which $u : \mathbb{R}\times\mathbb{R} \longrightarrow \mathbb{R}^m$ is an m-dimensional vector of conserved quantities or state variables and $f : \mathbb{R}^m \longrightarrow \mathbb{R}^m$ is the **flux function** or simply **flux** for the system of conservation laws. This function is supposed to be known and regular.

The particular form of the system (4.1) derives from physical considerations. In fact, supposed that u is regular and that the various components u_j

represent the *densities* of certain quantities that are conserved, the integral $\int_a^b u(x,t)\mathrm{d}x$ is the total quantity of this state variables in the interval $[a,b]$ at time t. The hypothesis that this quantity is conserved implies that its temporal variation in $[a,b]$ is governed by a flux function f which controls the variation of u through the border. This translates into the formula:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_a^b u(x,t)\mathrm{d}x = f(u(t,a)) - f(u(t,b)).$$

The system (4.1) is *hyperbolic* if, for each value of u, the eigenvalues of $f'(u)$ are real and the matrix is diagonalizable, where $f'(u)$ is the $m \times m$ Jacobian matrix of the flux function.

In *two spatial dimensions* a system of conservation laws takes the form

$$\frac{\partial}{\partial t}u(x,y,t) + \frac{\partial}{\partial x}f(u(x,y,t)) + \frac{\partial}{\partial y}g(u(x,y,t)) = 0, \tag{4.2}$$

in which $u : \mathbb{R}^2 \times \mathbb{R} \longrightarrow \mathbb{R}^m$ is the vector of state variables and $f, g : \mathbb{R}^m \longrightarrow \mathbb{R}^m$ are the two flux functions.

Applications

The hyperbolic systems of conservation laws are central in many applications such as metereology, astrophysical modeling, the physics of the plasmas, models for the flow of traffic, aerodynamics.

A system that is of particular importance in the field of systems of conservation laws is the system of **Euler equations of gas dynamics**, concerning the equations of the conservation of mass, energy and momentum in each direction. This system is hyperbolic and the eigenvalues are: $\lambda_1 = \mathbf{u}\cdot\mathbf{n} - \frac{c_s}{\varepsilon}$, $\lambda_2 = \mathbf{u}\cdot\mathbf{n}$ and $\lambda_3 = \mathbf{u}\cdot\mathbf{n} + \frac{c_s}{\varepsilon}$, where ε and c_s are the Mach number [6] and the sound speed respectively and $\mathbf{n}$ is the normal unit vector to the boundary.

In one spatial dimension the system of the Euler equations takes the following form:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2 + p\\ u(E+p)\end{pmatrix} = 0,$$

where $\rho = \rho(x,t)$ is the density, u is the velocity, ρu is the momentum, E is the energy and p is the pressure. The pressure p is calculated as a function of the other variables, according to the equation of state.

Much of the theory of conservation laws has been developed with such equations and many numerical methods have been developed specifically for this system. Therefore, good knowledge of Euler equations is needed to read much of the available literature and take advantage of these developments.

Analytical solution

The classical method for solving hyperbolic systems of conservation laws is given by the *method of characteristics*. However, difficulties may arise: discontinuities can be developed in finite time. Discontinuous solutions do not satisfy the systems in the classical sense at all points, since the derivatives are not defined at discontinuities. To overcome this problem is introduced a "weak form" of the differential equations that will be fundamental in the development of numerical methods. When are considered weak solutions, however, the uniqueness is less. Among all the (infinite) weak solutions possible for a given problem with initial conditions, the physically possible one is the only one that satisfies the entropy condition, obtained as the limit for $\varepsilon \to 0$ of a family of solutions (u_ε) of the convection-diffusion equation $\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}$.

Numerical solution

These problems can also occur on the numerical solution, for this reason numerical methods have been developed that are increasingly adapted to resolve the presence of possible discontinuities.

Linear problem The classical *finite difference methods* for linear problems (obtained by substituting the partial derivatives with the differences in the punctual values of the approximate solution) do not take into consideration the presence of possible discontinuity, therefore they can work well for smooth solutions but they may not give excellent results not even on a particular fine grid when the numerical method is applied to a linear problem with discontinuous initial data, for example a Riemann problem. A *Riemann problem* is a conservation law together with piecewise constant data having a single

discontinuity. For example the Riemann problem for the scalar advection equation [17] is:

$$u_t + au_x = 0, \quad -\infty < x < +\infty, \ t \geq 0$$

$$u_0(x) = \begin{cases} 1 & x < 0 \\ 0 & x > 0 \end{cases}.$$

The typical behavior is as follows: the first order methods give very smeared solutions while the second order methods give oscillations. For more details see [17].

Nonlinear problem When trying to solve numerically the non-linear conservation laws, there are additional difficulties. However, for smooth solutions to non-linear problems, the numerical method can often be linearized. We focus therefore our attention instead on discontinuous solutions for non-linear problems. The additional problem that occurs in the case of non-linear problems is that the method might converge to a function that is not a weak solution or this solution not satisfy the entropy condition. So that the numerical solution converges to a weak solution of the problem, we consider a conservative form that ensures convergence to a weak solution in the case in which the scheme is consistent and convergent. However, it does not ensure that this solution is the entropy solution, in fact there are many examples of conservative numerical methods that converge to weak solutions but they do not satisfy the entropy condition.

We discretize the x-t plane and we define the discrete mesh points (x_j, t^n) by

$$x_j = j\Delta x, \ j = \ldots, -1, 0, 1, \ldots$$

$$t = n\Delta t, \ n = 0, 1, 2, \ldots.$$

A numerical scheme is *conservative* if there is a function $\hat{h}$ (called **numerical flux function** and that depends both on the flux function $f(u)$ and on the type of discretization)

$$\hat{f}^n_{j+\frac{1}{2}} = \hat{h}(U^n_{j-p+1}, \ldots, U^n_{j+q}),$$

such that it is possible to write the scheme of the differences in the form (called conservative)

$$U^{n+1}_j = U^n_j - \frac{\Delta t}{\Delta x}\left(\hat{f}^n_{j+\frac{1}{2}} - \hat{f}^n_{j-\frac{1}{2}}\right), \tag{4.3}$$

where U_j^n is the numerical solution at the time n in the point x_j.
In general, for a first order scheme we choose $\hat{f}_{j+\frac{1}{2}} = \hat{h}(u_j, u_{j+1})$.

This difference scheme is a non-linear, explicit and two-level scheme, where the non-linearity is present in the function $f(u)$ and/or in the numerical flux function.

It is possible to show that, so that the scheme (4.3) is consistent, it is sufficient that the following property is satisfied

$$\hat{h}(u, u, \ldots, u) = f(u).$$

Conservative schemes are considered because it is proved that they tend to capture any discontinuities present in the solution in an appropriate manner, however, as already mentioned, they do not ensure convergence to the entropy solution. Another problem of conservative methods is that the convergence of the numerical scheme appears as a hypothesis of the theorem, and therefore must be verified by other means. For more details about the conservative schemes and the choice of the correct numerical flux function see [17].

An alternative to the conservative methods is provided by the *monotone schemes* [17]. The monotone methods for the conservation laws are TVD (Total Variation Diminishing) [17]. This property allow the oscillations to not grow over time and the non-oscillatory convergence to the entropy solution. However, monotone methods are at most first order accurate, giving poor accuracy solutions in the smooth regions of the flow. Moreover, shocks tend to be heavily smeared and poorly resolved on the grid. These effects are due to the large amount of numerical dissipation in monotone methods. Some dissipation is needed to give nonoscillatory shocks but monotone methods go overboard in this direction.

As already mentioned in the case of linear problems, the first order accurate numerical methods have a large amount of "numerical viscosity" that smoothes the solution (as physical viscosity). Thus, this methods give results that are smeared in the regions near the discontinuities. If instead we use a second order method we eliminate this numerical viscosity but introduce dispersive effects that lead to large oscillations in the numerical solution

itself.

Therefore numerical methods have been developed that produce sharp approximations to discontinuous solutions automatically. These methods are called **"shock capturing"** methods (or *high resolution methods*). These methods use a high order method, at least second order accurate on the smooth solutions, and increase the amount of numerical dissipation in the neighborhood of a discontinuity to give nonoscillatory solutions. Furthermore, they present an appropriate form of consistency with the weak form of the conservation law, required to converge to weak solutions, and a discrete form of the entropy condition, to alows the convergence at the physically correct weak solution.

We can define high order approximations to the flux at a cell boundary, using a piecewise linear approximation in defining the flux $\hat{f}_{j+\frac{1}{2}}$ using slope limiters: $\hat{f}_{j+\frac{1}{2}} = \hat{h}(u^L_{j+\frac{1}{2}}, u^R_{j+\frac{1}{2}})$, where $u^L_{j+\frac{1}{2}}$ and $u^R_{j+\frac{1}{2}}$ are the values of u on the right and left of $x_{j+\frac{1}{2}}$ and are determined from the reconstruction in cells j and $j+1$, respectively. See [17] for a discussion on TVD slope limiters.

Semi-discrete schemes

The discretization of the equations can be done through *fully discrete* schemes or *semi-discrete* schemes.
In the *fully discrete* schemes the discretizations of spatial and temporal derivatives occur simultaneously. It is obtained a scheme of differences with the following structure:

$$U^{n+1}_j = H(U^n_{j-p}, \ldots, U^n_{j+q}),$$

where H depends on the particular numerical method with which we have discretized the equations. These schemes allow to immediately calculate the solution to the node (or cell) j in terms of the values known at the previous time.

It is possible to consider the discretization process in two stages: first we discretize only in space, leaving the problem continuous in time and obtaining a system of ordinary differential equations (in time) and then we discretize in time using any standard numerical method for systems of ordinary differential equations.

The advantage in using semi-discrete methods lies in the fact that it is possible to obtain schemes with order of accuracy greater than 2, since it allows us to decouple the issues of spatial and temporal accuracy. Indeed, to improve the accuracy in time we can replaced the first order accurate Euler method with a higher order method and to improve the accuracy in space we can define high order approximations to the flux at a cell boundary. The classic approach to avoiding the onset of oscillations in semi-discrete schemes is to impose constraints on the reconstruction process to satisfy, for example, the TVD condition. In recent years have been developed modern schemes such as ENO (Essentially Non-Oscillatory) [17, 31] and WENO (Weighted Essentially Non-Oscillatory) [31].

In this Chapter we present explicit shock-capturing semi-discrete schemes for compressible Gas Dynamics, using first and second order accurate reconstructions both in time and space.

Explicit time discretization Most schemes used to compute the numerical solution of the systems of conservation laws are obtained by explicit time discretization, and the time step has to satisfy a stability condition, known as CFL condition, which states that the time step should be limited by the space step divided by the fastest wave speed:

$$\Delta t < \frac{\Delta x}{\lambda_{\max}} \approx \mathrm{O}(\varepsilon \Delta x), \tag{4.4}$$

where $\lambda_{\max} = \max_{\Omega} \left(|\mathbf{u}| + \frac{c_s}{\varepsilon} \right)$, $\mathbf{u}$ is the velocity wave, $c_{s_i} = \sqrt{\frac{\gamma p_i}{\rho_i}}$ is the sound speed, ρ_i and p_i are the density and pressure at the time step i respectively and ε represents a global Mach number. For low Mach number in the Gas Dynamics equations, the compressible case tends to a incompressible case.

We present two different approaches present in [17, 18, 24].

Methods for hyperbolic systems of conservation laws

There are schemes based on cell avarages (finite volume approach) and schemes based on point values (finite difference approach).

For semplicity, we consider the case of the single scalar equation

$$u_t + f(u)_x = 0. \tag{4.5}$$

Let Δx and Δt be the mesh widths. We introduce the grid points $x_j = j\Delta x$, $x_{j+\frac{1}{2}} = x_j + \frac{1}{2}\Delta x$, $j = \ldots, -1, 0, 1, \ldots$, and we use the standard notations

$$u_j^n = u(x_j, t^n), \ \bar{u}_j^n = \frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t^n)\mathrm{d}x.$$

Finite volumes (FV) methods

One of the most widely used methods of solving these problems is that of the *finite volumes method* in conservative form.

The inside of the domain is subdivided into many elementary volumes (called cells) $I_j = \left[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}\right]$, identifying with the index j the center of each cell. Through the integral form of the equations of the problem considered, are written the relationships between the various neighboring volumes.

There are two ways to discretize (4.5). The first way is to develop *fully discrete* (one-step) schemes, which is obtained by integrating both in space and in time.

Another way to discretize (4.5) is to keep the time variable t continuous and consider *semi-discrete* schemes. Integrating (4.5) with respect to x only we obtain the following system:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{u}_j(t) = -\frac{1}{\Delta x}\left[f(u(x_{j+\frac{1}{2}}, t)) - f(u(x_{j-\frac{1}{2}}, t))\right].$$

To convert this expression into a numerical scheme, it must be approximated on right hand side with a function of the cell averages $\{\bar{u}(t)\}_j$, which are the unknowns of the problem.

As already stated in the previous paragraphs, it is necessary to carry out a reconstruction of the unknown function $u(x, t)$ by a piecewise polynomial starting from cell averages $\bar{u}_j(t)$, and that must take into consider any discontinuities of the function $u(x, t)$. The accuracy order of the method depends on the order of the polynomial. The higher the order of the polynomial, the more accurate the numerical method is. For example, for first order schemes the reconstruction is piecewise constant, while second order schemes can be obtained by a piecewise linear reconstruction.

A piecewise polynomial reconstruction is obtained as follows:

$$\mathcal{R}(x) = \sum_j P_j(x)\chi_j(x),$$

where $P_j(x)$ is a polynomial satisfying some accuracy and non-oscillatory property and $\chi_j(x)$ is the indicator function of cell l_j.

The flux function at the edge of the cell can be computed by using a suitable numerical flux function, consistent with the analytical flux.
As already stated in the previous paragraphs, for a first order (in space) semi-discrete scheme we choose:

$$f(u(x_{j+\frac{1}{2}}, t)) \approx F(\bar{u}_j, \bar{u}_{j+1}),$$

and thus the scheme is:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{u}_j = -\frac{F(\bar{u}_j, \bar{u}_{j+1}) - F(\bar{u}_{j-1}, \bar{u}_j)}{\Delta x}.$$

An example of first order accurate numerical flux is the Local Lax-Friedrichs flux [17], which we will discuss later.
For a second order (in space) scheme we choose:

$$f(u(x_{j+\frac{1}{2}}, t)) \approx F(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}}),$$

where the quantities $u^{\pm}_{j+\frac{1}{2}}$ are obtained from the reconstruction in this way $u^{\pm}_{j+\frac{1}{2}} = \lim_{x \to x^{\pm}_{j+\frac{1}{2}}} \mathcal{R}(x)$. For example,

$$u^-_{j+\frac{1}{2}} = \bar{u}_j + \frac{\Delta x}{2} u'_j,$$

where the slope u'_j is a first order approximation of the space derivative of $u(x,t)$, and can be computed by suitable slope limiters [17]. Later we will introduce the generalized minmod limiter.
Thus, the scheme is:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{u}_j = -\frac{F(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}}) - F(u^-_{j-\frac{1}{2}}, u^+_{j-\frac{1}{2}})}{\Delta x}.$$

Finite-difference (FD) methods

We consider now a finite difference scheme in which the unknown is the pointwise value of the function, rather than its cell average.
We want to get a finite difference scheme in a conservative form (4.3). Therefore, given the equation (4.5), we write:

$$\frac{\partial f}{\partial x}(u(x,t)) = \frac{\hat{f}\left(u\left(x+\frac{\Delta x}{2}\right),t\right) - \hat{f}\left(u\left(x-\frac{\Delta x}{2}\right),t\right)}{\Delta x}.$$

To convert this expression into a numerical scheme we want to approximate the quantity $\hat{f}\left(u\left(x+\frac{\Delta x}{2}\right),t\right) - \hat{f}\left(u\left(x-\frac{\Delta x}{2}\right),t\right)$.

To do this, we find the relation between f and $\hat{f}$. In this regard, we consider the average operator

$$\bar{u}(x,t) = \frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} u(\xi,t)\mathrm{d}\xi$$

and differentiate it with respect to x, we obtain:

$$\frac{\partial \bar{u}}{\partial x} = \frac{1}{\Delta x}\left(\left(u\left(x+\frac{\Delta x}{2}\right),t\right) - \left(u\left(x-\frac{\Delta x}{2}\right),t\right)\right).$$

From the previous expressions we can observe that the relation between f and $\hat{f}$ is the same relation that there is between $\bar{u}(x,t)$ and $u(x,t)$, namely, the flux function f is the cell average of the function $\hat{f}$. Therefore to compute $\hat{f}(u(x_{j+\frac{1}{2}},t))$ from $f(u(x_j,t))$ we use the same technique used to compute $u_{j+\frac{1}{2}}$ from $\bar{u}_j$.
In the finite difference method it is the flux function which is computed at x_j and then reconstructed at $x_{j+\frac{1}{2}}$. But the reconstruction at $x_{j+\frac{1}{2}}$ may be discontinuous. To solve this problem we consider flux functions that can be split

$$f(u) = f^+(u) + f^-(u)$$

with the condition that

$$\frac{\mathrm{d}f^+(u)}{\mathrm{d}u} \geq 0, \quad \frac{\mathrm{d}f^-(u)}{\mathrm{d}u} \leq 0,$$

which defines a monotone consistent flux. For example, the local Lax-Friedrichs flux satisfies this condition.

A finite difference scheme therefore takes the following form

$$\frac{\mathrm{d}}{\mathrm{d}t}u_j = -\frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x},$$

where $\hat{f}_{j+\frac{1}{2}} = \hat{f}^+(u^-_{j+\frac{1}{2}}) + \hat{f}^-(u^+_{j+\frac{1}{2}})$.

The quantity $\hat{f}^+(u^-_{j+\frac{1}{2}})$ is obtained by

- computing $f^+(u_j)$ and interpret it as cell average of $\hat{f}^+$;
- performing pointwise reconstruction of $\hat{f}^+$ in cell j, and evaluate it in $x_{j+\frac{1}{2}}$;

and $\hat{f}^-(u^+_{j+\frac{1}{2}})$ is obtained by

- computing $f^-(u_j)$ and interpret it as cell average of $\hat{f}^-$;
- performing pointwise reconstruction of $\hat{f}^-$ in cell j, and evaluate it in $x_{j+\frac{1}{2}}$.

Remark 4.1: Both methods, discussed for the scalar equation, can be extended to systems, and both present advantages. The finite volumes methods, unlike the finite difference methods, can also be applied in the case of non-uniform grids. However, since the source is evaluated pointwise, finite difference schemes do not couple the cells, unlike the finite volumes schemes. This is a big advantage in the case we use a implicit step. Given these considerations, because in the next chapter we will present the semi-implicit methods on a regular Cartesian grid, we use the finite difference methods.

We apply the finite difference methods to the gas-dynamics equations both for the *Isentropic Euler Equations* and for the *Full Euler Equations*.

We start considering the isentropic gas dynamics case and successively we extend the results to the case of the full Euler system.

4.1 Isentropic Euler Equations

The *Isentropic Euler equations* in a d-dimensional space, with $x \in \Omega \subset \mathbb{R}^d$ and $t \geq 0$, are given by:

$$\begin{cases} \rho_t + \nabla \cdot (\rho \mathbf{u}) = 0 \\ (\rho \mathbf{u})_t + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \frac{\nabla p}{\varepsilon^2} = 0 \end{cases}, \tag{4.6}$$

where ρ is the density of the fluid, $\mathbf{u}$ is the velocity of the fluid and p is the pressure. To close the system we consider a equation of state, which in the case of a polytropic gas is:

$$p = \rho^\gamma,$$

where $\gamma = \frac{C_p}{C_v} > 1$ is the polytropic constant.
The parameter ε is the dimensionless reference Mach number.

Let $\mathbf{m} = \rho \mathbf{u}$ be the momentum, the system (4.6) can be written in this way:

$$\begin{cases} \rho_t + \nabla \cdot \mathbf{m} = 0 \\ \mathbf{m}_t + \nabla \cdot (\mathbf{m} \otimes \mathbf{u}) + \frac{\nabla p}{\varepsilon^2} = 0 \end{cases}. \tag{4.7}$$

4.1.1 Isentropic Euler Equations in 1D

In one-dimensional space the system (4.7) is:

$$\begin{cases} \rho_t + m_x = 0 \\ m_t + \left(\frac{m^2}{\rho} + \frac{p}{\varepsilon^2} \right)_x = 0 \end{cases}, \tag{4.8}$$

and it is closed by the equation of state $p = p(\rho) = \rho^\gamma$.

4.1.2 Isentropic Euler Equations in 2D

In two-dimensional space the system (4.7) is:

$$\begin{cases} \rho_t + m_x^{\mathbf{1}} + m_y^{\mathbf{2}} = 0 \\ m_t^{\mathbf{1}} + \left(\frac{{m^{\mathbf{1}}}^2}{\rho} + \frac{p}{\varepsilon^2} \right)_x + \left(m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \right)_y = 0 \\ m_t^{\mathbf{2}} + \left(m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \right)_x + \left(\frac{{m^{\mathbf{2}}}^2}{\rho} + \frac{p}{\varepsilon^2} \right)_y = 0 \end{cases}, \tag{4.9}$$

where $m^{\mathbf{1}}$ and $m^{\mathbf{2}}$ are the momentum along the x-axis and the momentum along the y-axis respectively, and it is closed by the equation of state $p = p(\rho) = \rho^\gamma$.

4.2 Full Euler Equations

The *Full Euler equations* in a d-dimensional space, with $x \in \Omega \subset \mathbb{R}^d$ and $t \geq 0$, are given by:

$$\begin{cases} \rho_t + \nabla \cdot (\rho \mathbf{u}) = 0 \\ (\rho \mathbf{u})_t + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \frac{\nabla p}{\varepsilon^2} = 0 \\ E_t + \nabla \cdot [(E+p)\mathbf{u}] = 0 \end{cases}, \tag{4.10}$$

where ρ is the density of the fluid, $\mathbf{u}$ is the velocity of the fluid, p is the pressure and E is the energy. To close the system we consider the equation of state, which for a polytropic gas is:

$$E = \frac{p}{\gamma - 1} + \frac{\varepsilon^2}{2} \rho |\mathbf{u}|^2.$$

Placing $\mathbf{m} = \rho \mathbf{u}$ the momentum and $h = \frac{E+p}{\rho}$ the total enthalpy, the system (4.10) can be written in this way:

$$\begin{cases} \rho_t + \nabla \cdot \mathbf{m} = 0 \\ \mathbf{m}_t + \nabla \cdot (\mathbf{m} \otimes \mathbf{u}) + \frac{\nabla p}{\varepsilon^2} = 0 \\ E_t + \nabla \cdot (h\mathbf{m}) = 0 \end{cases}. \tag{4.11}$$

4.2.1 Full Euler Equations in 1D

In one-dimensional space the system (4.11) is:

$$\begin{cases} \rho_t + m_x = 0 \\ m_t + \left(\frac{m^2}{\rho} + \frac{p}{\varepsilon^2} \right)_x = 0 \\ E_t + (hm)_x = 0 \end{cases}, \tag{4.12}$$

and it is closed by the equation of state $p = (\gamma - 1)\left(E - \frac{1}{2}\varepsilon^2 \frac{m^2}{\rho}\right)$.

4.2.2 Full Euler Equations in 2D

In two-dimensional space the system (4.11) is:

$$\begin{cases} \rho_t + m_x^{\mathbf{1}} + m_y^{\mathbf{2}} = 0 \\ m_t^{\mathbf{1}} + \left(\frac{{m^{\mathbf{1}}}^2}{\rho} + \frac{p}{\varepsilon^2} \right)_x + \left(m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \right)_y = 0 \\ m_t^{\mathbf{2}} + \left(m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \right)_x + \left(\frac{{m^{\mathbf{2}}}^2}{\rho} + \frac{p}{\varepsilon^2} \right)_y = 0 \\ E_t + (hm^{\mathbf{1}})_x + (hm^{\mathbf{2}})_y = 0 \end{cases}, \tag{4.13}$$

and it is closed by the equation of state $p = (\gamma - 1)\left(E - \frac{1}{2}\varepsilon^2\left(\frac{m^{\mathbf{1}^2}}{\rho} + \frac{m^{\mathbf{2}^2}}{\rho}\right)\right)$.

We apply the explicit finite-difference methods to the Euler gas-dynamics equations, both in the one-dimensional case and in the two-dimensional case.

