

The Shooting Method for the Numerical Solution of the Poisson Equation: A Control Viewpoint¹

C.E. SCHAEERER², E. KASZKUREWICZ³, NACAD-COPPE-UFRJ, Cx.P. 68.516, 21.945-970 Rio de Janeiro, RJ, Brazil.

N. MANGIAVACCHI⁴, Dept. of Mech. Eng., UERJ, Rua São Francisco Xavier 524, 20550-013 Rio de Janeiro, RJ, Brazil.

Abstract. A discrete-space state representation is used to provide a feedback control-theoretical formulation for the iterative shooting method. This method historically defined for boundary value problem in ordinary differential equations, now is presented for the numerical solution of elliptical partial differential equations, specifically it is focused on the numerical solution of the Poisson equation, which leads to a new a more efficient method for the solution of this type of equations.

1. Introduction

The objective of this paper is to discuss the concept of the shooting method using a control theoretical perspective, for the numerical solution of the Dirichlet problem for the elliptical second order partial differential equation (Poisson equation) expressed as

$$\Delta u = f \quad \text{in } \Omega. \quad (1.1)$$

The Poisson equation (1.1) is subjected to Dirichlet boundary conditions

$$u = g \quad \text{on } \partial\Omega, \quad \text{where } (x, y) \in \Omega.$$

These boundary conditions in a square domain $\Omega = \{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}$ are expressed as $u(0, y) = \alpha_a(y)$; $u(1, y) = \alpha_b(y)$; $u(x, 0) = \alpha_c(x)$; $u(x, 1) = \alpha_d(x)$. Since we are interested in the numerical solution of Poisson equation, we discretize the square domain Ω using an uniform mesh with spacing $h = 1/N$ in both directions (x and y), hence, we obtain a discrete square grid domain $\Omega^h = \{(hi, hj) : i, j = 0, 1, \dots, N\}$. In the interior of the discrete domain Ω^h the

¹Authors gratefully acknowledge the financial support given by FAPERJ and CNPq (grant: 460473/2001-8).

²Correspondence to Schaerer, cschaer@coep.ufrj.br

³eugenius@coep.ufrj.br

⁴norberto@uerj.br

well known “five points finite differences approximation” describes the relationship between the values of the variables at the grid points. This difference equation is normally arranged in a lexicographic ordering by rows of constant i , so that the unknowns corresponding to the row i of the grid may be arranged into a vector $u_i = [u_{i,1} \dots u_{i,n}]^T$ for $1 \leq i \leq n$, where $n = N - 1$. Similarly with the input function $f_i = [f_{i,1}, \dots, f_{i,n}]^T$, in consequence, the difference equation can be written in a form $Au = b$ with the structure

$$\begin{bmatrix} T & F & & & & & \\ F & T & F & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & T & F & \\ & & & & F & T & F \\ & & & & & F & T \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} b_1 + Fu_0 \\ b_2 \\ \vdots \\ b_{N-3} \\ b_{N-2} \\ b_{N-1} + Fu_N \end{bmatrix}, \quad (1.2)$$

where matrix A has dimension $A \in \mathbb{R}^{n^2 \times n^2}$, the unknown vector $u \in \mathbb{R}^{n^2}$ and the vector $b \in \mathbb{R}^{n^2}$. Matrices $T, F \in \mathbb{R}^{n \times n}$ where matrix $F = -I$ is an identity matrix and T is a tridiagonal matrix. In equation (1.2), the vector u_k corresponds to the unknowns at row k of the discrete domain, and u_0 and u_N correspond to the prescribed boundary values at the upper and lower borders of the square domain (row $i = 0$ and $i = N$), respectively, i.e. $u(0, hj) = \alpha_a(hj)$ and $u(1, hj) = \alpha_b(hj)$. The values of the subvector b_1, b_2, \dots, b_{N-1} correspond to the values of $u_{i,j}$ at the point of the boundary ($u(hi, 0) = \alpha_c(hi)$ and $u(hi, 1) = \alpha_d(hi)$) and the values of the function $f_{i,j}$ at the interior points of the domain Ω^h .