4.3 FD shock-capturing methods for Gas Dynamics Equations in 1D

We discretize in 1D the Euler systems (4.8) and (4.12) of conservation laws, that we can write in the following form:

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0.$$

with initial and boundary conditions. Here $\mathbf{U}(x,t)$ is the vector of unknown conservative variables and $\mathbf{F(U)}$ is the physical flux vector.

In the system (4.8)

$$\mathbf{U} = \begin{pmatrix} \rho \\ m \end{pmatrix} \text{ and } \mathbf{F(U)} = \begin{pmatrix} m \\ \frac{m^2}{\rho} + \frac{p}{\varepsilon^2} \end{pmatrix},$$

while in the system (4.12)

$$\mathbf{U} = \begin{pmatrix} \rho \\ m \\ E \end{pmatrix} \text{ and } \mathbf{F(U)} = \begin{pmatrix} m \\ \frac{m^2}{\rho} + \frac{p}{\varepsilon^2} \\ (E+p)u \end{pmatrix}.$$

Discretization in space

First we discretize the space $\Omega = [0,1]$ through a uniform grid characterized by a spatial step $\Delta x = x_{j+1} - x_j$, $\forall j$ $\left(\Delta x = \frac{1}{N}\right)$. At even time step n we have N cells of size Δx, with cell j centered at $x_j = \left(j - \frac{1}{2}\right)\Delta x$, $j = 1, \ldots, N$. The following figure (Figure 4.1) shows the discretization of the domain Ω:

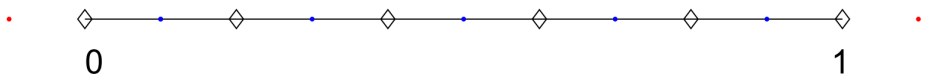


Figure 4.1: Discretization of the domain $[0,1]$ with $N=5$: the blue dots are the *internal points*, the red dots are the *ghost points*, in black diamonds we evaluate the numerical fluxes

The values of the numerical solution on the internal points $\mathbf{U}_i(t) \approx \mathbf{U}(x_i,t)$ evolve in time using the following semi-discrete scheme:

$$\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = -\frac{\hat{\boldsymbol{f}}_{i+\frac{1}{2}} - \hat{\boldsymbol{f}}_{i-\frac{1}{2}}}{\Delta x},\ i=1,2,\dots,N.$$

The numerical fluxes $\{\hat{\boldsymbol{f}}_{i+\frac{1}{2}}\}$ are computed as follows. We denote by $\hat{D}_x\boldsymbol{f}$ the flux derivatives that will be discretized as the difference of the numerical fluxes between $i+\frac{1}{2}$ and $i-\frac{1}{2}$.

First-order scheme

If we use a first-order scheme, to compute the numerical fluxes in $\hat{D}_x\boldsymbol{f}$ we use the local *Lax-Friedrichs fluxes*. We first split the flux function $\mathbf{F}$ as [31, 32, 33]

$$\mathbf{F}(\mathbf{U}_i) = \mathbf{F}_i^+ + \mathbf{F}_i^-,\ \mathbf{F}_i^\pm = \frac{1}{2}\left(\mathbf{F}(\mathbf{U}_i) \pm \alpha_i \mathbf{U}_i\right),$$

where $\alpha_i = |u_i| + \frac{c_i}{\varepsilon}$ is the spectral radius of the Jacobian matrix at $x=x_i$, $c_i = \sqrt{\frac{\gamma p_i}{\rho_i}}$ is the sound speed and ε is the dimensionless reference Mach number.
The numerical fluxes are given by: $\hat{\boldsymbol{f}}_{i+\frac{1}{2}} = \mathbf{F}_i^+ + \mathbf{F}_{i+1}^-$, that is:

$$\hat{\boldsymbol{f}}_{i+\frac{1}{2}} = \frac{\mathbf{F}(\mathbf{U}_{i+1}) + \mathbf{F}(\mathbf{U}_i)}{2} - \alpha_{i+\frac{1}{2}}\frac{\mathbf{U}_{i+1} - \mathbf{U}_i}{2},$$

with

$$\alpha_{i+\frac{1}{2}} = \max\left\{\alpha_i, \alpha_{i+1}\right\}.$$

Then the discrete derivatives are given by:

$$\hat{D}_x\boldsymbol{f} = \frac{\hat{\boldsymbol{f}}_{i+\frac{1}{2}} - \hat{\boldsymbol{f}}_{i-\frac{1}{2}}}{\Delta x},$$

where Δx is the space step discretization.

Second-order scheme

Second-order schemes in space are obtained by using a piecewise conservative linear reconstruction.
We refer to the scheme in [3, 8, 11].
For each conservative field u, we use the following reconstruction in cell j:

$$u_j^n(x) = \bar{u}_j^n + u'_j(x - x_j),$$

where the approximation of the derivative u'_j is computed using the generalized minmod limiter. The minmod function is defined by:

$$\text{minmod}(v_1, v_2, \ldots) := \begin{cases} \min(v_1, v_2, \ldots), & \text{if } v_i > 0 \ \forall i \\ \max(v_1, v_2, \ldots), & \text{if } v_i < 0 \ \forall i \\ 0 & \text{otherwise} \end{cases}.$$

Like in the previous case, we split the flux function $\mathbf{F}$ as [31, 32, 33]

$$\mathbf{F}(\mathbf{U}_i) = \mathbf{F}_i^+ + \mathbf{F}_i^-, \ \mathbf{F}_i^\pm = \frac{1}{2}\left(\mathbf{F}(\mathbf{U}_i) \pm \alpha_i \mathbf{U}_i\right).$$

To compute the values of $\mathbf{F}^+$ and $\mathbf{F}^-$ at the cell interfaces $x_{i+\frac{1}{2}}$ we use, as already mentioned, the non-oscillatory generalized minmod reconstruction, that is:

$$\mathbf{F}_i^{\text{E}} := \mathbf{F}_i^+ + \frac{\Delta x}{2}(\mathbf{F}_x)_i^+, \ \mathbf{F}_i^{\text{W}} := \mathbf{F}_i^- - \frac{\Delta x}{2}(\mathbf{F}_x)_i^-,$$

where the slopes are computed using the generalized minmod limiter:

$$(\mathbf{F}_x)_i^\pm = \text{minmod}\left(\theta\frac{\mathbf{F}_i^\pm - \mathbf{F}_{i-1}^\pm}{\Delta x}, \frac{\mathbf{F}_{i+1}^\pm - \mathbf{F}_{i-1}^\pm}{2\Delta x}, \theta\frac{\mathbf{F}_{i+1}^\pm - \mathbf{F}_i^\pm}{\Delta x}\right).$$

Finally, the numerical fluxes are given by:

$$\hat{\boldsymbol{f}}_{i+\frac{1}{2}} = \mathbf{F}_i^{\text{E}} + \mathbf{F}_{i+1}^{\text{W}}.$$

Remark 4.2: In [8] the authors explain how the choice of the values of θ and of α_i is fundamental to prevent oscillations. It is convenient not to use a high value of $\theta \in [1, 2]$, because it can control the amount of numerical dissipation: larger values of θ allow sharper resolution of discontinuities, but may cause some oscillations. In the same way, it is convenient replacing α_i with $\max_i \alpha_i$, as this choice allows an increase in the quantity of numerical diffusion.

Discretization in time

First-order scheme

If we use a first-order scheme, to compute the numerical solutions at time t^{n+1} we discretize time by a first order Euler method:

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \Delta t \mathbf{F}(\mathbf{U}^n)_x.$$

Second-order scheme

High order schemes in time are obtained through use of suitable implicit-explicit Runge-Kutta schemes. We use the technique described in [3, 5].
We define

$$\mathbf{U}_{\text{I}} = \mathcal{S}(\mathbf{U}_*, \mathbf{U}_{\text{E}}, \Delta t),$$

the function that is the solution of the problem

$$\mathbf{U}_{\text{I}} = \mathbf{U}_* - \Delta t \mathbf{F}(\mathbf{U}_{\text{E}})_x,$$

where $\mathbf{U}_{\text{E}}$ is the vector of the explicit variables of the system.
The method can be implemented in the following way:

$$\begin{aligned}
\mathbf{U}_{\text{E}}^{(1)} &= \mathbf{U}^n \\
\mathbf{U}_{\text{I}}^{(1)} &= \mathcal{S}(\mathbf{U}^n, \mathbf{U}^n, \beta\Delta t) \\
\mathbf{U}_{\text{E}}^{(2)} &= \left(1 - \frac{\hat{c}}{\beta}\right)\mathbf{U}^n + \frac{\hat{c}}{\beta}\mathbf{U}_{\text{I}}^{(1)} \\
\mathbf{U}_{*}^{(2)} &= \frac{2\beta - 1}{\beta}\mathbf{U}^n + \frac{1-\beta}{\beta}\mathbf{U}_{\text{I}}^{(1)} \\
\mathbf{U}_{\text{I}}^{(2)} &= \mathcal{S}(\mathbf{U}_{*}^{(2)}, \mathbf{U}_{\text{E}}^{(2)}, \beta\Delta t),
\end{aligned}$$

where $\beta = 1 - \frac{1}{\sqrt{2}}$ and $\hat{c} = \frac{1}{2\beta}$.
Finally, the numerical solution is computed as $\mathbf{U}^{n+1} = \mathbf{U}_{\text{I}}^{(2)}$.

We apply such discretizations to the Isentropic Euler equations and Full Euler equations in 1D.

Isentropic Euler Equations

Discretization of the system (4.8)

Finite-difference methods with a first-order scheme in time

$$\begin{cases} \rho^{n+1} = \rho^n - \Delta t \hat{D}_x m^n \\ m^{n+1} = m^n - \Delta t \hat{D}_x \left(\left(\frac{m^2}{\rho} \right)^n + \frac{p^n}{\varepsilon^2} \right) \end{cases},$$
$$p^{n+1} = \left(\rho^{n+1} \right)^\gamma .$$

We can compute $\hat{D}_x \boldsymbol{f}$ using a first or a second-order scheme in space.

Full Euler Equations

Discretization of the system (4.12)

Finite-difference methods with a first-order scheme in time

$$\begin{cases} \rho^{n+1} = \rho^n - \Delta t \hat{D}_x m^n \\ m^{n+1} = m^n - \Delta t \hat{D}_x \left(\left(\frac{m^2}{\rho} \right)^n + \frac{p^n}{\varepsilon^2} \right) \\ E^{n+1} = E^n - \Delta t \hat{D}_x \left(h^n m^n \right) \end{cases},$$
$$p^{n+1} = (\gamma - 1) \left(E^{n+1} - \frac{1}{2} \varepsilon^2 \left(\frac{m^2}{\rho} \right)^{n+1} \right).$$

We can compute $\hat{D}_x \boldsymbol{f}$ using a first or a second-order scheme in space.

Finite-difference methods with a second-order scheme in time

In the case of Full Euler Equations we also used a second-order discretization in time.
First, we compute:

$$\begin{aligned} \rho_{\mathrm{E}}^{(1)} &= \rho^n, \\ m_{\mathrm{E}}^{(1)} &= m^n, \\ E_{\mathrm{E}}^{(1)} &= E^n, \\ p_{\mathrm{E}}^{(1)} &= p^n. \end{aligned}$$

Now, we can write the equations for the density:

$$\begin{aligned}
\rho_{\text{I}}^{(1)} &= \rho^n - \beta\Delta t\hat{D}_x m_{\text{E}}^{(1)},\\
\rho_{\text{E}}^{(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)\rho^n + \frac{\hat{c}}{\beta}\rho_{\text{I}}^{(1)},\\
\rho_{*}^{(2)} &= \frac{2\beta-1}{\beta}\rho^n + \frac{1-\beta}{\beta}\rho_{\text{I}}^{(1)},
\end{aligned}$$

for the momentum:

$$\begin{aligned}
m_{\text{I}}^{(1)} &= m^n - \beta\Delta t\hat{D}_x\left(\frac{{m_{\text{E}}^{(1)}}^2}{\rho_{\text{E}}^{(1)}} + \frac{p_{\text{E}}^{(1)}}{\varepsilon^2}\right),\\
m_{\text{E}}^{(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)m^n + \frac{\hat{c}}{\beta}m_{\text{I}}^{(1)},\\
m_{*}^{(2)} &= \frac{2\beta-1}{\beta}m^n + \frac{1-\beta}{\beta}m_{\text{I}}^{(1)},
\end{aligned}$$

and for the energy:

$$\begin{aligned}
E_{\text{I}}^{(1)} &= E^n - \beta\Delta t\hat{D}_x\left(h_{\text{E}}^{(1)}m_{\text{E}}^{(1)}\right),\\
E_{\text{E}}^{(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)E^n + \frac{\hat{c}}{\beta}E_{\text{I}}^{(1)},\\
E_{*}^{(2)} &= \frac{2\beta-1}{\beta}E^n + \frac{1-\beta}{\beta}E_{\text{I}}^{(1)}.
\end{aligned}$$

Finally, we can compute:

$$\begin{aligned}
\rho_{\text{I}}^{(2)} &= \rho_{*}^{(2)} - \beta\Delta t\hat{D}_x m_{\text{E}}^{(2)},\\
m_{\text{I}}^{(2)} &= m_{*}^{(2)} - \beta\Delta t\hat{D}_x\left(\frac{{m_{\text{E}}^{(2)}}^2}{\rho_{\text{E}}^{(2)}} + \frac{p_{\text{E}}^{(2)}}{\varepsilon^2}\right),\\
E_{\text{I}}^{(2)} &= E_{*}^{(2)} - \beta\Delta t\hat{D}_x\left(h_{\text{E}}^{(2)}m_{\text{E}}^{(2)}\right),
\end{aligned}$$

and thus: $\rho^{n+1} = \rho_{\text{I}}^{(2)}$, $m^{n+1} = m_{\text{I}}^{(2)}$ and $E^{n+1} = E_{\text{I}}^{(2)}$.

We can compute $\hat{D}_x\boldsymbol{f}$ using a first or a second-order scheme in space.

Such methods can easily be extended to the two-dimensional case.

4.4 FD shock-capturing methods for Gas Dynamics Equations in 2D

We discretize in 2D the hyperbolic systems (4.9) and (4.13) of conservation laws, that we can write in the following form:

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = 0.$$

In the system (4.9)

$$\mathbf{U} = \begin{pmatrix} \rho \\ m^{\mathbf{1}} \\ m^{\mathbf{2}} \end{pmatrix}, \ \mathbf{F}(\mathbf{U}) = \begin{pmatrix} m^{\mathbf{1}} \\ \frac{{m^{\mathbf{1}}}^2}{\rho} + \frac{p}{\varepsilon^2} \\ m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \end{pmatrix} \text{ and } \mathbf{G}(\mathbf{U}) = \begin{pmatrix} m^{\mathbf{2}} \\ m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \\ \frac{{m^{\mathbf{2}}}^2}{\rho} + \frac{p}{\varepsilon^2} \end{pmatrix},$$

while in the system (4.13)

$$\mathbf{U} = \begin{pmatrix} \rho \\ m^{\mathbf{1}} \\ m^{\mathbf{2}} \\ E \end{pmatrix}, \ \mathbf{F}(\mathbf{U}) = \begin{pmatrix} m^{\mathbf{1}} \\ \frac{{m^{\mathbf{1}}}^2}{\rho} + \frac{p}{\varepsilon^2} \\ m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \\ h m^{\mathbf{1}} \end{pmatrix} \text{ and } \mathbf{G}(\mathbf{U}) = \begin{pmatrix} m^{\mathbf{2}} \\ m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \\ \frac{{m^{\mathbf{2}}}^2}{\rho} + \frac{p}{\varepsilon^2} \\ h m^{\mathbf{2}} \end{pmatrix}.$$

Discretization in space

We discretize the space $\Omega = [0,1]^2$ through a uniform grid characterized by a spatial step $\Delta x = x_{i+1} - x_i$, $\forall i$ along $\vec{x}$ axis $\left(\Delta x = \frac{1}{N_1}\right)$ and by a spatial step $\Delta y = y_{j+1} - y_j$, $\forall j$ along $\vec{y}$ axis $\left(\Delta y = \frac{1}{N_2}\right)$. In general $N_1 = N_2 = N$. At even time step n we have N^2 cells, in which each cell is centered at (x_i, y_j), with $x_i = \left(i - \frac{1}{2}\right)\Delta x$ and $y_j = \left(j - \frac{1}{2}\right)\Delta y$, $i, j = 1, \ldots, N$. The following figure (Figure 4.2) shows the discretization of the domain Ω:

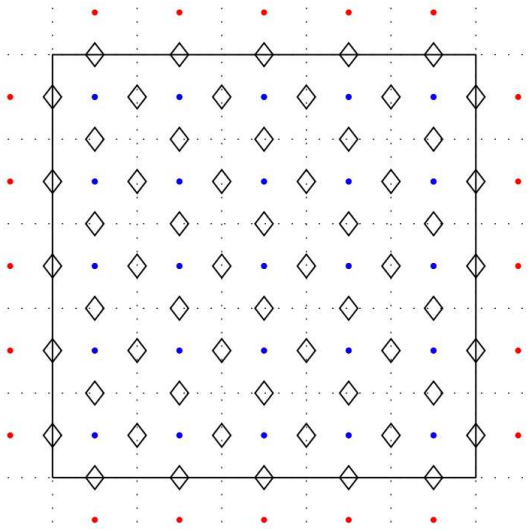


Figure 4.2: Discretization of the domain $[0,1]^2$ with $N = 5$

The values of the numerical solution on the internal points $\mathbf{U}_{i,j}(t) \approx \mathbf{U}(x_i, y_j, t)$ evolve in time using the following semi-discrete scheme:

$$\frac{\mathrm{d}\mathbf{U}_{i,j}}{\mathrm{d}t} = -\frac{\hat{\boldsymbol{f}}_{i+\frac{1}{2}} - \hat{\boldsymbol{f}}_{i-\frac{1}{2}}}{\Delta x} - \frac{\hat{\boldsymbol{g}}_{j+\frac{1}{2}} - \hat{\boldsymbol{g}}_{j-\frac{1}{2}}}{\Delta y}, \ i,j = 1,2,\ldots,N.$$

The numerical fluxes $\{\hat{\boldsymbol{f}}_{i+\frac{1}{2}}\}$ and $\{\hat{\boldsymbol{g}}_{j+\frac{1}{2}}\}$ are computed as follows. We denote by $\hat{D}_x\boldsymbol{f}$ the flux derivatives that will be discretized as the difference of the numerical fluxes between $i+\frac{1}{2}$ and $i-\frac{1}{2}$ and by $\hat{D}_y\boldsymbol{g}$ the flux derivatives that will be discretized as the difference of the numerical fluxes between $j+\frac{1}{2}$ and $j-\frac{1}{2}$.

First-order scheme

To compute the numerical fluxes in $\hat{D}_x\boldsymbol{f}$ and in $\hat{D}_y\boldsymbol{g}$ we use the local *Lax-Friedrichs fluxes*. We first split the flux functions $\mathbf{F}$ and $\mathbf{G}$ as

$$\mathbf{F}(\mathbf{U}_{i,j}) = \mathbf{F}^+_{i,j} + \mathbf{F}^-_{i,j}, \ \mathbf{F}^\pm_{i,j} = \frac{1}{2}\left(\mathbf{F}(\mathbf{U}_{i,j}) + \alpha_{i,j}\mathbf{U}_{i,j}\right),$$

$$\mathbf{G}(\mathbf{U}_{i,j}) = \mathbf{G}^+_{i,j} + \mathbf{G}^-_{i,j}, \ \mathbf{G}^\pm_{i,j} = \frac{1}{2}\left(\mathbf{G}(\mathbf{U}_{i,j}) + \beta_{i,j}\mathbf{U}_{i,j}\right),$$

where $\alpha_{i,j} = |u_{ij}| + \frac{c_{ij}}{\varepsilon}$ and $\beta_{i,j} = |v_{ij}| + \frac{c_{ij}}{\varepsilon}$, u and v are the velocities of the fluid along the x-axis and along the y-axis respectively, $c_{ij} = \sqrt{\frac{\gamma p_{ij}}{\rho_{ij}}}$ is the

sound speed and ε is the dimensionless reference Mach number.
The numerical fluxes are given by:

$$\hat{\boldsymbol{f}}_{i+\frac{1}{2}} = \mathbf{F}_i^+ + \mathbf{F}_{i+1}^-,\ \hat{\boldsymbol{g}}_{i+\frac{1}{2}} = \mathbf{G}_j^+ + \mathbf{G}_{j+1}^-,$$

that are:

$$\hat{\boldsymbol{f}}_{i+\frac{1}{2},j} = \frac{\mathbf{F}(\mathbf{U}_{i+1,j}) + \mathbf{F}(\mathbf{U}_{i,j})}{2} - \alpha_{i+\frac{1}{2},j}\frac{\mathbf{U}_{i+1,j} - \mathbf{U}_{i,j}}{2},$$

$$\hat{\boldsymbol{g}}_{i,j+\frac{1}{2}} = \frac{\mathbf{G}(\mathbf{U}_{i,j+1}) + \mathbf{G}(\mathbf{U}_{i,j})}{2} - \beta_{i,j+\frac{1}{2}}\frac{\mathbf{U}_{i,j+1} - \mathbf{U}_{i,j}}{2}.$$

In classical explicit schemes are chosen:

$$\alpha_{i+\frac{1}{2},j} = \max\left\{\alpha_{ij}, \alpha_{i+1,j}\right\},$$

$$\beta_{i,j+\frac{1}{2}} = \max\left\{\beta_{ij}, \beta_{i,j+1}\right\}.$$

Then the discrete derivatives are given by:

$$\hat{D}_x\boldsymbol{f} = \frac{\hat{\boldsymbol{f}}_{i+\frac{1}{2},j} - \hat{\boldsymbol{f}}_{i-\frac{1}{2},j}}{\Delta x},\ \hat{D}_y\boldsymbol{g} = \frac{\hat{\boldsymbol{g}}_{i,j+\frac{1}{2}} - \hat{\boldsymbol{g}}_{i,j-\frac{1}{2}}}{\Delta y},$$

where Δx and Δy are the space step discretization.

Second-order scheme

In the second order scheme the numerical fluxes $\hat{\boldsymbol{f}}_{i+\frac{1}{2},j}$ and $\hat{\boldsymbol{g}}_{i,j+\frac{1}{2}}$ are computed as follows.
Like in the first-order scheme, we split the flux functions $\mathbf{F}$ and $\mathbf{G}$ as

$$\mathbf{F}(\mathbf{U}_{i,j}) = \mathbf{F}_{i,j}^+ + \mathbf{F}_{i,j}^-,\ \mathbf{F}_{i,j}^\pm = \frac{1}{2}\left(\mathbf{F}(\mathbf{U}_{i,j}) + \alpha_{i,j}\mathbf{U}_{i,j}\right),$$

$$\mathbf{G}(\mathbf{U}_{i,j}) = \mathbf{G}_{i,j}^+ + \mathbf{G}_{i,j}^-,\ \mathbf{G}_{i,j}^\pm = \frac{1}{2}\left(\mathbf{G}(\mathbf{U}_{i,j}) + \beta_{i,j}\mathbf{U}_{i,j}\right).$$

To evaluate the values of $\mathbf{F}^\pm$ and $\mathbf{G}^\pm$ at the cell interfaces $\left(x_{i+\frac{1}{2}}, y_j\right)$ and $\left(x_i, y_{j+\frac{1}{2}}\right)$ respectively, we use a non-oscillatory generalized minmod reconstruction, that are:

$$\mathbf{F}_{i,j}^{\mathrm{E}} := \mathbf{F}_{i,j}^+ + \frac{\Delta x}{2}(\mathbf{F}_x)_{i,j}^+,\ \mathbf{F}_{i,j}^{\mathrm{W}} := \mathbf{F}_{i,j}^- - \frac{\Delta x}{2}(\mathbf{F}_x)_{i,j}^-,$$

$$\mathbf{G}^{\mathrm{N}}_{i,j} := \mathbf{G}^{+}_{i,j} + \frac{\Delta y}{2}(\mathbf{G}_y)^{+}_{i,j},\ \mathbf{G}^{\mathrm{S}}_{i,j} := \mathbf{G}^{-}_{i,j} - \frac{\Delta y}{2}(\mathbf{G}_y)^{-}_{i,j},$$

where the slopes are computed using the generalized minmod limiter:

$$(\mathbf{F}_x)^{\pm}_{i,j} = \mathrm{minmod}\left(\theta\frac{\mathbf{F}^{\pm}_{i,j} - \mathbf{F}^{\pm}_{i-1,j}}{\Delta x}, \frac{\mathbf{F}^{\pm}_{i+1,j} - \mathbf{F}^{\pm}_{i-1,j}}{2\Delta x}, \theta\frac{\mathbf{F}^{\pm}_{i+1,j} - \mathbf{F}^{\pm}_{i,j}}{\Delta x}\right),$$

$$(\mathbf{G}_y)^{\pm}_{i,j} = \mathrm{minmod}\left(\theta\frac{\mathbf{G}^{\pm}_{i,j} - \mathbf{G}^{\pm}_{i,j-1}}{\Delta y}, \frac{\mathbf{G}^{\pm}_{i,j+1} - \mathbf{G}^{\pm}_{i,j-1}}{2\Delta y}, \theta\frac{\mathbf{G}^{\pm}_{i,j+1} - \mathbf{G}^{\pm}_{i,j}}{\Delta y}\right).$$

The numerical fluxes are given by:

$$\hat{\boldsymbol{f}}_{i+\frac{1}{2},j} = \mathbf{F}^{\mathrm{E}}_{i,j} + \mathbf{F}^{\mathrm{W}}_{i+1,j},\ \hat{\boldsymbol{g}}_{i,j+\frac{1}{2}} = \mathbf{G}^{\mathrm{N}}_{i,j} + \mathbf{G}^{\mathrm{S}}_{i,j+1}.$$

Remark 4.3: As already discussed in the 1D case it is convenient to replace $\alpha_{i,j}$ with $\max_{i,j}\alpha_{i,j}$ and $\beta_{i,j}$ with $\max_{i,j}\beta_{i,j}$ to prevent oscillations.

Discretization in time

First-order scheme

If we use a first-order scheme, to compute the numerical solutions at time t^{n+1} we discretize time by a first order Euler method:

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \Delta t\mathbf{F}(\mathbf{U}^n)_x - \Delta t\mathbf{G}(\mathbf{U}^n)_y.$$

Second-order scheme

High order schemes in time can be constructed as follows, according to the scheme present in [3, 5].
We define

$$\mathbf{U}_{\mathrm{I}} = \mathcal{S}(\mathbf{U}_*, \mathbf{U}_{\mathrm{E}}, \Delta t),$$

the function that is the solution of the problem

$$\mathbf{U}_{\mathrm{I}} = \mathbf{U}_* - \Delta t\mathbf{F}(\mathbf{U}_{\mathrm{E}})_x - \Delta t\mathbf{G}(\mathbf{U}_{\mathrm{E}})_y,$$

where $\mathbf{U}_{\mathrm{E}}$ is the vector of explicit variables of the system.

Then, the method can be implemented in the following way:

$$\begin{aligned}
\mathbf{U}_{\mathrm{E}}^{(1)} &= \mathbf{U}^n \\
\mathbf{U}_{\mathrm{I}}^{(1)} &= \mathcal{S}(\mathbf{U}^n, \mathbf{U}^n, \beta\Delta t) \\
\mathbf{U}_{\mathrm{E}}^{(2)} &= \left(1 - \frac{\hat{c}}{\beta}\right)\mathbf{U}^n + \frac{\hat{c}}{\beta}\mathbf{U}_{\mathrm{I}}^{(1)} \\
\mathbf{U}_{*}^{(2)} &= \frac{2\beta - 1}{\beta}\mathbf{U}^n + \frac{1-\beta}{\beta}\mathbf{U}_{\mathrm{I}}^{(1)} \\
\mathbf{U}_{\mathrm{I}}^{(2)} &= \mathcal{S}(\mathbf{U}_{*}^{(2)}, \mathbf{U}_{\mathrm{E}}^{(2)}, \beta\Delta t),
\end{aligned}$$

where $\beta = 1 - \frac{1}{\sqrt{2}}$ and $\hat{c} = \frac{1}{2\beta}$.
At the end, the numerical solution is computed as $\mathbf{U}^{n+1} = \mathbf{U}_{\mathrm{I}}^{(2)}$.

We apply such discretizations to the Isentropic Euler equations and Full Euler equations in 2D.

Isentropic Euler Equations

Discretization of the system (4.9)

Finite-difference methods with a first-order scheme in time

$$\begin{cases}
\rho^{n+1} = \rho^n - \Delta t \hat{D}_x m^{\mathbf{1}^n} - \Delta t \hat{D}_y m^{\mathbf{2}^n} \\
m^{\mathbf{1}^{n+1}} = m^{\mathbf{1}^n} - \Delta t \hat{D}_x \left(\left(\frac{m^{\mathbf{1}^2}}{\rho} \right)^n + \frac{p^n}{\varepsilon^2} \right) - \Delta t \hat{D}_y \left(m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \right)^n \\
m^{\mathbf{2}^{n+1}} = m^{\mathbf{2}^n} - \Delta t \hat{D}_x \left(m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \right)^n - \Delta t \hat{D}_y \left(\left(\frac{m^{\mathbf{2}^2}}{\rho} \right)^n + \frac{p^n}{\varepsilon^2} \right)
\end{cases},$$

$$p^{n+1} = \left(\rho^{n+1}\right)^{\gamma}.$$

We can compute $\hat{D}_x \boldsymbol{f}$ and $\hat{D}_y \boldsymbol{g}$ using a first or a second-order scheme in space.

Full Euler Equations

Discretization of the system (4.13)

Finite-Difference methods with a first-order scheme in time

We proceed as in the 1D case:

$$\begin{cases} \rho^{n+1} = \rho^n - \Delta t \hat{D}_x m^{\mathbf{1}^n} - \Delta t \hat{D}_y m^{\mathbf{2}^n} \\ m^{\mathbf{1}^{n+1}} = m^{\mathbf{1}^n} - \Delta t \hat{D}_x \left(\left(\frac{m^{\mathbf{1}^2}}{\rho} \right)^n + \frac{p^n}{\varepsilon^2} \right) - \Delta t \hat{D}_y \left(m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \right)^n \\ m^{\mathbf{2}^{n+1}} = m^{\mathbf{2}^n} - \Delta t \hat{D}_x \left(m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \right)^n - \Delta t \hat{D}_y \left(\left(\frac{m^{\mathbf{2}^2}}{\rho} \right)^n + \frac{p^n}{\varepsilon^2} \right) \\ E^{n+1} = E^n - \Delta t \hat{D}_x \left(h^n m^{\mathbf{1}^n} \right) - \Delta t \hat{D}_y \left(h^n m^{\mathbf{2}^n} \right) \end{cases},$$

$$p^{n+1} = (\gamma - 1) \left(E^{n+1} - \frac{1}{2} \varepsilon^2 \left(\frac{m^{\mathbf{1}^2}}{\rho} + \frac{m^{\mathbf{2}^2}}{\rho} \right)^{n+1} \right).$$

We can compute $\hat{D}_x \boldsymbol{f}$ and $\hat{D}_y \boldsymbol{g}$ using a first or a second-order scheme in space.

Finite-difference methods with a second-order scheme in time

First we compute:

$$\begin{aligned} \rho_{\mathrm{E}}^{(1)} &= \rho^n, \\ m_{\mathrm{E}}^{\mathbf{1}(1)} &= m^{\mathbf{1}^n}, \\ m_{\mathrm{E}}^{\mathbf{2}(1)} &= m^{\mathbf{2}^n}, \\ E_{\mathrm{E}}^{(1)} &= E^n, \\ p_{\mathrm{E}}^{(1)} &= p^n. \end{aligned}$$

Now, we can write the equations for the density:

$$\begin{aligned} \rho_{\mathrm{I}}^{(1)} &= \rho^n - \beta \Delta t \hat{D}_x m_{\mathrm{E}}^{\mathbf{1}(1)} - \beta \Delta t \hat{D}_y m_{\mathrm{E}}^{\mathbf{2}(1)}, \\ \rho_{\mathrm{E}}^{(2)} &= \left(1 - \frac{\hat{c}}{\beta} \right) \rho^n + \frac{\hat{c}}{\beta} \rho_{\mathrm{I}}^{(1)}, \\ \rho_{*}^{(2)} &= \frac{2\beta - 1}{\beta} \rho^n + \frac{1 - \beta}{\beta} \rho_{\mathrm{I}}^{(1)}, \end{aligned}$$

for the momentums:

$$
\begin{aligned}
m_{\mathrm{I}}^{\mathbf{1}(1)} &= m^{\mathbf{1}^n} - \beta\Delta t\hat{D}_x\left(\frac{m_{\mathrm{E}}^{\mathbf{1}(1)^2}}{\rho_{\mathrm{E}}^{(1)}} + \frac{p_{\mathrm{E}}^{(1)}}{\varepsilon^2}\right) - \beta\Delta t\hat{D}_y\left(m_{\mathrm{E}}^{\mathbf{2}(1)}\frac{m_{\mathrm{E}}^{\mathbf{1}(1)}}{\rho_{\mathrm{E}}^{(1)}}\right),\\
m_{\mathrm{I}}^{\mathbf{2}(1)} &= m^{\mathbf{2}^n} - \beta\Delta t\hat{D}_x\left(m_{\mathrm{E}}^{\mathbf{1}(1)}\frac{m_{\mathrm{E}}^{\mathbf{2}(1)}}{\rho_{\mathrm{E}}^{(1)}}\right) - \beta\Delta t\hat{D}_y\left(\frac{m_{\mathrm{E}}^{\mathbf{2}(1)^2}}{\rho_{\mathrm{E}}^{(1)}} + \frac{p_{\mathrm{E}}^{(1)}}{\varepsilon^2}\right),\\
m_{\mathrm{E}}^{\mathbf{1}(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)m^{\mathbf{1}^n} + \frac{\hat{c}}{\beta}m_{\mathrm{I}}^{\mathbf{1}(1)},\\
m_{\mathrm{E}}^{\mathbf{2}(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)m^{\mathbf{2}^n} + \frac{\hat{c}}{\beta}m_{\mathrm{I}}^{\mathbf{2}(1)},\\
m_{*}^{\mathbf{1}(2)} &= \frac{2\beta-1}{\beta}m^{\mathbf{1}^n} + \frac{1-\beta}{\beta}m_{\mathrm{I}}^{\mathbf{1}(1)},\\
m_{*}^{\mathbf{2}(2)} &= \frac{2\beta-1}{\beta}m^{\mathbf{2}^n} + \frac{1-\beta}{\beta}m_{\mathrm{I}}^{\mathbf{2}(1)},
\end{aligned}
$$

and for the energy:

$$
\begin{aligned}
E_{\mathrm{I}}^{(1)} &= E^n - \beta\Delta t\hat{D}_x\left(h^n m_{\mathrm{E}}^{\mathbf{1}(1)}\right) - \beta\Delta t\hat{D}_y\left(h^n m_{\mathrm{E}}^{\mathbf{2}(1)}\right),\\
E_{\mathrm{E}}^{(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)E^n + \frac{\hat{c}}{\beta}E_{\mathrm{I}}^{(1)},\\
E_{*}^{(2)} &= \frac{2\beta-1}{\beta}E^n + \frac{1-\beta}{\beta}E_{\mathrm{I}}^{(1)}.
\end{aligned}
$$

Finally, we can compute:

$$
\begin{aligned}
\rho_{\mathrm{I}}^{(2)} &= \rho_{*}^{(2)} - \beta\Delta t\hat{D}_x m_{\mathrm{E}}^{\mathbf{1}(2)} - \beta\Delta t\hat{D}_y m_{\mathrm{E}}^{\mathbf{2}(2)},\\
m_{\mathrm{I}}^{\mathbf{1}(2)} &= m_{*}^{\mathbf{1}(2)} - \beta\Delta t\hat{D}_x\left(\frac{m_{\mathrm{E}}^{\mathbf{1}(2)^2}}{\rho_{\mathrm{E}}^{(2)}} + \frac{p_{\mathrm{E}}^{(2)}}{\varepsilon^2}\right) - \beta\Delta t\hat{D}_y\left(m_{\mathrm{E}}^{\mathbf{2}(2)}\frac{m_{\mathrm{E}}^{\mathbf{1}(2)}}{\rho_{\mathrm{E}}^{(2)}}\right),\\
m_{\mathrm{I}}^{\mathbf{2}(2)} &= m_{*}^{\mathbf{2}(2)} - \beta\Delta t\hat{D}_x\left(m_{\mathrm{E}}^{\mathbf{1}(2)}\frac{m_{\mathrm{E}}^{\mathbf{2}(2)}}{\rho_{\mathrm{E}}^{(2)}}\right) - \beta\Delta t\hat{D}_y\left(\frac{m_{\mathrm{E}}^{\mathbf{2}(2)^2}}{\rho_{\mathrm{E}}^{(2)}} + \frac{p_{\mathrm{E}}^{(2)}}{\varepsilon^2}\right),\\
E_{\mathrm{I}}^{(2)} &= E_{*}^{(2)} - \beta\Delta t\hat{D}_x\left(h^n m_{\mathrm{E}}^{\mathbf{1}(2)}\right) - \beta\Delta t\hat{D}_y\left(h^n m_{\mathrm{E}}^{\mathbf{2}(2)}\right),
\end{aligned}
$$

and thus: $\rho^{n+1} = \rho_{\mathrm{I}}^{(2)}$, $m^{\mathbf{1}^{n+1}} = m_{\mathrm{I}}^{\mathbf{1}(2)}$, $m^{\mathbf{2}^{n+1}} = m_{\mathrm{I}}^{\mathbf{2}(2)}$ and $E^{n+1} = E_{\mathrm{I}}^{(2)}$.

We can compute $\hat{D}_x\boldsymbol{f}$ and $\hat{D}_y\boldsymbol{g}$ using a first or a second-order scheme in space.

Boundary conditions The boundary conditions depend on the particular test under consideration, and thus on the initial conditions. We have dealt with the case in which there is the presence of a *wall*, that includes:

- *Dirichlet boundary conditions:* in general on the x component of the momentum (because the fluid comes back as soon as it meets an obstacle, like a wall)
- *Neumann boundary conditions:* in general on the density, on the pressure (and thus on the energy) and on the y component of the momentum (for the 2D problems - to prevent the fluid from leaking from the duct in which it moves)

We close this Chapter dedicated to the finite-difference methods, applying it to the case in which the domain is arbitrary. As already seen in Chapter 1, in this case it is possible that the border does not adapt to the grid. It is therefore necessary a particular discretization of the boundary conditions.

4.5 Coco-Russo method for the Full Euler Equations on arbitrary domains

In this Section we present an application of the Coco-Russo method to the Full Euler equations. This application is present in [8], in which the authors describe how to construct a second-order finite-difference shock-capturing method for the numerical solution of the Euler equations of gas dynamics on arbitrary two dimensional domain Ω. This domain can be fixed or moving. We limit ourselves to dealing the case of a fixed domain.

A similar work is presented in [4], in which the authors have developed a finite difference method based on the discretization of the equations on a normal Cartesian grid and on the application of Lagrange interpolation with a filter for the detection of discontinuities that permits a data dependent extrapolation, with higher order at smooth regions and essentially non oscillatory properties near discontinuities. In [8] the authors use a different extrapolation technique, which is similar to that adopted in the context of

elliptic problems [9, 10], and that we present below.

We discretize the domain Ω with a uniform Cartesian grid with a size mesh h along the two axes respectively ($N = \frac{1}{h}$). We identify internal and ghost points. We indicate with N_{I} and N_{G} the number of internal points and of ghost points respectively. The following figure (Figure 4.3) shows the discretization of a square domain with a circular obstacle:

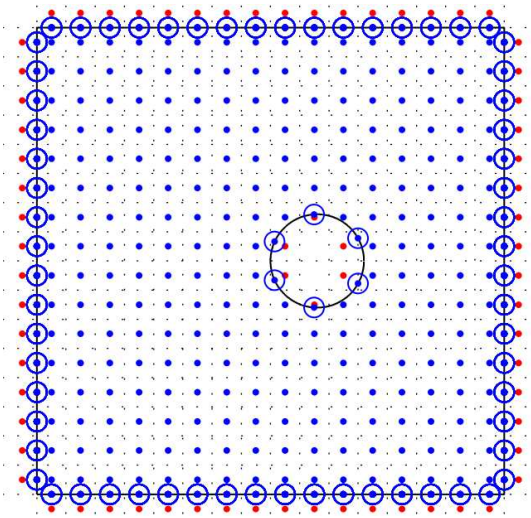


Figure 4.3: Discretization of a domain $\Omega \subset [0,1]^2$ with $N = 16$: the blue dots are the internal point, the red dots are the ghost points. We impose the boundary conditions in the blue circles

We discretize the Full Euler equations for each internal point (see system (4.13)). As already seen in Chapter 1 for the Poisson equation, the discretized equations on the internal points involve the values of the numerical solution on the neighboring ghost points. In this way we obtain a system of $4N_{\text{I}}$ equations with $4(N_{\text{I}} + N_{\text{G}})$ unknowns. To close the system we proceed as in the Coco-Russo method, thus we write the $4N_{\text{G}}$ equations for the N_{G} ghost points, which come from the discretization of the boundary conditions. For each ghost point G we consider its projection F on the border of the domain and in this point F we impose the boundary conditions appropriately discretized on the density, on the momentums and on the energy.

- For each ghost point G outside the square domain (whose projection is precisely on the border of the unit square) the boundary conditions are imposed in the usual way:

1. Dirichlet: $\mathrm{U}(x_{\mathrm{G}}, y_{\mathrm{G}}) = -\mathrm{U}(x_{\mathrm{I}}, y_{\mathrm{I}})$ (on the x component of the momentum),
2. Neumann: $\mathrm{U}(x_{\mathrm{G}}, y_{\mathrm{G}}) = \mathrm{U}(x_{\mathrm{I}}, y_{\mathrm{I}})$ (on the density, pressure, energy and y component of the momentum),

where $\mathrm{U}(x_{\mathrm{G}}, y_{\mathrm{G}})$ is the value of the numerical solution on the ghost point $\mathrm{G}(x_{\mathrm{G}}, y_{\mathrm{G}})$ and $\mathrm{U}(x_{\mathrm{I}}, y_{\mathrm{I}})$ is the value of the numerical solution on the neighbor internal point $\mathrm{I}(x_{\mathrm{I}}, y_{\mathrm{I}})$.

- For each ghost point G inside the circular obstacle we describe how the boundary conditions are set and discretized in the case of a fixed boundary. For more details see [8].

We denote by $\mathbf{n}$ and τ the normal and tangential unit vectors to the boundary $\partial\Omega$, respectively, and by κ the signed curvature of $\partial\Omega$ (see Figure 4.4). In the case of a circular domain (our obstacle) $\kappa = \frac{1}{R}$, where R is the radius of the circumference.

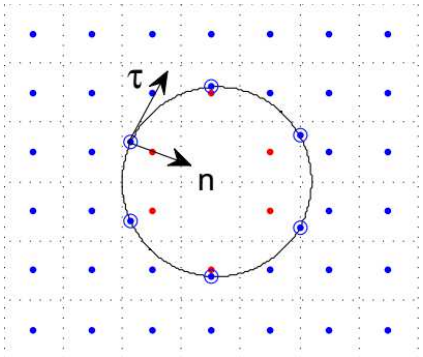


Figure 4.4: Normal and tangential unit vectors to the boundary $\partial\Omega$ in the projection of the ghost point

We denote by $\mathbf{u} := (u, v)^{\mathrm{T}}$ the velocity vector and by $u_n = \mathbf{u} \cdot \mathbf{n}$ and $u_\tau = \mathbf{u} \cdot \tau$ its normal and tangential components, respectively.

We consider the simpler case of a fixed boundary and set four boundary conditions on the normal and tangential components of the velocity, pressure and density.

1. Condition on the normal component of the velocity

$$u_n = 0$$

2. Condition on the pressure

$$\frac{\partial p}{\partial \mathbf{n}} = \kappa \rho u_\tau^2$$

3. Condition on the density

$$\frac{\partial \rho}{\partial \mathbf{n}} = \frac{\rho}{\gamma p}\frac{\partial p}{\partial \mathbf{n}}$$

4. Condition on the tangential component of the velocity

$$\frac{\partial u_\tau}{\partial \mathbf{n}} = -\kappa u_\tau$$

We see now how it is possible to discretize numerically the boundary conditions ***1.***, ***2.***, ***3.*** and ***4.***. In [8] the authors indicated with $\mathcal{Q}[\psi;\mathcal{S}]$ the biquadratic interpolant (described in Chapter 1) of a grid function $\psi_{i,j}$ in the stencil $\mathcal{S}$. The discretization of the boundary conditions is obtained by approximating in the equations ***1.***, ***2.***, ***3.*** and ***4.*** the values of pressure, density, normal and tangential components of the velocity at point F with the corresponding biquadratic interpolants:

$$\mathcal{Q}[u_n;\mathcal{S}_{\mathrm{G}}](\mathrm{F}) = 0 \tag{4.14}$$

$$\frac{\partial \mathcal{Q}[p;\mathcal{S}_{\mathrm{G}}]}{\partial \mathbf{n}}(\mathrm{F}) = \kappa \mathcal{Q}[\rho;\mathcal{S}_{\mathrm{G}}](\mathrm{F})\left(\mathcal{Q}[u_\tau;\mathcal{S}_{\mathrm{G}}](\mathrm{F})\right)^2 \tag{4.15}$$

$$\frac{\partial \mathcal{Q}[\rho;\mathcal{S}_{\mathrm{G}}]}{\partial \mathbf{n}}(\mathrm{F}) = \frac{\mathcal{Q}[\rho;\mathcal{S}_{\mathrm{G}}](\mathrm{F})}{\gamma \mathcal{Q}[p;\mathcal{S}_{\mathrm{G}}](\mathrm{F})}\frac{\partial \mathcal{Q}[p;\mathcal{S}_{\mathrm{G}}]}{\partial \mathbf{n}}(\mathrm{F}) \tag{4.16}$$

$$\frac{\partial \mathcal{Q}[u_\tau;\mathcal{S}_{\mathrm{G}}]}{\partial \mathbf{n}}(\mathrm{F}) = -\kappa \mathcal{Q}[u_\tau;\mathcal{S}_{\mathrm{G}}](\mathrm{F}) \tag{4.17}$$

We can observe that we cannot solve separately the 4×4 systems obtained from the equations (4.14), (4.15), (4.16) and (4.17) for each ghost point G, since each 4×4 system may be coupled with the corresponding 4×4 systems obtained at other ghost points.

To solve this problem, in [8], the authors use an iterative scheme. They first transform the system of boundary conditions into a time-dependent problem with a fictitious time σ:

$$\begin{cases} \frac{\partial u_n}{\partial \sigma} = -u_n \\ \frac{\partial u_\tau}{\partial \sigma} = -\mu_2\left(\frac{\partial u_\tau}{\partial \mathbf{n}} + \kappa u_\tau\right) \\ \frac{\partial p}{\partial \sigma} = -\mu_1\left(\frac{\partial p}{\partial \mathbf{n}} - \kappa\rho u_\tau^2\right) \\ \frac{\partial \rho}{\partial \sigma} = -\mu_3\left(\frac{\partial \rho}{\partial \mathbf{n}} - \frac{1}{c_s^2}\frac{\partial p}{\partial \mathbf{n}}\right) \end{cases}, \tag{4.18}$$

where μ_1, μ_2 and μ_3 are suitable constants. Then, to obtain the iterative scheme, they discretize the system (4.18) in space and time. The partial derivatives with respect to σ are discretized at the ghost point G using the first-order forward Euler method. Therefore, the iterative scheme is:

$$
\begin{aligned}
u_n^{(m+1)}(\mathrm{G}) &= u_n^{(m)}(\mathrm{G}) - \Delta\sigma\mathcal{Q}[u_n^{(m)};\mathcal{S}_\mathrm{G}](\mathrm{F}),\\
u_\tau^{(m+1)}(\mathrm{G}) &= u_\tau^{(m)}(\mathrm{G}) - \mu_2\Delta\sigma\left(\frac{\partial\mathcal{Q}[u_\tau^{(m)};\mathcal{S}_\mathrm{G}](\mathrm{F})}{\partial\mathbf{n}} + \kappa\mathcal{Q}[u_\tau^{(m)};\mathcal{S}_\mathrm{G}](\mathrm{F})\right),\\
p^{(m+1)}(\mathrm{G}) &= p^{(m)}(\mathrm{G}) - \mu_1\Delta\sigma\left(\frac{\partial\mathcal{Q}[p^{(m)};\mathcal{S}_\mathrm{G}]}{\partial\mathbf{n}}(\mathrm{F}) - \kappa\mathcal{Q}[\rho^{(m)};\mathcal{S}_\mathrm{G}](\mathrm{F})\left(\mathcal{Q}[u_\tau^{(m)};\mathcal{S}_\mathrm{G}](\mathrm{F})\right)^2\right),\\
\rho^{(m+1)}(\mathrm{G}) &= \rho^{(m)}(\mathrm{G}) - \mu_3\Delta\sigma\left(\frac{\partial\mathcal{Q}[\rho^{(m)};\mathcal{S}_\mathrm{G}]}{\partial\mathbf{n}}(\mathrm{F}) - \frac{\mathcal{Q}[\rho^{(m)};\mathcal{S}_\mathrm{G}](\mathrm{F})}{\gamma\mathcal{Q}[p^{(m)};\mathcal{S}_\mathrm{G}](\mathrm{F})}\frac{\partial\mathcal{Q}[p^{(m)};\mathcal{S}_\mathrm{G}]}{\partial\mathbf{n}}(\mathrm{F})\right),
\end{aligned}
$$

in which the time step $\Delta\sigma$ and the constants μ_i, $i = 1, 2, 3$, are chosen so as to satisfy the CFL conditions:

$$\Delta\sigma < 1 \text{ and } \mu_i\Delta\sigma < \min(\Delta x, \Delta y),\ i = 1, 2, 3.$$

However, the FD methods described in this Chapter have a limit: the time step must satisfy the CFL condition. In the following Chapter we will present a class of numerical methods for systems of conservation laws which, as we shall see, will be stable also for larger time steps.