2. The state space representation

Taking an arbitrary row i in equation (1.2), we have

$$Fu_{i-1} + Tu_i + Fu_{i+1} = b_i. \quad (2.1)$$

From this equation, it is possible that the values of $u_{i,j}$ at the row $i + 1$ (which is given by the vector u_{i+1}) be computed using the values of the two previous rows (rows i and $i - 1$). Defining the state space vector z_i at row i , the output vector $y_i \in \mathbb{R}^n$ and the input vector $q_i \in \mathbb{R}^n$ as

$$z_i := [u_i \ u_{i-1}]^T, \quad z_1 := [u_1 \ u_0]^T, \quad y_i := u_{i+1}, \quad q_i := -b_{i+1}.$$

Hence, the difference equation (2.1) can be arranged into a state space representation of the form

$$\begin{aligned} z_{i+1} &= Wz_i + Bq_i, \\ y_i &= Cz_i, \quad i = 1, \dots, n, \end{aligned} \quad (2.2)$$

where $z_i \in \mathbb{R}^{2n}$, $q_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}^n$ are respectively the state variable, the input and the output of the system at the i^{th} iteration. The matrices W, B and C have

dimensions $2n \times 2n$; $2n \times n$; and $n \times 2n$, respectively. Let I be an identity matrix with dimension $n \times n$. The matrices W , B and C , and the boundary conditions of the state space representation (2.2), are defined as follows:

$$W := \begin{bmatrix} T & I \\ I & 0 \end{bmatrix}, \quad B := \begin{bmatrix} I \\ 0 \end{bmatrix}, \quad C := [I \quad 0], \quad \begin{array}{l} \alpha_0 := u_0 = [\alpha_a(h) \dots \alpha_a(hn)]^T, \\ \alpha_N := u_N = [\alpha_b(h) \dots \alpha_b(hn)]^T. \end{array}$$

With this new definition the boundary conditions are understood as a two point boundary value conditions, i.e. the boundary conditions are specified at the “points $i = 0$ and $i = N$ ”. Hence, we define equation (2.2), as the state space representation of the two point boundary value problem (TPBVP) associated to the Dirichlet problem expressed by equation (1.1). The solution of this state-space representation is that of the difference equation (2.1) or, equivalently, the solution of the linear system (1.2).

3. A simple shooting method

Once the Dirichlet problem has been transformed into an associated TPBVP, we are interested in solving it by using a shooting procedure. A state space representation is used in the shooting method as it was presented in [4] for the case of ODEs. The iterative simple shooting method initially transforms the boundary value problem into an initial value problem (IVP) [4] by arbitrating some initial values which are not specified by the boundary conditions [3]. After this transformation, the numerical solution can be obtained via numerical integration which here will be called the “marching” process [1]. The estimated “initial values”, usually, do not correspond to the correct ones and, as a consequence, there is a discrepancy between the computed values, by the marching process, and the values given by the boundary conditions of the problem. This discrepancy is defined as the “error” of the shooting procedure. Consequently, in each iteration of the shooting, these values are adjusted in order to minimize the discrepancy (error) in the next iteration. In the context of this work, the adjustment of adopted values is performed via a feedback control law.

3.1. The state space representation of the iterative simple shooting method

Marching part. For equation (2.2) be an IVP, all boundary conditions need to be specified at one single point, and consequently all elements of the initial state vector z_0 would be known. In the case of two point boundary value problems, only the subvector $u_0 = \alpha_0$, that corresponds to the values that are associated to the first row of the grid, is known. Consequently, in order to solve (iterate) the equation (2.2), the subvector u_1 needs to be estimated. As a consequence, in order to transform the TPBVP associated to the Dirichlet problem, into an IVP, the “initial conditions” vector z_0 is split into two subvectors as follows:

$$z_0 = \begin{bmatrix} z_0^a \\ z_0^g \end{bmatrix} = \begin{bmatrix} u_1 \\ u_0 \end{bmatrix}, \quad (3.3)$$

where the subvectors $z_0^g \in \mathfrak{R}^n$ and $z_0^a \in \mathfrak{R}^n$ correspond, respectively, to the given initial values and the values that need to be supplied. With this initial condition vector (3.3), the solution of the initial value problem associated to the Dirichlet problem is given at the k^{th} iteration of the equation (2.2) by [2]

$$\begin{aligned} z_k &= W^k z_0 + \sum_{j=0}^{k-1} W_{k-j-1} B q_j, \\ y_k &= C z_k \quad k = 0, \dots, n. \end{aligned} \quad (3.4)$$