5

Semi-Implicit Methods for the Gas Dynamics Equations

In this Chapter we present the idea of [3] for the construction of alternative methods for the Gas Dynamics that do not suffer the usual CFL stability restriction of explicit schemes proposed in the previous Chapter. Such methods are known as *Semi-Implicit methods*.

Semi-Implicit methods

All hyperbolic systems of conservation laws develop waves that propagate at finite speeds. Therefore, if we want to accurately compute all the waves in a hyperbolic system, is necessary to resolve all the space and time scales that characterize it.

We have seen that in the explicit methods for the hyperbolic systems of conservation laws the time step must satisfy the CFL condition (4.4). If the order of accuracy in time is the same of the order of accuracy in space, accuracy and stability restrictions are almost the same, and the systems are *not stiff*. However, there are cases in which some of the waves are not particularly relevant and one is not interested in resolving them. For example, if we consider the Euler equations of compressible gas dynamics, in the low Mach number regimes the acoustic waves carry a negligible amount of energy and thus have a negligible influence on the solution, but since classical CFL condition on the time step is determined by the acoustic waves, they impose a very restrictive CFL condition if one uses an explicit scheme to solve them.

The system becomes *stiff*. This restriction on the time step results in an increasingly large computational time for smaller and smaller ε.

Implicit time discretization To remedy the problem of stiffness, it can possible consider implicit time discretizations, which avoid the acoustic CFL restriction and allow the use of a much larger time step. However, the implicit schemes to solve the systems present two problems.

- They are more difficult to solve because they are highly nonlinear.
- They may introduce an excessive numerical dissipation on the slow wave, causing loss of accuracy.

To solve the problems given by both explicit and implicit techniques, the strategy is to use *semi-implicit methods*. A semi-implicit scheme avoids the CFL condition for the acoustic waves and maintains a good accuracy on the more important features of the flow.

In [3] the authors propose a family of second-order accurate schemes for the numerical solution of Euler equations of gas dynamics that are (linearly) implicit in the acoustic waves, eliminating the acoustic CFL restriction on the time step. In [3] the explicit differential operators in space relative to convective or material speeds are discretized by local Lax-Friedrics fluxes and the linear implicit operators, pertaining to acoustic waves, are discretized by central differences. The results show that these schemes do not introduce excessive numerical dissipation at low Mach number providing an accurate solution in such regimes. They perform reasonably well also when the Mach number are not too small.

Therefore, if for the standard explicit schemes is necessary that $\mathrm{CFL} \leq \frac{1}{2}$, for these semi-implicit schemes CFL can be quite large than 1. However, since the material wave is treated explicitly, we have a restriction on the material CFL, that is, $\mathrm{CFL}_u \leq \frac{1}{2}$, where

$$\mathrm{CFL}_u = \frac{\max|\mathbf{u}|\Delta t}{\Delta x} = \frac{\mathrm{CFL}\max|\mathbf{u}|}{\lambda_{\max}}.$$

5.1 Semi-Implicit methods for Gas Dynamics Equations in 1D

We discretize in 1D the hyperbolic systems (4.8) and (4.12) of conservation laws, that we can write in the following form:

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0.$$

We discretize the space as in the Section 4.3.

In the case of semi-implicit methods, we denote by $\hat{D}_x \boldsymbol{f}$ the flux derivatives for non-stiff terms that will be discretized as the difference of the numerical fluxes between $i + \frac{1}{2}$ and $i - \frac{1}{2}$ and by $D_x \boldsymbol{f}$ the flux derivatives for stiff terms discretized with a centered scheme. Both in the case of a first-order scheme and in the case of a second-order scheme in space, for non-stiff terms, the numerical fluxes $\{\hat{\boldsymbol{f}}_{i+\frac{1}{2}}\}$ are computed as in the Section 4.3.

Therefore, for non-stiff terms, the discrete derivatives are given by:

$$\hat{D}_x \boldsymbol{f} = \frac{\hat{\boldsymbol{f}}_{i+\frac{1}{2}} - \hat{\boldsymbol{f}}_{i-\frac{1}{2}}}{\Delta x}.$$

For stiff terms, instead, we have:

$$D_x \boldsymbol{f} = \frac{\boldsymbol{f}_{i+\frac{1}{2}} - \boldsymbol{f}_{i-\frac{1}{2}}}{\Delta x} = \frac{\boldsymbol{f}_{i+1} - \boldsymbol{f}_{i-1}}{2\Delta x}.$$

Since the acoustic waves are treated implicitly, we use α proportional to the material speed. We expect that for very low Mach number, $\alpha \approx |u|$ should be sufficient, while for Mach number larger than one, the speed of sound is bounded by the fluid speed. For this reason, we choose

$$\alpha_i = |u_i|,$$

for low Mach number.

Discretization in time

First-order scheme

If we use a first-order scheme, to compute the numerical solutions at time t^{n+1} we discretize time by a first order Euler method: stiff terms are evaluated at time t^{n+1}, while non-stiff terms are evaluated at time $t^n = n\Delta t$, $n = 0, 1, \ldots$:

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \Delta t \mathbf{F}(\mathbf{U}_{\text{E}}^n, \mathbf{U}_{\text{I}}^{n+1})_x,$$

where $\mathbf{U}_{\text{E}}$ is the vector of non-stiff terms and $\mathbf{U}_{\text{I}}$ is the vector of stiff terms of the system.

Second-order scheme

High order schemes in time are obtained with the same technique proposed by [3, 5] and described in the previous chapter for explicit methods. Now, it is only a question of adapting the technique to semi-implicit methods.
We define

$$\mathbf{U}_{\text{I}} = \mathcal{S}(\mathbf{U}_*, \mathbf{U}_{\text{E}}, \Delta t),$$

the function that is the solution of the problem

$$\mathbf{U}_{\text{I}} = \mathbf{U}_* - \Delta t \mathbf{F}(\mathbf{U}_{\text{E}}, \mathbf{U}_{\text{I}})_x.$$

Then, the method can be implemented in the following way:

$$\begin{aligned}
\mathbf{U}_{\text{E}}^{(1)} &= \mathbf{U}^n \\
\mathbf{U}_{\text{I}}^{(1)} &= \mathcal{S}(\mathbf{U}^n, \mathbf{U}^n, \beta\Delta t) \\
\mathbf{U}_{\text{E}}^{(2)} &= \left(1 - \frac{\hat{c}}{\beta}\right)\mathbf{U}^n + \frac{\hat{c}}{\beta}\mathbf{U}_{\text{I}}^{(1)} \\
\mathbf{U}_*^{(2)} &= \frac{2\beta - 1}{\beta}\mathbf{U}^n + \frac{1-\beta}{\beta}\mathbf{U}_{\text{I}}^{(1)} \\
\mathbf{U}_{\text{I}}^{(2)} &= \mathcal{S}(\mathbf{U}_*^{(2)}, \mathbf{U}_{\text{E}}^{(2)}, \beta\Delta t),
\end{aligned}$$

where $\beta = 1 - \frac{1}{\sqrt{2}}$ and $\hat{c} = \frac{1}{2\beta}$.
Finally, the numerical solution is computed as $\mathbf{U}^{n+1} = \mathbf{U}_{\text{I}}^{(2)}$.

We apply such discretizations to the Isentropic Euler equations and Full Euler equations in 1D.

Isentropic Euler Equations

Discretization of the system (4.8)

Semi-Implicit methods with a first-order scheme in time

$$\begin{cases} \rho^{n+1} = \rho^n - \Delta t \tilde{D}_x m^{n+1} & (a) \\ m^{n+1} = m^n - \Delta t \tilde{D}_x \left(\left(\frac{m^2}{\rho} \right)^n + \frac{p^{n+1}}{\varepsilon^2} \right) & (b) \end{cases},$$
$$p^{n+1} = \left(\rho^{n+1} \right)^\gamma.$$

We consider the equation (b) on the momentum:

$$\begin{aligned} m^{n+1} &= m^n - \Delta t \tilde{D}_x \left(\left(\frac{m^2}{\rho} \right)^n + \frac{p^{n+1}}{\varepsilon^2} \right) \\ &= m^* - \Delta t D_x \frac{p^{n+1}}{\varepsilon^2}, \end{aligned}$$

with $m^* = m^n - \Delta t \hat{D}_x \left(\frac{m^2}{\rho} \right)^n$.
Plugging this expression in the equation (a) on the density, we obtain:

$$\rho^{n+1} = \rho^n - \Delta t \hat{D}_x m^* + \frac{\Delta t^2}{\varepsilon^2} D_x^2 p^{n+1} \tag{5.1}$$
$$= \rho^* + \frac{\Delta t^2}{\varepsilon^2} D_x^2 p^{n+1}, \tag{5.2}$$

with $\rho^* = \rho^n - \Delta t \hat{D}_x m^*$.
ρ^{n+1} can now be computed by solving the non-linear tridiagonal system (5.2) and plugged into the momentum equation to find m^{n+1}. To solve the system, we use a system linearization method:

$$\begin{aligned} D_x^2 p(\rho)^{n+1} &= D_x \left(\frac{\partial p^n}{\partial \rho} D_x \rho^{n+1} \right) \\ &= D_x \left(\gamma \left(\rho^n \right)^{\gamma - 1} D_x \rho^{n+1} \right) \\ &= \gamma D_x \left(\left(\rho^n \right)^{\gamma - 1} D_x \rho^{n+1} \right), \end{aligned}$$

and thus we have:

$$\rho_i^{n+1} - \frac{\gamma}{\varepsilon^2}\frac{\Delta t^2}{\Delta x^2}\left((\rho^n)_{i+\frac{1}{2}}^{\gamma-1}(\rho_{i+1}^{n+1} - \rho_i^{n+1}) - (\rho^n)_{i-\frac{1}{2}}^{\gamma-1}(\rho_i^{n+1} - \rho_{i-1}^{n+1})\right) = \rho_i^*,$$

where $(\rho^n)_{i+\frac{1}{2}}^{\gamma-1} = \frac{(\rho^n)_i^{\gamma-1}+(\rho^n)_{i+1}^{\gamma-1}}{2}$ and $(\rho^n)_{i-\frac{1}{2}}^{\gamma-1} = \frac{(\rho^n)_{i-1}^{\gamma-1}+(\rho^n)_i^{\gamma-1}}{2}$ are known terms from the previous step (at time n).

Therefore, the algorithm is:

- Compute m^* at time n
- Compute ρ^* at time n
- Compute ρ^{n+1}
- Compute p^{n+1}
- Compute m^{n+1}

We can compute $\hat{D}_x \boldsymbol{f}$ using a first or a second-order scheme in space.

Full Euler Equations

Discretization of the system (4.12)

Semi-Implicit methods with a first-order scheme in time

We present an approach described in [3] called "*Pressure splitting*" and that is based on an explicit treatment of the convective terms and an implicit treatment of the pressure terms:

$$\begin{cases} \rho^{n+1} = \rho^n - \Delta t \hat{D}_x m^n & (a) \\ m^{n+1} = m^n - \Delta t \tilde{D}_x \left(\left(\frac{m^2}{\rho}\right)^n + \frac{p^{n+1}}{\varepsilon^2}\right) & (b) \\ E^{n+1} = E^n - \Delta t \tilde{D}_x \left(h^n m^{n+1}\right) & (c) \end{cases},$$

$$p^{n+1} = (\gamma - 1)\left(E^{n+1} - \frac{1}{2}\varepsilon^2\left(\frac{m^2}{\rho}\right)^n\right).$$

In this scheme ρ^{n+1} is computed explicity, while E^{n+1} and p^{n+1} are linearly

linked through the equation of state, thus it is possible solve the system both on the energy and on the pressure. In [3] the authors explored the use of scheme obtained by solving the linear system on the pressure, and they found it to be less accurate and more oscillatory.

First, we replace the equation of state p^{n+1} in the equation (b) of the momentum:

$$\begin{aligned} m^{n+1} &= m^n - \frac{3-\gamma}{2}\Delta t \hat{D}_x \left(\frac{m^2}{\rho}\right)^n - \frac{\gamma-1}{\varepsilon^2}\Delta t D_x E^{n+1} \\ &= m^* - \frac{\gamma-1}{\varepsilon^2}\Delta t D_x E^{n+1}, \ (d) \end{aligned}$$

with $m^* = m^n - \frac{3-\gamma}{2}\Delta t \hat{D}_x \left(\frac{m^2}{\rho}\right)^n$.

Plugging this expression in the equation (c) on the energy, we obtain:

$$E^{n+1} = E^n - \Delta t \hat{D}_x \left(h^n m^*\right) + \frac{\gamma-1}{\varepsilon^2}\Delta t^2 D_x(h^n(D_x E^{n+1})) \tag{5.3}$$

$$= E^* + \frac{\gamma-1}{\varepsilon^2}\Delta t^2 D_x(h^n(D_x E^{n+1})), \tag{5.4}$$

with $E^* = E^n - \Delta t \hat{D}_x \left(h^n m^*\right)$.

To compute E^{n+1} we solve the tridiagonal system (5.4):

$$E_i^{n+1} - \frac{\gamma-1}{\varepsilon^2}\frac{\Delta t^2}{\Delta x^2}\left(h^n_{i+\frac{1}{2}}(E^{n+1}_{i+1} - E^{n+1}_i) - h^n_{i-\frac{1}{2}}(E^{n+1}_i - E^{n+1}_{i-1})\right) = E_i^*$$

$$E_i^{n+1} - \frac{\gamma-1}{\varepsilon^2}\frac{\Delta t^2}{\Delta x^2}\left(\frac{h^n_i + h^n_{i+1}}{2}(E^{n+1}_{i+1} - E^{n+1}_i) - \frac{h^n_{i-1} + h^n_i}{2}(E^{n+1}_i - E^{n+1}_{i-1})\right) = E_i^*.$$

We now insert E^{n+1} into the momentum equation (d) to find m^{n+1}.

Therefore the algorithm is:

- Compute m^* at time n
- Compute E^* at time n
- Compute E^{n+1}
- Compute p^{n+1}

- Compute ρ^{n+1}
- Compute m^{n+1}

We can compute $\hat{D}_x \boldsymbol{f}$ using a first or a second-order scheme in space.

We also propose the case in which we solve the system on the pressure, just do not replace the equation of state in the equation of the momentum. The alghoritm in this case is:

- Compute $m^* = m^n - \Delta t \hat{D}_x \left(\frac{m^2}{\rho}\right)^n$ at time n
- Compute $E^* = E^n - \Delta t \hat{D}_x \left(h^n m^*\right) - \frac{1}{2}\varepsilon^2 \left(\frac{m^2}{\rho}\right)^n$ at time n
- Compute p^{n+1} solving the tridiagonal system $\frac{p^{n+1}}{\gamma - 1} - \frac{\Delta t^2}{\varepsilon^2} D_x^2 \left(h^n p^{n+1}\right) = E^*$
- Compute E^{n+1}
- Compute $\rho^{n+1} = \rho^n - \Delta t \hat{D}_x m^n$
- Compute m^{n+1}

This method can be extended to the two-dimensional case and also to the case in which we use second-order reconstructions in time. However, unless otherwise specified, in our numerical tests we will use the method based on the energy.

Semi-Implicit methods with a second-order scheme in time

In the case of Full Euler Equations we also used a second-order discretization in time.
First, we compute:

$$\begin{aligned} \rho_{\mathrm{E}}^{(1)} &= \rho^n, \\ m_{\mathrm{E}}^{(1)} &= m^n, \\ E_{\mathrm{E}}^{(1)} &= E^n. \end{aligned}$$

We can write the equations for the density:

$$\begin{aligned}
\rho_{\mathrm{I}}^{(1)} &= \rho^n - \beta\Delta t\hat{D}_x m_{\mathrm{E}}^{(1)},\\
\rho_{\mathrm{E}}^{(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)\rho^n + \frac{\hat{c}}{\beta}\rho_{\mathrm{I}}^{(1)},\\
\rho_{*}^{(2)} &= \frac{2\beta-1}{\beta}\rho^n + \frac{1-\beta}{\beta}\rho_{\mathrm{I}}^{(1)}.
\end{aligned}$$

We write now the equation for the momentum. As we have already described for the first-order scheme replacing the equation of state in the equation of the momentum we obtain:

$$\begin{aligned}
m_{\mathrm{I}}^{(1)} &= m^n - \beta\Delta t\frac{3-\gamma}{2}\hat{D}_x\frac{{m_{\mathrm{E}}^{(1)}}^2}{\rho_{\mathrm{E}}^{(1)}} - \beta\Delta t\frac{\gamma-1}{\varepsilon^2}D_x E_{\mathrm{I}}^{(1)}\\
&= M_{\mathrm{E}}^{(1)} - \beta\Delta t\frac{\gamma-1}{\varepsilon^2}D_x E_{\mathrm{I}}^{(1)},
\end{aligned}$$

with $M_{\mathrm{E}}^{(1)} = m^n - \beta\Delta t\frac{3-\gamma}{2}\hat{D}_x\frac{{m_{\mathrm{E}}^{(1)}}^2}{\rho_{\mathrm{E}}^{(1)}}$.

Plugging this expression in the equation on the energy, we obtain:

$$\begin{aligned}
E_{\mathrm{I}}^{(1)} &= E^n - \beta\Delta t\hat{D}_x\left(hM_{\mathrm{E}}^{(1)}\right) + \beta^2\Delta t^2\frac{\gamma-1}{\varepsilon^2}D_x\left(h\left(D_x E_{\mathrm{I}}^{(1)}\right)\right) &(5.5)\\
&= E^* + \beta^2\Delta t^2\frac{\gamma-1}{\varepsilon^2}D_x\left(h\left(D_x E_{\mathrm{I}}^{(1)}\right)\right), &(5.6)
\end{aligned}$$

with $E^* = E^n - \beta\Delta t\hat{D}_x\left(hM_{\mathrm{E}}^{(1)}\right)$.

$E_{\mathrm{I}}^{(1)}$ can now be computed by solving the tridiagonal system (5.6) and plugged into the momentum equation to find $m_{\mathrm{I}}^{(1)}$, and thus:

$$\begin{aligned}
m_{\mathrm{E}}^{(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)m^n + \frac{\hat{c}}{\beta}m_{\mathrm{I}}^{(1)},\\
m_{*}^{(2)} &= \frac{2\beta-1}{\beta}m^n + \frac{1-\beta}{\beta}m_{\mathrm{I}}^{(1)},\\
\rho_{\mathrm{I}}^{(2)} &= \rho_{*}^{(2)} - \beta\Delta t\hat{D}_x m_{\mathrm{E}}^{(2)},\\
E_{\mathrm{E}}^{(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)E^n + \frac{\hat{c}}{\beta}E_{\mathrm{I}}^{(1)},\\
E_{*}^{(2)} &= \frac{2\beta-1}{\beta}E^n + \frac{1-\beta}{\beta}E_{\mathrm{I}}^{(1)}.
\end{aligned}$$

To compute $m_{\mathrm{I}}^{(2)}$ and $E_{\mathrm{I}}^{(2)}$ we use the same procedure used to compute $m_{\mathrm{I}}^{(1)}$ and $E_{\mathrm{I}}^{(1)}$. We have:

$$\begin{aligned} m_{\mathrm{I}}^{(2)} &= m_*^{(2)} - \beta\Delta t\frac{3-\gamma}{2}\hat{D}_x\frac{{m_{\mathrm{E}}^{(2)}}^2}{\rho_{\mathrm{E}}^{(2)}} - \beta\Delta t\frac{\gamma-1}{\varepsilon^2}D_xE_{\mathrm{I}}^{(2)} \\ &= M_{\mathrm{E}}^{(2)} - \beta\Delta t\frac{\gamma-1}{\varepsilon^2}D_xE_{\mathrm{I}}^{(2)}, \end{aligned}$$

with $M_{\mathrm{E}}^{(2)} = m_*^{(2)} - \beta\Delta t\frac{3-\gamma}{2}\hat{D}_x\frac{{m_{\mathrm{E}}^{(2)}}^2}{\rho_{\mathrm{E}}^{(2)}}$, and

$$\begin{aligned} E_{\mathrm{I}}^{(2)} &= E_*^{(2)} - \beta\Delta t\hat{D}_x\left(hM_{\mathrm{E}}^{(2)}\right) + \beta^2\Delta t^2\frac{\gamma-1}{\varepsilon^2}D_x\left(h\left(D_xE_{\mathrm{I}}^{(2)}\right)\right) \\ &= E^{**} + \beta^2\Delta t^2\frac{\gamma-1}{\varepsilon^2}D_x\left(h\left(D_xE_{\mathrm{I}}^{(2)}\right)\right), \end{aligned}$$

with $E^{**} = E_*^{(2)} - \beta\Delta t\hat{D}_x\left(hM_{\mathrm{E}}^{(2)}\right)$.

Therefore the algorithm is:

- Compute $\rho_{\mathrm{E}}^{(1)}$, $m_{\mathrm{E}}^{(1)}$, $E_{\mathrm{E}}^{(1)}$
- Compute $\rho_{\mathrm{I}}^{(1)}$, $\rho_{\mathrm{E}}^{(2)}$, $\rho_*^{(2)}$
- Compute $M_{\mathrm{E}}^{(1)}$, E^*, $E_{\mathrm{I}}^{(1)}$
- Compute $m_{\mathrm{I}}^{(1)}$, $m_{\mathrm{E}}^{(2)}$, $m_*^{(2)}$, $\rho_{\mathrm{I}}^{(2)}$, $E_{\mathrm{E}}^{(2)}$, $E_*^{(2)}$
- Compute $M_{\mathrm{E}}^{(2)}$, E^{**}, $E_{\mathrm{I}}^{(2)}$, $m_{\mathrm{I}}^{(2)}$
- Compute $E^{n+1} = E_{\mathrm{I}}^{(2)}$
- Compute p^{n+1}
- Compute $\rho^{n+1} = \rho_{\mathrm{I}}^{(2)}$
- Compute $m^{n+1} = m_{\mathrm{I}}^{(2)}$

We can compute $\hat{D}_x\boldsymbol{f}$ using a first or a second-order scheme in space.

Such methods can easily be extended to the two-dimensional case.

5.2 Semi-Implicit methods for Gas Dynamics Equations in 2D

We discretize in 2D the hyperbolic systems (4.9) and (4.13) of conservation laws, that we can write in the following form:

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = 0.$$

We discretize the space as in the Section 4.4.

In the case of semi-implicit methods we denote by $\hat{D}_x\boldsymbol{f}$ and $\hat{D}_y\boldsymbol{g}$ the flux derivatives for non-stiff terms that will be discretized as the difference of the numerical fluxes between $i+\frac{1}{2}$ and $i-\frac{1}{2}$ and by $D_x\boldsymbol{f}$ and $D_y\boldsymbol{g}$ the flux derivatives for stiff terms discretized with a centered scheme. Both in the case of a first-order scheme and in the case of a second-order scheme in space, for non-stiff terms, the numerical fluxes $\{\hat{\boldsymbol{f}}_{i+\frac{1}{2}}\}$ and $\{\hat{\boldsymbol{g}}_{j+\frac{1}{2}}\}$ are computed as in the Section 4.4.

Therefore, for non-stiff terms, the discrete derivatives are given by:

$$\hat{D}_x\boldsymbol{f} = \frac{\hat{\boldsymbol{f}}_{i+\frac{1}{2},j} - \hat{\boldsymbol{f}}_{i-\frac{1}{2},j}}{\Delta x},\ \hat{D}_y\boldsymbol{g} = \frac{\hat{\boldsymbol{g}}_{i,j+\frac{1}{2}} - \hat{\boldsymbol{g}}_{i,j-\frac{1}{2}}}{\Delta y}.$$

For stiff terms, instead, we have:

$$D_x\boldsymbol{f} = \frac{\boldsymbol{f}_{i+\frac{1}{2},j} - \boldsymbol{f}_{i-\frac{1}{2},j}}{\Delta x} = \frac{\boldsymbol{f}_{i+1,j} - \boldsymbol{f}_{i-1,j}}{2\Delta x},$$

$$D_y\boldsymbol{g} = \frac{\boldsymbol{g}_{i,j+\frac{1}{2}} - \boldsymbol{g}_{i,j-\frac{1}{2}}}{\Delta y} = \frac{\boldsymbol{g}_{i,j+1} - \boldsymbol{g}_{i,j-1}}{2\Delta y}.$$

As in 1D case, for high Mach number we choose:

$$\alpha_{i,j} = |u_{i,j}| + \frac{c_{i,j}}{\varepsilon},$$

$$\beta_{i,j} = |v_{i,j}| + \frac{c_{i,j}}{\varepsilon},$$

while, for low Mach number we have:

$$\alpha_{i,j} = |u_{i,j}|,$$

$$\beta_{i,j} = |v_{i,j}|.$$

Discretization in time

First-order scheme

If we use a first-order scheme, to compute the numerical solutions at time t^{n+1} we discretize time by a first order Euler method: stiff terms are evaluated at time t^{n+1}, while non-stiff terms are evaluated at time $t^n = n\Delta t$, $n = 0, 1, \ldots$:

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \Delta t \mathbf{F}(\mathbf{U}_{\text{E}}^n, \mathbf{U}_{\text{I}}^{n+1})_x - \Delta t \mathbf{G}(\mathbf{U}_{\text{E}}^n, \mathbf{U}_{\text{I}}^{n+1})_y,$$

where $\mathbf{U}_{\text{E}}$ is the vector of non-stiff terms and $\mathbf{U}_{\text{I}}$ is the vector of stiff terms of the system.

Second-order scheme

High order schemes in time are obtained like in 1D case.
We define

$$\mathbf{U}_{\text{I}} = \mathcal{S}(\mathbf{U}_*, \mathbf{U}_{\text{E}}, \Delta t),$$

the function that is the solution of the problem

$$\mathbf{U}_{\text{I}} = \mathbf{U}_* - \Delta t \mathbf{F}(\mathbf{U}_{\text{E}}, \mathbf{U}_{\text{I}})_x - \Delta t \mathbf{G}(\mathbf{U}_{\text{E}}, \mathbf{U}_{\text{I}})_y;$$

Then, the method can be implemented in the following way:

$$\begin{aligned}
\mathbf{U}_{\text{E}}^{(1)} &= \mathbf{U}^n \\
\mathbf{U}_{\text{I}}^{(1)} &= \mathcal{S}(\mathbf{U}^n, \mathbf{U}^n, \beta\Delta t) \\
\mathbf{U}_{\text{E}}^{(2)} &= \left(1 - \frac{\hat{c}}{\beta}\right)\mathbf{U}^n + \frac{\hat{c}}{\beta}\mathbf{U}_{\text{I}}^{(1)} \\
\mathbf{U}_*^{(2)} &= \frac{2\beta - 1}{\beta}\mathbf{U}^n + \frac{1-\beta}{\beta}\mathbf{U}_{\text{I}}^{(1)} \\
\mathbf{U}_{\text{I}}^{(2)} &= \mathcal{S}(\mathbf{U}_*^{(2)}, \mathbf{U}_{\text{E}}^{(2)}, \beta\Delta t),
\end{aligned}$$

where $\beta = 1 - \frac{1}{\sqrt{2}}$ and $\hat{c} = \frac{1}{2\beta}$.
Finally, the numerical solution is computed as $\mathbf{U}^{n+1} = \mathbf{U}_{\text{I}}^{(2)}$.

We apply such discretizations to the Isentropic Euler equations and Full Euler equations in 2D.

In 2D the CFL condition is $\text{CFL} = \lambda_{1_{\max}} \frac{\Delta t}{\Delta x} + \lambda_{2_{\max}} \frac{\Delta t}{\Delta y}$, from which

$$\Delta t = \frac{\text{CFL}}{\frac{\lambda_{1_{\max}}}{\Delta x} + \frac{\lambda_{2_{\max}}}{\Delta y}}.$$

Isentropic Euler Equations

Discretization of the system (4.9)

Semi-Implicit methods with a first-order scheme in time

$$\begin{cases} \rho^{n+1} = \rho^n - \Delta t \tilde{D}_x m^{\mathbf{1}^{n+1}} - \Delta t \tilde{D}_y m^{\mathbf{2}^{n+1}} & (a) \\ m^{\mathbf{1}^{n+1}} = m^{\mathbf{1}^n} - \Delta t \tilde{D}_x \left(\left(\frac{m^{\mathbf{1}^2}}{\rho} \right)^n + \frac{p^{n+1}}{\varepsilon^2} \right) - \Delta t \hat{D}_y \left(m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \right)^n & (b) \\ m^{\mathbf{2}^{n+1}} = m^{\mathbf{2}^n} - \Delta t \hat{D}_x \left(m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \right)^n - \Delta t \tilde{D}_y \left(\left(\frac{m^{\mathbf{2}^2}}{\rho} \right)^n + \frac{p^{n+1}}{\varepsilon^2} \right) & (c) \end{cases},$$

$$p^{n+1} = \left(\rho^{n+1} \right)^\gamma.$$

First, we consider the equation (b) and (c) on the momentums:

$$\begin{aligned} m^{\mathbf{1}^{n+1}} &= m^{\mathbf{1}^n} - \Delta t \tilde{D}_x \left(\left(\frac{m^{\mathbf{1}^2}}{\rho} \right)^n + \frac{p^{n+1}}{\varepsilon^2} \right) - \Delta t \hat{D}_y \left(m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \right)^n \\ &= m^{\mathbf{1}^*} - \Delta t D_x \frac{p^{n+1}}{\varepsilon^2}, \quad (d) \\ m^{\mathbf{2}^{n+1}} &= m^{\mathbf{2}^n} - \Delta t \hat{D}_x \left(m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \right)^n - \Delta t \tilde{D}_y \left(\left(\frac{m^{\mathbf{2}^2}}{\rho} \right)^n + \frac{p^{n+1}}{\varepsilon^2} \right) \\ &= m^{\mathbf{2}^*} - \Delta t D_y \frac{p^{n+1}}{\varepsilon^2}, \quad (e) \end{aligned}$$

with $m^{\mathbf{1}^*} = m^{\mathbf{1}^n} - \Delta t \hat{D}_x \left(\frac{m^{\mathbf{1}^2}}{\rho} \right)^n - \Delta t \hat{D}_y \left(m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \right)^n$

and $m^{\mathbf{2}^*} = m^{\mathbf{2}^n} - \Delta t \hat{D}_x \left(m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \right)^n - \Delta t \hat{D}_y \left(\frac{m^{\mathbf{2}^2}}{\rho} \right)^n$.

Plugging these expressions in the equation (a) on the density, we obtain:

$$\rho^{n+1} = \rho^n - \Delta t \hat{D}_x m^{\mathbf{1}^*} - \Delta t \hat{D}_y m^{\mathbf{2}^*} + \frac{\Delta t^2}{\varepsilon^2} D_x^2 p^{n+1} + \frac{\Delta t^2}{\varepsilon^2} D_y^2 p^{n+1} \tag{5.7}$$

$$= \rho^* + \frac{\Delta t^2}{\varepsilon^2} D_x^2 p^{n+1} + \frac{\Delta t^2}{\varepsilon^2} D_y^2 p^{n+1}, \tag{5.8}$$

with $\rho^* = \rho^n - \Delta t \hat{D}_x m^{\mathbf{1}^*} - \Delta t \hat{D}_y m^{\mathbf{2}^*}$.

ρ^{n+1} can now be computed by solving the tridiagonal system (5.8) and plugged into the momentums equations to find $m^{\mathbf{1}^{n+1}}$ and $m^{\mathbf{2}^{n+1}}$. As in 1D case, we use a system linearization method to solve the system (5.8).

Therefore, the algorithm is:

- Compute $m^{\mathbf{1}^*}$ and $m^{\mathbf{2}^*}$ at time n
- Compute ρ^* at time n
- Compute ρ^{n+1}
- Compute p^{n+1}
- Compute $m^{\mathbf{1}^{n+1}}$ and $m^{\mathbf{2}^{n+1}}$

We can compute $\hat{D}_x \boldsymbol{f}$ and $\hat{D}_y \boldsymbol{g}$ using a first or a second-order scheme in space.

Full Euler Equations

Discretization of the system (4.13)

Semi-Implicit methods with a first-order scheme in time

We proceed as in the 1D case:

$$\begin{cases} \rho^{n+1} = \rho^n - \Delta t \hat{D}_x m^{\mathbf{1}^n} - \Delta t \hat{D}_y m^{\mathbf{2}^n} & (a) \\ m^{\mathbf{1}^{n+1}} = m^{\mathbf{1}^n} - \Delta t \tilde{D}_x \left(\left(\frac{m^{\mathbf{1}^2}}{\rho} \right)^n + \frac{p^{n+1}}{\varepsilon^2} \right) - \Delta t \hat{D}_y \left(m^{\mathbf{1}} \frac{m^{\mathbf{2}}}{\rho} \right)^n & (b) \\ m^{\mathbf{2}^{n+1}} = m^{\mathbf{2}^n} - \Delta t \hat{D}_x \left(m^{\mathbf{2}} \frac{m^{\mathbf{1}}}{\rho} \right)^n - \Delta t \tilde{D}_y \left(\left(\frac{m^{\mathbf{2}^2}}{\rho} \right)^n + \frac{p^{n+1}}{\varepsilon^2} \right) & (c) \\ E^{n+1} = E^n - \Delta t \tilde{D}_x \left(h^n m^{\mathbf{1}^{n+1}} \right) - \Delta t \tilde{D}_y \left(h^n m^{\mathbf{2}^{n+1}} \right) & (d) \end{cases},$$

$$p^{n+1} = (\gamma - 1)\left(E^{n+1} - \frac{1}{2}\varepsilon^2\left(\frac{m^{\mathbf{1}^2}}{\rho} + \frac{m^{\mathbf{2}^2}}{\rho}\right)^n\right).$$

In this case ρ^{n+1} is computed explicity. First, we replace the equation of state p^{n+1} in the equations (*b*) and (*c*) of the momentums:

$$\begin{aligned}
m^{\mathbf{1}^{n+1}} &= m^{\mathbf{1}^n} - \Delta t\hat{D}_x\left(\frac{3-\gamma}{2}\frac{m^{\mathbf{1}^2}}{\rho} + \frac{1-\gamma}{2}\frac{m^{\mathbf{2}^2}}{\rho}\right)^n - \Delta t\hat{D}_y\left(m^{\mathbf{1}}\frac{m^{\mathbf{2}}}{\rho}\right)^n - \Delta t\frac{\gamma-1}{\varepsilon^2}D_x\mathrm{E}^{n+1} \\
&= m^{\mathbf{1}^*} - \Delta t\frac{\gamma-1}{\varepsilon^2}D_xE^{n+1}, \\
m^{\mathbf{2}^{n+1}} &= m^{\mathbf{2}^n} - \Delta t\hat{D}_x\left(m^{\mathbf{2}}\frac{m^{\mathbf{1}}}{\rho}\right)^n - \Delta t\hat{D}_y\left(\frac{3-\gamma}{2}\frac{m^{\mathbf{2}^2}}{\rho} + \frac{1-\gamma}{2}\frac{m^{\mathbf{1}^2}}{\rho}\right)^n - \Delta t\frac{\gamma-1}{\varepsilon^2}D_y\mathrm{E}^{n+1} \\
&= m^{\mathbf{2}^*} - \Delta t\frac{\gamma-1}{\varepsilon^2}D_yE^{n+1},
\end{aligned}$$

with $m^{\mathbf{1}^*} = m^{\mathbf{1}^n} - \Delta t\hat{D}_x\left(\frac{3-\gamma}{2}\frac{m^{\mathbf{1}^2}}{\rho} + \frac{1-\gamma}{2}\frac{m^{\mathbf{2}^2}}{\rho}\right)^n - \Delta t\hat{D}_y\left(m^{\mathbf{1}}\frac{m^{\mathbf{2}}}{\rho}\right)^n$
and $m^{\mathbf{2}^*} = m^{\mathbf{2}^n} - \Delta t\hat{D}_x\left(m^{\mathbf{2}}\frac{m^{\mathbf{1}}}{\rho}\right)^n - \Delta t\hat{D}_y\left(\frac{3-\gamma}{2}\frac{m^{\mathbf{2}^2}}{\rho} + \frac{1-\gamma}{2}\frac{m^{\mathbf{1}^2}}{\rho}\right)^n$.

Plugging these expressions in the equation (*d*) on the energy, we obtain:

$$\begin{aligned}
E^{n+1} =& E^n - \Delta t\hat{D}_x\left(h^n m^{\mathbf{1}^*}\right) - \Delta t\hat{D}_y\left(h^n m^{\mathbf{2}^*}\right) + && (5.9)\\
& \frac{\gamma-1}{\varepsilon^2}\Delta t^2\left(D_x(h^n(D_xE^{n+1})) + D_y(h^n(D_yE^{n+1}))\right) && (5.10)\\
=& E^* + \frac{\gamma-1}{\varepsilon^2}\Delta t^2\left(D_x(h^n(D_xE^{n+1})) + D_y(h^n(D_yE^{n+1}))\right), && (5.11)
\end{aligned}$$

with $E^* = E^n - \Delta t\hat{D}_x\left(h^n m^{\mathbf{1}^*}\right) - \Delta t\hat{D}_y\left(h^n m^{\mathbf{2}^*}\right)$.

E^{n+1} can now be computed by solving the tridiagonal system (5.11) and plugged into the momentums equations to find $m^{\mathbf{1}^{n+1}}$ and $m^{\mathbf{2}^{n+1}}$.

Therefore the algorithm is:

- Compute $m^{\mathbf{1}^*}$ and $m^{\mathbf{2}^*}$ at time n
- Compute E^* at time n
- Compute E^{n+1}
- Compute p^{n+1}

- Compute ρ^{n+1}
- Compute $m^{\mathbf{1}^{n+1}}$ and $m^{\mathbf{2}^{n+1}}$

We can compute $\hat{D}_x \boldsymbol{f}$ and $\hat{D}_y \boldsymbol{g}$ using a first or a second-order scheme in space.

Semi-Implicit methods with a second-order scheme in time

First we compute:

$$\begin{aligned}
\rho_{\mathrm{E}}^{(1)} &= \rho^n, \\
m_{\mathrm{E}}^{\mathbf{1}(1)} &= m^{\mathbf{1}^n}, \\
m_{\mathrm{E}}^{\mathbf{2}(1)} &= m^{\mathbf{2}^n}, \\
E_{\mathrm{E}}^{(1)} &= E^n.
\end{aligned}$$

We can write the equations for the density:

$$\begin{aligned}
\rho_{\mathrm{I}}^{(1)} &= \rho^n - \beta\Delta t \hat{D}_x m_{\mathrm{E}}^{\mathbf{1}(1)} - \beta\Delta t \hat{D}_y m_{\mathrm{E}}^{\mathbf{2}(1)}, \\
\rho_{\mathrm{E}}^{(2)} &= \left(1 - \frac{\hat{c}}{\beta}\right)\rho^n + \frac{\hat{c}}{\beta}\rho_{\mathrm{I}}^{(1)}, \\
\rho_{*}^{(2)} &= \frac{2\beta - 1}{\beta}\rho^n + \frac{1-\beta}{\beta}\rho_{\mathrm{I}}^{(1)}.
\end{aligned}$$

We write now the equations for the momentums. As we have already described for the first-order scheme replacing the equation of state in the equations of the momentums we obtain:

$$\begin{aligned}
m_{\mathrm{I}}^{\mathbf{1}(1)} =& m^{\mathbf{1}^n} - \beta\Delta t \hat{D}_x \left(\frac{3-\gamma}{2}\frac{m_{\mathrm{E}}^{\mathbf{1}(1)^2}}{\rho_{\mathrm{E}}^{(1)}} + \frac{1-\gamma}{2}\frac{m_{\mathrm{E}}^{\mathbf{2}(1)^2}}{\rho_{\mathrm{E}}^{(1)}}\right) - \\
& \beta\Delta t \hat{D}_y \left(m_{\mathrm{E}}^{\mathbf{1}(1)}\frac{m_{\mathrm{E}}^{\mathbf{2}(1)}}{\rho_{\mathrm{E}}^{(1)}}\right) - \beta\Delta t \frac{\gamma - 1}{\varepsilon^2} D_x E_{\mathrm{I}}^{(1)} \\
=& M_{\mathrm{E}}^{\mathbf{1}(1)} - \beta\Delta t \frac{\gamma - 1}{\varepsilon^2} D_x E_{\mathrm{I}}^{(1)}, \\
m_{\mathrm{I}}^{\mathbf{2}(1)} =& m^{\mathbf{2}^n} - \beta\Delta t \hat{D}_y \left(\frac{3-\gamma}{2}\frac{m_{\mathrm{E}}^{\mathbf{2}(1)^2}}{\rho_{\mathrm{E}}^{(1)}} + \frac{1-\gamma}{2}\frac{m_{\mathrm{E}}^{\mathbf{1}(1)^2}}{\rho_{\mathrm{E}}^{(1)}}\right) - \\
& \beta\Delta t \hat{D}_x \left(m_{\mathrm{E}}^{\mathbf{2}(1)}\frac{m_{\mathrm{E}}^{\mathbf{1}(1)}}{\rho_{\mathrm{E}}^{(1)}}\right) - \beta\Delta t \frac{\gamma - 1}{\varepsilon^2} D_y E_{\mathrm{I}}^{(1)} \\
=& M_{\mathrm{E}}^{\mathbf{2}(1)} - \beta\Delta t \frac{\gamma - 1}{\varepsilon^2} D_y E_{\mathrm{I}}^{(1)},
\end{aligned}$$

with $M_{\mathrm{E}}^{\mathbf{1}(1)} = m^{\mathbf{1}^n} - \beta\Delta t\hat{D}_x\left(\frac{3-\gamma}{2}\frac{m_{\mathrm{E}}^{\mathbf{1}(1)^2}}{\rho_{\mathrm{E}}^{(1)}} + \frac{1-\gamma}{2}\frac{m_{\mathrm{E}}^{\mathbf{2}(1)^2}}{\rho_{\mathrm{E}}^{(1)}}\right) - \beta\Delta t\hat{D}_y\left(m_{\mathrm{E}}^{\mathbf{1}(1)}\frac{m_{\mathrm{E}}^{\mathbf{2}(1)}}{\rho_{\mathrm{E}}^{(1)}}\right)$
and $M_{\mathrm{E}}^{\mathbf{2}(1)} = m^{\mathbf{2}^n} - \beta\Delta t\hat{D}_y\left(\frac{3-\gamma}{2}\frac{m_{\mathrm{E}}^{\mathbf{2}(1)^2}}{\rho_{\mathrm{E}}^{(1)}} + \frac{1-\gamma}{2}\frac{m_{\mathrm{E}}^{\mathbf{1}(1)^2}}{\rho_{\mathrm{E}}^{(1)}}\right) - \beta\Delta t\hat{D}_y\left(m_{\mathrm{E}}^{\mathbf{2}(1)}\frac{m_{\mathrm{E}}^{\mathbf{1}(1)}}{\rho_{\mathrm{E}}^{(1)}}\right)$.

Plugging these expressions in the equation on the energy, we obtain:

$$\begin{aligned} E_{\mathrm{I}}^{(1)} =& E^n - \beta\Delta t\hat{D}_x\left(hM_{\mathrm{E}}^{\mathbf{1}(1)}\right) - \beta\Delta t\hat{D}_y\left(hM_{\mathrm{E}}^{\mathbf{2}(1)}\right) + && (5.12)\\ & \beta^2\Delta t^2\frac{\gamma-1}{\varepsilon^2}\left(D_x\left(h\left(D_xE_{\mathrm{I}}^{(1)}\right)\right) + D_y\left(h\left(D_yE_{\mathrm{I}}^{(1)}\right)\right)\right) && (5.13)\\ =& E^* + \beta^2\Delta t^2\frac{\gamma-1}{\varepsilon^2}\left(D_x\left(h\left(D_xE_{\mathrm{I}}^{(1)}\right)\right) + D_y\left(h\left(D_yE_{\mathrm{I}}^{(1)}\right)\right)\right), && (5.14)\end{aligned}$$

with $E^* = E^n - \beta\Delta t\hat{D}_x\left(hM_{\mathrm{E}}^{\mathbf{1}(1)}\right) - \beta\Delta t\hat{D}_y\left(hM_{\mathrm{E}}^{\mathbf{2}(1)}\right)$.

$E_{\mathrm{I}}^{(1)}$ can now be computed by solving the tridiagonal system (5.14) and plugged into the momentums equations to find $m_{\mathrm{I}}^{\mathbf{1}(1)}$ and $m_{\mathrm{I}}^{\mathbf{2}(1)}$. Thus, we can compute:

$$\begin{aligned} m_{\mathrm{E}}^{\mathbf{1}(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)m^{\mathbf{1}^n} + \frac{\hat{c}}{\beta}m_{\mathrm{I}}^{\mathbf{1}(1)},\\ m_{\mathrm{E}}^{\mathbf{2}(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)m^{\mathbf{2}^n} + \frac{\hat{c}}{\beta}m_{\mathrm{I}}^{\mathbf{2}(1)},\\ m_{*}^{\mathbf{1}(2)} &= \frac{2\beta-1}{\beta}m^{\mathbf{1}^n} + \frac{1-\beta}{\beta}m_{\mathrm{I}}^{\mathbf{1}(1)},\\ m_{*}^{\mathbf{2}(2)} &= \frac{2\beta-1}{\beta}m^{\mathbf{2}^n} + \frac{1-\beta}{\beta}m_{\mathrm{I}}^{\mathbf{2}(1)},\\ \rho_{\mathrm{I}}^{(2)} &= \rho_{*}^{(2)} - \beta\Delta t\hat{D}_xm_{\mathrm{E}}^{\mathbf{1}(2)} - \beta\Delta t\hat{D}_ym_{\mathrm{E}}^{\mathbf{2}(2)},\\ E_{\mathrm{E}}^{(2)} &= \left(1-\frac{\hat{c}}{\beta}\right)E^n + \frac{\hat{c}}{\beta}E_{\mathrm{I}}^{(1)},\\ E_{*}^{(2)} &= \frac{2\beta-1}{\beta}E^n + \frac{1-\beta}{\beta}E_{\mathrm{I}}^{(1)}.\end{aligned}$$

To compute $m_{\mathrm{I}}^{\mathbf{1}(2)}$, $m_{\mathrm{I}}^{\mathbf{2}(2)}$ and $E_{\mathrm{I}}^{(2)}$ we use the same procedure used to compute $m_{\mathrm{I}}^{\mathbf{1}(1)}$, $m_{\mathrm{I}}^{\mathbf{2}(1)}$ and $E_{\mathrm{I}}^{(1)}$.