Equation of the error. If the given subvector has been set correctly, the elements of the computed vector u_N must be equal to the elements of the vector α_N which, by its turn, is given by the boundary condition $\alpha_b(hi)$ for $i = 1, \dots, n$. We define the error equation, in each iteration of the shooting associated to the iteration counter “ m ”, by the discrepancy between the prescribed correct value α_N and the computed vector $u_N(m)$, i.e.,

$$e(m) = \alpha_N - u_N(m), \quad (3.5)$$

where $e(m) \in \mathfrak{R}^n$. Considering $u_N(m) = y_n(m)$ and defining $w(m) := z_0(m)$, and using (2.2) and (3.5) at the m^{th} , an error equation as a function of the arbitrated subvector z^a is obtained as follows:

$$e(m) = \alpha_N - (CW^m w(m) + C\bar{b}), \quad \text{where} \quad \bar{b} := \sum_{j=0}^{m-1} W^{m-j-1} B q_j. \quad (3.6)$$

Feedback control law. In order to obtain a state-space representation for the shooting method, it is necessary to define an updating law for the corresponding state variables (arbitrated initial conditions) as a function of the error $e(m)$. This can be done through a so called “dynamic feedback law” of the form

$$\begin{aligned} \delta(m+1) &= A_c \delta(m) + B_c K e(m), \\ w(m) &= C_c \delta(m) + v, \quad m = 0, 1, 2, \dots, \end{aligned} \quad (3.7)$$

where $\delta := z_0^a$ is the state vector which corresponds to the subvector of z_0 . In the specific case of the Poisson equation that we are dealing with, the initial vector is u_1 . On other hand, the vector v in equation (3.7) corresponds to the part of the vector z_0 that is given by the boundary conditions, i.e. $v := [0 \quad z_0^g]^T$ is a constant vector.

The matrix $K \in \mathfrak{R}^{n \times n}$ expresses a gain matrix and the matrices $B_c \in \mathfrak{R}^{n \times n}$ and $C_c \in \mathfrak{R}^{2n \times n}$ play, respectively, the role of an input and output matrix of this dynamic controller in the sense that $\delta(m)$ does not affect all elements of the initial values vector $w(m)$. Combining (3.4), (3.5) and (3.7) the following state-space recurrent equation is obtained.

$$\begin{aligned} \delta(m+1) &= (A_c - B_c K C W^N C_c) \delta(m) + B_c K (\alpha_N - C W^N v - C \bar{b}), \\ w(m) &= C_c \delta(m) + v, \quad m = 0, 1, 2, \dots \end{aligned} \quad (3.8)$$

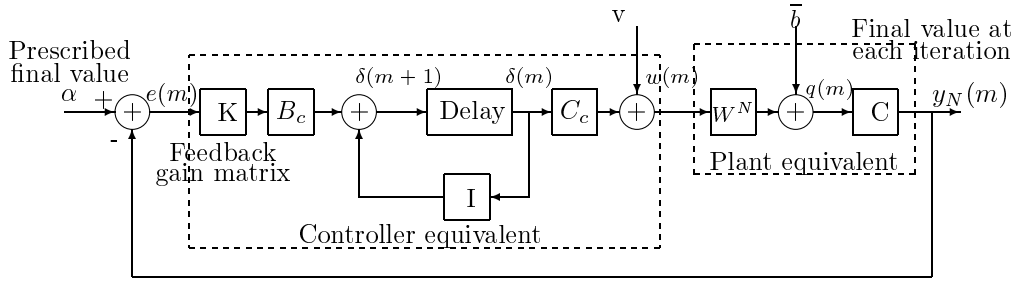


Figure 1: Block diagram of the shooting method from a control feedback perspective when using to solve numerically the Poisson equation