We have:

$$\begin{aligned}
m_{\mathrm{I}}^{\mathbf{1}(2)} =& m_*^{\mathbf{1}(2)} - \beta\Delta t\hat{D}_x\left(\frac{3-\gamma}{2}\frac{{m_{\mathrm{E}}^{\mathbf{1}(2)}}^2}{\rho_{\mathrm{E}}^{(2)}} + \frac{1-\gamma}{2}\frac{{m_{\mathrm{E}}^{\mathbf{2}(2)}}^2}{\rho_{\mathrm{E}}^{(2)}}\right) - \\
& \beta\Delta t\hat{D}_y\left(m_{\mathrm{E}}^{\mathbf{1}(2)}\frac{m_{\mathrm{E}}^{\mathbf{2}(2)}}{\rho_{\mathrm{E}}^{(2)}}\right) - \beta\Delta t\frac{\gamma-1}{\varepsilon^2}D_x E_{\mathrm{I}}^{(2)} \\
=& M_{\mathrm{E}}^{\mathbf{1}(2)} - \beta\Delta t\frac{\gamma-1}{\varepsilon^2}D_x E_{\mathrm{I}}^{(2)}, \\
m_{\mathrm{I}}^{\mathbf{2}(2)} =& m_*^{\mathbf{2}(2)} - \beta\Delta t\hat{D}_y\left(\frac{3-\gamma}{2}\frac{{m_{\mathrm{E}}^{\mathbf{2}(2)}}^2}{\rho_{\mathrm{E}}^{(2)}} + \frac{1-\gamma}{2}\frac{{m_{\mathrm{E}}^{\mathbf{1}(2)}}^2}{\rho_{\mathrm{E}}^{(2)}}\right) - \\
& \beta\Delta t\hat{D}_x\left(m_{\mathrm{E}}^{\mathbf{2}(2)}\frac{m_{\mathrm{E}}^{\mathbf{1}(2)}}{\rho_{\mathrm{E}}^{(2)}}\right) - \beta\Delta t\frac{\gamma-1}{\varepsilon^2}D_y E_{\mathrm{I}}^{(2)} \\
=& M_{\mathrm{E}}^{\mathbf{2}(2)} - \beta\Delta t\frac{\gamma-1}{\varepsilon^2}D_y E_{\mathrm{I}}^{(2)},
\end{aligned}$$

with $M_{\mathrm{E}}^{\mathbf{1}(2)} = m_*^{\mathbf{1}(2)} - \beta\Delta t\hat{D}_x\left(\frac{3-\gamma}{2}\frac{{m_{\mathrm{E}}^{\mathbf{1}(2)}}^2}{\rho_{\mathrm{E}}^{(2)}} + \frac{1-\gamma}{2}\frac{{m_{\mathrm{E}}^{\mathbf{2}(2)}}^2}{\rho_{\mathrm{E}}^{(2)}}\right) - \beta\Delta t\hat{D}_y\left(m_{\mathrm{E}}^{\mathbf{1}(2)}\frac{m_{\mathrm{E}}^{\mathbf{2}(2)}}{\rho_{\mathrm{E}}^{(2)}}\right)$,
$M_{\mathrm{E}}^{\mathbf{2}(2)} = m_*^{\mathbf{2}(2)} - \beta\Delta t\hat{D}_y\left(\frac{3-\gamma}{2}\frac{{m_{\mathrm{E}}^{\mathbf{2}(2)}}^2}{\rho_{\mathrm{E}}^{(2)}} + \frac{1-\gamma}{2}\frac{{m_{\mathrm{E}}^{\mathbf{1}(2)}}^2}{\rho_{\mathrm{E}}^{(2)}}\right) - \beta\Delta t\hat{D}_y\left(m_{\mathrm{E}}^{\mathbf{2}(2)}\frac{m_{\mathrm{E}}^{\mathbf{1}(2)}}{\rho_{\mathrm{E}}^{(2)}}\right)$, and

$$\begin{aligned}
E_{\mathrm{I}}^{(2)} =& E_*^{(2)} - \beta\Delta t\hat{D}_x\left(hM_{\mathrm{E}}^{\mathbf{1}(2)}\right) - \beta\Delta t\hat{D}_y\left(hM_{\mathrm{E}}^{\mathbf{2}(2)}\right) + \\
& \beta^2\Delta t^2\frac{\gamma-1}{\varepsilon^2}\left(D_x\left(h\left(D_x E_{\mathrm{I}}^{(2)}\right)\right) + D_y\left(h\left(D_y E_{\mathrm{I}}^{(2)}\right)\right)\right) \\
=& E^{**} + \beta^2\Delta t^2\frac{\gamma-1}{\varepsilon^2}\left(D_x\left(h\left(D_x E_{\mathrm{I}}^{(2)}\right)\right) + D_y\left(h\left(D_y E_{\mathrm{I}}^{(2)}\right)\right)\right),
\end{aligned}$$

with $E^{**} = E_*^{(2)} - \beta\Delta t\hat{D}_x\left(hM_{\mathrm{E}}^{\mathbf{1}(2)}\right) - \beta\Delta t\hat{D}_y\left(hM_{\mathrm{E}}^{\mathbf{2}(2)}\right)$.

Therefore the algorithm is:

- Compute $\rho_{\mathrm{E}}^{(1)}$, $m_{\mathrm{E}}^{\mathbf{1}(1)}$, $m_{\mathrm{E}}^{\mathbf{2}(1)}$, $E_{\mathrm{E}}^{(1)}$
- Compute $\rho_{\mathrm{I}}^{(1)}$, $\rho_{\mathrm{E}}^{(2)}$, $\rho_*^{(2)}$
- Compute $M_{\mathrm{E}}^{\mathbf{1}(1)}$, $M_{\mathrm{E}}^{\mathbf{2}(1)}$, E^*, $E_{\mathrm{I}}^{(1)}$
- Compute $m_{\mathrm{I}}^{\mathbf{1}(1)}$, $m_{\mathrm{E}}^{\mathbf{1}(2)}$, $m_*^{\mathbf{1}(2)}$, $m_{\mathrm{I}}^{\mathbf{2}(1)}$, $m_{\mathrm{E}}^{\mathbf{2}(2)}$, $m_*^{\mathbf{2}(2)}$, $\rho_{\mathrm{I}}^{(2)}$, $E_{\mathrm{E}}^{(2)}$, $E_*^{(2)}$
- Compute $M_{\mathrm{E}}^{\mathbf{1}(2)}$, $M_{\mathrm{E}}^{\mathbf{2}(2)}$, E^{**}, $E_{\mathrm{I}}^{(2)}$, $m_{\mathrm{I}}^{\mathbf{1}(2)}$, $m_{\mathrm{I}}^{\mathbf{2}(2)}$
- Compute $E^{n+1} = E_{\mathrm{I}}^{(2)}$

- Compute p^{n+1}
- Compute $\rho^{n+1} = \rho_{\text{I}}^{(2)}$
- Compute $m^{\mathbf{1}^{n+1}} = m_{\text{I}}^{\mathbf{1}(2)}$, $m^{\mathbf{2}^{n+1}} = m_{\text{I}}^{\mathbf{2}(2)}$

We can compute $\hat{D}_x \boldsymbol{f}$ and $\hat{D}_y \boldsymbol{g}$ using a first or a second-order scheme in space.

Now, just as with the finite-difference methods, we want to apply the Semi-Implicit methods at the Full Euler equations on a bidimensional arbitrary domains. We will therefore eliminate the time step restriction present in [8].

5.3 Coco-Russo method for the Full Euler Equations on arbitrary domains

In this last Section, we apply the semi-implicit methods, discussed in the previous Section, to the Full Euler equations on bidimensional domains in presence of obstacles, eliminating the restriction on the time step present in [8], of which we have already spoken in the Section 4.5.

In addition to presenting the advantage of eliminating the restriction on the time step, in this method we provide an alternative technique to that proposed in [8] to impose the boundary conditions on the obstacle. Instead of using the iterative method (4.18) we will solve a simple linear system, which turns out to be computationally faster.

In this regard, we rewrite the equations (4.14), (4.15), (4.16) and (4.17) so that only the quantities $\mathcal{Q}[u; \mathcal{S}_{\text{G}}]$, $\mathcal{Q}[v; \mathcal{S}_{\text{G}}]$, $\mathcal{Q}[\rho; \mathcal{S}_{\text{G}}]$ and $\mathcal{Q}[p; \mathcal{S}_{\text{G}}]$ appear. Unlike the method proposed in [8], we have used bilinear interpolations, which we indicate with $\mathcal{B}[\psi_{ij}; \mathcal{S}_{\text{G}}]$, rather than biquadratic interpolations.

To do this, we make the following considerations.

The condition (4.14) is equivalent to $\mathcal{B}[u; \mathcal{S}_{\text{G}}](\text{F})\text{n}_x + \mathcal{B}[v; \mathcal{S}_{\text{G}}](\text{F})\text{n}_y = 0$, where

n_x and n_y are computed as follows. We define the level set function [23, 27]:

$$\phi(x,y) = \begin{cases} -\mathrm{d}((x,y),\partial\Omega) & (x,y) \in \Omega \\ \mathrm{d}((x,y),\partial\Omega) & (x,y) \notin \Omega \end{cases}.$$

Given the two upwind close points to G, Q_x and Q_y (see Figure 5.1), we have:

$$\mathrm{n}_x = \frac{\partial\phi}{\partial x} = \frac{\phi_x}{\sqrt{\phi_x^2+\phi_y^2}},$$

$$\mathrm{n}_y = \frac{\partial\phi}{\partial y} = \frac{\phi_y}{\sqrt{\phi_x^2+\phi_y^2}},$$

with

$$\phi_x = \frac{\phi_{\mathrm{G}} - \phi_{\mathrm{Q}_x}}{h}\mathrm{sgn}\left(\mathrm{G}^{(x)} - \mathrm{Q}_x^{(x)}\right),$$

$$\phi_y = \frac{\phi_{\mathrm{G}} - \phi_{\mathrm{Q}_y}}{h}\mathrm{sgn}\left(\mathrm{G}^{(y)} - \mathrm{Q}_y^{(y)}\right).$$

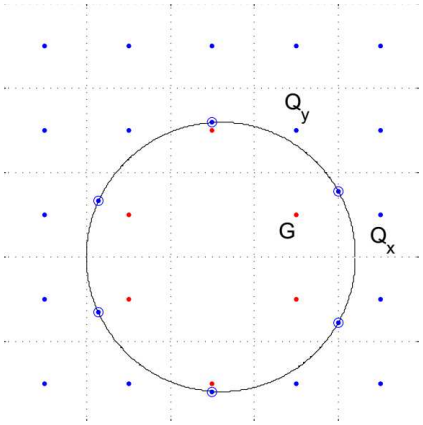


Figure 5.1: Upwind close points to G

While $\tau = \left(\frac{\partial\phi}{\partial y}, -\frac{\partial\phi}{\partial x}\right)$ and $u_\tau = \mathcal{B}[u;\mathcal{S}_{\mathrm{G}}]\mathrm{n}_y - \mathcal{B}[v;\mathcal{S}_{\mathrm{G}}]\mathrm{n}_x$.

In the condition (4.17) the term $\frac{\partial\mathcal{B}[u_\tau;\mathcal{S}_{\mathrm{G}}]}{\partial\mathbf{n}}(\mathrm{F})$ is approximated for simplicity with the following formula which still ensures a first-order accuracy:

$$\frac{\partial u_\tau}{\partial\mathbf{n}}(\mathrm{F}) = \frac{1}{h}\left(\left(u_{\tau_{\mathrm{G}}} - u_{\tau_{\mathrm{Q}_x}}\right)|\mathrm{n}_x| + \left(u_{\tau_{\mathrm{G}}} - u_{\tau_{\mathrm{Q}_y}}\right)|\mathrm{n}_y|\right).$$

In the conditions (4.15) and (4.16) the non-linear terms $(\mathcal{B}[u_\tau;\mathcal{S}_{\mathrm{G}}](\mathrm{F}))^2$ and $\frac{\mathcal{B}[\rho;\mathcal{S}_{\mathrm{G}}](\mathrm{F})}{\gamma\mathcal{B}[p;\mathcal{S}_{\mathrm{G}}](\mathrm{F})}$ are approximated with the values they assume at the internal point closest to the ghost point in question. That is, if I is the internal point closest to ghost point G, the two quantities are approximated respectively with $u_\tau(\mathrm{I})^2$ and $\frac{\rho(\mathrm{I})}{\gamma p(\mathrm{I})}$.

In this way the equations, for each ghost point, become:

$$\mathcal{B}[u;\mathcal{S}_{\mathrm{G}}](\mathrm{F})\mathrm{n}_x + \mathcal{B}[v;\mathcal{S}_{\mathrm{G}}](\mathrm{F})\mathrm{n}_y = 0 \tag{5.15}$$

$$\frac{\partial\mathcal{B}[p;\mathcal{S}_{\mathrm{G}}]}{\partial\mathbf{n}}(\mathrm{F}) = \kappa\mathcal{B}[\rho;\mathcal{S}_{\mathrm{G}}](\mathrm{F})u_\tau(\mathrm{I})^2 \tag{5.16}$$

$$\frac{\partial \mathcal{B}[\rho; \mathcal{S}_{\text{G}}]}{\partial \mathbf{n}}(\text{F}) = \frac{\rho(\text{I})}{\gamma p(\text{I})} \frac{\partial \mathcal{B}[p; \mathcal{S}_{\text{G}}]}{\partial \mathbf{n}}(\text{F}) \tag{5.17}$$

$$\frac{\partial u_\tau}{\partial \mathbf{n}}(\text{F}) = -\kappa \left(\mathcal{B}[u; \mathcal{S}_{\text{G}}](\text{F}) \text{n}_y - \mathcal{B}[v; \mathcal{S}_{\text{G}}](\text{F}) \text{n}_x \right) \tag{5.18}$$

which is a linear system with unknowns the values of tangential and normal velocities, pressure and density on the ghost point G.

As already discussed in Section 4.5, we can observe that we cannot solve separately the 4×4 systems obtained from the equations (5.15), (5.16), (5.17) and (5.18) for each ghost point G, since each 4×4 system may be coupled with the corresponding 4×4 systems obtained at other ghost points. Thus, to extend the numerical solution on the ghost points, we solve a linear system of $4N_{\text{G}}$ equations in $4N_{\text{G}}$ unknowns, whose unknowns are u, v, p and ρ on the ghost points. While, the known terms are the values of normal and tangential components of velocity, pressure and density on the internal points involved in the stencils.

We denote with $\mathbf{U}_{\text{I}}$ and $\mathbf{U}_{\text{G}}$ the vectors of numerical solution on internal points and on ghost points, respectively. The numerical method has the following form.

If we develop an explicit method, we have:

$$\mathbf{U}_{\text{I}}^{n+1} = \mathbf{U}_{\text{I}}^{n} - \Delta t \mathbf{F}(\mathbf{U}_{\text{I}}^{n}, \mathbf{U}_{\text{G}}^{n}) - \Delta t \mathbf{G}(\mathbf{U}_{\text{I}}^{n}, \mathbf{U}_{\text{G}}^{n}),$$

where $\mathbf{U}_{\text{G}}^{n}$ as computed solving the linear system

$$\text{A}\mathbf{U}_{\text{G}}^{n} = \text{b}(\mathbf{U}_{\text{I}}^{n}),$$

and $\text{A} \in \mathbb{R}^{4N_{\text{G}} \times 4N_{\text{G}}}$ is the matrix of coefficients, $\text{b}(\mathbf{U}_{\text{I}}^{n})$ is the vector of known terms.

If we develop a semi-implicit method, we have:

$$\mathbf{U}_{\text{I}}^{n+1} = \mathbf{U}_{\text{I}}^{n} - \Delta t \mathbf{F}(\mathbf{U}_{\text{I}}^{n+1}, \mathbf{U}_{\text{G}}^{n+1}) - \Delta t \mathbf{G}(\mathbf{U}_{\text{I}}^{n+1}, \mathbf{U}_{\text{G}}^{n+1}),$$

where $\mathbf{U}_{\text{G}}^{n+1}$ as computed solving the linear system

$$\text{A}\mathbf{U}_{\text{G}}^{n+1} = \text{b}(\mathbf{U}_{\text{I}}^{n+1}).$$

We close this Chapter by showing some numerical results that highlight the stability characteristics of Semi-Implicit methods.

5.4 Numerical tests for the Semi-Implicit methods

In this Section we present some numerical tests to highlight the robust characteristics of the Semi-Implicit method for the Gas Dynamics equations even for low Mach number values. The schemes are accurate for a wide range of values of the Mach number.

For each test we monitor the classical Courant number

$$\mathrm{CFL} = \frac{\lambda_{\max}\Delta t}{\Delta x}. \tag{5.19}$$

For all the numerical tests we give well prepared initial values and unlike the texts proposed in [3] and [6] we will not adopt periodic boundary conditions but we suppose there is a *wall*.

5.4.1 Numerical tests for the Isentropic Gas Dynamics

We start by presenting the numerical results for the Isentropic Gas Dynamics, in which we use a first-order reconstruction in time and a second-order reconstruction in space with $\theta = 1.5$.

Example 1: Two colliding acoustic waves

Consider the evolution of two colliding acoustic waves, with the following well prepared initial data:

$$p(\rho_\varepsilon) = \rho_\varepsilon^\gamma \text{ with } \gamma = 1.4,$$

$$\rho_\varepsilon(x,0) = 0.955 + \frac{\varepsilon}{2}\left(1 - \cos(2\pi x)\right),\ u_\varepsilon(x,0) = -\mathrm{sgn}(x)\sqrt{\gamma}\left(1 - \cos(2\pi x)\right).$$

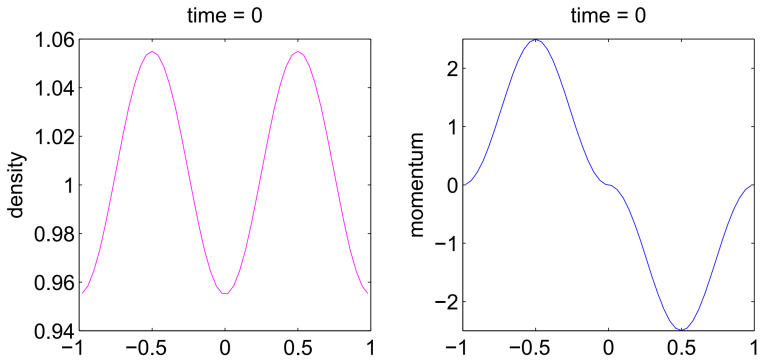


Figure 5.2: Initial Conditions: density (left panel) and momentum (right panel)

These acoustic pulses, one right-running and one left-running, collide and superpose and then separate again and during the whole procedure no shock forms. We present the numerical results choosing, as in [6], the spatial step $\Delta x = \frac{1}{50}$, $\varepsilon = 0.1$ and the following boundary conditions:

1. void Neumann boundary conditions on the density and thus on the pressure

2. void Dirichlet boundary conditions on the momentum

The time step is computed by (5.19) with CFL $= 0.5$.

In [6] the authors use a staggered grid, with which a second-order accuracy in space is automatically guaranteed.

The following figures show the numerical results on the density (left panel) and on the momentum (right panel) for different final times T.

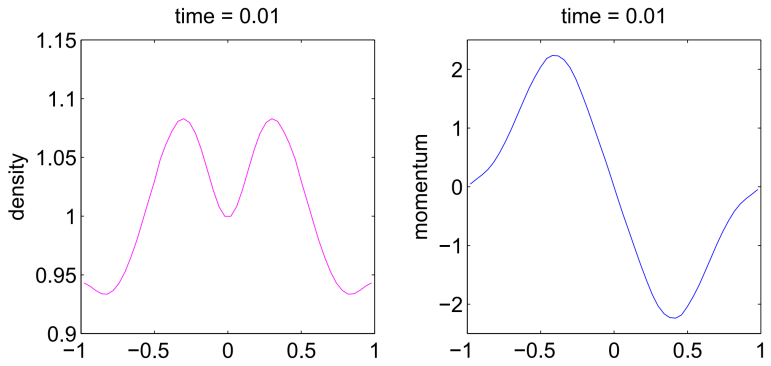


Figure 5.3: Density (left panel) and momentum (right panel) at final time $T = 0.01$, $N = 50$, $\varepsilon = 0.1$, CFL $= 0.5$

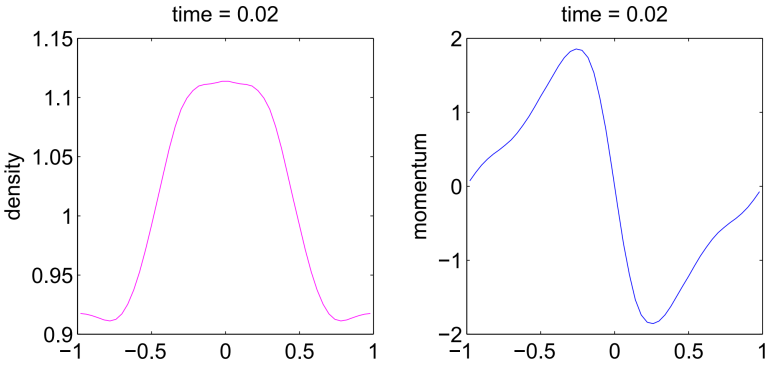


Figure 5.4: Density (left panel) and momentum (right panel) at final time $T = 0.02$, $N = 50$, $\varepsilon = 0.1$, CFL $= 0.5$

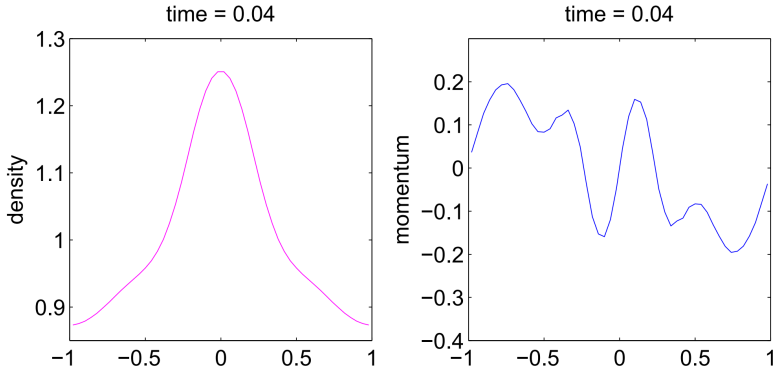


Figure 5.5: Density (left panel) and momentum (right panel) at final time $T = 0.04$, $N = 50$, $\varepsilon = 0.1$, CFL $= 0.5$

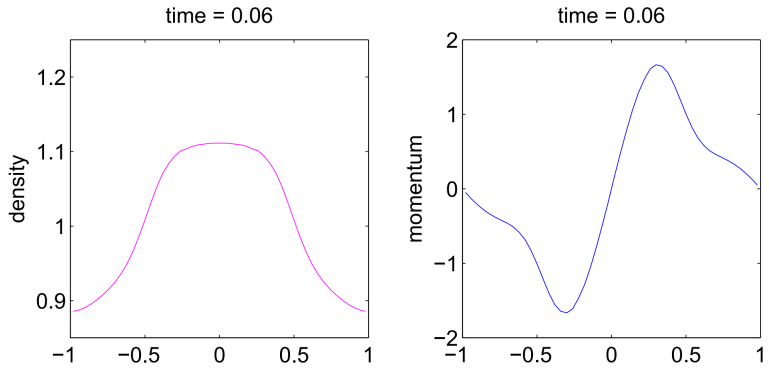


Figure 5.6: Density (left panel) and momentum (right panel) at final time $T = 0.06$, $N = 50$, $\varepsilon = 0.1$, CFL $= 0.5$

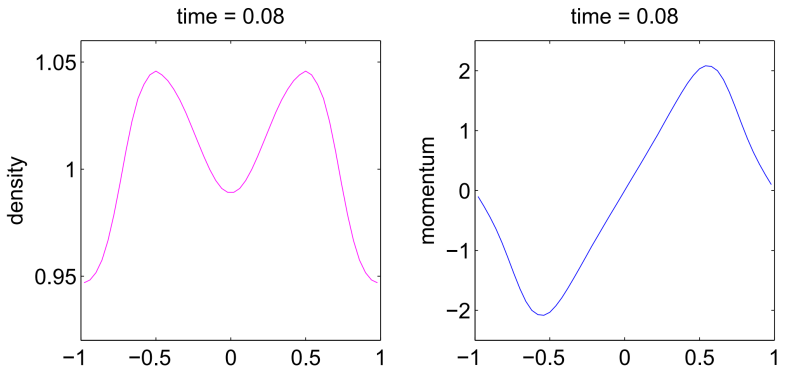


Figure 5.7: Density (left panel) and momentum (right panel) at final time $T = 0.08$, $N = 50$, $\varepsilon = 0.1$, CFL $= 0.5$

Example 2: 2D Isentropic Problem

We focus on a two dimensional case with $p(\rho) = \rho^2$. The computational domain is $\Omega = [0, 1]^2$. We consider the numerical test with initial conditions:

$$\begin{cases} \rho(x, y, 0) = e^{\frac{(x-0.5)^2+(y-0.5)^2}{0.1^2}} \\ \rho(x, y, 0)u(x, y, 0) = 0 \\ \rho(x, y, 0)v(x, y, 0) = 0 \end{cases} .$$

We present the numerical results by choosing the spatial step $\Delta x = \Delta y = \frac{1}{100}$, CFL $= 0.5$, $\varepsilon = 0.05$, final time $T = 0.002$ and the following boundary conditions:

1. void Neumann boundary conditions on the density and thus on the pressure

2. void Dirichlet boundary conditions on the momentums

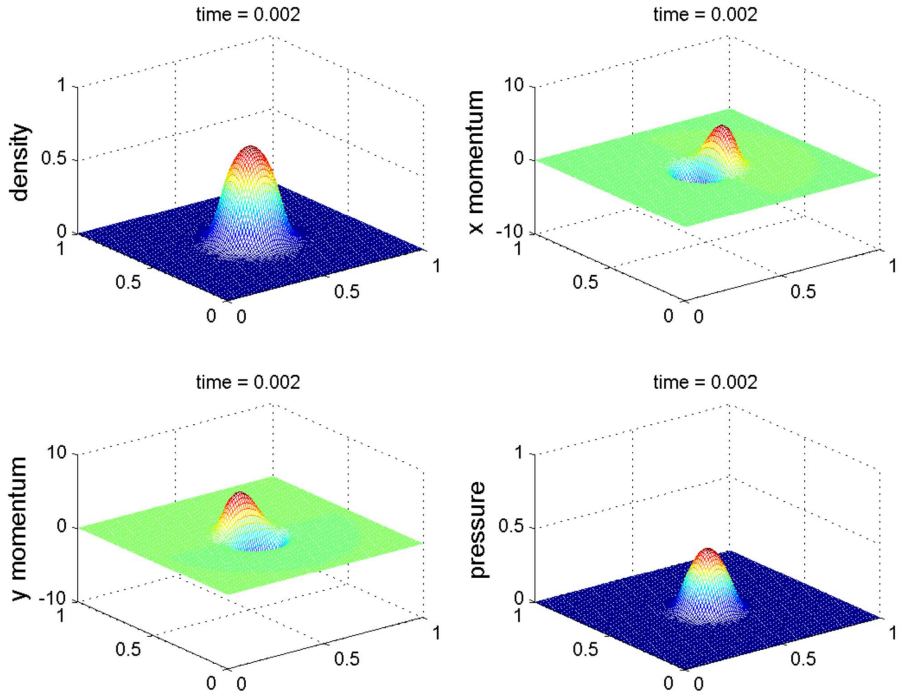


Figure 5.8: Numerical results at final time $T = 0.002$, $N = 100$, $\varepsilon = 0.05$, CFL $= 0.5$

5.4.2 Numerical tests for the Full Euler Equations

We proceed by presenting the numerical results for the Full Euler Equations, in which we use a second-order reconstruction both in time and in space with $\theta = 1$.

Example 1: Two Colliding Acoustic Pulses

We consider two colliding acoustic pulses in a weakly compressible regime. This test is present in [3] and [6]. The initial conditions are:

$$\rho_\varepsilon(x,0) = \rho_0 + \frac{\varepsilon}{2}\rho_1\left(1-\cos\left(\frac{4\pi x}{L}\right)\right),\ \rho_0 = 0.955,\ \rho_1 = 2,$$

$$u_\varepsilon(x,0) = -\frac{1}{2}u_0\mathrm{sgn}(x)\left(1-\cos\left(\frac{4\pi x}{L}\right)\right),\ u_0 = 2\sqrt{\gamma},$$

$$p(x,0) = p_0 + \frac{\varepsilon}{2}p_1\left(1-\cos\left(\frac{4\pi x}{L}\right)\right),\ p_0 = 1,\ p_1 = 2\gamma,$$

where L is the length of the domain, that in this case is $\left[-\frac{2}{\varepsilon}, \frac{2}{\varepsilon}\right]$.
We impose the following boundary conditions: void Neumann boundary conditions on the density, on the energy and thus on the pressure and void Dirichlet boundary conditions on the momentum.

We consider the numerical results obtained with $N = 440$ at different times $T = 0.815$ and $T = 1.63$. We choose $\varepsilon = \frac{1}{11}$ and $\varepsilon = 10^{-4}$, $\gamma = 1.4$ and CFL $= 0.5$.

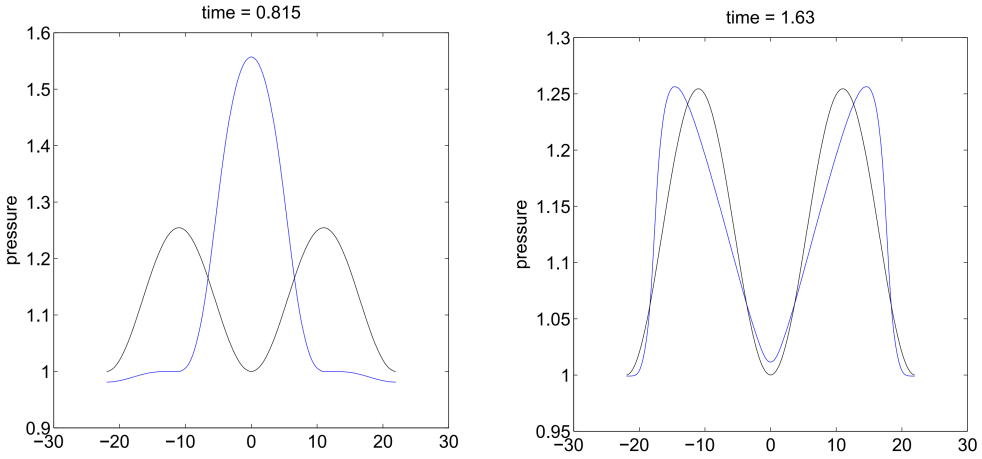


Figure 5.9: Pressure at two different final times $T = 0.815$ (left panel) and $T = 1.63$ (right panel), $N = 440$, $\varepsilon = \frac{1}{11}$, CFL $= 0.5$

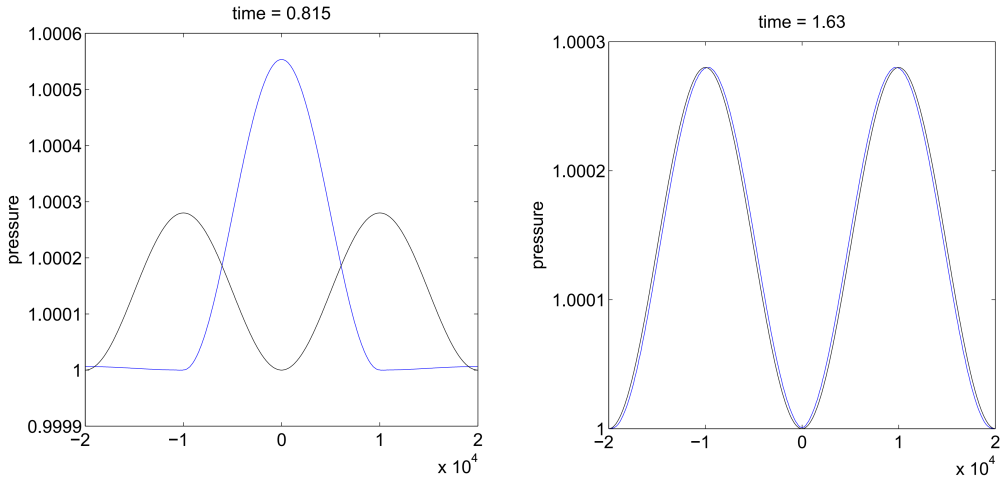


Figure 5.10: Pressure at two different final times $T = 0.815$ (left panel) and $T = 1.63$ (right panel), $N = 440$, $\varepsilon = 10^{-4}$, CFL $= 0.5$

In the case of a semi-implicit method we can choose a CFL higher than 1. If we choose CFL $= 3$ which corresponds to a $\text{CFL}_u = 0.44$ we obtain:

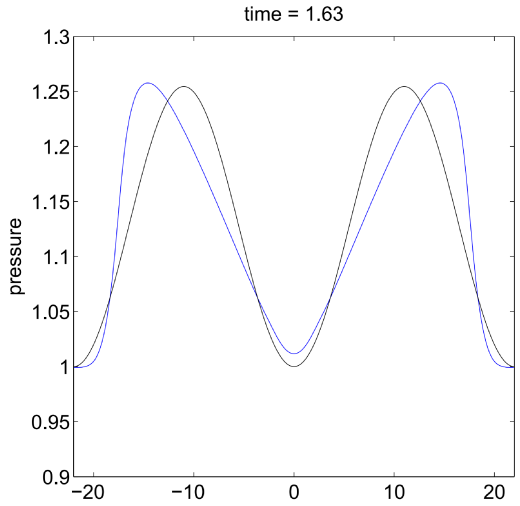


Figure 5.11: Pressure at final time $T = 1.63$, $N = 440$, $\varepsilon = \frac{1}{11}$, CFL $= 3$

The black line in the figures above is the initial condition.

Remark 5.1 - on the stability of semi-implicit methods: Now we suppose to move the right wall to the left. The situation is shown in the following figure (Figure 5.12):

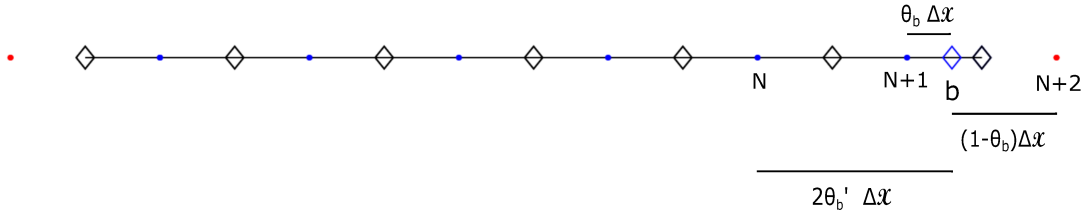


Figure 5.12: The wall has moved into the blue diamond

In this case to impose the Dirichlet boundary condition in the right wall we consider the interpolation procedure at the nodes x_{N+2} and x_{N+1}:

$$\theta_{\mathrm{b}}\mathbf{U}_{i,j}(x_{N+2}) + (1-\theta_{\mathrm{b}})\mathbf{U}_{i,j}(x_{N+1}) = 0,$$

from which:

$$\mathbf{U}_{i,j}(x_{N+2}) = -\frac{1-\theta_{\mathrm{b}}}{\theta_{\mathrm{b}}}\mathbf{U}_{i,j}(x_{N+1}), \tag{5.20}$$

where $\theta_{\mathrm{b}} = \frac{\mathrm{b}-x_{N+1}}{\Delta x}$.
When $\theta_{\mathrm{b}} = 0.5$ we have the classic case that we discussed previously.

We can observe that in the condition (5.20) when θ_{b} is very small, stability problems can occur. For example, if we choose $\theta_{\mathrm{b}} = 0.005$, $N = 440$, $\varepsilon = \frac{1}{11}$, we have stability problems. To remedy this problem due to small values of θ_{b}, we perform the linear interpolation at the nodes x_{N+2} and x_N:

$$\theta'_{\mathrm{b}}\mathbf{U}_{i,j}(x_{N+2}) + (1-\theta'_{\mathrm{b}})\mathbf{U}_{i,j}(x_N) = 0,$$

from which:

$$\mathbf{U}_{i,j}(x_{N+2}) = -\frac{1-\theta'_{\mathrm{b}}}{\theta'_{\mathrm{b}}}\mathbf{U}_{i,j}(x_N),$$

in which the new value of θ_{b} is $\theta'_{\mathrm{b}} = \frac{\mathrm{b}-x_N}{2\Delta x}$, that it is bigger than θ_{b}, thus eliminating the problem of stability.
For example, if $\theta_{\mathrm{b}} = 0.005$, $\theta'_{\mathrm{b}} = 0.5025$.

Example 2: 2D Sod Shock Tube Problem

Finally, we show the results obtained on a problem with radial symmetry [20]. We consider a shock tube initial condition with radial symmetry, that is

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} (1, 0, 0, 1) & (x-0.5)^2 + (y-0.5)^2 \leq (0.2)^2 \\ (0.125, 0, 0, 0.1) & \text{otherwise} \end{cases}.$$

We have imposed the following boundary conditions:

- void Dirichlet boundary conditions on the momentums
- void Neumann boundary conditions on the density, energy end pressure

The computational domain is $[0, 1]^2$. The Figure 5.13 shows the achieved results at final time $T = 0.1$ choosing $N = 100$ (on the top) and $N = 200$ (on the bottom) grid points in each direction and CFL $= 0.5$. The scatter plots are computed rewriting the solution (ρ, u, v, p) as a function of $r = \sqrt{(x-0.5)^2 + (y-0.5)^2}$.

This numerical test has been proposed by [20], in which the authors present a fourth-order central scheme for two-dimensional hyperbolic systems of conservation laws.

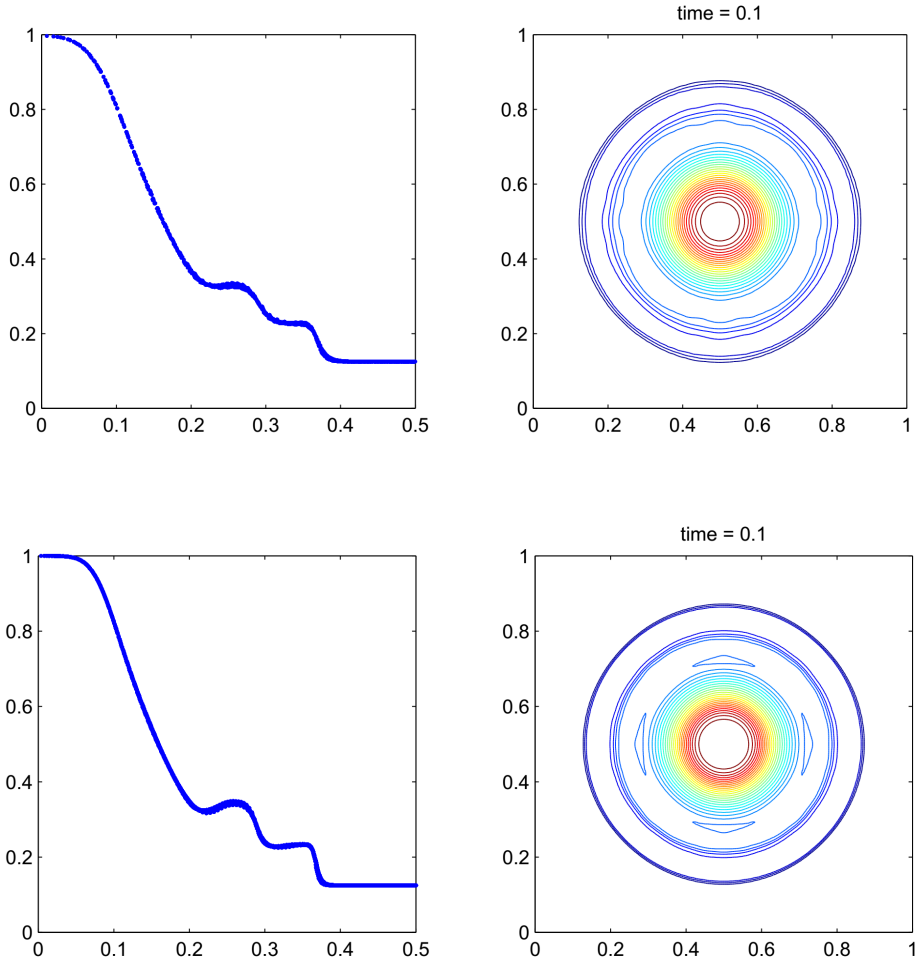


Figure 5.13: Solution with circular symmetry at final time $T = 0.1$. Scatter plots (left panels) and contour plots (right panels) for the density, for $N = 100$ (top panels) and $N = 200$ (bottom panels), CFL $= 0.5$, $\varepsilon = 1$. The contour plots have 30 equally spaced contour lines

We performed all the numerical tests mentioned above also solving with respect to the pressure. This is to make sure that numerical results are not altered, bacause in the next Section, when working on arbitrary domains, it is more convenient to work on pressure rather than energy. We obtained the same behavior.

5.5 Numerical tests for the Semi-Implicit methods on arbitrary domains

In this last Section we show the numerical results for the Full Euler Equations on two-dimensional arbitrary domains, which we talked about in the Section 5.3.

We consider the example present in [8], that is a simple wave that propagates around a steady disk with $x_C = 0.6$ and $y_C = 0.5$.

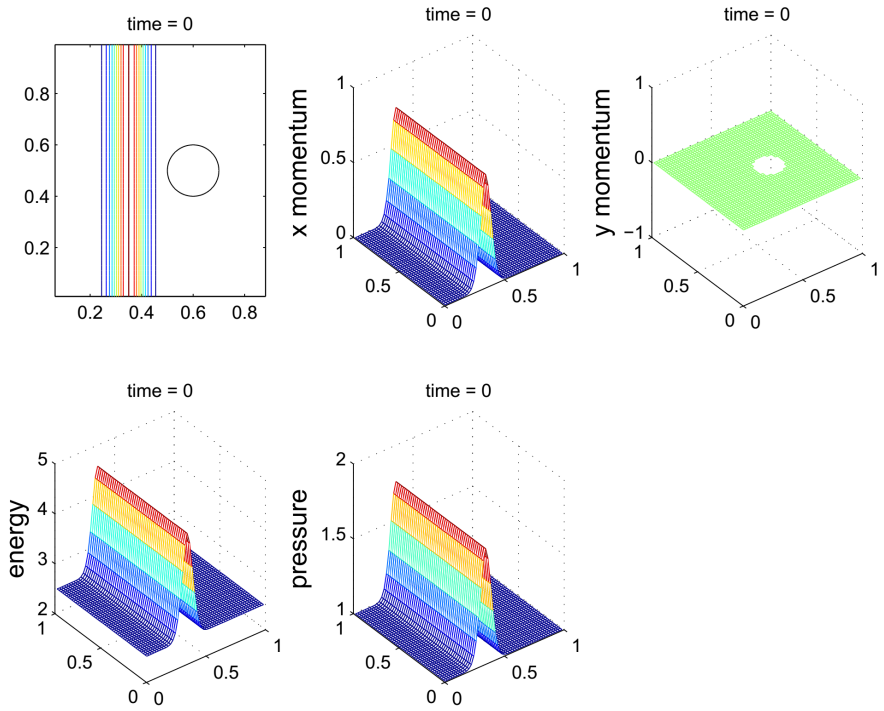


Figure 5.14: Initial conditions with a circular obstacle. The first figure on the upper left shows the contour plots of density

The initial conditions are given by:

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} (\tilde{\rho}, \tilde{u}, 0, \tilde{p})(x, y) & |x - 0.35| < 0.25 \\ (\rho_0, 0, 0, p_0) & \text{otherwise} \end{cases}, \tag{5.21}$$

where

$$\tilde{u}(x, y) = 0.5e^{-\frac{(x-0.35)^2}{0.005}},$$

$$\tilde{\rho}(x,y) = \rho_0 \left(1 + \frac{\gamma - 1}{2}\frac{\tilde{u}(x,y)}{c_0}\right)^{\frac{2}{\gamma-1}},$$

$$\tilde{p}(x,y) = p_0 \left(\frac{\tilde{\rho}(x,y)}{\rho_0}\right)^{\gamma}.$$

We set $\rho_0 = p_0 = 1$ and $c_0 = \sqrt{\frac{\gamma p_0}{\rho_0}} = \sqrt{1.4}$.

Absence of the obstacle

We have first solved the problem in the absence of the obstacle. We use a second-order reconstruction both in time and in space with $\theta = 1$. The following figure shows the results obtained choosing $N = 100$ along each axis, $\varepsilon = 1$, CFL $= 0.5$ and final time $T = 0.1$:

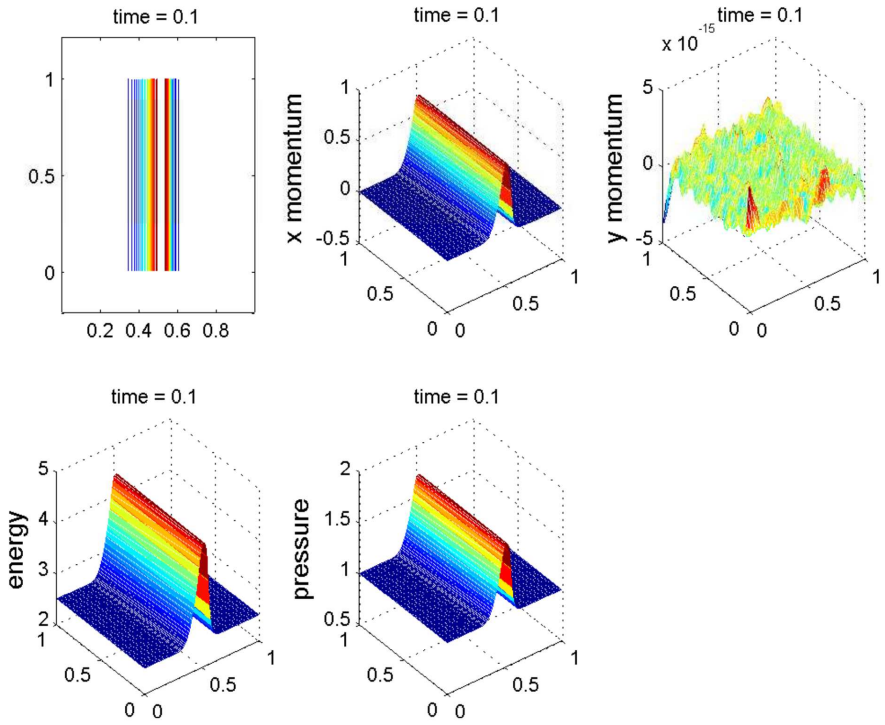


Figure 5.15: Solution at final time $T = 0.1$. The first figure on the upper left shows the contour plots of density. The contour plots have 10 equally spaced contour lines

We have imposed void Neumann boundary conditions on the momentums, density, energy and pressure.

We now introduce an obstacle. We found the numerical results by first introducing a square obstacle and then a circular obstacle. In particular, we have solved the problem on the pressure and not on the energy, to more easily impose the boundary conditions (5.15), (5.16), (5.17) and (5.18). We have imposed the following boundary conditions.

1. For the ghost points inside the obstacle we have imposed the conditions (5.15), (5.16), (5.17) and (5.18)
2. For the ghost points outside the unit square we have imposed
   - void Neumann boundary conditions on the energy, pressure, density and v (tangential velocity)
   - void Dirichlet boundary conditions on u (normal velocity)

We use a second-order reconstruction both in time and in space, with $\theta = 1$.

Square obstacle

We first solved the problem by considering a square obstacle. This is because in the case of a square obstacle the conditions (5.15), (5.16), (5.17) and (5.18) are much simpler (they coincide with the boundary conditions that we impose on the border of the unitary square). This is an example of a square obstacle:

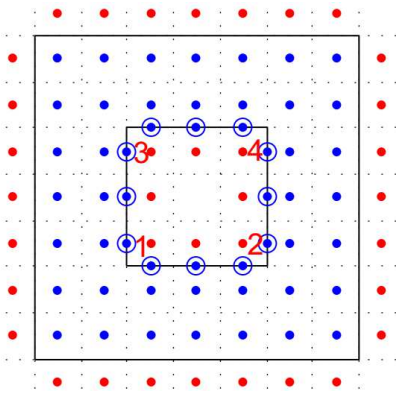


Figure 5.16: Square obstacle

In the case of the square obstacle for the ghost points that we have indicated with **1**, **2**, **3** and **4** we have two unknowns: one along the x axis and one along the y axis.