In this case, the dynamics of this iterative system is ruled by the feedback law given by (3.7) and the error, in each iteration, is defined by (3.5). Therefore, equation (3.8) describes a “state space representation of the iterative shooting method”. Figure 1 presents the block diagram of the simple shooting for the numerical solution of the Poisson equation (1). A control problem for the shooting method consists in building the feedback control law (3.7) specified by the controller matrix A_c , the input matrix B_c , the output matrix C_c and the feedback gain matrix K . The problem consists in choosing a set of these matrices in order that equation (3.8) will converge to the desired solution. The condition to obtain the convergence of the system (3.7) is the Schwarz condition for the iteration matrix $H := (A_c - B_c K C W^N C_c)$. This condition establishes that all the eigenvalues of the matrix H in modulus are less than one, or in other words, the spectral radius is less than one, i.e. $\rho(H) < 1$.

3.2. Multiple subdomains: a multiple shooting method

The multiple shooting is performed by subdividing the domain in several subdomains and using the simple shooting in each of these subdomains. In the case of ODEs, the domain of the independent variable x , defined as an interval or domain $\theta := [0, 1]$, it is subdivided into p intervals: $0 = x_0 < x_2 < \dots < x_{p-1} < x_p = 1$, and the simple shooting is performed in each subinterval $\theta_k := [x_k, x_{k+1}]$, $0 \leq k \leq p - 1$. Here we generalize this concept to the square domain Ω in which the Poisson equation is defined. In this case, as the domain Ω represents a 2-D domain, we determine a direction for the marching process of the shooting. We choose the x direction, and in consequence, we subdivide the 2-D domain Ω into p “strips”. Therefore, we can define the subdomain $\Omega_k = \{(x, y) : x \in \theta_k, 0 \leq y \leq 1\}$ where $0 \leq k \leq p - 1$. With the domain Ω subdivided in this way, the Dirichlet problem for the Poisson equation in each subdomain is defined as

$$\Delta u^k = f^k \quad \text{in } \Omega_k, \tag{3.9}$$

where u^k is the value of $u(x, y)$ such that the point $(x, y) \in \Omega_k$. Hence, the boundary conditions for a k^{th} subdomain are defined as

$$u_{x_k,y}^k = \alpha_a^k(y), \quad u_{x_{k+1},y}^k = \alpha_b^k(y), \quad u_{x,0}^k = \alpha_c^k(x), \quad u_{x,1}^k = \alpha_d^k(x).$$

3.3. The discrete two point boundary value problem in a subdomain

Since we are interested in the numerical solution of the PDE in the domain Ω , now this is subdivided in the x direction in p subdomains such that $p = N/N_s$, where N is the number of points in the x direction, and N_s is smaller than the maximum number of points that the simple shooting can solve in this direction. Hence, the k^{th} discrete strip subdomain is defined as $\Omega_k^h = \{(hi + d_k, hj) : i = 0, 1, \dots, N_s \text{ and } j = 0, 1, \dots, N\}$, where $d_k := hkN_s$.

Defining the state space vector z for the k^{th} subdomain as z^k , then we define the state space representation for the associated two point boundary value problem in the k^{th} subdomain as:

$$\begin{aligned} z_{i+1}^k &= Wz_i^k + Bq_i^k, \\ y_i^k &= Cz_i^k, \quad i = 1, \dots, n_s, \end{aligned} \tag{3.10}$$

subject to the initial condition $\alpha_0^k := u_0^k = [\alpha_a^k(h) \dots \alpha_a^k(hn_s)]^T$ and the final condition $\alpha_{N_s}^k := u_{N_s}^k = [\alpha_b^k(h) \dots \alpha_b^k(hn_s)]^T$.

The initial condition α_0^0 in the first subdomain ($k = 0$) and the final condition $\alpha_{N_s}^{p-1}$ in the last subdomain ($k = p-1$) correspond to the boundary conditions given by the Dirichlet to which the Poisson equation is subjected. In other words, $\alpha_0^0 = \alpha_0$ and $\alpha_{N_s}^{p-1} = \alpha_N$, respectively. Equation (3.10) is the TPBVP in the subdomain k^{th} . This problem is solved using the simple shooting method presented in section 3.