We have verified that the explicit method and the semi-implicit method give similar results.
The following figure shows the results obtained choosing $N = 100$ along each axis, $\varepsilon = 1$, CFL $= 0.5$ and final time $T = 0.2$, in case we use a semi-implicit method:

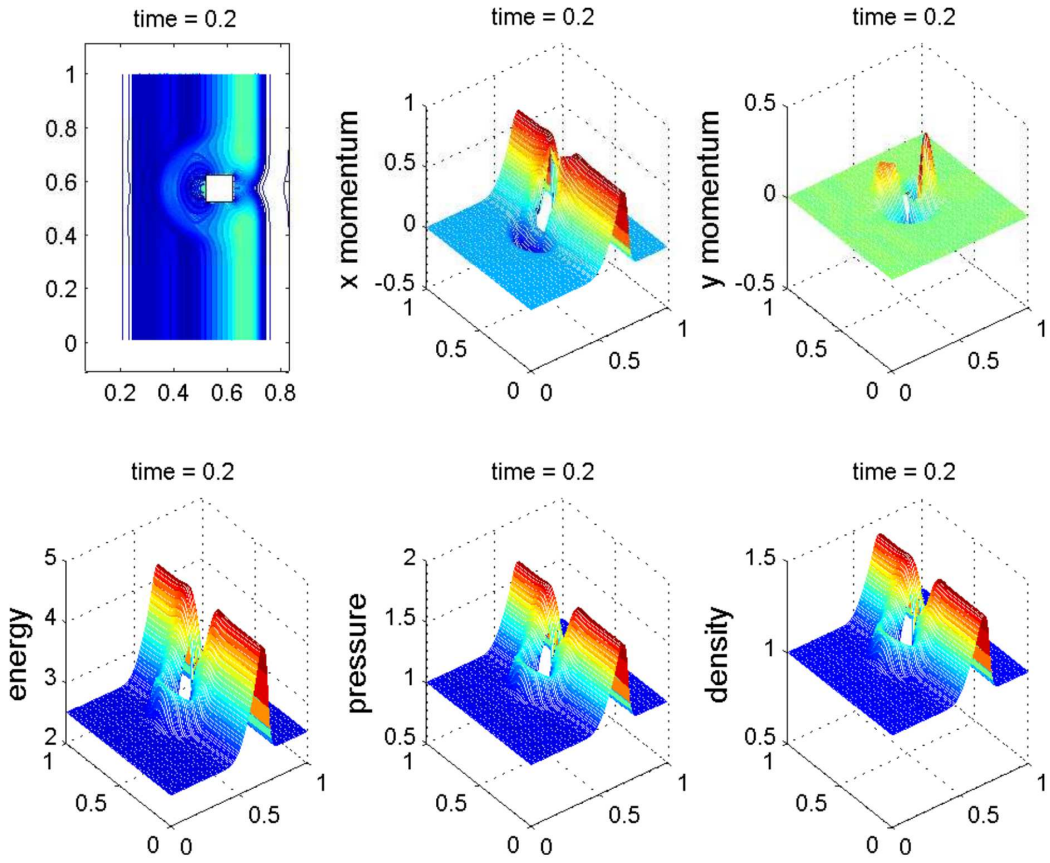


Figure 5.17: Numerical solution at final time $T = 0.2$ with a square obstacle, $N = 100$, $\varepsilon = 1$, CFL $= 0.5$. The first figure on the upper left shows the contour plots of density. The contour plots have 10 equally spaced contour lines

Circular obstacle

Finally, we have solved the problem in a domain with a circular obstacle whose initial conditions are the (5.21).

We have verified that the explicit method and the semi-implicit method give similar results.
The following figure shows the results obtained choosing $N = 100$ along each axis, $\varepsilon = 1$, CFL $= 0.5$ and final time $T = 0.2$, in case we use a semi-implicit method:

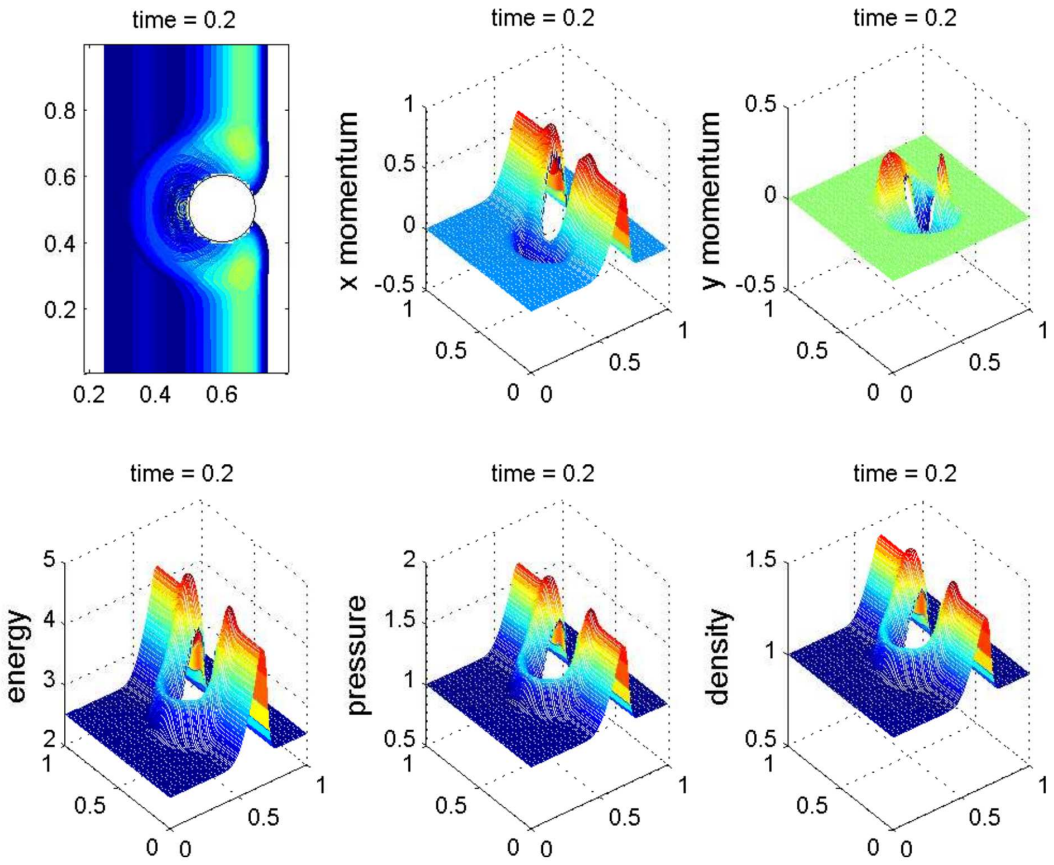


Figure 5.18: Numerical solution at final time $T = 0.2$ with a circular obstacle, $N = 100$, $\varepsilon = 1$, CFL $= 0.5$. The first figure on the upper left shows the contour plots of density. The contour plots have 10 equally spaced contour lines

We show that the method works well for CFL higher than one. In the following figure we show a comparison between the results obtained with CFL = 0.5 (left panel) and those obtained with CFL = 2 (right panel).

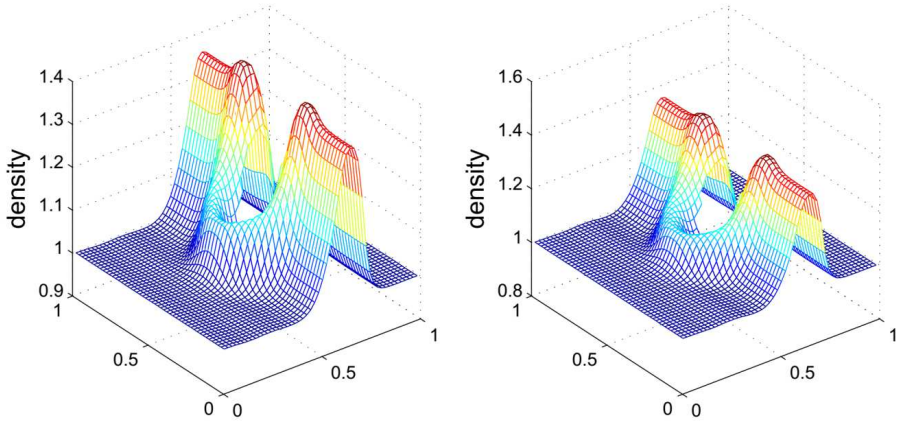


Figure 5.19: Density at final time $T = 0.2$ with CFL $= 0.5$ (left panel) and CFL $= 2$ (right panel), $N = 50$, $\varepsilon = 1$

To compute the accuracy order of this method, we proceed as follows. We first compute the solutions on 3 uniform grids with $\Delta x = \frac{1}{50}$, $\Delta x = \frac{1}{100}$ and $\Delta x = \frac{1}{200}$. Then we compute the L^1- and L^∞-norms of the differences between the solutions computed on two consecutive grids. Thus, to compute the experimental rates of convergence at the uniform grid of size Δx we use the Aitken's formula [2]:

$$r_p = \log_2 \left(\frac{\|\rho_{2\Delta x} - \rho_{4\Delta x}\|_{L^p}}{\|\rho_{\Delta x} - \rho_{2\Delta x}\|_{L^p}} \right), \ p = 1, \infty.$$

In the following table we show the experimental rates of convergence at final time $T = 0.1$, with $\varepsilon = 1$ and CFL $= 0.5$:

$\Delta x = \Delta y$	r_1	r_∞
$\frac{1}{100}$	-	-
$\frac{1}{200}$	1.2010	0.6402

First, we can note that the L^∞ convergence rate is lower with respect the L^1 convergence rate. This can be explained by the clipping phenomenon typical for the generalized minmod reconstruction.

The table shows that the method is first-order accurate. Probably this loss of accuracy compared to the method proposed in [8] is due to the use of bilinear interpolations rather than biquadratic interpolations. Maybe increasing the number of grid points to 400 (for each axis) we could get a value that is close to 2 also with bilinear interpolations, but the method turns out to be slow computationally due to the high number of grid points. For greater precision on the accuracy order of the method it is therefore advisable to work using biquadratic interpolations.

Remark 5.2: Thanks to **d'Alembert's solution** we know the exact solution in the case of an irrotational and stationary current of a perfect fluid in uniform motion on an object immersed in this fluid, in the absence of external forces. For simplicity we present the two-dimensional case with the object of circular shape, but the results are independent of the shape of the object.

Suppose we have a perfect two-dimensional incompressible fluid with the boundary conditions:

$$\mathbf{v}(x, y) \to Ue_1 \text{ if } (x, y) \to \infty,$$

$$\mathbf{v} \cdot \mathbf{n} = 0 \text{ on the border of the obstacle.}$$

Assuming the circular-shaped obstacle of radius R and introducing the polar coordinates (centered in the center of the obstacle) we have:

$$\lim_{r\to\infty} \mathbf{v}(r, \vartheta) = U(\cos\vartheta e_r - \sin\vartheta e_\vartheta),\ \mathbf{v}_r(R, \vartheta) = 0.$$

The irrotationality of the motion leads us to introduce the kinetic potential $\varphi(r, \vartheta)$ and for the incompressibility we have $\Delta\varphi = 0$. The solution is

$$\varphi(r, \vartheta) = U\left(\frac{R^2}{r} + r\right)\cos\vartheta.$$

From this the components of the velocity are derived:

$$\mathbf{v}_r(r, \vartheta) = U\left(1 - \frac{R^2}{r^2}\right)\cos\vartheta,\ \mathbf{v}_\vartheta(r, \vartheta) = U\left(1 + \frac{R^2}{r^2}\right)\sin\vartheta.$$

Now we want to find an approximation of exact solution (in Cartesian coordinates) supposing that the solution repeats itself periodically (periodic conditions).
We place $\varphi_{i,j}(x,y) := \varphi(x-2\pi i, y-2\pi j) : [-\pi i, \pi j]^2 \longrightarrow \mathbb{R}$ and we consider, fixed a certain value n, the following function:

$$\Phi_n(x,y) = \sum_{i,j=-n,\ldots,n} \varphi_{i,j}(x,y) - [(2n+1)^2 - 1]x. \tag{5.22}$$

In doing so we have extended (periodically) the solution φ of the d'Alembert Paradox in $(2n+1)^2$ squares of width 2π along each axis. The following figure show the extension of d'Alembert's solution on nine squares ($n = 1$):

$\varphi_{-1,1}$	$\varphi_{-0,1}$	$\varphi_{1,1}$
$\varphi_{-1,0}$	$\varphi_{0,0}$	$\varphi_{1,0}$
$\varphi_{-1,-1}$	$\varphi_{0,-1}$	$\varphi_{1,-1}$

Figure 5.20: Extension of d'Alembert's solution on nine squares ($n = 1$)

We can suppose that the function Φ_n ($n \to \infty$) is an approximation of exact solution with periodic conditions. In the following figure (Figure 5.21), we shows the velocity plot (quiver) with $n = 50$:

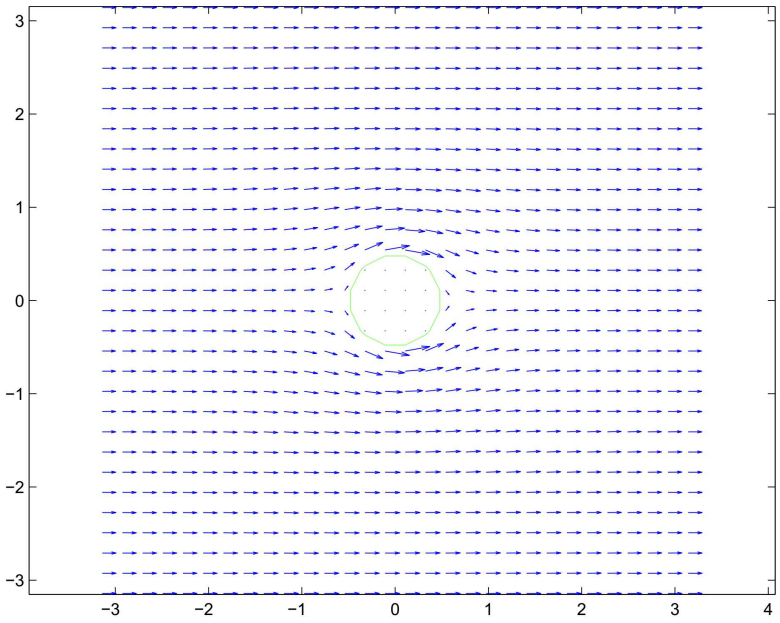


Figure 5.21: Velocity plot with $n = 50$ ($N = 30$: number of grid points along each axes on each square)

Our future aim is to prove that the the error given by the difference between Φ_n (which we think is an approximation of the exact solution) and the numerical solution tends to zero as n increases.

Conclusions and future works

In this thesis we have presented a finite-difference ghost-point method to solve elliptic and hyperbolic equations on arbitrary domains. The equations are discretized on a uniform Cartesian grid.
At first we applied the Coco-Russo method, which represents a generalization of the finite-difference method for the elliptic equations on arbitrary domains, at the resolution of the Poisson equation. This method proposes a polynomial interpolation technique to impose boundary conditions and therefore the interpolation error can influence the accuracy order of the method itself. We have proposed linear and bilinear interpolation techniques. These conditions are imposed on the projections of the ghost points on the border of the domain. We have also presented a rigorous proof of the stability and convergence of the numerical method in the one-dimensional case. The proof of the stability of the Coco-Russo method in the two-dimensional case is instead a still open problem, since the matrix associated to the method does not have a well-defined structure. Even the extension of the proofs seen in the one-dimensional case to the two-dimensional case has not obtained results because the matrix in the two-dimensional case is not a M-matrix as is the matrix associated to the method in the one-dimensional case. However, the numerical tests performed on the behaviors of the inverse matrix of the method, of the inverse of the Schur complement, of the error and of the consistency error confirm the stability and convergence of the Coco-Russo method in 1D, 2D and 3D, in the case of Dirichlet problems and in the case of mixed problems. The results obtained suggest us to look for the stability of the method in the ∞-norm, since we have obtained:

- $\|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$,
- $\rho(\mathrm{A}_h^{-1}) \to \mathrm{c}$, where c is a constant,
- $\|\mathrm{S}_h^{-1}\|_\infty \sim \mathrm{O}(N^0)$,

where S_h is the Schur complement and $N = \frac{1}{h}$. In particular in the one-dimensional case numerical and theoretical results show that $\|\mathrm{A}_h^{-1}\|_\infty \sim \mathrm{O}(h^{-\frac{1}{p}})$, $p = 1, 2, \infty$.
The results on the behaviors of the error and of the consistency error depend instead on the problem (Dirichlet or mixed) and on the interpolation procedures performed.
For the Dirichlet problems we have obtained:

- $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$,

both in the case of linear interpolations and in the case of quadratic interpolations.
For the mixed problems the accuracy orders of $\|\tau_h\|_{L^p}$ and of $\|e_h\|_{L^p}$ depend on the procedure with which we impose the Neumann boundary conditions. If we perform the trilinear interpolation procedures we have:

- $\|\tau_h\|_{L^1} \sim \mathrm{O}(h^2),\ \|\tau_h\|_{L^2} \sim \mathrm{O}(h^{\frac{3}{2}}),\ \|\tau_h\|_{L^\infty} \sim \mathrm{O}(h)$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h),\ p = 1, 2, \infty$,

because the Neumann conditions confers a first-order accuracy at the border in the case in which we perform the trilinear interpolations. If we perform the triquadratic interpolation procedures we have:

- $\|\tau_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$;
- $\|e_h\|_{L^p} \sim \mathrm{O}(h^2),\ p = 1, 2, \infty$.

Therefore, in the Dirichlet problem both the linear interpolations and the quadratic interpolations provide a second-order accuracy, while in mixed problem only the quadratic interpolations provide such accuracy.

Once we certain of the convergence and stability of the Coco-Russo method,

our interest it has moved to the study of the Euler equations of the gas dynamic. This is because in the last decades many numerical methods have been developed for the resolution of hyperbolic systems of conservation laws. A recent work [3] we have referred to has presented semi-implicit methods for the resolution of such systems, which have the advantage of eliminating the restriction on the spatial step for the acoustic waves that suffer the explicit methods. They also have advantages with respect implicit methods, since the completely implicit methods are more complex to solve and introduce excessive numerical dissipation. In the semi-implicit methods the differential operators in space relative to convective of material fluxes are explicitely discretized with Lax-Friedrichs fluxes and the operators relative to acoustic waves are discretized implicitely with central differences.
These methods have suggested the development of semi-implicit methods for Euler equations on domains that have obstacles, in which the Coco-Russo method is applied to impose boundary conditions in a manner similar to elliptic equations. This method being semi-implicit overcomes the problem of spatial restriction present in [8], in which the authors apply the Coco-Russo method for the resolution of Euler equations on domains with obstacles but applying explicit methods.

Future works

The future developments of the present thesis are different.

As regards the first part of the thesis is concerned, as we have already said, the problem of the stability of the Coco-Russo method in the two-dimensional case is still open. A rigorous proof of the stability of the method is necessary to confirm the numerical evidence.

As regards the second part of the thesis, it is possible to substitute the bilinear interpolations with the biquadratic interpolations in the semi-implicit method for hyperbolic equations on arbitrary domains described in Section 5.3. We want also to prove that the solution (5.22) found as an extension of d'Alembert's solution is actually an approximation of the exact solution of Euler equations on a domain with a circular obstacle in the incompressible regime. That is to say, it must be proved, that for a sufficiently large value of the final time, the error between numerical solution and exact solution tends

to zero as the spatial step decreases.

It is also possible to find an application of the semi-implied methods in [1], in which the author develops completely implicit methods for the simulation of flows of compressible materials, both fluid and elastic solids. The author discretize the equations on a regular Cartesian grid.

Finally, it is possible to further improve the stability and precision properties of semi-implicit methods by performing an automatic step control technique [2, 16], since the error committed in the discretization method primarily depends on the time-step size, which is varied along the solution in order to minimize the computational effort.

Bibliography

[1] E. Abbate. *Numerical methods for all-speed flows in fluid-dynamics and non-linear elasticity*. PhD thesis, Università dell'Insubria, Como, 2018.

[2] K. E. Atkinson. *An introduction to numerical analysis*. John Wiley & Sons Inc., New York, second ed., 1989.

[3] S. Avgerinos, F. Bernard, A. Iollo and G. Russo. Linearly Implicit All Mach Number Shock Capturing Schemes. Submitted to *Journal of Computational Physics*.

[4] A. Baeza, P. Mulet and D. Zorío. High order boundary extrapolation technique for finite difference methods on complex domains with Cartesian meshes. *J. Sci. Comput.*, 66(2):761-791, 2016.

[5] S. Boscarino, F. Filbet and G. Russo. High order semi-implicit schemes for time dependent partial differential equations. *Journal of Scientific Computing*, 1-27, 2016.

[6] S. Boscarino, G. Russo and L. Scandurra. All Mach Number Second Order Semi-Implicit Scheme for the Euler Equations of Gas Dynamics. *Journal of Scientific Computing*, preprint 2018.

[7] A. Brandt. Generalized Local Maximum Principles for Finite-Difference Operators. *Mathematic of Computation*, 27:685-718, 1973.

[8] A. Chertock, A. Coco, A. Kurganov and G. Russo. A Second-Order Finite-Difference Method for Compressible Fluids in Domains with Moving Boundaries. *Journal of Computational Physics*, 23(1):230-263, 2018.

[9] A. Coco. *Finite-Difference Ghost-Cell Multigrid Methods for Elliptic Problems with Mixed Boundary Conditions and Discontinuous Coefficients*. PhD thesis, Università di Catania, 2011.

[10] A. Coco and G. Russo. Finite-Difference Ghost-Point Multigrid Methods on Cartesian Grids for Elliptic Problems in Arbitrary Domains. *Journal of Computational Physics*, 241:464-501, 2013.

[11] F. Cordier, P. Degond and A. Kumbaro. An Asymptotic-Preserving all-speed scheme for the Euler and Navier-Stokes equations. *Journal of Computational Physics*, 231(17):5685-5704, 2012.

[12] S. Dellacherie. Analysis of Godunov type schemes applied to the compressible Euler system at low Mach number. *Journal of Computational Physics*, 229(4):978-1016, 2010.

[13] F. Gibou, R. Fedkiw, L.-T. Cheng and M. Kang. A Second-Order-Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains. *Journal of Computational Physics*, 176(1):205-227, 2002.

[14] E. Godlewski and P.-A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer, 2014.

[15] W. Hackbusch. *Elliptic Differential Equations: Theory and Numerical Treatment*. Springer, 2003.

[16] S. Ilie, G. Söderlind and R. M.Corless. Adaptivity and computational complexity in the numerical solution of ODEs. *Journal of Complexity*, 24(3):341-361, 2008.

[17] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Lectures in Mathematics, Birkhauser Verlag, Basel, 1992.

[18] R. J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

[19] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, 2007.

[20] D. Levy, G. Puppo and G. Russo. A Fourth-Order Central WENO Scheme for Multidimensional Hyperbolic Systems of Conservation Laws. *SIAM Journal of Scientific Computing*, 24(2):480-506, 2002.

[21] K. W. Morton and D. Mayers. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, 2005.

[22] G. Naldi, L. Pareschi and G. Russo. *Introduzione al Calcolo Scientifico. Metodi e Applicazioni con Matlab*. McGraw -Hill, 2001.

[23] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. vol. 153 of Applied Mathematical Sciences, Springer-Verlag, New York, 2003.

[24] L. Pareschi and G. Russo. Implicit-Explicit Runge-Kutta Schemes and Applications to Hyperbolic Systems with Relaxation. *Journal of Scientific Computing*, 25(1):129-155, 2005.

[25] A. Quarteroni and R. Sacco. *Numerical approximation of partial differential equations*. Springer, 1994.

[26] A. Quarteroni, R. Sacco and F. Saleri. *Numerical mathematics*. Springer, 2000.

[27] J. A. Sethian. *Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. vol. 3 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, second ed., 1999.

[28] J. Sherman and W. Adjustment of an Inverse Matrix Corresponding to Changes in the Elements of a Given Column or a Given Row of the Original Matrix (abstract). *Annals of Mathematical Statistics*, 20(4):620-624, 1949.

[29] J. Sherman and W. Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *Annals of Mathematical Statistics*, 21(1):124-127, 1950.

[30] G. H. Shortley and R. Weller. The numerical solution of Laplace's equation. *J. Appl. Phys.*, 9(5):334-348, 1938.

[31] C.-W. Shu. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws. *Advanced numerical approximation of non-linear hyperbolic equations*, Lecture Notes in Mathematics, 1697:325-432, 2000.

[32] C.-W. Shu. High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems. *SIAM Rev.*, 51(1):82-126, 2009.

[33] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83(1):32-78, 1989.

[34] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Third edition, 2009.