The continuity of the solution: Once the simple shooting reaches the solution in each subdomain Ω^k , the local solutions are patched up to form an overall global solution over the entire domain Ω . By using (3.10), the continuity (or matching) conditions for the solutions associated to the subdomains are given by [5]

$$z_{n_s}^k - z_0^{k+1} = 0, \quad \text{for } k = 0, 1, 2, \dots, p-2. \tag{3.11}$$

Using expression (3.4) in each subdomain k^{th} (vector z_i^k at the i^{th} iteration for $i = n_s$ (marching process at each subdomain)), then the continuity conditions expressed by (3.11) for the k^{th} subdomain take the form

$$z_{n_s}^k - z_0^{k+1} = A^{n_s} z_0^k + \overline{b}^k - z_0^1 = 0, \quad \overline{b}^k := \sum_{j=0}^{n-1} W^{n-j-1} Bq_j^k. \tag{3.12}$$

In order to arrange the equations (3.12) and (3.10), we define matrices C_0, C_N, B_0 and B_N as expressed at equation (3.13) below

$$C_0 z_0^0 = \alpha_0, \quad C_N z_{n_s}^{p-1} = \alpha_N, \quad B_0 C_0 z_0^0 + B_N C_N z_{n_s}^{p-1} = \begin{bmatrix} \alpha_0 \\ \alpha_N \end{bmatrix} = \beta. \tag{3.13}$$

Hence, equations (3.12), (3.10) and (3.13) can be arranged in an $Ax = b$ form as: $A_{ms}\xi = b_{ms}$, where $\xi \in \mathfrak{R}^{2n(p-1)}$ is the unknown vector, $b_{ms} \in \mathfrak{R}^{2n(p-1)}$ is a given vector and the matrix $A_{ms} \in \mathfrak{R}^{2n(p-1) \times 2n(p-1)}$ presents the structure below

$$A_{ms} = \begin{bmatrix} W^{n_s} & -I & & & & \\ & W^{n_s} & -I & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & W^{n_s} & -I \\ B_0 C_0 & & & & & & B_N C_N W_s^{n_s} \end{bmatrix} \quad (3.14)$$

and the vectors ξ and b_{ms} are given by $\xi = [z_0^0 \ \dots \ z_0^{p-1}]^T$ and $b_{ms} = [\bar{b}^0 \ \dots \ \bar{b}^{p-2} \ \beta - B_N C_N \bar{b}^{p-1}]^T$. Here, after having solved the linear system of equations $A_{ms}\xi = b_{ms}$, the solution is formed by iterating the marching process (3.10) in each subdomain Ω_k using the corresponding initial vector z_0^k . Clearly, the linear system of equations $A_{ms}\xi = b_{ms}$ can be solved directly or iteratively. The matrix A_{ms} is block diagonal dominant [1]. Direct methods for the solution of equation $A_{ms}\xi = b_{ms}$ which ignore the special structure of the system, are not a good choice. In the next subsection we introduce the so called iterative multiple shooting method for the solution of equation $A_{ms}\xi = b_{ms}$.

The iterative multiple shooting method: Defining, for the subdomain Ω_k , the reduction matrix $R_k \in \mathfrak{R}^{2n \times 2n(p-1)}$ and expansion matrix R_k^T as: reduction matrix $R_k : z_0^k = R_k \xi$; then, the simple shooting in a specific k^{th} subdomain can be expressed as

$$z_0^k(m+1) = z_0^k(m) + M_k R_k (b_{ms} - A_{ms} \xi(m)), \quad (3.15)$$

for $k = 0, 1, \dots, p-1$ where the preconditioner matrix $M_k \in \mathfrak{R}^{2n \times 2n}$ is defined as: $M_k := C_c B_c K C$ for $k = 0$, and $M_k := C_c B_c K C - B_N C_0$ for $k = 1, \dots, p-1$.

In the expression for the matrix M_k the matrices B_c and C_c are the input and output matrices of the feedback control law. The matrix K is a feedback gain matrix associated to the feedback control law which was defined in section 3. and matrix C is the output matrix. The $B_c K$ is a preconditioner matrix for $A_s = C W^n C_c$, and for the particular case of the dead-beat controller, we choose the input B_c as an identity matrix and the gain matrix K as the inverse of A_s . In terms of the vector ξ , and using the simple shooting to solve each subdomain k^{th} , the iterative multiple shooting is described by the expression

$$\xi(m + \frac{k}{p-1}) = \xi(m + \frac{k-1}{p-1}) + R_k^T M_k R_k \left(b_{ms} - A_{ms} \xi(m + \frac{k-1}{p-1}) \right), \quad (3.16)$$

$$k = 0, 1, 2, \dots, p-1.$$

Note that, the m^{th} iteration of the multiple shooting occurs when the simple shooting has been completed in all the subdomains Ω_k .

4. Computational results for the Poisson equation

The iterative shooting method described in this work was implemented in Fortran 77 using double precision arithmetic, and was tested on a four node IBM-SP2 parallel computer at NACAD/COPPE/UFRJ. The test problem was the Poisson equation (1.1) subject to Dirichlet boundary conditions: $u(0, y) = 0$; $u(1, y) = 0$; $u(x, 0) = 1$; $u(x, 1) = 1$, with domain $\Omega^h = \{(ih, jh) : i, j = 0, 1, \dots, N\}$. The convergence criterion for all computations is $\|e(m)\| < Tolerance$ where the *error* was defined by equation (3.5) in subsection 3.1.

Results for the iterative simple shooting method: For the matrix $B_c K$, the input matrix B_c was taken as an identity matrix and the gain matrix K was taken formally as the “inverse” of $A_s = CW^n C_c$. The inverse does not need to be explicitly computed, since it is more efficient to perform the computations using the LU decomposition of this matrix.

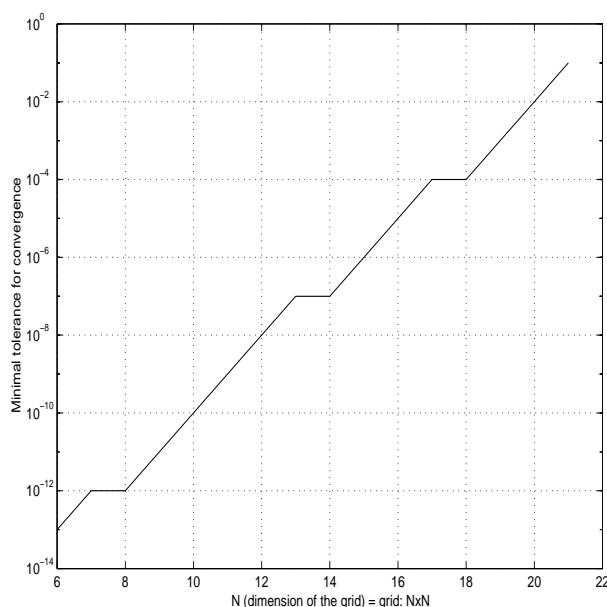


Figure 2: Admissible tolerance to obtain the convergence of the iterative simple shooting method in each subdomain

Even though, the iterative simple shooting method corresponds to a direct method when using infinite precision arithmetic in the computations, the method still remains an iterative method in finite precision arithmetic. From Figure 2, it is possible to see that admissible tolerance to assure convergence depends on the dimension of the grid N .

Additionally, we can see that the “maximum length of the shot” for the iterative simple shooting method, when using double precision arithmetic, is $N = 22$ or, equivalently, the maximum power for the matrix W is $n = 21$ ($n = N - 1$). We can conclude that, for the numerical solution of the Poisson equation, the method becomes numerically unstable when the grid is refined beyond some limit. This instability comes from the fact that in order to compute the marching part it is necessary to compute powers of matrix W . This matrix can be mildly ill conditioned; however its powers will be progressively ill conditioned. Additionally, this limit, using double precision for the computations, is a grid of $\Omega^h = (hi, hj) : i, j = 0, 1, 2, \dots, N$ for $N = 22$, and with this grid, it finds a high precision solution (up to round off) for the adopted initial values.

An iterative multiple shooting method: For the iterative multiple shooting method, we recall that the convergence criterion is $\|b - Ax(m)\|_2 / \|b - Ax(0)\|_2 \leq \textit{tolerance}$, where we adopt the tolerance for the convergence as 10^{-7} .

Iterative multiple shooting					Gauss-Seidel		SOR		Alt. Schwarz GS	
N_s	p	N	it.	time	it.	time	it.	time	it.	time
18	2	34	60	0.8e-1	499	0.23	178	0.11	196	0.11
18	3	50	101	0.27	1004	0.94	367	0.40	360	0.75
18	4	66	152	0.60	1648	2.56	609	0.99	545	1.84
18	5	82	213	1.47	2413	6.26	899	2.20	739	4.03
18	6	98	280	3.16	3290	12.46	1234	4.7	934	7.55
18	7	114	354	6.00	4268	21.28	1610	8.21	1125	12.54

Table 1: Comparison between the iterative multiple shooting, the Gauss-Seidel method and the iterative multiplicative Schwarz method with 5 iterations of the Gauss-Seidel in each subdomain. The item time in the table corresponds to CPU time (it. means iterations).

A comparison between the iterative multiple shooting procedure, Gauss-Seidel method, the successive over relaxation (SOR) and the alternating Schwarz technique using the Gauss-Seidel method as a solver for each subdomain is presented at table 1. In this table, the dimension N_s of the shooting is maintained constant although the dimension of the grid is increased. Consequently, in order to refine the grid the number of subdomains is increased. It can be noted that, the iterative multiple shooting is faster, in CPU time and number of iterations, than the Gauss-Seidel, the SOR and the alternating Schwarz Gauss-Seidel method. So, the iterative multiple shooting is 4 times faster than the Gauss-Seidel, 1.5 times the SOR and 2.7 times the alternating Schwarz Gauss-Seidel.

5. Conclusions

In this paper, a control theoretical interpretation for the shooting method is presented for the numerical solution of the Poisson equation; the TPBVP is defined now in the context of PDEs. In the feedback controller discussion, it can be seen that choosing the matrices $B_c = I$ and $K = A_s^{-1}$, the shooting method can be viewed theoretically as a direct method. However, in practice, due to roundoff errors, which

are present in the computations, the method remains an iterative method. Therefore, the simple shooting method presents the property of having an optimal order of complexity but limited to the size of the grid. This property makes the simple shooting method a strong candidate for a solver in a domain decomposition context.

The iterative multiple shooting method is understood as a kind of domain decomposition method, and overcomes the numerical instability, by subdividing the domain in subdomains and performing the simple shooting in each subdomain. Additionally, if the feedback gain matrix K is pre-computed, the order of complexity for the simple shooting in each subdomain corresponds to the order of complexity of one matrix vector multiplication only ($O((N - 1)^2)$). This conduces to an iterative method for the Poisson equation which, in each iteration, has an optimal order of complexity. Finally, it is important to remark that this new point of view for the numerical solution of elliptical PDEs leads to new and more efficient methods by using some initial value methods, so far used only in the context of ODEs.

References

- [1] U. Ascher, R.M.M. Mattheij and R.D. Russell, "Numerical Solution of Boundary Value Problems for Ordinary Differential Equations", SIAM Classic in Applied Mathematics, 1995.
- [2] K.J. Åström and B. Wittenmark, "Computer-Controlled Systems: Theory and Design", Second Edition, Prentice-Hall International Inc., Englewood Cliffs, N.J. 07632, 1990.
- [3] H. Keller, "Numerical Solution of Two Point Boundary Value Problems", SIAM, Number 24, 1976.
- [4] C.E. Schaerer and E. Kaszkurewicz, The shooting method for the solution of ODE's: a control theoretical perspective", *International Journal of Systems Science*, **32**, No.8 (2001), 1047-1053.
- [5] M. Hermann, D. Kaiser, Shooting methods for two-point BVPs with partially separated endcondition, *Zeitschrift für Angewandte Mathematik und Mechanik*, **75** (1995), 651-668.
- [6] C.E. Schaerer, E. Kaszkurewicz and N. Mangiavacchi, A Multilevel Schwarz Shooting Method for the solution of the Poisson Equation in Two Dimensional Incompressible Flow Simulations", *Applied Mathematics and Computation*, to appear.
- [7] C.H. Lai, On domain decomposition and shooting methods for two point boundary value problems, *Contemporary Mathematics*, **180** (1994), 257-263.
- [8] S.N. Ha, A nonlinear Shooting Method for two-point boundary value problems, *International Journal of Computers and Mathematics with applications*, **42** (2001), 1411-1420.